NAS 12-2035

CR 86303

DYNAMICS RESEARCH CORPORATION

60 Concord Street
Wilmington, Massachusetts

NAS 12-2150

R-106U

OPERATIONAL ALIGNMENT AND CALIBRATION OF THE ISU FOR PHASE 2 OF THE V/STOL PROGRAM

Volume II Alignment

By Donald O. Benson, Jr., and Dr. H. James Rome

December 1969

Prepared under Contract No. NAS12-2150 by

DYNAMICS RESEARCH CORPORATION
60 Concord Street
Wilmington, Massachusetts 01887

for Electronics Research Center

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

TABLE OF CONTENTS

TABLE OF CONTENTS (cont'd)

OPERATIONAL ALIGNMENT AND CALIBRATION OF THE ISU FOR PHASE 2 OF THE V/STOL PROGRAM

Volume II Alignment

By Donald O. Benson, Jr., and Dr. H. James Rome

DYNAMICS RESEARCH CORPORATION
60 Concord Street
Wilmington, Massachusetts 01887

1. SUMMARY

This volume of the final report presents a detailed design and study of an alignment scheme to be used for the strapdown inertial system for Phase 2 of the V/STOL Program. The alignment scheme which was selected had to be capable of being implemented on the on-board, fixed point computer and also had to yield alignment errors which were close to their theoretical minima within 15 minutes of alignment time. A three-stage alignment scheme which employs least squares algorithms was found which meets these criteria. Both a theoretical error analyses employing Kalman filter estimation theory and a computer simulation have been used to demonstrate the effectiveness of the scheme. Confidence in the results has been established by the fact that the theoretical covariance results and the simulation results are consistent. The performance of the three-stage scheme is shown to compare favorably with an optimal alignment scheme in the theoretical covariance analysis.

A detailed equation set for the self-contained alignment scheme is described, and alternative computer implementations for batch processing of the data are discussed. The ultimate choice of which alternative to implement will depend on the amount of computer memory available. The

range and resolution of the computed quantities are estimated from the simulation results in order to determine if single or double precision is required. Rough estimates of the time required to perform the computations are made.

2. INTRODUCTION

The purpose of these studies is to develop operational pre-flight alignment and calibration procedures for the strapdown inertial navigator to be used during Phase 2 of the V/STOL Program. The calibration procedure is designed to be one which can be implemented using the limited test equipment which will be available in the hangar. The alignment procedure is designed to be one which can be accomplished in the helicopter using the fixed point, on-board flight computer. Since the calibration and alignment procedure developments can be discussed separately and since many personnel at NASA-ERC are primarily interested in only one of these developments, the developments are discussed and documented in separate volumes: Volume 1 for the calibration procedure and Volume 2 for the alignment procedure.

Error analyses are presented which indicate the expected accuracies of the developed procedures. These error analyses, of course, are dependent on the assumed input error models. Developing models for the inertial sensors was one of the tasks performed as part of the overall study.

To the extent that they were known, input parameters for the study were supplied by NASA-ERC. These parameters describe the following types of inputs:

- specifications on the performance of the accelerometers and gyros,
- a description of the dynamic environment in which the alignment and calibration must be performed,

- a description of the test equipment that will be available for calibration in the hangar, and
- a description of the coordinate frames and the mounting of the inertial sensors in the strapdown system.

In those cases where the specifications and descriptions were incomplete, the associated error analyses must be considered preliminary in nature. In those cases where assumptions had to be made, the resultant error analysis provides information on how to set the performance specifications.

ACKNOWLEDGEMENT

The authors wish to extend their appreciation to P. J. McKinnon of DRC for her effort in programming the simulation, and to D. M. Garmer of DRC for his helpful suggestions.

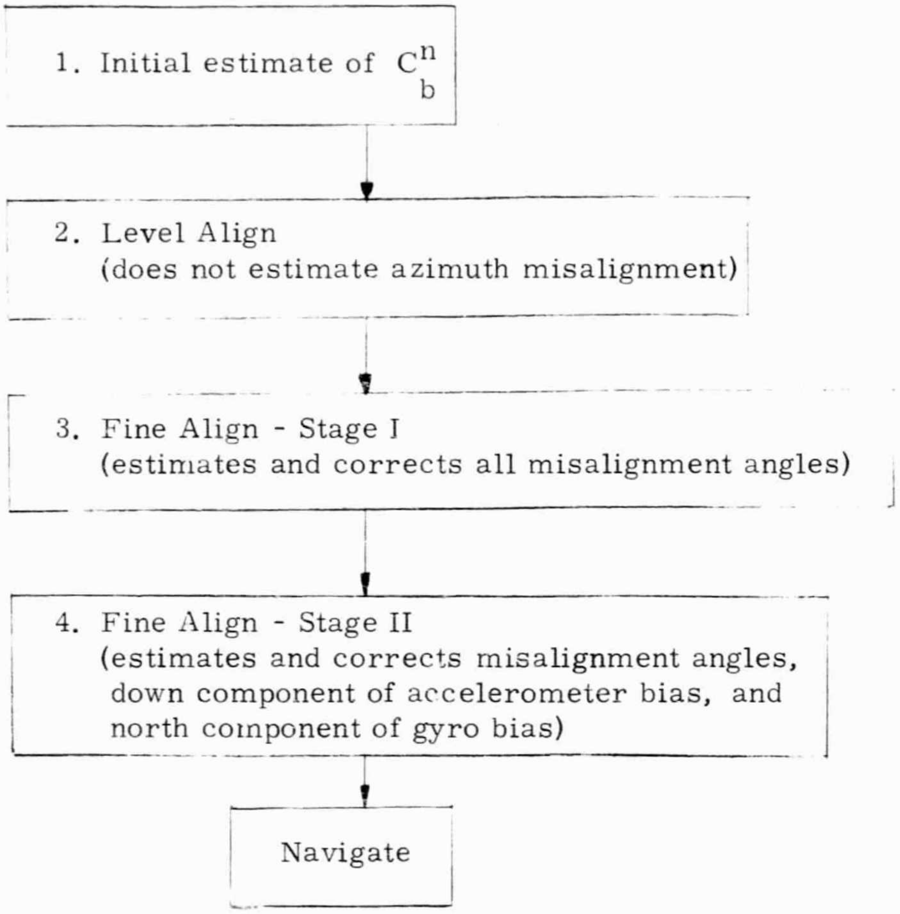
3. GLOSSARY

$\underline{\epsilon}$	misalignment angles
$\underline{E}$	misalignment matrix
w_{ie}^{n}	earth rate coordinatized in the n frame
n	local level frame
$\tilde{n}$	nominal level frame
b	orthogonal b frame
$\underline{A}$	accelerometer errors
$\underline{R}$	gyro errors
C_b^n	direction cosine matrix coordinatizing a vector in the n frame from the b frame
Ω	skew symmetric angular velocity matrix
ΔV	accelerometer incremental velocity output
$\Delta\theta$	gyro incremental angle output
q^G	gyro quantization level
q^A	accelerometer quantization level
Φ	transition matrix
$\underline{g}^n$	gravity coordinatized in the n frame
$\underline{\delta f}$	translational vibrations
$\underline{w}_{nb}^{b}$	rotational vibrations (angular velocity of b frame with respect to n frame coordinatized in b frame)
D_1 D_2 E_1 E_2 GGI_1 GGI_2	matrices used for fine align filter
$\wedge$	denotes estimate
$\sim$	denotes estimation error, e.g., $\underline{\tilde{\epsilon}} = \underline{\hat{\epsilon}} - \underline{\epsilon}$

4. ALIGNMENT OF THE V/STOL STRAPDOWN INERTIAL SYSTEM

The function of alignment of a strapdown inertial system is to determine the coordinate transformation between the inertial sensor input axes and the navigation frame (or other related frame) immediately before the navigate mode is entered. Let the transformation at any time be $C_b^n(t)$. Due to measurement errors the exact transformation cannot be determined. Instead, a coordinate transformation $C_b^{\tilde{n}}$ is determined where the $\tilde{n}$ frame is "close" to the n frame. The angles determiring the relation between the n and $\tilde{n}$ frames are called the misalignment angles.

The design and analysis of the fine alignment scheme assumes that the misalignment angles are small enough for linear analysis. To realize this assumption, a coarse alignment scheme is implemented. The following block diagram indicates the time-sequencing of the alignment scheme.


1. Initial estimate of C_b^n
2. Level Align
(does not estimate azimuth misalignment)
3. Fine Align - Stage I
(estimates and corrects all misalignment angles)
4. Fine Align - Stage II
(estimates and corrects misalignment angles, down component of accelerometer bias, and north component of gyro bias)
Navigate

4.1 INITIAL ESTIMATE OF C_b^n

From previous calibration, that of determining the sensor input rates with respect to each other and relative to the body (helicopter) axes, the following transformations are known:

T_s^b - the transformation which takes sensor outputs into an orthogonal frame given by the "b" axes

C_h^b - the orthogonal transformation which relates the "h" orthogonal helicopter axes to the orthogonal "b" axes

As previously stated, the accuracy of C_h^b depends upon the accuracy of the mechanical mounting of the ISU to the helicopter.

For initialization, first assume that the ground under the helicopter is approximately level. Then the angle θ between z and down is known. Secondly, assume that the approximate heading (H) of the helicopter is known from a magnetic compass or from a comparison to some bench mark in the hangar.

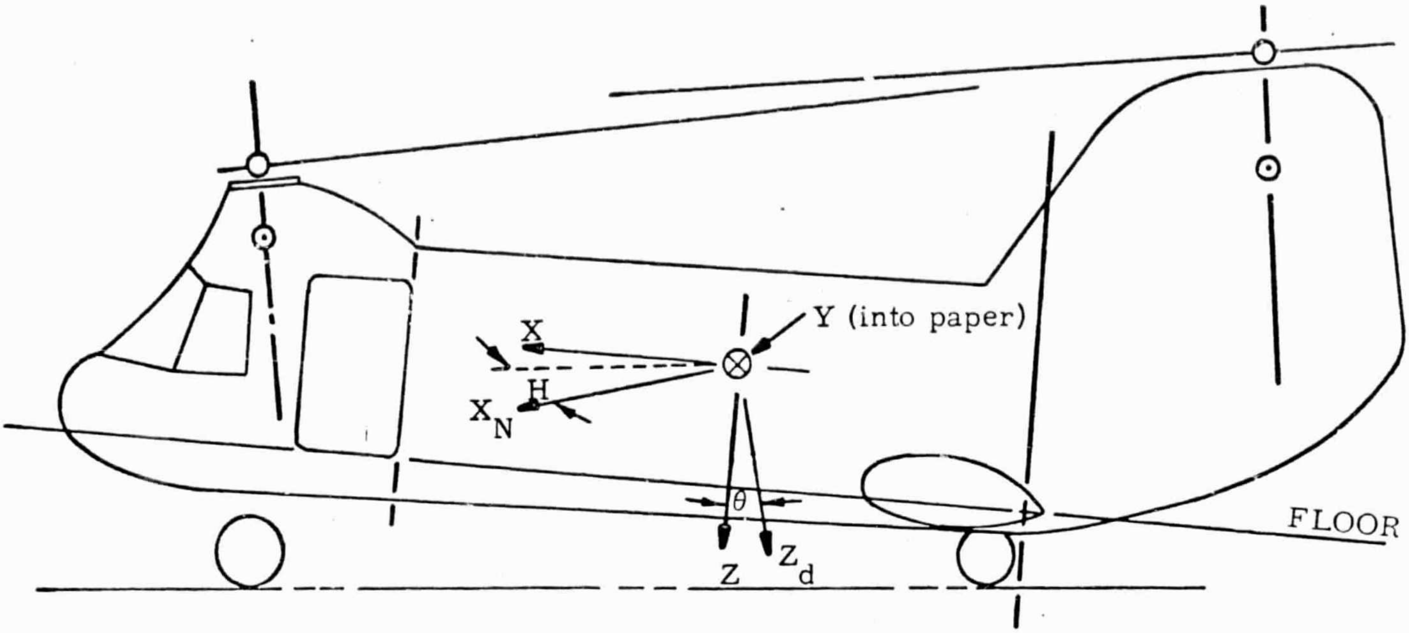


Figure 4.1-1 Helicopter Axes and North, East, Down Axes

From the properties of transformation matrices

$$C_n^b = C_h^b \, C_n^h$$

The orientation of the helicopter with respect the the "n" frame is determined from a positive rotation of angle H about Z_d followed by a positive rotation of angle θ about Y.

The direction cosine matrix C_n^h is then

$$C_n^h = \begin{bmatrix} \cos\theta \cos H & \cos\theta \sin H & -\sin\theta \\ -\sin H & \cos H & 0 \\ \sin\theta \cos H & \sin\theta \sin H & \cos\theta \end{bmatrix}$$

The simple scheme is fast and requires little computation, but it can suffere large errors, perhaps as much as 5° of misalignment about each axis. Also note

$$C_b^n = (C_n^b)^T$$

4.2 LEVEL ALIGNMENT

The outputs of the accelerometers are averaged over a short time. The vector is resolved into the $\tilde{n}$ frame and compared with what it should be in the n frame.

Let the accelerometer output be denoted as $\underline{Z}_a$.

$$\underline{Z}_a = -\underline{g}^b + \underline{\delta f}^b + \underline{A}$$

where

$\underline{Z}_a$	accelerometer output
$\underline{g}^n = \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix}$	gravity in n frame
$\underline{\delta f}^b$	vibratory accelerations
$\underline{A}$	accelerometer errors

Let

$$\underline{\bar{f}}^b = \frac{1}{T} \int_o^T \underline{Z}_a dt$$

$$\underline{\bar{f}}^{\tilde{n}} = C_b^{n-} \underline{\bar{f}}^b$$

Let

$$u_{\tilde{n}} = \frac{\underline{f}^{\tilde{n}}}{\left|\underline{f}^{\tilde{n}}\right|}$$

u in the n frame is given as

$$u_n = \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix}$$

The C_b^{n-} matrix is corrected by rotating the $u_{\tilde{n}}$ vector into the u_n vector. Then (Ref. 23)

$$C_b^{n+} = [I - (1 - u_{\tilde{n}}^T u_n) (I - \frac{kk^T}{k^T k}) + K] C_b^{n-}$$

where

$$k = u_{\tilde{n}} \times u_n$$

$$K = \begin{bmatrix} 0 & -k_z & k_y \\ -k_z & 0 & -k_x \\ -k_y & k_x & 0 \end{bmatrix}$$

Heading or azimuth cannot be estimated by this method. Any attempt to estimate heading by averaging the gyro outputs to estimate earth rate will be futile because the rotational vibration amplitude overwhelms earth rate (1°/sec compared to 15°/hr).

4.3 FINE ALIGN

After the level align is finished, the fine align mode is entered. In this mode the direction cosine matrix (DCM) is updated using the gyro outputs. The DCM ($C_b^{\tilde{n}}$) determines the time-varying coordinate between the b frame and the nominal $\tilde{n}$ frame.

The accelerometer outputs resolved into the $\tilde{n}$ frame are used as measurements. The misalignment angles, gyro and accelerometer errors determine the state vector. At the end of the first stage, the three misalignment angles are estimated, and the direction cosine matrix is reset to begin the second stage of fine align. The reset relinearizes the equations.

The direction cosine matrix determining the transformation between the n and the b frame satisfies

$$\dot{C}_b^n = -\Omega_{in}^n C_b^n + C_b^n \Omega_{ib}^b \tag{4.3-1}$$

where the Ω matrices are skew symmetric angular velocity matrices

$$\Omega_{ib}^b = \begin{bmatrix} 0 & -w_z & w_y \\ w_z & 0 & -w_x \\ -w_y & w_x & 0 \end{bmatrix} \tag{4.3-2}$$

$$\Omega_{in}^n = \Omega_{ie}^n + \Omega_{en}^n$$

where w_{en}^n is the angular velocity of the ISU location over the earth.

This term can be dropped since it is small compared to other terms. To show this, the acceleration model over the earth is

$$\begin{bmatrix} \delta f_x \\ \delta f_y \\ \delta f_z \end{bmatrix} = \begin{bmatrix} .1g\ \sin(wt + \phi_1) \\ .1g\ \sin(wt + \phi_2) \\ .1g\ \sin(wt + \phi_3) \end{bmatrix} \qquad (4.3\text{-}3)$$

The resulting velocity is

$$\delta V_x = -\frac{.1g}{w} \cos(wt + \phi_1) \qquad (4.3\text{-}4)$$

for $w = 25$ Hz

$$\delta V_{x\ max} \approx .021 \text{ ft/sec}$$

hence

$$w_{en} = \frac{.021}{r}$$

$$r = 21 \times 10^6 \text{ ft (earth radius)}$$

$$w_{en} \approx 2 \times 10^{-4} \text{ deg/hr}$$

Thus the true DCM satisfies

$$\dot{C}_b^n = -\Omega_{ie}^n C_b^n + C_b^n \Omega_{ib}^b \qquad (4.3\text{-}5)$$

where

$$\Omega^n_{ie} = \begin{bmatrix} 0 & w_{ie}\sin L & 0 \\ -w_{ie}\sin L & 0 & -w_{ie}\cos L \\ 0 & w_{ie}\cos L & 0 \end{bmatrix} \quad (4.3\text{-}6)$$

w_{ie}	earth rate
L	Latitude

Let the gyro outputs be

$$\underline{z}_g = (\underline{w}^b_{ib})_g = \underline{w}^b_{ib} + \underline{R} = \underline{w}^b_{ie} + \underline{w}^b_{nb} + \underline{R} \quad (4.3\text{-}7)$$

In skew symmetric form this can be written as

$$(\Omega^b_{ib})_g = \Omega^b_{ib} + R_m \quad (4.3\text{-}8)$$

The gyro outputs are used to implement a nominal solution. The nominal DCM satisfies the following differential equation

$$\dot{C}^{\tilde{n}}_b = -\Omega^n_{ie}\, C^{\tilde{n}}_b + C^{\tilde{n}}_b\, (\Omega^b_{ib})_g \quad (4.3\text{-}9)$$

The misalignment is the transformation between the n frame and the $\tilde{n}$ frame.

Let

$$C^n_{\tilde{n}} = I + E \quad (4.3\text{-}10)$$

Then

$$C_b^n = C_{\tilde{n}}^n \, C_b^{\tilde{n}} = (I + E) C_b^{\tilde{n}} \tag{4.3-11}$$

$$\dot{C}_b^n = (I + E)\, \dot{C}_b^{\tilde{n}} + \dot{E}\, C_b^{\tilde{n}} \tag{4.3-12}$$

Substitute Eq. (4.3-3) and Eq. (4.3-7) in the above equation and after some algebra

$$\dot{E} = E\Omega_{ie}^n - \Omega_{ie}^n E - C_b^{\tilde{n}} R_m C_{\tilde{n}}^b - E C_b^{\tilde{n}} R_m C_{\tilde{n}}^b \tag{4.3-13}$$

As yet no assumption of linearity has been made. A linear equation is obtained if the term $E C_b^{\tilde{n}} R_m C_{\tilde{n}}^b$ is dropped. This term is small compared to the other terms. Then

$$\boxed{\begin{array}{c} \dot{E} = E\Omega_{ie}^n - \Omega_{ie}^n E - C_b^{\tilde{n}} R_m C_{\tilde{n}}^b \\ \text{system equation} \end{array}} \tag{4.3-14}$$

The accelerometer output is

$$\underline{z}_a = -\underline{g}^b + \delta f^b + \underline{A} \tag{4.3-15}$$

$$C_b^{\tilde{n}} \underline{z}_a = -C_n^{\tilde{n}} C_b^n \underline{g}^b + C_b^{\tilde{n}} \underline{\delta f}^b + C_b^{\tilde{n}} \underline{A}$$

$$= -C_n^{\tilde{n}} g^n + C_b^{\tilde{n}} \delta f^b + C_b^{\tilde{n}} \underline{A}$$

$$\boxed{\begin{array}{c} C_b^{\tilde{n}} \underline{z}_a + \underline{g}^n = -E^T \underline{g}^n + C_b^{\tilde{n}} \underline{\delta f}^b + C_b^{\tilde{n}} \underline{A} \\ \text{measurement} \end{array}} \tag{4.3-16}$$

Note that E is a 3 x 3 matrix. If it is further assumed that E is small, then it is skew symmetric, i. e.,

$$E = \begin{bmatrix} 0 & -\epsilon_z & \epsilon_E \\ \epsilon_z & 0 & -\epsilon_N \\ -\epsilon_E & \epsilon_N & 0 \end{bmatrix} \qquad (4.3\text{-}17)$$

Now define

$$\underline{\epsilon} = \begin{bmatrix} \epsilon_N \\ \epsilon_E \\ \epsilon_z \end{bmatrix} \qquad (4.3\text{-}18)$$

Thenthe system and measurement equations can be written as

$$\boxed{\begin{aligned} \dot{\underline{\epsilon}} &= -\Omega^n_{ie}\,\underline{\epsilon} - C^{\tilde{n}}_b R && (4.3\text{-}19) \\ C^{\tilde{n}}_b \underline{z}_a + \underline{g}^n &= \begin{bmatrix} 0 & g & 0 \\ -g & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \epsilon_N \\ \epsilon_E \\ \epsilon_z \end{bmatrix} + C^{\tilde{n}}_b \underline{\delta f}^b + C^{\tilde{n}}_b \underline{A} && (4.3\text{-}20) \end{aligned}}$$

If instrument biases are to be estimated, they are biases in the b frame. However for the present problem, the biases can be assumed constant in the n or $\tilde{n}$ frame (the changes in the elements of $C^{\tilde{n}}_b$ are small). After they are estimated in the n frame, they can be transformed into the b frame. With this assumption, the filter can be designed on the basis of a time invariant system and measurement. The simulation of the alignment scheme verified this assumption.

Vertical deflections can be compensated (if they are known) at the end of the last stage of fine align, since they appear as equivalent accelerometer biases.

A block diagram of the fine align scheme is shown below.

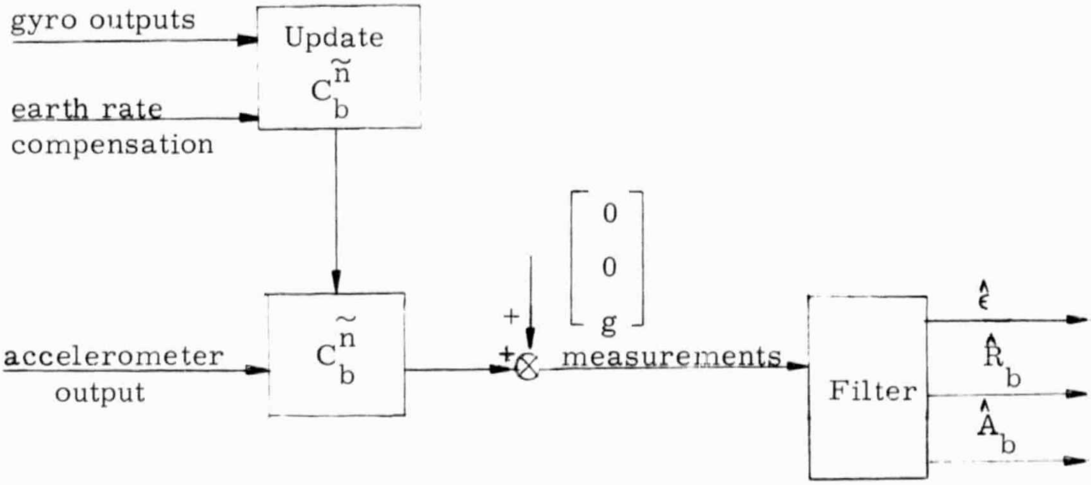


Figure 4.3-1 Block Diagram of Fine Align

5. CHOICE OF AN ALIGNMENT SCHEME

There are two distinct approaches which can be taken toward the alignment scheme. One is the Kalman filter approach and the other is the least squares estimation approach. The measurements in either case are processed accelerometer outputs. The differences between these approaches are discussed in the following paragraphs.

The Kalman filter attempts to estimate all states of the system including all tilts, accelerometer biases, gyro biases, and any random error states in the model. In order to implement the Kalman filter, one must model all the error states of the system including those above, specify the measurements in terms of the states, and specify the measurement noise. In addition, the covariance of the intial uncertainties of all the states must be specified. One can then apply this information to the standard Kalman algorithms to obtain a rule (filter) which specifies the optimal operation on the measurement data to estimate the states of the system. Once the states are estimated, they can be used to reset or align the strapdown system.

The other approach to the alignment problem is called the least squares approach. In this approach it is assumed that the system can be modelled as a deterministic system; i. e., platform tilts driven by gyro biases only. In addition, it is assumed that the accelerometer errors are also pure biases. Thus the errors for all time can be written or modelled as the appropriate transition matrix* times the initial error states. Least squares fitting involves choosing the initial error states such that the mean

* This is known a priori.

squared difference between the estimated measurements as computed from the initial conditions, and the actual measurements observed, is a minimum. Alignment is accomplished by propagating the estimated initial error states forward to the time of reset, then resetting with this information.

The least squares technique is suboptimal because (1) it does not incorporate every ramification of the error model, (2) it does not account for the fact that something is known about the initial bounds on the various errors. It has the advantage over Kalman filtering in that it is considerably simpler to implement. It should be pointed out however, that both the least squares technique and the Kalman filter can be implemented recursively.

In this chapter both Kalman filtering and least squares estimation are considered. Numerical results are given in order to justify the overall advantage of the alignment technique finally decided upon. The analysis and results obtained in this chapter lead to the equations which will be used in the alignment scheme.

5.1 OBSERVABLE STATES

Not all error states of the strapdown inertial system can be estimated at the time of alignment. The states that cannot be reliably estimated are called unobservable states. For instance, the North and East accelerometer biases are unobservable. It is important for the designer to have knowledge of what states are unobservable so that he can intelligently construct a suboptimal filter which is as simple as possible, but close to optimal. In this section the observability of the various error states are studied.

Kalman filtering techniques offer the most convenient way of studying this problem. If a state cannot be estimated, it will be demonstrated by observing that the initial covariance is not reduced by the measurements. The nominal system specified below has been chosen for study.

Two types of acceleration environments have been considered:

Acceleration environment 1	0. 35 g, 25 cps sinusoidal
Acceleration environment 2	0.1g, rms random[*]

Other system specifications are given below.

Vehicle location	42° N Latitude	
Gyro error	0. 05°/hr bias	
Accelerometer error	20 μg bias	
Gyro Quantization level	1 $\widehat{\text{sec}}$	(5. 1-1)
Acc. quantization level	0. 0025 ft/sec	
C matrix update time	0. 01 sec	

Two sampling intervals, 10 sec and 28. 8 sec, respectively, are considered. It is assumed that a measurement consists of the North and East accelerom-

[*] The spectrum of the acceleration is flat from 2 cps to about 25 cps. It falls off on both sides of these limits.

eter outputs averaged over the sampling interval.* Other specifications are:

$$\begin{array}{ll} \text{Initial tilt error} & 0.081^\circ \ ^{**} \\ \text{Initial azimuth error} & 5^\circ \end{array} \qquad (5.1\text{-}2)$$

Since the acceleromete: outputs of the ISU are space-stabilized through the C matrix, no modelling of the rotational motion is really necessary. One must remember, though, that the inertial instruments are fixed to the body of the airframe, thus the contributing errors from the instruments vary with time. Since rotation about any particular axis is expected to be a degree or less, the time-varying contribution is very small and can be ignored.

The actual equations used in the study can be found in Appendix E. These equations have been processed by DRC's state-space programs to determine the optimal covariance of the errors in estimating the state of the system for various filtering times.

The results of the study are tabulated in Fig. 5.1. In this table the optimal (minimum) errors associated with estimating the error states of the system have been tabulated for 12 minutes and 24 minutes of filtering.

* It should be pointed out that the z accelerometer output -g (gravity) can also be measured. This measurement can be used to obtain the z accelerometer bias. This measurement is independent of the others which enables one to use a simple averaging to obtain the bias. Because of the simplicity of the estimation in this case, it is not considered here.

** A 20 sec crude align will specify the tilts to within this accuracy.

RUN	Estim. Time	Samp. Inter.	Acceler-ation	$\tilde{\epsilon}_N$ sec	$\tilde{\epsilon}_E$ sec	$\tilde{\epsilon}_z$ deg	$\tilde{R}_N$ deg/hr	$\tilde{R}_E$ deg/hr	$\tilde{R}_z$ deg/hr	$\tilde{A}_N$ μg	$\tilde{A}_E$ μg
1	12 min	10 sec	.35g	4.12	4.12	.254	.0002	.0495	.0062	20μg	20μg
	24 min		25 cps	4.12	4.12	.254	.000192	.0495	.00081		
2	12 min	28.8 sec	.35g	4.12	4.12	.254	.000204	.0495	.0142	20μg	20μg
	24 min		25 cps	4.12	4.12	.254	.0002	.0495	.00124		
3	12 min	10 sec	.1g	4.12	4.12	.255	.00045	.0495	.045	20μg	20μg
	24 min		random	4.12	4.12	.254	.00022	.0495	.012		
4	12 min	28.8 sec	.1g	4.35	4.35	.255	.00348	.0495	.0495	20μg	20μg
	24 min		random	4.15	4.15	.255	.00071	.0495	.0455		

Figure 5.1 Results of Optimal Filtering Studies: RMS Errors In Estimating Various States

From the results in the table, and basic knowledge of the system-measurement configuration, it can be easily seen that in general:

East tilt can be estimated to approximately $A_{bN/g}$

North tilt can be estimated to approximately $A_{bE/g}$

Heading can be estimated to approximately $R_E/w_{ie}\cos L$

In addition, the effective North gyro bias can be estimated to within 5% of its original value, or better. The accuracy depends on the acceleration environment, on the length of the sampling interval, and the time available for estimation.

The results show that the North and East accelerometer biases cannot be estimated at all, since the error in estimating the biases after filtering is the same value as the original bias. In addition, the results show that the East gyro bias cannot be estimated, for the same reason. The fact that the estimation error after filtering is slightly less than the original bias error is due to the finite initial distribution of the error covariance. Aside from the use of this artificially conceived a priori knowledge, there is no estimation of this state.

In the case where the acceleration environment consists of a 25 cps signal, the effective z-gyro bias can be estimated significantly. In the case where the acceleration environment is random, the z-gyro bias is estimated to a much smaller extent. The degree of estimation depends upon the length of the estimation time and the sampling interval used. When the sampling interval is 28.8 sec, there is essentially no estimation even for an estimation time of 24 minutes. When the sampling interval is 10 sec, there is significant estimation when the sampling time is 24 minutes.

The 25 cps results are overly optimistic because the sampling intervals are an even multiple of the frequency. Therefore the effects of the acceleration tend to cancel over an averaging interval.

The results obtained by modelling the acceleration environment as a random process are more relevant to the true alignment environment. Therefore these results should carry the most weight in any decision to reset z-gyro bias.

DRC has decided not to reset the z-gyro bias in the alignment scheme for three reasons. First, the final alignment scheme should be workable for the minimum alignment time of 15 minutes. Even at the high sampling rate of ten seconds, there is little estimation of this state. If such an estimate were attempted, there is a significant probability that the bias estimate would contain a greater error than the actual bias. Second, the acceleration environment may be worse than that assumed. Thus the estimate would be further deteriorated. Third, results which were previously obtained (see Appendix C) for a different measurement configuration indicated that, if there is any random-type noise on the gyro drift rate, the estimate of the z-gyro bias deteriorates disproportionately with the magnitude of the random noise. In summary, any attempt to estimate the z-gyro drift will be unreliable. There is a high probability that the system performance will be deteriorated by resetting the z-gyro bias.

One can conclude from the results presented in this chapter that the following states can be estimated and reset during alignment:

1. North and East tilt
2. Heading
3. North Gyro drift rate bias.

Any suboptimal technique incorporated to perform alignment should be geared to estimating these quantities. The fact that other quantities cannot be estimated, and indeed, seem to have no effect on the estimation algorithms used, can be used to simplify any finalized schemes.

5.2 KALMAN FILTERING VS. LEAST SQUARES ESTIMATION

As was mentioned in the Introduction, there appear to be two workable techniques to effect alignment. One is implementation of the Kalman estimation algorithms and the other is least squar≥s estimation of tilts and biases. Kalman filtering, being optimal, has the potential of being more accurate. However, the least squares technique is simpler to implement. In this section equations are developed and numerical results given to justify the use of the simpler technique, the least squares approach.

5.2.1 Ground Rules for Alignment

Before one can evaluate either approach, he must set up a set of ground rules on which to base the estimation techniques. He must be able to justify his choice. Specifically, the following has to be chosen.

1. Form of measurements to be used.
2. Number of stages in the alignment (if there is more than one).
3. Sampling rate used in each stage.
4. Estimation time in each stage (with the possibility of the last stage having an increasing estimation time).
5. Estimated error states which will be reset.

The specification of ground rules is based primarily on preliminary studies. In the studies (see Appendix C) measurements were assumed to be the doubly-integrated accelerometer outputs. A relatively harsh instrument error environment was assumed. It was assumed that the system was aligned to within 1/2 degree in attitude before alignment was started. However, when sensitivity analyses were performed on the optimal Kalman estimators, it was found that the estimation accuracy deteriorated if there were small changes in the initial covariance. This effect was more pronounced with a longer sampling time. The same type of effect was observed when an approximate least squares estimation scheme was used. The immediate solution to the problem was to incorporate a three-stage alignment with reduced sampling rates. This eliminated the 'P sensitivity'. The three stages consisted of a crude align and two fine aligns.

It should also be pointed out that there is another good reason for going to a three-stage alignment. Kalman filtering and least squares estimation are based on linear theory. When there are large misalignment angles in the strapdown system error model, it cannot be properly modelled as a linear system. Thus the accuracy of the estimation is in doubt. By using a three-stage alignment,one can be assured that at least the errors in the last stage can be adequately represented by a linear system.

In the latter stages of the development of the alignment scheme it was decided that the measurements should be the averaged accelerometer outputs. Such measurements are more easily incorporated into a simple least squares technique than are the doubly-integrated accelerometer measurements. The concept of the three-stage alignment and the concern for the length of sampling intervals were carried forward with the new measurement scheme.

As was discussed in Section 5.1, the states to be reset are North and East tilt, heading and North gyro bias (also z-accelerometer bias). The others can be ignored, insofar as reset is concerned.

The preceding facts lead in part to the following ground rules for the alignment scheme. The type of measurements are accelerometer outputs averaged over a measurement interval. The alignment operation can be described by the following table.

Stage	Name	Length of Sampling Interval	Total Time of Stage	Quantities Reset at End of Stage
1	Crude Align	20 sec	20 sec	North & East Tilt
2	First Fine Align	10 sec	200 sec	All 4(5) Error States
3	Second Fine Align	28.8 sec	12 min	All 4(5) Error States of Interest

The above scheme meets the specification of an alignment scheme in that it requires no more than 15 minutes, and does not require a huge amount of data processing. In this section least squares estimation of the error states will be compared to optimal Kalman filtering on a covariance analysis basis.

5.2.2 Least Squares Fitting

Let us briefly review the algorithms which are to be used in the least squares technique. The measurements are averaged North and East accelerometer outputs. However, the averaged outputs are treated as the instantaneous accelerometer outputs at the middle of the sampling interval. The measurements can be written

$$z_{1i} = \frac{1}{\Delta T} \int_{t_{i-1/2}}^{t_{i+1/2}} (\dot{V}_x(t)+A_{bN} + g\epsilon_E)\, dt$$

$$z_{2i} = \frac{1}{\Delta T} \int_{t_{i-1/2}}^{t_{i+1/2}} (\dot{V}_y(t)+A_{bE} - g\epsilon_N)\, dt \qquad (5.2\text{-}1)$$

$\dot{V}_x$, $\dot{V}_y$ are true accelerations; A_{bN}, A_{bE} are North and East accelerometer biases, and ϵ_N, ϵ_E are North and East tilts, respectively.

The results in Section 5.1 revealed that the tilts are estimated to the accuracy of the accelerometer biases. The biases, in turn, are not estimated at all. Therefore it can be reasonably assumed that the estimator is not affected by the bias, and for this reason the bias can be ignored.

Now consider the system itself. Using similar reasoning it can be shown that the East gyro bias need not be modelled. Essentially, heading plus (East gyro bias)/w_{ie}cos L, is estimated. In addition the z-gyro drift does not seem to affect the estimation accuracy of the three error angles and the North gyro drift. Therefore it has been decided to neglect the z-gyro drift in the modelling. The finalized model used in the least squares fitting is the following:

$$\begin{bmatrix} \dot{\epsilon}_N \\ \dot{\epsilon}_E \\ \dot{\epsilon}_z \\ \dot{R}_N \end{bmatrix} = \begin{bmatrix} 0 & -w_{ie} \sin L & 0 & -1 \\ w_{ie} \sin L & 0 & w_{ie} \cos L & 0 \\ 0 & -w_{ie} \cos L & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \epsilon_N \\ \epsilon_E \\ \epsilon_z \\ R_N \end{bmatrix} \tag{5.2-2}$$

with the expected measurement

$$\mu_i = \begin{bmatrix} 0 & g & 0 & 0 \\ -g & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \epsilon_{Ni} \\ \epsilon_{Ei} \\ \epsilon_{zi} \\ R_{Ni} \end{bmatrix} \tag{5.2-3}$$

$$\mu_i = G \cdot \underline{\epsilon}^{\dagger}(t_i)$$

from Eq. (5.2-2), it can be shown that

$$\underline{\epsilon}^{\dagger}(t_i) = \Phi(t_i)\underline{\epsilon}^{\dagger}(0) \tag{5.2-4}$$

where $\epsilon^{\dagger}$ is the vector in (5.2-2) and

$$\Phi(t) = \begin{bmatrix} s^2 \cos w_{ie} t + c^2 & -s \sin w_{ie} t & s\,c(\cos w_{ie} t - 1) & \frac{-s^2 \sin w_{ie} t}{w_{ie}} - c^2 t \\ s \sin w_{ie} t & \cos w_{ie} t & c \sin w_{ie} t & -\frac{-s(1-\cos w_{ie} t)}{w_{ie}} \\ sc(\cos w_{ie} t - 1) & -c \sin w_{ie} t & c^2 \cos w_{ie} t + s^2 & -sc \frac{\sin w_{ie} t}{w_{ie}} - t \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.2-5}$$

and

$$c = \cos L$$
$$s = \sin L$$

Define $E(t) = G\,\Phi(t)$. (5.2-6)

Then $E(t)\overset{\dagger}{\epsilon}(0) = \mu$ = the expected measurement at time t. By definition of the least squares technique, choose $\overset{\dagger}{\epsilon}(0)$ such that it minimizes:

$$V_r = \sum_{i=1}^{N} \left(E(t_i)\overset{\dagger}{\epsilon}(0) - \underline{Z}_i\right)^T \left(E(t_i)\,\overset{\dagger}{\epsilon}(0) - \underline{Z}_i\right) \tag{5.2-7}$$

which is accomplished when

$$\underline{\overset{\wedge\dagger}{\epsilon}}(0) = W^{-1} \sum_{i=1}^{N} E^T(t_i)\,\underline{Z}_i \tag{5.2-8}$$

$\underline{Z}_i$ is a two vector made up of measurements in Eq. (5.2-1). Also

$$W = \sum_{i=1}^{N} E^T(t_i)\,E(t_i) \tag{5.2-9}$$

The value used to reset the strapdown system is the initial condition extrapolated to the time of reset:

$$\overset{\wedge\dagger}{\epsilon}(T) = \Phi(T)\,\underline{\overset{\wedge\dagger}{\epsilon}}(0) \tag{5.2-10}$$

T is the time of reset.

The values used to reset the inertial system can be rewritten as a linear combination of the inputs:

$$\overset{\wedge\dagger}{\epsilon}(T) = U_i\,\underline{Z}_i \tag{5.2-11}$$

where

$$U_i = \Phi(T)\,W^{-1}\,E^T(t_i) \tag{5.2-12}$$

5.2.3 Covariance Analysis

A covariance analysis has been performed to analyze the least squares estimation performance when used in the three-stage alignment scheme defined above. Statistics describing the error equations and quantiz ation errors were modelled. The full model and description of the measurement equations can be found in Appendix E. Conceptually, the system can be shown as in Fig. 5.2-1.

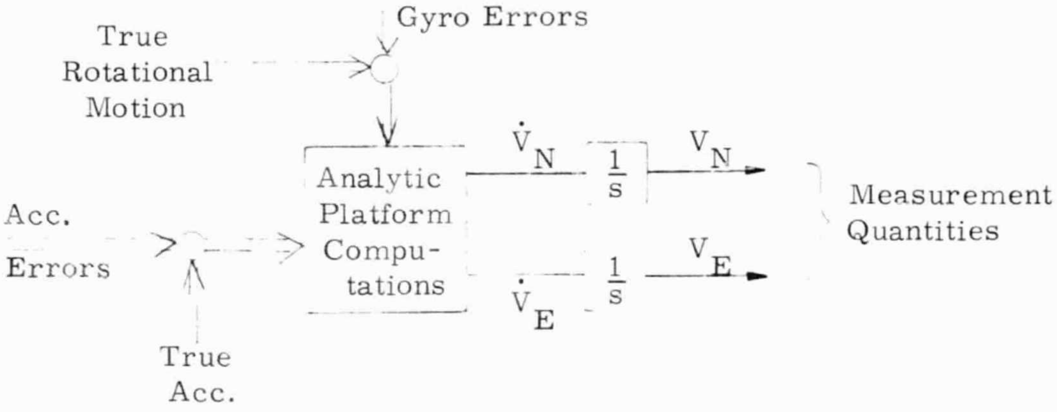


Figure 5.2-1 Basic Strapdown System

$\dot{V}_N$ and $\dot{V}_E$ are respectively the North and East accelerometer outputs.

The covariance analysis is carried out in the following way: First consider the crude alignment which estimates the tilts via a simple average

$$\hat{\epsilon}_N = (V_N(T) - V_N(0))/gT$$

$$\hat{\epsilon}_E = (V_E(T) - V_E(0))/gT \qquad (5.2\text{-}13)$$

Using the full statistical model for the system errors including the initial covariance of the states, P_o , compute

$$C_{cr} = E\left[\left(\begin{bmatrix}\epsilon_N(T_1)\\ \epsilon_E(T_1)\\ \epsilon_z(T_1)\\ R_N\end{bmatrix} - \begin{bmatrix}\hat{\epsilon}_N(T_1)\\ \hat{\epsilon}_E(T_1)\\ 0\\ 0\end{bmatrix}\right)\left(\begin{bmatrix}\epsilon_N(T)\\ \epsilon_E(T)\\ \epsilon_z(T)\\ R_N\end{bmatrix} - \begin{bmatrix}\hat{\epsilon}_N(T)\\ \hat{\epsilon}_E(T)\\ 0\\ 0\end{bmatrix}\right)^T\right] \qquad (5.2\text{-}14)$$

Replace the appropriate elements of the initial covariance of the states with the matrix C_{cr}. This indicates the covariance of the states after the crude align reset. Call this matrix P_C, where

$$P_C = \begin{bmatrix} C_{cr} & \\ & P_o \end{bmatrix} \qquad (5.2\text{-}15)$$

In the first fine align, the following procedures are followed.

Sampling interval is $\Delta T = 10$ sec

Total time to alignment is $T = 200$ sec

Obtain the least squares coefficients, $U_1, U_2, \ldots, U_{20}$, associated with the measurements. They are obtained from Eq. (5.2-12). Estimate

$$\begin{bmatrix}\hat{\epsilon}_N(T_2)\\ \hat{\epsilon}_E(T_2)\\ \hat{\epsilon}_z(T_2)\\ \hat{R}_N(T_2)\end{bmatrix} = \sum_{i=1}^{20} \frac{U_i}{\Delta T}\begin{bmatrix} V_N[(i+1)\Delta T] - V_N(i\Delta T)\\ V_E[(i+1)\Delta T] - V_E(i\Delta T)\end{bmatrix} \qquad (5.2\text{-}16)$$

Using the covariance, P_C, as the initial covariance, compute

$$C_{F1} = E\left[\left(\begin{bmatrix} \epsilon_N(T_2) \\ \epsilon_E(T_2) \\ \epsilon_z(T_2) \\ R_N \end{bmatrix} - \begin{bmatrix} \hat{\epsilon}_N(T_2) \\ \hat{\epsilon}_E(T_2) \\ \hat{\epsilon}_z(T_2) \\ \hat{R}_N(T_2) \end{bmatrix}\right)\left(\begin{bmatrix} \epsilon_N(T_2) \\ \epsilon_E(T_2) \\ \epsilon_z(T_2) \\ R_N \end{bmatrix} - \begin{bmatrix} \hat{\epsilon}_N(T_2) \\ \hat{\epsilon}_E(T_2) \\ \hat{\epsilon}_z(T_2) \\ \hat{R}_N(T_2) \end{bmatrix}\right)^T\right] \quad (5.2\text{-}16)$$

Replace the appropriate elements of the initial covariance of the states with the matrix C_{F1}. This indicates the covariance of the error states after the first fine align reset. Call this matrix P_{F1},

where

$$P_{F1} = \begin{bmatrix} C_{F1} & \\ & P_o \end{bmatrix} \quad (5.2\text{-}17)$$

The second fine align is similar to the first fine align.

Sampling interval is now $\Delta T = 28.8$ sec.

Total Time is $T_3 = 12$ min

Obtain the least squares coefficients associated with the measurements, i.e., $U_1, U_2, \ldots, U_{25}$. They are obtained from Eq. (5.2-12). Estimate

$$\begin{bmatrix} \hat{\epsilon}_N(T_3) \\ \hat{\epsilon}_E(T_3) \\ \hat{\epsilon}_z(T_3) \\ \hat{R}_N(T_3) \end{bmatrix} = \sum_{i=1}^{25} \frac{U_i}{\Delta T} \begin{bmatrix} V_N[(i+1)\Delta T] - V_N(i\Delta T) \\ V_E[(i+1)\Delta T] - V_E(i\Delta T) \end{bmatrix}$$

Using the covariance P_{F1} as the initial covariance, compute

$$C_{F2} = E \left[\left(\begin{bmatrix} \epsilon_N(T_3) \\ \epsilon_E(T_3) \\ \epsilon_z(T_3) \\ R_N \end{bmatrix} - \begin{bmatrix} \hat{\epsilon}_N(T_3) \\ \hat{\epsilon}_E(T_3) \\ \hat{\epsilon}_z(T_3) \\ \hat{R}_N(T_3) \end{bmatrix} \right) \left(\begin{bmatrix} \epsilon_N(T_3) \\ \epsilon_E(T_3) \\ \epsilon_z(T_3) \\ R_N \end{bmatrix} - \begin{bmatrix} \hat{\epsilon}_N(T_3) \\ \hat{\epsilon}_E(T_3) \\ \hat{\epsilon}_z(T_3) \\ \hat{R}_N(T_3) \end{bmatrix} \right)^T \right]$$

The square root of the diagonal elements of C_{F2} are the rms alignment errors after the second fine align.

The nominal system considered in Section 5.1 (Eqs. 5.1-1, 2, 3) has been used here for comparison of the least squares three-stage alignment technique and the Kalman filter approach. Below is a table of results. The results pertaining to Kalman filtering assume only two stages, - the crude stage and one fine stage where the sampling interval is constant over the length of the fine stage. For each acceleration environment, results are given for both 10 sec and 28.8 sec sampling intervals.

Results on the table indicate that the least squares technique yields errors which are slightly worse than the optimal results. The deterioration caused by using the suboptimal estimation scheme is small however. It is negligible in the case where acceleration is modelled as a 25 cps sinusoid. In the case where there is a random acceleration environment, the deterioration is considerably more. There is about 20% deterioration in tilt when there is a random acceleration. However, the absolute value of the resultant tilt error is so small that even the deteriorated accuracy is acceptable.

Run	Opt. or Lst. Sq. Align	Samp. Inter.	Stage	Total Time of Filt.	Accel. Environment	$\tilde{\epsilon}_N$ sec	$\tilde{\epsilon}_E$ sec	$\tilde{\epsilon}_z$ deg	$\tilde{R}_N$ deg/hr
1	Lst. Sq.	10 sec	2	4 min	.35g's	4.12	4.12	.256	.005
	Opt.	10 sec	2	4 min	25 cps	4.11	4.11	.255	.00173
	Lst. Sq.	28.8sec	3	15min		4.12	4.12	.254	.0005
	Opt	10 sec	2	15min		4.11	4.11	.254	.00018
	Opt	28.8sec	2	15min		4.11	4.11	.254	.00018
2	Lst. Sq.	10 sec	2	4 min	.1g	4.35	5.7	.312	.015
	Opt	10 sec	2	4 min	ran.	4.16	4.16	.254	.005
	Lst. Sq.	28.8sec	3	15min		5.2	5.2	.257	.008
	Opt	10 sec	2	15min		4.10	4.10	.251	.000328
	Opt	28.8sec	2	15min		4.3	4.3	.254	.0024
3	Lst. Sq.	10 sec	2	4 min	.1g	8.2	8.2	.435	.06
	Lst. Sq.	28.8sec	3	15min	12.611 cps	4.14	4.14	.258	.0029
	Opt	28.8sec	2	15min		4.11	4.11	.255	.0002

Table 5.2-1 Comparison of Optimal Alignment and Least Squares Alignment

The North gyro bias accuracy is considerably less than optimal. However, when least squares is used, the estimate is sufficient for alignment purposes. One should also remember that the optimal estimation error is probably overly optimistic because the time-varying C matrix has not been modelled in the system. If it were modelled, one can be reasonably sure that estimation of North gyro drift would be not much better than 10% of 0.05, or about 0.005°/hr.

5.2.4 Conclusions

If the time of alignment were to be fixed, either a three-stage least squares estimation procedure or a three-stage Kalman estimation procedure could be implemented with equal ease. All that would be necessary would be a priori specification of the weighting coefficients (U's) associated with the measurements. The values of the coefficients would be slightly different depending on the technique used in defining them. However, it is desirable to utilize an alignment scheme where the time of estimation in the last stage can be extended with little additional effort. Both Kalman filtering and least square estimation can be accomplished recursively for the case when additional data is available. They therefore both meet this additional specification. Also, with both techniques, a recursive equation can be implemented to specify the linear combination of the last best estimates and the present measurements which specify the present estimate. The least squares algorithms, however, require less computation and probably less memory space than the Kalman algorithms.* On the other hand, if batch processing techniques are to be used, least squares estimation requires a modest amount of additional computation to determine the weighting coefficients. However, converting the Kalman algorithms into batch processing coefficients requires more computation than can be easily accommodated in a flight computer.

Because of the general ease of computation involved in the least squares estimation algorithms when they must be implemented online, it has been decided that the least square approach should be used in the alignment scheme. The price paid for reduced accuracy does not appear to be critical.

* The dimension of the system is lower. Also in the least squares technique the Φ matrix can be computed analytically.

5.3 TWO-STAGE VERSUS THREE-STAGE LEAST SQUARES ALIGNMENT

As was previously stated, the concept of using a three-stage alignment originated from two lines of reasoning. The first originated from preliminary studies which were performed on a system where the measurements were considered to be the doubly integrated accelerometer outputs. A harsh environment was modelled. Such a measurement had the advantage that the estimation error was not affected by increasing the sampling interval, or by increasing the acceleration environment. It had the disadvantage that there appeared to be a P sensitivity, - an effect created by uncertainty in the initial covariance matrix. This problem was solved quite adequately by proceeding to a three-stage alignment. The results of this study are described in Appendix C.

The measurement scheme described above had the disadvantage that additional state: would be necessary in either least squares estimation algorithms or Kalman estimation algorithms. Therefore, it was decided to use averaged accelerometer outputs as measurements. The estimation accuracy then became sensitive to the sampling interval and to the acceleration environment. Considering the environments expected, the degradation of accuracy was not excessive. When the new measurement scheme was incorporated into an alignment scheme, the concept of three-stage alignment was carried over because of the confidence which had already been established in using it.

The second line of reasoning which favored a three-stage alignment is based on the fact that the error models describing the inertial system can be modelled as a linear system only when the errors are small.

By using a crude align and a short first fine align, one can be assured that on the second fine align the equations describing the error are linear. Thus the linear estimator will predictably estimate the error states.

In order to establish the actual need of the three-stage alignment when using the averaged accelerometer outputs as measurements, several additional covariance and simulation runs have been made for comparing a three-stage alignment to a two-stage alignment consisting of one crude and one fine align. The total filtering time is the same in both cases. The standard statistics given in Eqs. (5.1-1) and (5.1-2) are used as a basis. However, the initial heading error is assumed to be 10° and the acceleration environment is assumed 0.1g at 12.611 cps sinusoidal.

Run #	Situation	$\tilde{\epsilon}_N$ sec	$\tilde{\epsilon}_E$ sec	$\tilde{\epsilon}_z$ deg	$\tilde{R}_N$ deg/hr
1	3-stage (covariance)	4.14	4.19	.258	.0029
2	2-stage (covariance)	4.14	4.21	.258	.0029
3	3-stage (simulation)	3.8	2.9	.252	.00061
4	2-stage (simulation)	3.3	3.0	.256	.16

Figure 5.3-1 Comparison of 2- and 3-Stage Least Squares Alignment Results

Note that with this measurement scheme the covariance analysis indicates only a minute degradation of accuracy in going from a three-stage alignment to a two-stage alignment. Thus it appears that the P sensitivity observed using the doubly integrated accelerometer outputs is not observed here. This is probably explicable for two reasons. (1) The environment used is not as severe in the present results, and (2) doubly integrated measurements may have unusually high values for significant tilt angles. The estimator then can be represented by the addition and subtraction of large numbers. Although the estimator might do well on a percentage basis, it is not accurate on an absolute value basis. Intermediate reset is necessary. On the other hand, averaged accelerometer outputs never become very large.

The simulation results indicate that only North gyro drift is deteriorated in the two-stage alignment. This indicates the effect of non-linearity.

DRC still recommends the three-stage alignment because it is necessary to obtain a reliable estimate of North gyro drift rate and it allows for a safety factor in the case when there may be more severe environments.

6. ALIGNMENT EQUATIONS

6.1 INITIALIZATION

Input:

H	heading angle

Assumed known quantities:

θ

C_b^h (see Section 4.1)

Output:

$$C_b^{n-} = \begin{bmatrix} \cos\theta \cos H & -\sin H & \sin\theta \cos H \\ \cos\theta \sin H & \cos H & \sin\theta \sin H \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} C_b^h \qquad (6.1)$$

6.2 LEVEL ALIGN

Input: Average of ΔV (accelerometer outputs) to give acceleration

$$\begin{bmatrix} A_n \\ A_e \\ A_z \end{bmatrix} = C_b^{n-} \begin{bmatrix} \frac{1}{N\Delta t} \sum_{i=1}^{N} \Delta V_x(i) \\ \frac{1}{N\Delta t} \sum_{i=1}^{N} \Delta V_y(i) \\ \frac{1}{N\Delta t} \sum_{i=1}^{N} \Delta V_z(i) \end{bmatrix} \qquad (6.2)$$

Output:

$$C_b^{n+} = \begin{bmatrix} 1 - (1+UNT_3)\dfrac{UNT_1^2}{UNT_1^2+UNT_2^2} & -(1+UNT_3)\dfrac{UNT_1 UNT_2}{UNT_1^2+UNT_2^2} & UNT_1 \\ -(1+UNT_3)\dfrac{UNT_1 UNT_2}{UNT_1^2+UNT_2^2} & 1-(1+UNT_3)\dfrac{UNT_2^2}{UNT_2^2+UNT_2^2} & UNT_2 \\ -UNT_1 & -UNT_2 & -UNT_3 \end{bmatrix} C_b^{n-} \tag{6.3}$$

where

$$\begin{bmatrix} UNT_1 \\ UNT_2 \\ UNT_3 \end{bmatrix} = \begin{bmatrix} A_n/\sqrt{A_n^2 + A_e^2 + A_z^2} \\ A_e/\sqrt{A_n^2 + A_e^2 + A_z^2} \\ A_z/\sqrt{A_n^2 + A_e^2 + A_z^2} \end{bmatrix} \tag{6.4}$$

6.3 FINE ALIGN - STAGE I

Only the misalignment angles need to be estimated since the DCM is reset and the biases are not compensated. The following computations must be performed for batch processing techniques.

$$\begin{bmatrix}\hat{\epsilon}_N\\ \epsilon_E\\ \epsilon_z\end{bmatrix}_{tf_1} = D_1(tf_1)\,[GGI_1]^{-1}\sum_{i=1}^{NA_1} E_1^T\left(\Delta T_1 i - \frac{\Delta T_1}{2}\right)\begin{bmatrix}Z_N(i)\\ Z_E(i)\end{bmatrix} \qquad (6.5)$$

where

$$\Delta T_1 = \frac{tf_1}{NA_1}$$

tf_1 is total time for Stage I

NA_1 is number of data vectors in the estimate

$\begin{bmatrix}Z_N(i)\\ Z_E(i)\end{bmatrix}$ is the average indicated acceleration resolved in the $\tilde{n}$ frame over the time interval $\Delta T_1 i-1 \leq t \leq \Delta T_1 i$.

$s = \sin L$

$c = \cos L$

$$GGI_1 = \sum_{i=1}^{NA_1} E_1^T\left(\Delta T_1 i - \frac{\Delta T_1}{2}\right) E_1\left(\Delta T_1 i - \frac{\Delta T_1}{2}\right) \qquad (6.6)$$

$$E_1(t) = g\begin{bmatrix} s\cdot\sin w_{ie}t & \cos w_{ie}t & c\cdot\sin w_{ie}t & -\dfrac{s(1-\cos w_{ie}t)}{w_{ie}} \\ -s^2\cos w_{ie}t - c^2 & s\cdot\sin w_{ie}t & -sc(\cos w_{ie}t-1) & \dfrac{s^2\sin w_{ie}t}{w_{ie}} + c^2 t \end{bmatrix} \qquad (6.7)$$

$$D_1(t) = \begin{bmatrix} s^2\cos w_{ie}t + c^2 & -s\,\sin w_{ie}t & sc(\cos w_{ie}t - 1) & \dfrac{-s^2\sin w_{ie}t}{w_{ie}} - c^2 t \\ s\,\sin w_{ie}t & \cos w_{ie}t & c\,\sin w_{ie}t & -s\dfrac{(1-\cos w_{ie}t)}{w_{ie}} \\ sc(\cos w_{ie}t - 1) & -c\sin w_{ie}t & c^2\cos w_{ie}t + s^2 & -sc\left[\dfrac{\sin w_{ie}t}{w_{ie}} - t\right] \end{bmatrix} \tag{6.8}$$

A number of computational techniques can be used to perform the above calculations. These are discussed in Sec. 8.

At the end of Stage I, the direction cosine matrix can be reset using the following computation

$$C_b^{\tilde{n}} = \begin{bmatrix} \cos\hat{\epsilon}_z & -\sin\hat{\epsilon}_z & \hat{\epsilon}_y \\ \sin\hat{\epsilon}_z & \cos\hat{\epsilon}_z & -\hat{\epsilon}_x \\ -\hat{\epsilon}_y & \hat{\epsilon}_x & 1 \end{bmatrix} C_b^{\tilde{n}} \tag{6.9}$$

The resulting matrix is made orthonormal in the direction cosine update routine.

Output:

Reset DCM

Input:

Depends on computational technique (see Sec. 8)

6.4 FINE ALIGN - STAGE II

At the end of the second stage of fine align, the following computations are performed from the filter estimates.

Output:

- Reset direction cosine matrix
- Compensate north component of gyro bias
- Compensate vertical component of accelerometer bias

$$\begin{bmatrix} \hat{\epsilon}_N \\ \epsilon_E \\ \epsilon_z \\ R_N \end{bmatrix}_{tf_2} = D_2(tf_2)(GGI_2)^{-1} \sum_{i=1}^{NA_2} E_1^T \left(\Delta T_2 i - \frac{\Delta T_2}{2}\right) \begin{bmatrix} Z_n(i) \\ Z_E(i) \end{bmatrix} \qquad (6.10)$$

$$\hat{A}_z = \frac{1}{NA_2} \sum_{i=1}^{NA_2} (Z_z(i) + g) \qquad (6.11)$$

where

$$\Delta T_2 = \frac{tf_2}{NA_2}$$

tf_2 is total time for Stage II

NA_2 is number of data vectors in the estimate

$\begin{bmatrix} Z_{N(i)} \\ Z_{E(i)} \\ Z_{t(i)} \end{bmatrix}$ is the average indicated acceleration resolved in the $\tilde{n}$ frame over the time interval $\Delta T_2\ i-1 \le t < \Delta T_2\ i$

$s = \sin L$

$c = \cos L$

$$GGI_2 = \sum_{i=1}^{NA_2} E_1^T(\Delta T_2 i - \frac{\Delta T_2}{2})\, E_1(\Delta T_2 i - \frac{\Delta T_2}{2}) \tag{6.12}$$

$$D_2(t) = \begin{bmatrix} s^2 \cos w_{ie} t + c^2 & -s \sin w_{ie} t & s\,c(\cos w_{ie} t - 1) & \frac{s^2 \sin w_{ie} t}{w_{ie}} - c^2 t \\ s \sin w_{ie} t & \cos w_{ie} t & c \sin w_{ie} t & \frac{-s(1 - \cos w_{ie} t)}{w_{ie}} \\ sc(\cos w_{ie} t - 1) & -c \sin w_{ie} t & c^2 \cos w_{ie} t + s^2 & -sc \frac{\sin w_{ie} t}{w_{ie}} - t \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6.13}$$

R_N is north component of gyro bias

A_z is vertical component of accelerometer bias

The computational alternatives of the above equations are discussed in Sec. 8.

At the end of Stage II the direction cosine matrix is reset

$$\tilde{C}_b^n = \begin{bmatrix} \cos \hat{\epsilon}_z & -\sin \hat{\epsilon}_z & \hat{\epsilon}_y \\ \sin \hat{\epsilon}_z & \cos \hat{\epsilon}_z & -\hat{\epsilon}_x \\ -\hat{\epsilon}_y & \hat{\epsilon}_x & 1 \end{bmatrix} \tilde{C}_b^n \tag{6.14}$$

The estimate of the north gyro bias component in b axes is given as

$$\begin{bmatrix} \hat{R}_x \\ R_y \\ R_z \end{bmatrix} = \begin{bmatrix} \tilde{C}_b^n \end{bmatrix}^T \begin{bmatrix} \hat{R}_N \\ 0 \\ 0 \end{bmatrix} \tag{6.15}$$

The three gyros are rebiased using the above components.

The estimate of the vertical component of accelerometer bias in b axes is given as

$$\begin{bmatrix} \hat{A}_{bx} \\ A_{by} \\ A_{bz} \end{bmatrix} = \begin{bmatrix} \tilde{C}_b^n \end{bmatrix}^T \begin{bmatrix} 0 \\ 0 \\ \hat{A}_z \end{bmatrix} \tag{6.16}$$

and the three accelerometers are rebiased using the above components.

7. COMPUTER SIMULATION RESULTS AND TRADE-OFF STUDIES

Trade-off studies are presented in this section based on results obtained using the digital computer simulation program described in Appendix A. Of particular interest are the effects of sensor quantization and calibration accuracy on alignment time and accuracy. The effect of computer word length (single precision versus double precision) is discussed later in the next section along with the estimated computer requirements.

The results presented here are for the 3-stage alignment scheme which begins with a coarse alignment and ends with two stages of fine alignment. The reasons for choosing this 3-stage alignment scheme over a two-stage scheme were discussed earlier in Section 5.3. The different results obtained by these two schemes using the computer simulation are now discussed.

7.1 TWO-STAGE VERSUS THREE-STAGE ALIGNMENT

A computer simulation study was performed using the nominal set of input parameters described in Table 7.2-1 of the next subsection. However, there was one exception to this nominal set. The initial azimuth tilt error was raised from 2 degrees to 10 degrees in order to induce nonlinear effects into the results.

Two simulations were run using identical input parameters: one using two stages of alignment and one using three stages of alignment.

By the end of 920 seconds there was no appreciable difference between the platform tilt errors of the two runs. However, there was a substantial difference in the bias estimates produced by the two schemes. This difference is illustrated in Fig. 7.1-1 where the estimated biases of the north component of gyro drift rate and the vertical component of accelerometer bias are plotted as a function of the alignment time. The estimated biases using the two-stage scheme are shown as dashed lines, while the estimated biases using the three-stage scheme are shown as solid lines.

The biases values inserted into the simulation were 0.05 deg/hr for the gyro and 20 μg for the accelerometer. It is seen from Fig. 7.1-1 that by the middle of the second stage of fine align the three-stage scheme has closed in on the correct bias values. On the other hand, the two-stage scheme using only one stage of fine align has deviated substantially from the correct values.

Thus the simulation results have presented us with another reason for selecting a three-stage alignment over a two-stage scheme. The difference noted here was not obtained when using the linear covariance program. Thus, the difficulty that the two-stage alignment scheme had in estimating the biases is attributed to the nonlinearities inherent in the computer simulation.

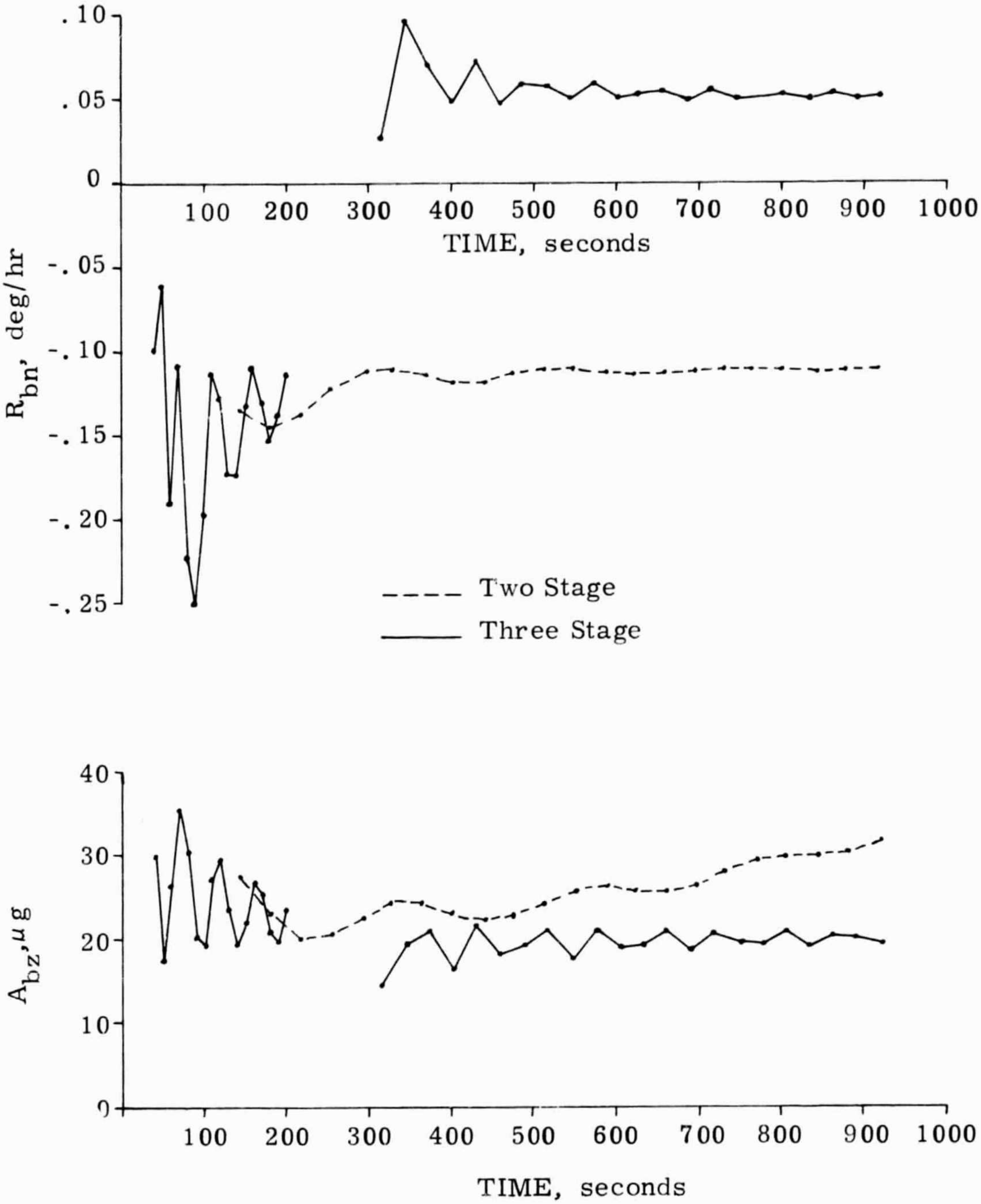


Figure 7.1-1 Difference in Bias Values Estimated by the Two Stage and Three Stage Alignment Schemes.

7.2 RESULTS FOR THE NOMINAL CASE

The input parameters for the nominal case are listed in Table 7.2-1. The nominal case is the one which is most likely to represent the operating environment. The effect of other possible environments on the alignment results is discussed in the trade-off studies presented in Sections 7.3 and 7.4.

The input parameters that were most difficult to define for this analysis were those associated with the vibrational motion (rotation and translation). A separate study being run concurrently with this one was to provide us with a vibrational spectrum based on actual measurements taken aboard the helicopter. However, the full data processing of these measurements was not completed in time, and instead DRC was instructed to perform its study with a simple sinusoidal model. Preliminary analysis of the measurements taken aboard the helicopter indicated that the vibrational spectrum was peaking about 25 hz.

To simulate this frequency an update interval of approximately 0.02 seconds is required. The inverse of the product (.02)(25) produces two samples per cycle. In order to decrease the running time for the computer simulation, it was agreed that the frequency used for the vibrational motion model could be (1/2) 25 or 12.5 hz. This would allow a doubling of the update interval to 0.04 seconds, which cuts the computer running time approximately in half.

Latitude	42°
Gravity	32.17 ft/sec^2
Gyro Biases	0.05°/hr
Accelerometer Biases	20 μg
Initial Estimates	2° about each axis
Accelerometer Quantization	.0025 ft/sec
Gyro Quantization	.84 $\widehat{\text{sec}}$
Translational Vibration	.1g along each axis
Rotational Vibration	1°/sec about each axis
Course (level) Align Time	20 sec
Stage I Fine Align	200 sec with 20 samples
Stage II Fine Align	720 sec with 25 samples

Initial true b axes along North, East, Vertical

Table 7.2-1 Input Parameters for Nominal Case

In Run 1 for the nominal case the update interval was 0.04 seconds and the vibrational motion was modeled by a sinusoid with a frequency of 12.5 hz. The platform tilt errors for this run are graphed in Fig. 7.2-1 as a function of time. For reference and comparison, also shown are the tilt errors which occur when there is no quantization error and no vibrational motion. Under this no-noise environment, the tilt errors immediately achieve their theoretical limits: the accelerometer biases in level and the east pointing gyro bias divided by earth rate times cos L in azimuth. A visual inspection of Fig. 7.2-1 reveals that the results of Run 1 are not much different from the theoretical limits. The reason for these overly optimistic results is the uniform sampling of the vibratory motion sinusoid from cycle to cycle. With identical sampling in each cycle the disrupting effect of this sinusoid on the accelerometer measurements will tend to be eliminated when these measurements are averaged. Since this would not be the case if the vibratory motion were being modeled in a more sophisticated manner (i.e., a spectrum rather than a simple sinusoid), the frequency of the sinusoid was changed to 12.611 hz to avoid identical sampling during each cycle of the sinusoid. The results of this change (identified as Run 2) on the level platform tilts are shown in Fig. 7.2-2. Here it is seen that the results of Run 2 are considerably different from the theoretical limits during the initial portion of alignment, but they tend to approach the theoretical limit near the end of the second stage of fine alignment.

The phase of the 12.611 hz sinusoid with respect to the 0.04 updating intervals is taken from a set of random numbers generated from

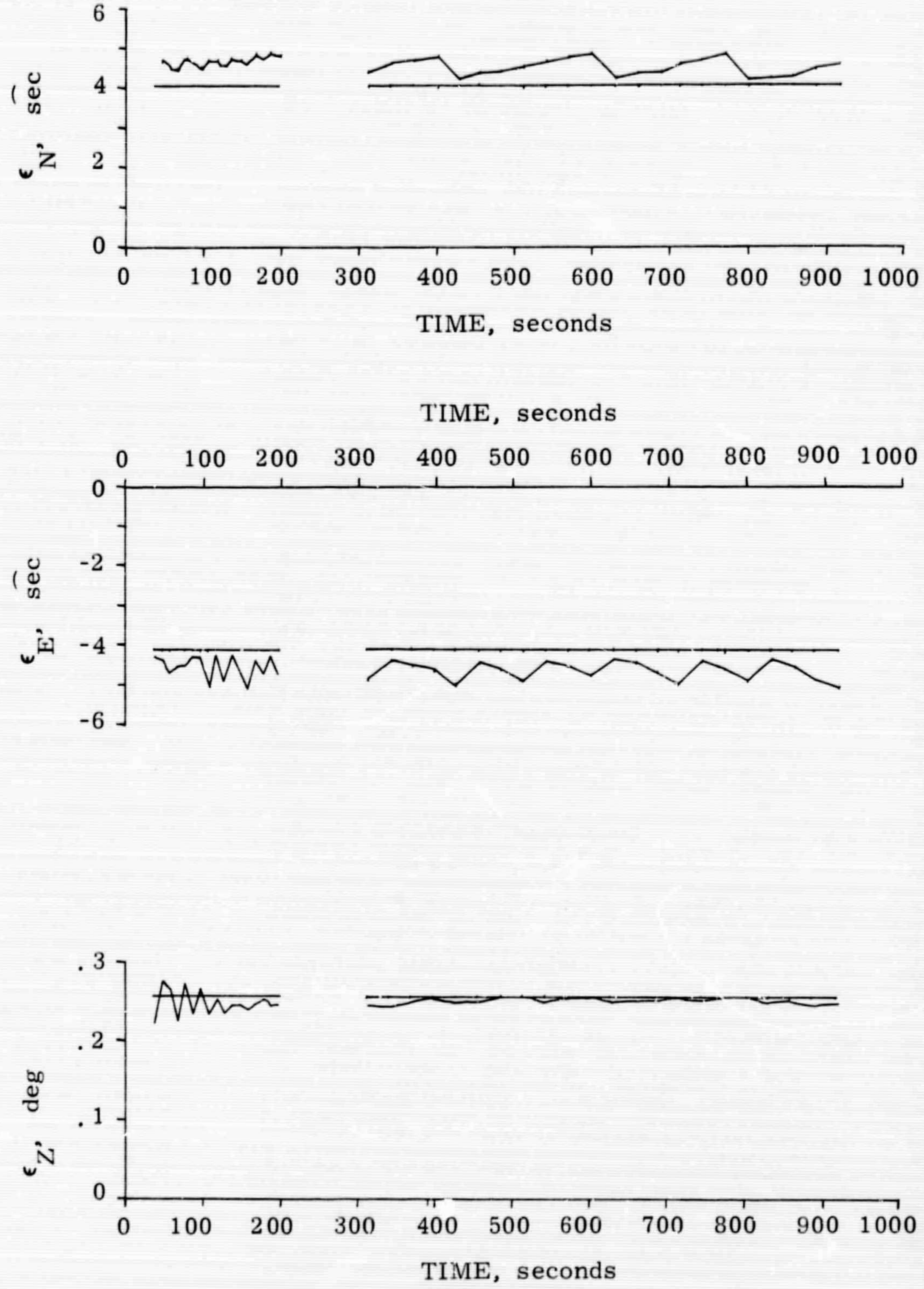


Figure 7.2-1 Platform Tilt Errors Versus Time
Nominal Case - Run 1

a uniform distribution. Another simulation run with a different phase angle for the 12.611 hz sinusoid was performed, and the level tilt error results for this run (designated Run 3) are presented in Fig. 7.2-3. In Run 3 the initial level tilt errors are about twice as large as they were in Run 2, but they still approach their theoretical limit by the end of the second stage of fine alignment.

The azimuth tilt errors for Runs 2 and 3 are presented in Fig. 7.2-4. In both runs the initial azimuth tilt error was close to 3 degrees, but again the tilt error was approximately that of the theoretical limit by the end of the second fine alignment stage. Actually, in this case the theoretical limit was essentially achieved after 500 seconds of alignment time.

The results of estimating the effective bias levels in the Z (vertical) accelerometer and north gyros are graphed in Fig. 7.2-5. The horizontal lines at 20 μg for the accelerometer and 0.05 deg/hr for the gyro represent the results for the no quantization, no vibrational motion case. The solid line curves for Run 1 are extremely close to those for the no noise case, and in fact, the reader will find it difficult to distinguish the Run 1 curves from the horizontal lines in parts of the graph. The dashed line curves for Run 2 represent the results that are more likely to occur. The bias estimates for Run 2 are poor during the first stage of fine alignment, but become quite good by the end of the second stage of fine alignment. For the gyro, the initial bias estimate after 40 seconds of alignment time is -0.8 deg/hr, which is well off the graph. The curve goes off the graph

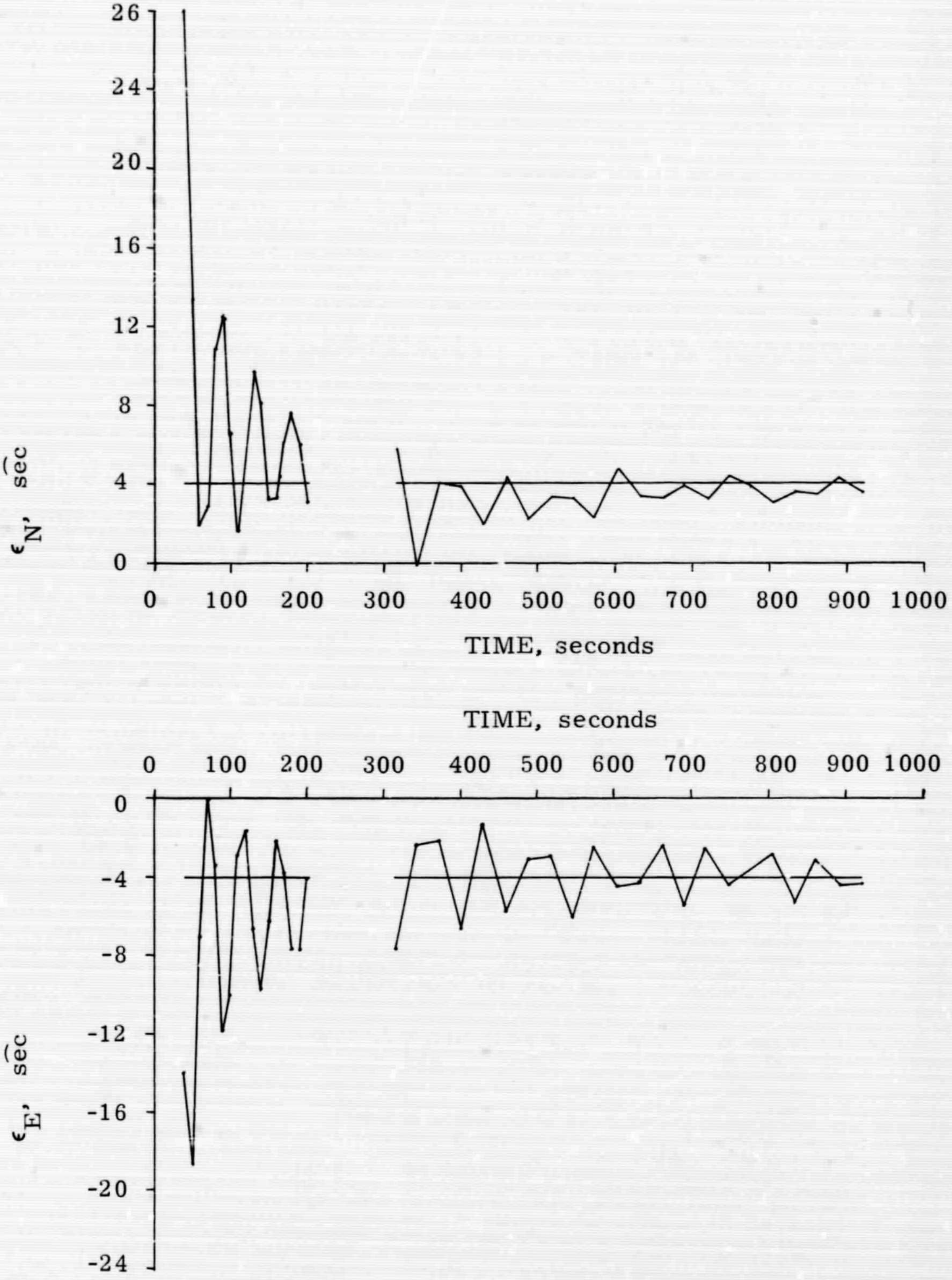


Figure 7.2-3 Level Tilt Errors Versus Time
Nominal Case - Run 3

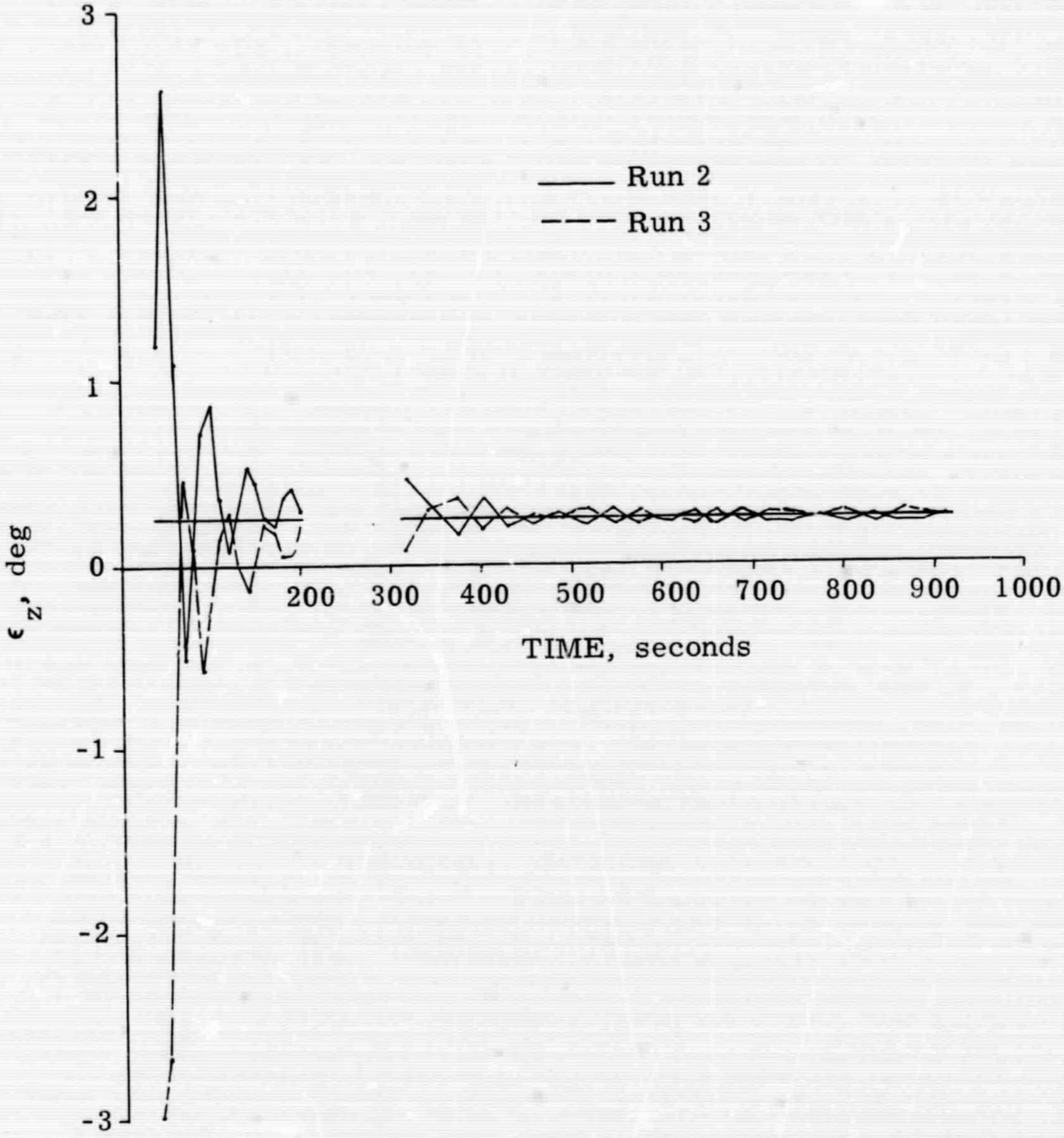
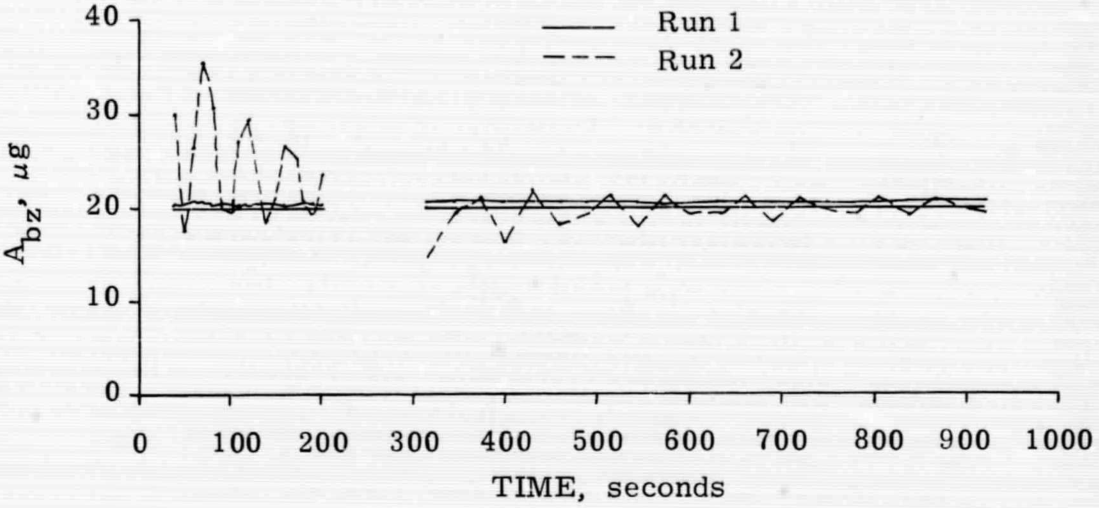
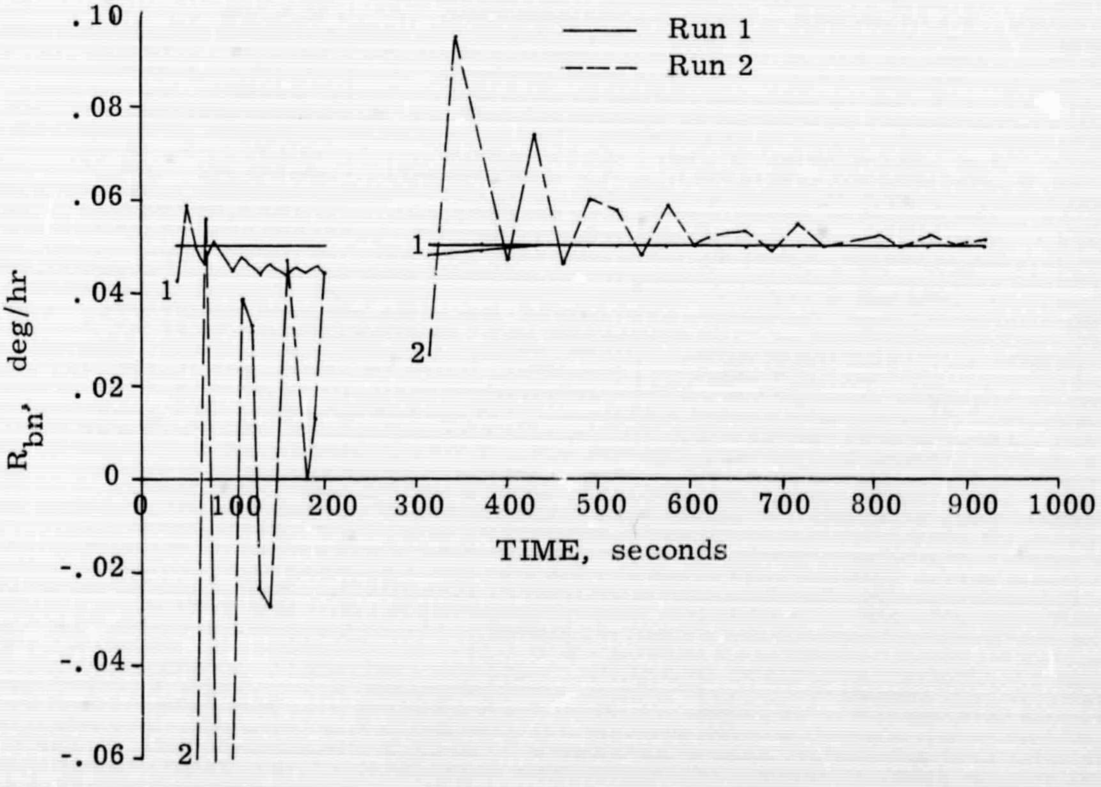


Figure 7. 2-4 Azimuth Tilt Errors Versus Time Nomina' Case-Runs 2 and 3.

Figure 7.2-5 Running Estimates of the Vertical Component of Accelerometer Bias and the North Component of Gyro Bias for the Nominal Case





again at 90 seconds, the bias estimate here being -0.14 deg/hr. Thus, with the two stages of fine alignment the relative improvement of the gyro bias estimate during the alignment period is most dramatic.

7.3 EFFECT OF VARYING INERTIAL INSTRUMENT QUANTIZATION LEVELS ON PLATFORM TILT ERRORS

In the nominal case, the inertial instrument quantization levels were:

0.84 $\widehat{\text{sec}}$ for the gyros

0.0025 ft/sec for the accelerometers.

In this subsection these quantization levels are increased in order to observe the degradation which takes place in the alignment.

Accelerometer

The quantization levels in the accelerometers were raised to 0.25 ft/sec, which is 100 times greater than that used in the nominal case. The effects of this increase on the platform tilts $\epsilon_N, \epsilon_E, \epsilon_z$ are graphed in Figs. 7.3-1, 7.3-2, and 7.3-3, respectively. For comparison purposes, the corresponding tilt errors for the nominal case are plotted on these graphs in dashed lines.

It can be seen that the level tilt errors during the first stage of fine alignment have been greatly effected by the increase in the quantization level. However, by the end of the second stage of fine alignment, the level tilt errors are only slightly higher than in the nominal case.

The increased quantization produced an increased azimuth tilt error during the first stage, but its degrading effect was negligible by the end of the second stage.

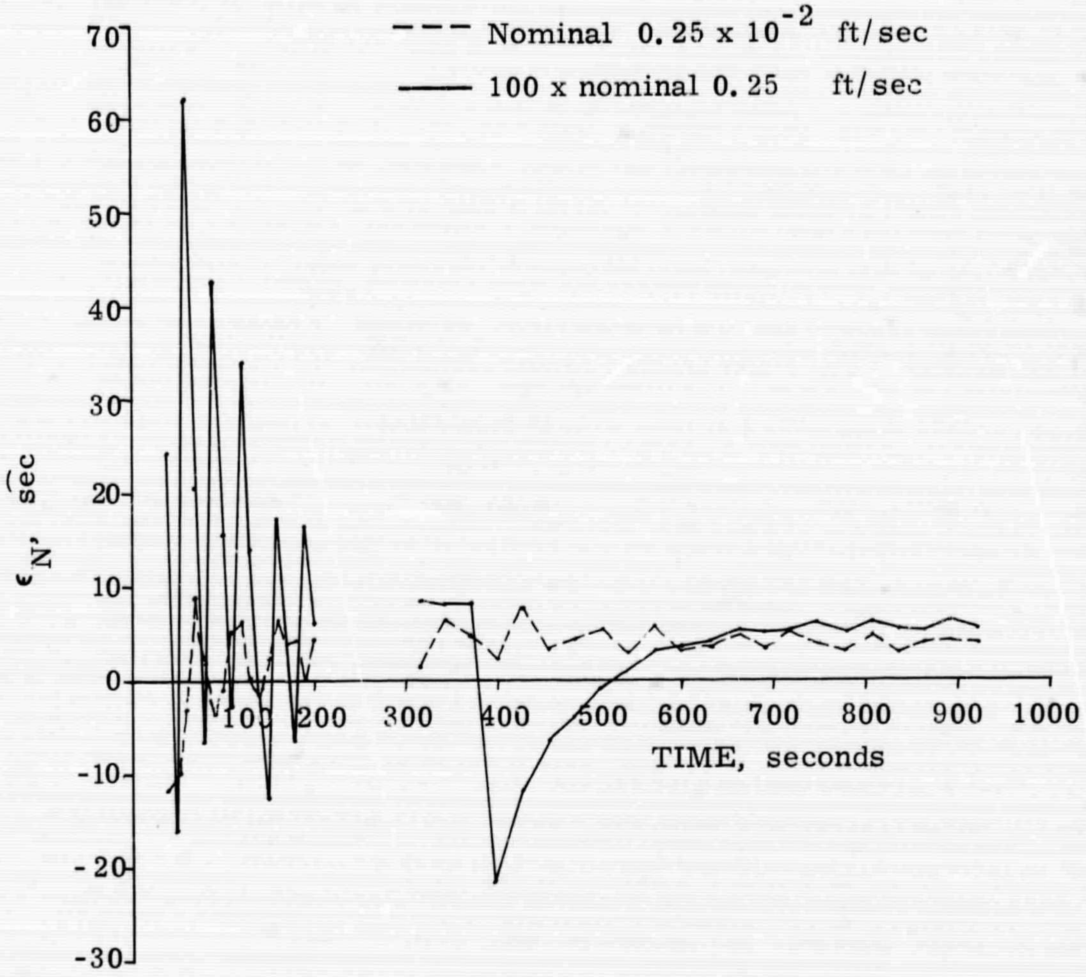


Figure 7.3-1 Effect of Increasing the Accelerometer Quantization Level on the North Platform Tilt Error

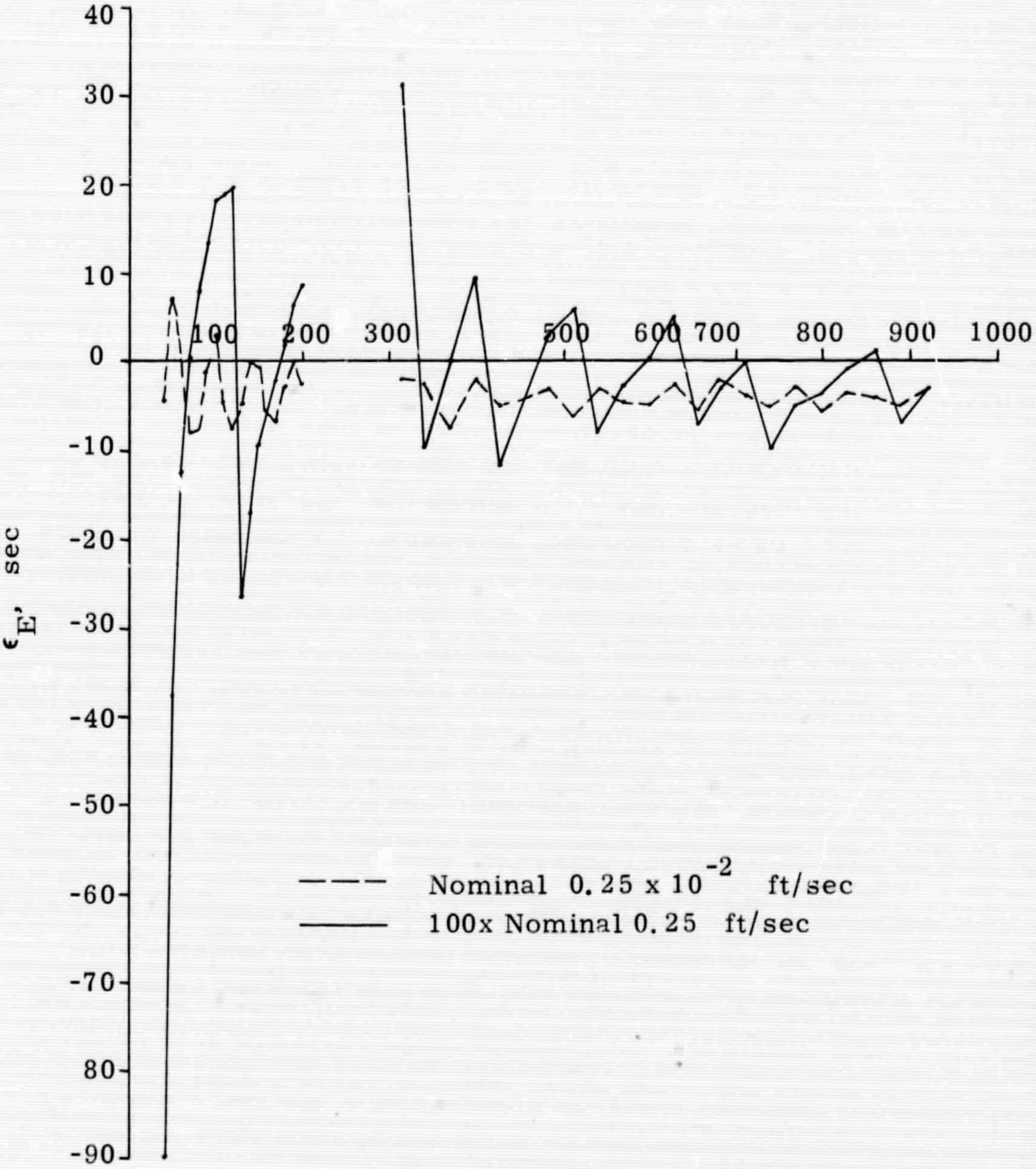


Figure 7.3-2 Effect of Increasing the Accelerometer Quantization Level on the East Platform Tilt Error

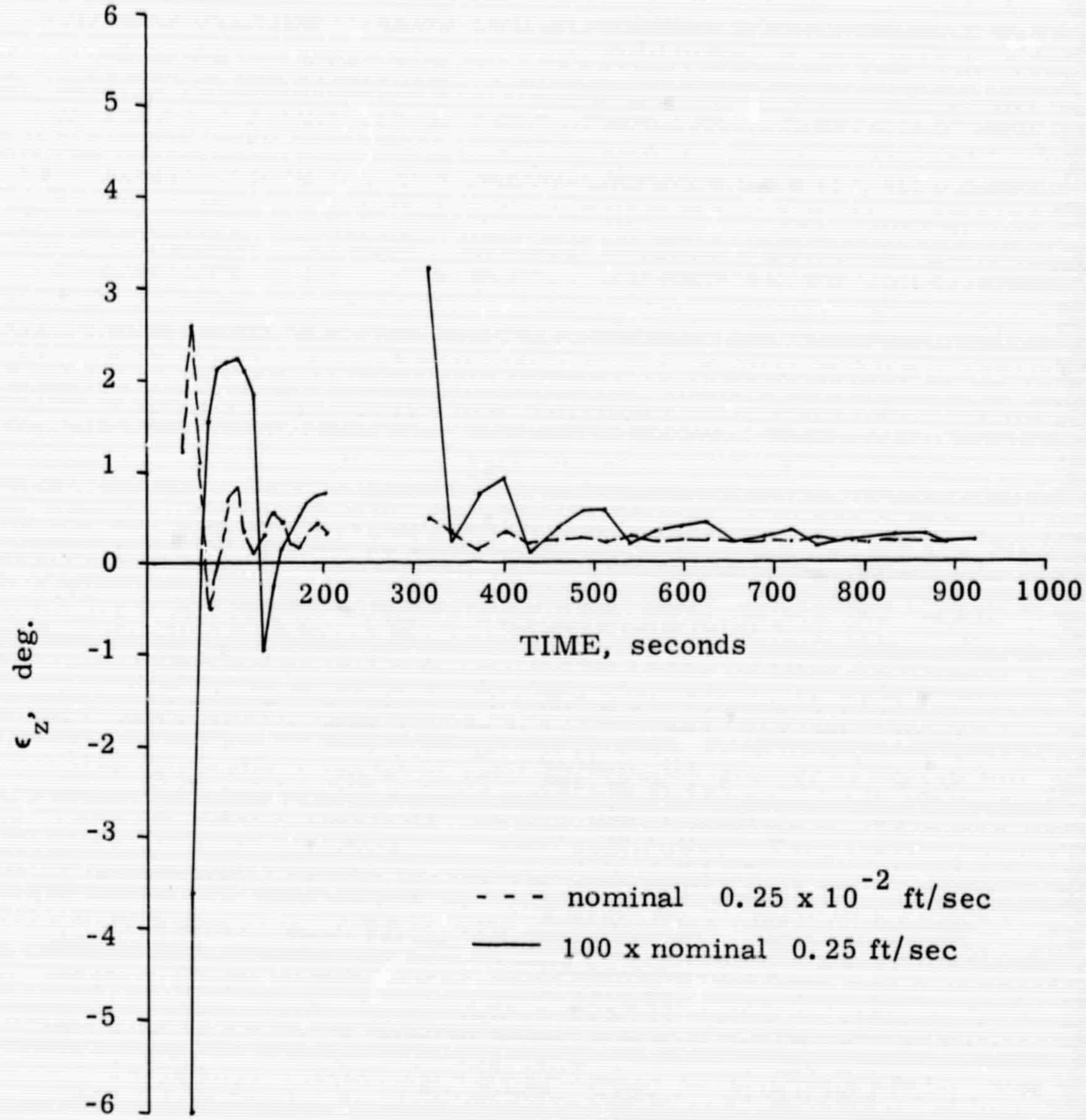


Figure 7.3-3 Effect of Increasing the Accelerometer Quantization Level on the Azimuth Platform Tilt Error

These simulation results are in general agreement with results obtained from the covariance error analysis program. A comparison of the simulation and covariance results are presented in Table 7.3-1 for the end points of the two stages of fine alignment.

	ϵ_N ($\widehat{\text{sec}}$)	ϵ_E ($\widehat{\text{sec}}$)	ϵ_z deg
End of First Stage: Covariance	13.3	13.3	0.67
Simulation	6.1	8.3	0.78
End of Second Stage: Covariance	4.9	5.0	0.26
Simulation	5.8	3.0	0.27

Table 7.3-1 Comparison of Covariance Results With One Simulation Run

Any interpretation of the results in Table 7.3-1 must take into account that the covariance results represent an rms error over some large ensemble, whereas the simulation result is only for one run. Time and cost considerations did not allow a full Monte Carlo type of analysis to be performed using the simulation; this type of study would have generated rms errors which could have been compared directly to the covariance rms errors.

Gyro

The quantization levels in the gyros were raised to 16.8 $\widehat{\text{sec}}$, a value which is 20 times greater than that used in the nominal case. The effect of this increase on the level platform tilts is illustrated in Fig. 7.3-4.

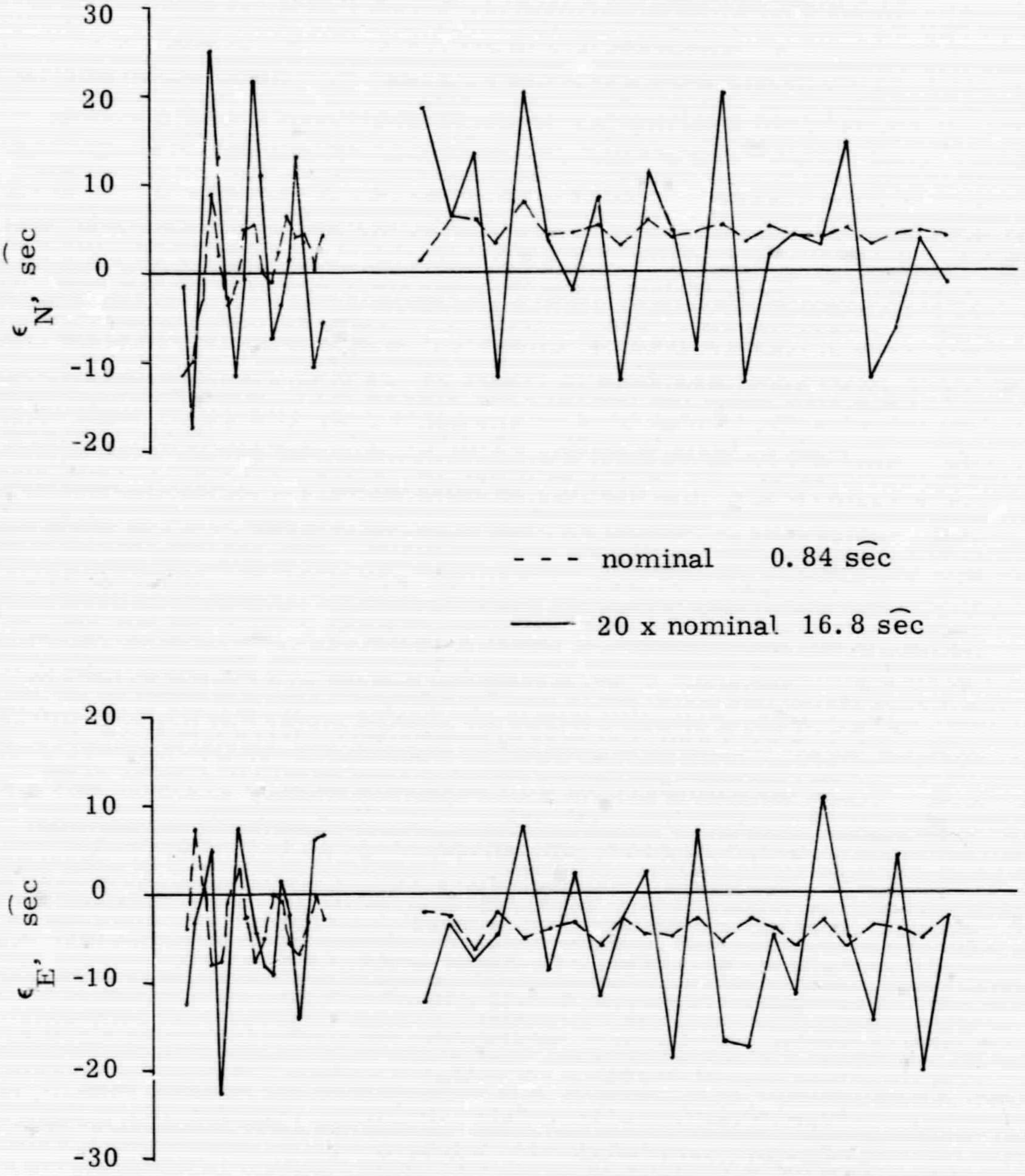


Figure 7. 3-4 Effect of Increasing the Gyro Quantization Level On the Level Platform Tilt Errors

For comparison purposes, the results for the nominal case are plotted on these graphs using dashed lines. A visual inspection reveals that the increased quantization level affected the level alignment errors significantly during both stages of fine alignment. This is to be expected since the magnitude of the quantization level essentially acts as a white noise addition to the tilt errors. If more sample points had been taken during the run, the curves would have appeared more noisy than they do.

The effect of this increased quantization level on the azimuth tilt error was negligible since the additive 20 $\widehat{\text{sec}}$ noise was insignificant compared to the 0.26 deg error in azimuth tilt which existed prior to increasing the quantization level. Thus the azimuth tilt error curve for the increased quantization level is essentially identical to the one already shown for Run 2 in Fig. 7.2-4, and another graph is unnecessary.

Again the simulation results are in general agreement with those from the covariance program. The covariance program indicates that with a 20 times increase in gyro quantization level, there is approximately a 50% increase in the level tilt errors; i.e., an error increase from 4.3 $\widehat{\text{sec}}$ rms to 6.4 $\widehat{\text{sec}}$ rms at the end of the second stage of fine alignment. However, there is no increase at all in the rms azimuth tilt error.

7.4 ALIGNMENT ACCURACY VERSUS CALIBRATION ACCURACY

In the nominal case the alignment accuracy by the end of the second stage of fine alignment was approximately equal to the theoretical limits. These theoretical limits are governed by the level accelerometer and east gyro equivalent bias accuracies. The assumed calibration accuracies of 20 μg for the accelerometer biases results in level alignment errors of

4.1 $\widehat{\text{sec}}$. The assumed calibration accuracy of 0.05 deg/hr for the equivalent east gyro results in an azimuth alignment error of 0.26 deg. For any change in these bias accuracies, there will be a proportional change in the resulting tilt errors. The resulting straight line accuracy curves are presented in Fig. 7.4-1.

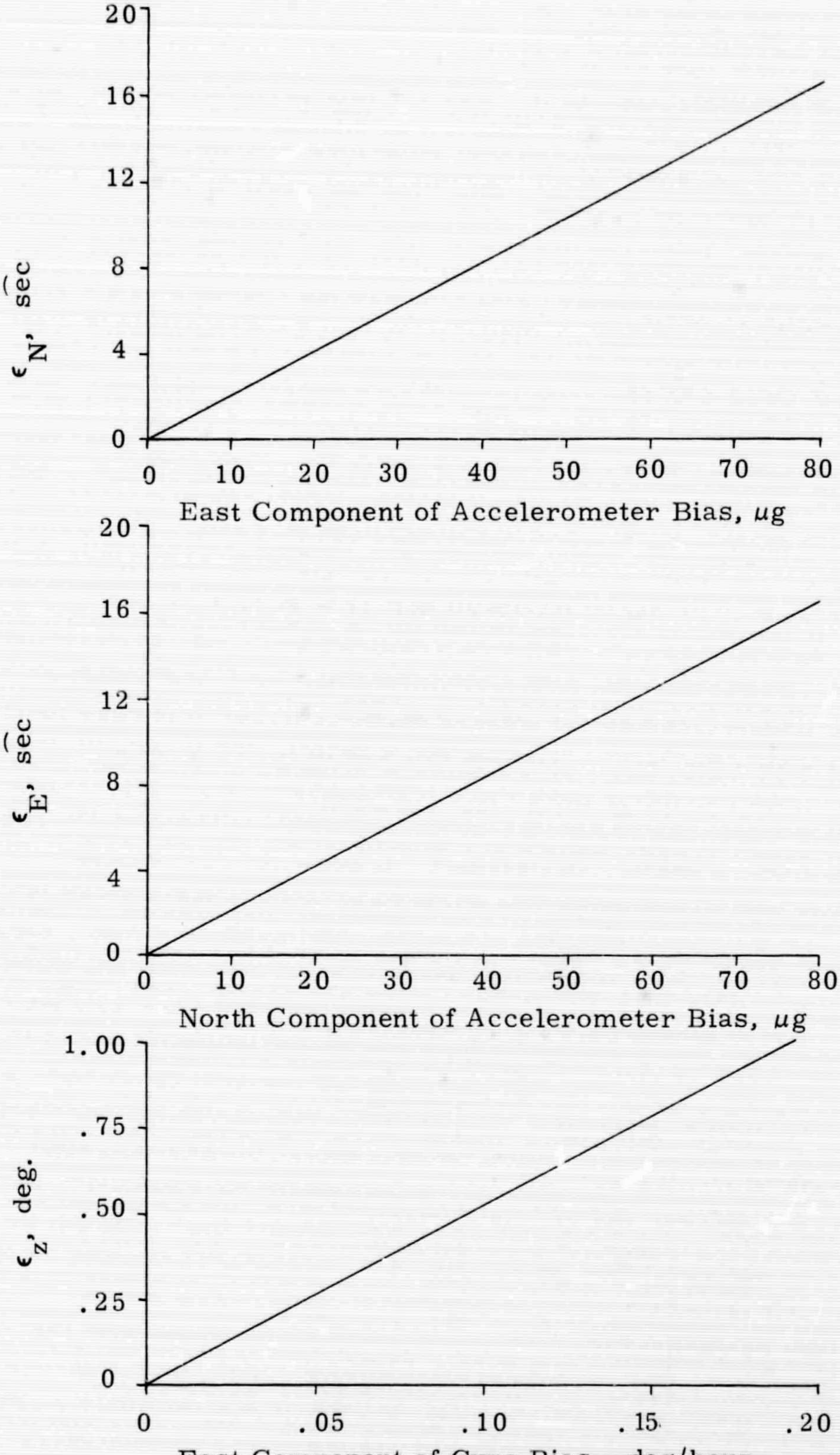


Figure 7.4-1 Alignment Accuracy Versus Calibration Accuracy

8. ESTIMATED COMPUTER REQUIREMENTS

The following information is available on the 4 PI computer to be used on the V/STOL helicopter.

Single precision (SP)	15 bits and sign bit
Double precision (DP)	31 bits and sign bit

(μs microseconds)

DP	ADD	3.8 μs
DP	MULT	11.5 μs
DP	DIV	46.3 μs
SP	LOAD	3.75 μs
DP	LOAD	5.00 μs
SP	STORE	4.58 μs
DP	STORE	5.83 μs
SHIFT		1.88 + 1.2N μs
	N is the number of bits shifted	
SIN		244 μs
COS		230 μs
SQRT		250 μs

Single precision numbers are good to 1 part in $2^{15}-1$ or 1 part in 32767. Double precision numbers are good to 1 part in $2^{31}-1$ or 1 part in 2,147,483,647.

Initialization

The accuracy of the computations should be compatible with the accuracy of the input and output numbers. For instance H, (the heading angle) has a range of $0 \leq H \leq 360°$. The initial estimate of H has an expected accuracy of at least ±2° (resolution). Hence H can be stored in a word length good to 1 part in 180. In this case single precision will do well.

		Required			Memory	Time
	Range	Resolution	SP	DP	words	µs
θ	±10°	±1°	x			
SIN θ			x		1	
COS θ			x		1	

Sin θ and Cos θ can be precomputed and stored in single precision words.

H (heading)	0 to 360°	±2°	x		1	
H in radians			x		1	11.5
SIN H			x		1	244
COS H			x		1	230

Note that all the elements of a direction cosine matrix have a range $-1 \leq C_{ij} \leq 1$. The resolution given by single precision words is compatible with the accuracy of the input and output quantities for the initialization.

The computation of C_b^{n-} (see Eq. (6.1) requires

		Time μs
MULT	28	322
ADD	15	57
LOAD	37	138.75
STORE	37	169.46
		687.21 μs

Memory required - 27 words and sin, cos subroutine.

Level Align

	Range	Required Resolution	SP	DP
A_n	±.3g	3×10^{-4}g	x	
A_e	±.3g	3×10^{-4}g	x	
A_z	±1.5g	3×10^{-4}g	x	

Range A_n = .2 rad x 1.5g = .3g

The level align is basically an averaging scheme. Hence even with perfect accelerometers, the estimate of level will be off due to rotational motion. The amount can be calculated from the model of rotational motion. The model is

$$w_x = w_0 \sin (wt + \phi)$$

where

$$w_0 = 1°/\text{sec}$$

$$w = 25 \text{ cps}$$

hence

$$\theta = -\theta_0 \cos(wt \times \phi)$$

where

$$\theta_0 = \frac{1°/\text{sec}}{2\pi \cdot 25} \approx 1 \times 10^{-4} \text{rad}$$

$$\frac{.2 \times 1.5g \text{ (range)}}{1 \times 10^{-4} \times 1.5g \text{ (resolution)}} \approx .2 \times 10^{4} = 2 \times 10^{3}$$

Hence single precision is consistent with the above result.

	Range	Required Resolution	SP	DP
UNT_1	±.2	1×10^{-4}	x	
UNT_2	±.2	1×10^{-4}	x	
UNT_3	±1.0	1×10^{-4}	x	

	Mult	Time	Add	Time	Div	Time	Sqrt	Time	Load	Time	Store	Time
UNT_1	6	69	3	11.4	1	46.3	1	250	11	41.25	11	50.38
UNT_2	6	69	3	11.4	1	46.3	1	250	11	41.25	11	50.38
UNT_3	6	69	3	11.4	1	46.3	1	250	11	41.25	11	50.38

Total Time 1404.99 μs

	Range	Required Resolution
$\dfrac{UNT_1^2}{UNT_2^2 + UNT_1^2}$	0 to 1	1×10^{-4}
$\dfrac{UNT_1 \; UNT_2}{UNT_1^2 + UNT_2^2}$	± 1/2	1×10^{-4}
$\dfrac{UNT_2^2}{UNT_1^2 + UNT_2^2}$	0 to 1	1×10^{-4}

Possible computational problems could arise in the above ratios so it is suggested that double precision be used. Also if $UNT_1^2 + UNT_2^2 < 10^{-8}$, then set

$$C_b^{n^+} = C_b^{n-}$$

From Eq. (6.3)

$$C_b^{n^+} = M \, C_b^{n^-}$$

The computation of M requires:

Mult	Time	Add	Time	Div	Time	Load	Time	Store	Time
16	184μs	10	38μs	4	185.2μs	20	75μs	20	91.6μs

Computation of $C_b^{n^+} = M\,C_b^{n^-}$

Mult	Time	Add	Time	Load	Time	Store	Time
27	310.5	18	68.4	45	168.75	45	206.1

Total Time 1327.55 μs

Total Memory 47 words and sqrt subroutine

$C_b^{n^+}$ is then converted to double precision for the direction cosine update.

Fine Align

A number of computational alternatives are available to determine the estimates given by Eq. (6.5) and Eq. (6.10).

1. Precompute all weighting coefficients.
2. Precompute $D(GGI)^{-1}$, $E_1^T(\Delta T_1 i)$ and $E_1^T(\Delta T_2 i)$ are evaluated in the onboard computer.
3. Perform all computations in the onboard computer.

Fine Align - Stage I

1. Precompute all weighting coefficients by evaluating the terms

$$D_1(tf_1)(GGI_1)^{-1}\,E_1^T(\Delta T_1 i - \frac{\Delta T_1}{2})$$

$i = 1, 20$

For L = 42° and g = 32.17 ft/sec^2 the results are:

Fine Align - Stage I

```
SAMPLE NUMBER          1

U                  1                2                3
    1   0,1373279393E-04 0,2886421372E-02 0,0000000000E+00
    2   -,2886434194E-02 0,1373278844E-04 0,0000000000E+00
    3   -,8193980871E+00 0,2598955250E-02 0,0000000000E+00
    4   -,1408488395E-06 -,4440675087E-04 0,0000000000E+00
    5   0,0000000000E+00 0,0000000000E+00 0,5000000000E-01
```

```
SAMPLE NUMBER          2

U                  1                2                3
    1   0,1091867979E-04 0,2418988075E-02 0,0000000000E+00
    2   -,2419001914E-02 0,1091866576E-04 0,0000000000E+00
    3   -,7331477132E+00 0,2220163861E-02 0,0000000000E+00
    4   -,1203204640E-06 -,3973238442E-04 0,0000000000E+00
    5   0,0000000000E+00 0,0000000000E+00 0,5000000000E-01
```

```
SAMPLE NUMBER          3

U                  1                2                3
    1   0,8332655549E-05 0,1951553461E-02 0,0000000000E+00
    2   -,1951567605E-02 0,8332635873E-05 0,0000000000E+00
    3   -,6468969495E+00 0,1883460043E-02 0,0000000000E+00
    4   -,1020729997E-06 -,3505800852E-04 0,0000000000E+00
    5   0,0000000000E+00 0,0000000000E+00 0,5000000000E-01
```

```
SAMPLE NUMBER          4

U                  1                2                3
    1   0,5974721247E-05 0,1484117640E-02 0,0000000000E+00
    2   -,1484131517E-02 0,5974698377E-05 0,0000000000E+00
    3   -,5606458417E+00 0,1588843733E-02 0,0000000000E+00
    4   -,8610644288E-07 -,3038362426E-04 0,0000000000E+00
    5   0,0000000000E+00 0,0000000000E+00 0,5000000000E-01
```

```
SAMPLE NUMBER        5

U                1                2                3
     1  0,3844878507E-05 0,1016680725E-02 0,0000000000E+00
     2  -,1016693897E-02 0,3844854448E-05 0,0000000000E+00
     3  -,4743944357E+00 0,1336315155E-02 0,0000000000E+00
     4  -,7242080576E-07 -,2570923277E-04 0,0000000000E+00
     5  0,0000000000E+00 0,0000000000E+00 0,5000000000E-01
```

```
SAMPLE NUMBER        6

U                1                2                3
     1  0,1943128949E-05 0,5492428260E-03 0,0000000000E+00
     2  -,5492549964E-03 0,1943105256E-05 0,0000000000E+00
     3  -,3881427775E+00 0,1125874531E-02 0,0000000000E+00
     4  -,6101610046E-07 -,2103483516E-04 0,0000000000E+00
     5  0,0000000000E+00 0,0000000000E+00 0,5000000000E-01
```

```
SAMPLE NUMBER        7

U                1                2                3
     1  0,2694741906E-06 0,8180405470E-04 0,0000000000E+00
     2  -,8181506152E-04 0,2694519742E-06 0,0000000000E+00
     3  -,3018909128E+00 0,9575220875E-03 0,0000000000E+00
     4  -,5189233913E-07 -,1636043254E-04 0,0000000000E+00
     5  0,0000000000E+00 0,0000000000E+00 0,5000000000E-01
```

```
SAMPLE NUMBER        8

U                1                2                3
     1  -,1176085711E-05 -,3856354777E-03 0,0000000000E+00
     2  0,3856256585E-03 -,1176105787E-05 0,0000000000E+00
     3  -,2156388876E+00 0,8312577586E-03 0,0000000000E+00
     4  -,4504951828E-07 -,1168602602E-04 0,0000000000E+00
     5  0,0000000000E+00 0,0000000000E+00 0,5000000000E-01
```

```
SAMPLE NUMBER        9

U                1                 2                 3
    1   -,2393552258E-05 -,8530756598E-03 0,0000000000E+00
    2   0,8530669148E-03 -,2393569976E-05 0,0000000000E+00
    3   -,1293867477E+00 0,7470811930E-03 0,0000000000E+00
    4   -,4048761885E-07 -,7011616730E-05 0,0000000000E+00
    5   0,0000000000E+00 0,0000000000E+00 0,5000000000E-01
```

```
SAMPLE NUMBER        10

U                1                 2                 3
    1   -,3382920714E-05 -,1320516380E-02 0,0000000000E+00
    2   0,1320508459E-02 -,3382936302E-05 0,0000000000E+00
    3   -,4313453899E-01 0,7049931900E-03 0,0000000000E+00
    4   -,3820668416E-07 -,2337205763E-05 0,0000000000E+00
    5   0,0000000000E+00 0,0000000000E+00 0,5000000000E-01
```

```
SAMPLE NUMBER        11

U                1                 2                 3
    1   -,4144195699E-05 -,1787957528E-02 0,0000000000E+00
    2   0,1787950043E-02 -,4144209833E-05 0,0000000000E+00
    3   0,4311769265E-01 0,7049928220E-03 0,0000000000E+00
    4   -,3820666393E-07 0,2337205761E-05 0,0000000000E+00
    5   0,0000000000E+00 0,0000000000E+00 0,5000000000E-01
```

```
SAMPLE NUMBER        12

U                1                 2                 3
    1   -,4677370917E-05 -,2255398992E-02 0,0000000000E+00
    2   0,2255391417E-02 -,4677384717E-05 0,0000000000E+00
    3   0,1293699013E+00 0,7470811765E-03 0,0000000000E+00
    4   -,4048761711E-07 0,7011616728E-05 0,0000000000E+00
    5   0,0000000000E+00 0,0000000000E+00 0,5000000000E-01
```

```
SAMPLE NUMBER        13

U                  1                 2                 3
     1    -,4982449429E-05  -,2722840660E-02  0,0000000000E+00
     2    0,2722832334E-02  -,4982464465E-05  0,0000000000E+00
     3    0,2156220412E+00  0,8312576136E-03  0,0000000000E+00
     4    -,4504950902E-07  0,1168602602E-04  0,0000000000E+00
     5    0,0000000000E+00  0,0000000000E+00  0,5000000000E-01


SAMPLE NUMBER        14

U                  1                 2                 3
     1    -,5059429619E-05  -,3190282421E-02  0,0000000000E+00
     2    0,3190272543E-02  -,5059447903E-05  0,0000000000E+00
     3    0,3018740665E+00  0,9575223570E-03  0,0000000000E+00
     4    -,5189235178E-07  0,1636043254E-04  0,0000000000E+00
     5    0,0000000000E+00  0,0000000000E+00  0,5000000000E-01


SAMPLE NUMBER        15

U                  1                 2                 3
     1    -,4908311428E-05  -,3657724164E-02  0,0000000000E+00
     2    0,3657711798E-02  -,4908335422E-05  0,0000000000E+00
     3    0,3881259312E+00  0,1125875342E-02  0,0000000000E+00
     4    -,6101614192E-07  0,2103483516E-04  0,0000000000E+00
     5    0,0000000000E+00  0,0000000000E+00  0,5000000000E-01


SAMPLE NUMBER        16

U                  1                 2                 3
     1    -,4529101041E-05  -,4125165778E-02  0,0000000000E+00
     2    0,4125149850E-02  -,4529133650E-05  0,0000000000E+00
     3    0,4743775894E+00  0,1336315355E-02  0,0000000000E+00
     4    -,7242081355E-07  0,2570923277E-04  0,0000000000E+00
     5    0,0000000000E+00  0,0000000000E+00  0,5000000000E-01
```

```
SAMPLE NUMBER        17

U                  1                 2                 3
      1   -,3921789037E-05  -,4592607151E-02  0,0000000000E+00
      2   0,4592586448E-02  -,3921833616E-05  0,0000000000E+00
      3   0,5606289954E+00  0,1588844058E-02  0,0000000000E+00
      4   -,8610645683E-07  0,3038362426E-04  0,0000000000E+00
      5   0,0000000000E+00  0,0000000000E+00  0,5000000000E-01
```

```
SAMPLE NUMBER        18

U                  1                 2                 3
      1   -,3086380042E-05  -,5060048172E-02  0,0000000000E+00
      2   0,5060021347E-02  -,3086440388E-05  0,0000000000E+00
      3   0,6468801033E+00  0,1883460522E-02  0,0000000000E+00
      4   -,1020730214E-06  0,3505800851E-04  0,0000000000E+00
      5   0,0000000000E+00  0,0000000000E+00  0,5000000000E-01
```

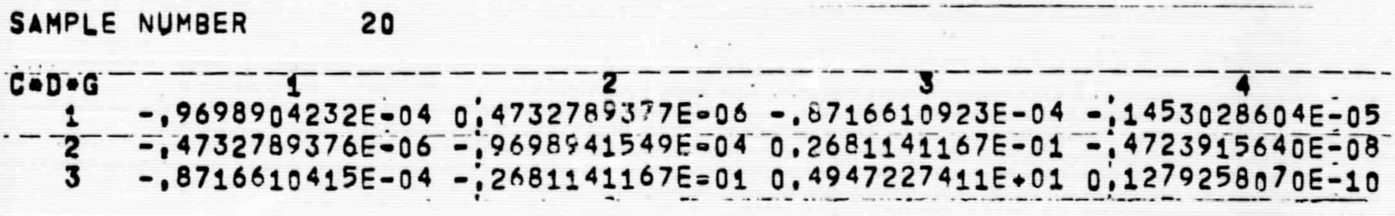
```
SAMPLE NUMBER        19

U                  1                 2                 3
      1   -,2022877116E-05  -,5527488730E-02  0,0000000000E+00
      2   0,5527454295E-02  -,2022957477E-05  0,0000000000E+00
      3   0,7331308671E+00  0,2220164109E-02  0,0000000000E+00
      4   -,1203204727E-06  0,3973238442E-04  0,0000000000E+00
      5   0,0000000000E+00  0,0000000000E+00  0,5000000000E-01
```

```
SAMPLE NUMBER        20

U                  1                 2                 3
      1   -,7312755230E-06  -,5994928713E-02  0,0000000000E+00
      2   0,5994885046E-02  -,7313805894E-06  0,0000000000E+00
      3   0,8193812410E+00  0,2598955618E-02  0,0000000000E+00
      4   -,1408488541E-06  0,4440675087E-04  0,0000000000E+00
      5   0,0000000000E+00  0,0000000000E+00  0,5000000000E-01
```

2. If $D_1(GGI_1)^{-1}$ is precomputed the result is:



```
SAMPLE NUMBER       20

C*D*G         1                 2                 3                 4
  1    -.9698904232E-04  0.4732789377E-06  -.8716610923E-04  -.1453028604E-05
  2    -.4732789376E-06  -.9698941549E-04  0.2681141167E-01  -.4723915640E-08
  3    -.8716610415E-04  -.2681141167E-01  0.4947227411E+01  0.1279258070E-10
```

Simulation results show that

$$|Z_N(i)| \leq .1 \times 10^{-1} \quad \text{ft/sec}^2$$

$$|Z_e(i)| \leq .1 \times 10^{-1} \quad \text{ft/sec}^2$$

The simulation also shows that the misalignment estimates are approximately 100 times larger than the estimation errors. Hence single precision computations are consistent with these results. However, due to the number of numerical operations, and the resultant round-off error, it is recommended that double precision be used (see the discussion at the end of this section).

Fine Align - Stage II

1. Precompute all weighting coefficients. The results are:

```
SAMPLE NUMBER         1

U                1                2                  3
    1   0,4034103940E-04  0,2342999844E=02  0,0000000000E+00
    2   -,2343131845E-02  0,4034087268E=04  0,0000000000E+00
    3   -,1838225232E+00  0,2109748411E=02  0,0000000000E+00
    4   -,1143419481E-06  -,9962521817E=05  0,0000000000E+00
    5   0,0000000000E+00  0,0000000000E+00  0,4000000000E-01
```

```
SAMPLE NUMBER         2

U                1                2                  3
    1   0,3375953831E-04  0,2044177413E=02  0,0000000000E+00
    2   -,2044319054E-02  0,3375909774E=04  0,0000000000E+00
    3   -,1685088506E+00  0,1862208995E=02  0,0000000000E+00
    4   -,1009260613E-06  -,9132380800E=05  0,0000000000E+00
    5   0,0000000000E+00  0,0000000000E+00  0,4000000000E-01
```

```
SAMPLE NUMBER         3

U                1                2                  3
    1   0,2759791058E-04  0,1745346029E=02  0,0000000000E+00
    2   -,1745492329E-02  0,2759726852E=04  0,0000000000E+00
    3   -,1531944347E+00  0,1636191140E=02  0,0000000000E+00
    4   -,8867657795E-07  -,8302221748E=05  0,0000000000E+00
    5   0,0000000000E+00  0,0000000000E+00  0,4000000000E-01
```

```
SAMPLE NUMBER         4

U                1                2                  3
    1   0,2185618189E-04  0,1446506280E=02  0,0000000000E+00
    2   -,1446652987E-02  0,2185540220E=04  0,0000000000E+00
    3   -,1378793432E+00  0,1431695765E=02  0,0000000000E+00
    4   -,7759354769E-07  -,7472046301E=05  0,0000000000E+00
    5   0,0000000000E+00  0,0000000000E+00  0,4000000000E-01
```

```
SAMPLE NUMBER        5

U                1                 2                   3
    1   0,1653437795E-04 0,1147658757E-02|0,0000000000E+00
    2   -,1147802347E-02 0,1653351593E-04|0,0000000000E+00
    3   -,1225636434E+00 0,1248723792E-02|0,0000000000E+00
    4   -,6767702051E-07 -,6641856098E-05|0,0000000000E+00
    5   0,0000000000E+00 0,0000000000E+00 0,4000000000E-01
```

```
SAMPLE NUMBER        6

U                1                 2                   3
    1   0,1163252270E-04 0,8488040510E-03|0,0000000000E+00
    2   -,8489417267E-03 0,1163162515E-04|0,0000000000E+00
    3   -,1072474031E+00 0,1087276052E-02|0,0000000000E+00
    4   -,5892704148E-07 -,5811652778E-05|0,0000000000E+00
    5   0,0000000000E+00 0,0000000000E+00 0,4000000000E-01
```

```
SAMPLE NUMBER        7

U                1                 2                   3
    1   0,7150637336E-05 0,5499427507E-03|0,0000000000E+00
    2   -,5500724447E-03 0,7149742530E-05|0,0000000000E+00
    3   -,9193068987E-01 0,9473532362E-03|0,0000000000E+00
    4   -,5134364799E-07 -,4981437982E-05|0,0000000000E+00
    5   0,0000000000E+00 0,0000000000E+00 0,4000000000E-01
```

```
SAMPLE NUMBER        8

U                1                 2                   3
    1   0,3088741309E-05 0,2510754470E-03|0,0000000000E+00
    2   -,2511958191E-03 0,3087878995E-05|0,0000000000E+00
    3   -,7661357109E-01 0,8289559443E-03|0,0000000000E+00
    4   -,4492687262E-07 -,4151213348E-05|0,0000000000E+00
    5   0,0000000000E+00 0,0000000000E+00 0,4000000000F-01
```

```
SAMPLE NUMBER         9

U                 1                 2                 3
      1    -,5531459264E-06  -,4779726996E-04  0,0000000000E+00
      2    0,4768683151E-04  -,5539545224E-06  0,0000000000E+00
      3    -,6129611441E-01  0,7320847781E-03  0,0000000000E+00
      4    -,3967674794E-07  -,3320980516E-05  0,0000000000E+00
      5    0,0000000000E+00  0,0000000000E+00  0,4000000000E-01
```

```
SAMPLE NUMBER        10

U                 1                 2                 3
      1    -,3775010111E-05  -,3466748099E-03  0,0000000000E+00
      2    0,3465741888E-03  -,3775752288E-05  0,0000000000E+00
      3    -,4597838740E-01  0,6567400722E-03  0,0000000000E+00
      4    -,3559329210E-07  -,2490741126E-05  0,0000000000E+00
      5    0,0000000000E+00  0,0000000000E+00  0,4000000000E-01
```

```
SAMPLE NUMBER        11

U                 1                 2                 3
      1    -,6576835949E-05  -,6455565825E-03  0,0000000000E+00
      2    0,6454649345E-03  -,6577507529E-05  0,0000000000E+00
      3    -,3066045763E-01  0,6029222148E-03  0,0000000000E+00
      4    -,3267652611E-07  -,1660496817E-05  0,0000000000E+00
      5    0,0000000000E+00  0,0000000000E+00  0,4000000000E-01
```

```
SAMPLE NUMBER        12

U                 1                 2                 3
      1    -,8958611952E-05  -,9444419975E-03  0,0000000000E+00
      2    0,9443577500E-03  -,8959217281E-05  0,0000000000E+00
      3    -,1534239265E-01  0,5706313987E-03  0,0000000000E+00
      4    -,3092646042E-07  -,8302492282E-06  0,0000000000E+00
      5    0,0000000000E+00  0,0000000000E+00  0,4000000000E-01
```

```
SAMPLE NUMBER        13

U                1                2                3
     1   -,1092032732E-04 -,1243330464E-02 0,0000000000E+00
     2   0,1243251317E-02 -,1092087927E-04 0,0000000000E+00
     3   -,2426005243E-04 0,5598677810E-03 0,0000000000E+00
     4   -,3034310356E-07 -,1665334537E-15 0,0000000000E+00
     5   0,0000000000E+00 0,0000000000E+00 0,4000000000E-01
```

```
SAMPLE NUMBER        14

U                1                2                3
     1   -,1246197267E-04 -,1542221393E-02 0,0000000000E+00
     2   0,1542144317E-02 -,1246249263E-04 0,0000000000E+00
     3   0,1529387261E-01 0,5706314478E-03 0,0000000000E+00
     4   -,3092646017E-07 0,8302492282E-06 0,0000000000E+00
     5   0,0000000000E+00 0,0000000000E+00 0,4000000000E-01
```

```
SAMPLE NUMBER        15

U                1                2                3
     1   -,1358354204E-04 -,1841114193E-02 0,0000000000E+00
     2   0,1841035431E-02 -,1358405993E-04 0,0000000000E+00
     3   0,3061193776E-01 0,6029223079E-03 0,0000000000E+00
     4   -,3267652531E-07 0,1660496817E-05 0,0000000000E+00
     5   0,0000000000E+00 0,0000000000E+00 0,4000000000E-01
```

```
SAMPLE NUMBER        16

U                1                2                3
     1   -,1428503019E-04 -,2140008275E-02 0,0000000000E+00
     2   0,2139923342E-02 -,1428558445E-04 0,0000000000E+00
     3   0,4592986783E-01 0,6567402342E-03 0,0000000000E+00
     4   -,3559329209E-07 0,2490741126E-05 0,0000000000E+00
     5   0,0000000000E+00 0,0000000000E+00 0,4000000000E-01
```

```
SAMPLE NUMBER        17

U                 1                 2                 3
     1    -,1456643429E-04  -,2438903046E=02  0,0000000000E+00
     2    0,2438806731E-02  -,1456707188E=04  0,0000000000E+00
     3    0,6124759526E-01  0,7320849754E=03  0,0000000000E+00
     4    -,3967674689E-07  0,3320980516E=05  0,0000000000E+00
     5    0,0000000000E+00  0,0000000000E+00  0,4000000000E-01
```

```
SAMPLE NUMBER        18

U                 1                 2                 3
     1    -,1442775188E-04  -,2737797918E=02  0,0000000000E+00
     2    0,2737684278E-02  -,1442852828E=04  0,0000000000E+00
     3    0,7656505247E-01  0,8289562625E=03  0,0000000000E+00
     4    -,4492687511E-07  0,4151213348E=05  0,0000000000E+00
     5    0,0000000000E+00  0,0000000000E+00  0,4000000000E-01
```

```
SAMPLE NUMBER        19

U                 1                 2                 3
     1    -,1386898567E-04  -,3036692300E=02  0,0000000000E+00
     2    0,3036554667E-02  -,1386996490E=04  0,0000000000E+00
     3    0,9188217189E-01  0,9473535601E=03  0,0000000000E+00
     4    -,5134364774E-07  0,4981437982E=05  0,0000000000E+00
     5    0,0000000000E+00  0,0000000000E+00  0,4000000000E-01
```

```
SAMPLE NUMBER        20

U                 1                 2                 3
     1    -,1289013631E-04  -,3335585602E=02  0,0000000000E+00
     2    0,3335416578E-02  -,1289139090E=04  0,0000000000E+00
     3    0,1071988859E+00  0,1087276439E=02  0,0000000000E+00
     4    -,5892704152E-07  0,5811652778E=05  0,0000000000E+00
     5    0,0000000000E+00  0,0000000000E+00  0,4000000000E-01
```

```
SAMPLE NUMBER       21

U                 1                 2                 3
     1   -,1149120860E-04 -,3634477233E-02 0,0000000000E+00
     2   0,3634268694E-02 -,1149281961E-04 0,0000000000E+00
     3   0,1225151271E+00 0,1248724257E-02 0,0000000000E+00
     4   -,6767702164E-07 0,6641856098E-05 0,0000000000E+00
     5   0,0000000000E+00 0,0000000000E+00 0,4000000000E-01
```

```
SAMPLE NUMBER       22

U                 1                 2                 3
     1   -,9672209793E-05 -,3933366602E-02 0,0000000000E+00
     2   0,3933109696E-02 -,9674266792E-05 0,0000000000E+00
     3   0,1378308278E+00 0,1431696249E-02 0,0000000000E+00
     4   -,7759354657E-07 0,7472046301E-05 0,0000000000E+00
     5   0,0000000000E+00 0,0000000000E+00 0,4000000000E-01
```

```
SAMPLE NUMBER       23

U                 1                 2                 3
     1   -,7433145716E-05 -,4232253120E-02 0,0000000000E+00
     2   0,4231938265E-02 -,7435746802E-05 0,0000000000E+00
     3   0,1531459205E+00 0,1636191717E-02 0,0000000000E+00
     4   -,8867657860E-07 0,8302221748E-05 0,0000000000E+00
     5   0,0000000000E+00 0,0000000000E+00 0,4000000000E-01
```

```
SAMPLE NUMBER       24

U                 1                 2                 3
     1   -,4774028461E-05 -,4531136197E-02 0,0000000000E+00
     2   0,4530753084E-02 -,4777280252E-05 0,0000000000E+00
     3   0,1684603376E+00 0,1862209648E-02 0,0000000000E+00
     4   -,1009260627E-06 0,9132380800E-05 0,0000000000E+00
     5   0,0000000000E+00 0,0000000000E+00 0,4000000000E-01
```

```
SAMPLE NUMBER        25

U                    1                  2                  3
        1   -,1694868730E-05   -,4830015241E-02   0,0000000000E+00
        2   0,4829552835E-02   -,1698886365E-05   0,0000000000E+00
        3   0,1837740116E+00   0,2109749097E-02   0,0000000000E+00
        4   -,1143419477E-06   0,9962521817E-05   0,0000000000E+00
        5   0,0000000000E+00   0,0000000000E+00   0,4000000000E-01
```

2. If $D_2(GGI_2)^{-1}$ is precomputed the result is:

```
SAMPLE NUMBER        25

C*D*G              1                  2                  3                  4
      1   -,7747614288E-04   0,1361182235E-05   -,6967946355E-04   -,3225857958E-06
      2   -,1361182235E-05   -,7748003117E-04   0,5952100082E-02   -,3776415080E-08
      3   -,6967946278E-04   -,5952100082E-02   0,3050348586E+00   0,3682757170E-10
      4   0,3225857958E-06   -,3776415038E-08   -,36827572[illegible]E-10   0,8961637653E-09
```

Alignment Accuracy vs Data Sampling Date

The simulation results show virtually no degradation in performance of the filter as the sampling time is increased. However, covariance results using worse environmental conditions show a degradation in performance.

Effect of Data Sampling Rate - Simulation Results

Sampling Time (sec)	ϵ_N μ rad	ϵ_E μ rad	ϵ_z μ rad	$\hat{R}_N$ rad/sec	Comments
28.8	22.008	-25.513	$.43106x10^{-2}$	$.244956x10^{-6}$	Total time 911.2 sec
57.6	22.534	-26.167	$.42753x10^{-2}$	$.246457x10^{-6}$	I. C. $\epsilon_N=\epsilon_E=2°;\epsilon_z=10°$
115.2	22.253	-25.875	$.42908x10^{-2}$	$.245651x10^{-6}$	Stage I Fine Align Sample
172.8	21.507	-25.031	$.43359x10^{-2}$	$.243516x10^{-6}$	Time = 10 sec
					.1g vibration
28.8	20.900	-21.113	$.43892x10^{-2}$	$.242111x10^{-6}$	Total time 911.2 sec
57.6	20.651	-20.509	$.44215x10^{-2}$	$.241407x10^{-6}$	I. C. $\epsilon_N=\epsilon_E=\epsilon_z=2°$
115.2	20.718	-20.828	$.44042x10^{-2}$	$.241590x10^{-6}$	Stage I Fine Align Sample
172.8	21.103	-21.632	$.43607x10^{-2}$	$.242696x10^{-6}$	Time = 10 sec New random phase .1g vibration
28.8	22.009	-25.514	$.43025x10^{-2}$	$.244783x10^{-6}$	Total time 911.2 sec
57.6	22.535	-26.168	$.42672x10^{-2}$	$.246280x10^{-6}$	I. C. $\epsilon_N=\epsilon_E=\epsilon_z=2°$
115.2	22.256	-25.876	$.42827x10^{-2}$	$.245476x10^{-6}$	Stage I Fine Align
172.8	21.514	-25.031	$.43279x10^{-2}$	$.243345x10^{-6}$	Sample time = 40 sec .1g vibration

Effect of Data Sampling Rate - Covariance Results

Sampling Time	$\lvert\epsilon_N\rvert$ μ rad	$\lvert\epsilon_E\rvert$ μ rad	$\lvert\epsilon_z\rvert$ rad	Comments
30 sec	24.9	24.9	$.444\times10^{-2}$	Vibration .35g
1 min	39.8	39.8	$.498\times10^{-2}$	
2 min	51.8	51.8	$.524\times10^{-2}$	

Fine Align Computer Requirements

If all the weighting coefficients are precomputed, then Stage I of fine align requires:

	Mult	Time	Add	Time	Load	Time	Store	Time	DP
$\hat{\epsilon}_N$	40	460	39	148.2	79	395	79	460.57	x
$\hat{\epsilon}_E$	40	460	39	148.2	79	395	79	460.57	x
$\hat{\epsilon}_z$	40	460	39	148.2	79	395	79	460.57	x

Memory required 270 words

Total time per cycle 219.57 μs

Reset DCM

	Time
SIN $\hat{\epsilon}_z$	244 μs
COS $\hat{\epsilon}_z$	230 μs

DCM Reset Matrix Multiply

Mult	Time	Add	Time	Load	Time	Store	Time	DP
24	276	27	102.6	51	255	51	297.33	x

Memory required 46

Total time 1404.93

If all the weighting coefficients are precomputed then Stage II of fine align requires:

	Mult	Time	Add	Time	Load	Time	Store	Time	DP
$\hat{\epsilon}_N$	50	575	49	186.2	99	495	99	577.17	x
$\hat{\epsilon}_E$	50	575	49	186.2	99	495	99	577.17	x
$\hat{\epsilon}_z$	50	575	49	186.2	99	495	99	577.17	x
$\hat{R}_N$	50	575	49	186.2	99	495	99	577.17	x

Memory required 440 words

Total time per cycle 309.34 μs

Reset DCM

SIN $\hat{\epsilon}_z$	244 μs
COS $\hat{\epsilon}_z$	230 μs

DC M Reset Matrix Multiply

Mult	Time	Add	Time	Load	Time	Store	Time	DP
24	276	27	102. 6	51	255	51	297. 33	x

Memory required	46
Total time	1404. 93

Options at End of Stage II of Fine Align

If it is not desired to enter the navigate mode after the final time for alignment, the following alternatives are available:

1. If none of the matrices are precomputed, the data sum Eq. (6.7) can be extended, and the matrix $D_2(GGI_2)^{-1}$ can be computed when it is desired to enter the navigate mode. A disadvantage of this scheme is the scaling problem arising from adding more data to the sum, and the matrix inversion.

2. Reset the DCM, rebias the gyros and accelerometers and do nothing (except, of course, update the DCM). It is easily shown that this is the same as not resetting and rebiasing, but extrapolating the estimates and then resetting and rebiasing. A disadvantage is the increasing errors since new data is not used.

3. Perform the resets and enter Stage II of fine align again. Use a recursive scheme, or a weighting scheme with a finite initial covariance.

9. REFERENCES

1. "Strapdown Calibration and Alignment Study: Volume 1 - Development Document", UNIVAC Federal Systems Division, prepared for Electronics Research Center, NASA (NAS-12-577), 1968.

2. "Strapdown Calibration and Alignment Study: Volume 2 - Procedural and Parametric Trade-off Analyses Document", UNIVAC Federal Systems Division, prepared for Electronics Research Center, NASA (NAS 12-577), 1968.

3. "Strapdown Calibration and Alignment Study: Volume 3 - Laboratory Procedures Manual", UNIVAC Federal Systems Division, prepared for Electronics Research Center, NASA (NAS 12-577), 1968.

4. "Summary of Test Results: Honeywell GG 334A Gyro", M. H. Lanman, memorandum, NASA-ERC, March 26, 1969.

5. "D4E Vibrating String Accelerometer Research for Strapdown Navigation", J. J. Mihm, NASA-ERC, GSC-67-02, October 1967.

6. "Axis Definitions for CH-46C Helicopter and Inertial Sensing Unit", A. H. Lipton, memorandum, NASA-ERC, January 3, 1969.

7. Inertial Guidance , Course 16. 42 notes, W. R. Markey and K. Britting, MIT Department of Aeronautics and Astronautics.

8. "Vibration Monitor for Flight Vehicles", L. Sher, NASA-ERC Working Paper, January 1969.

9. "Math Models for NASA/Huntsville Strapdown System Tests", W. H. Fincke, DRC Report E-1453U, May 27, 1968.

10. "SD-53 Strapdown Inertial Guidance System Lab. Procedure", W. H. Fincke, DRC Report E-1478U, 3 July 1968.

11. "Data Reduction Equations and Error Analysis for SD-53 Strapdown System Lab. Tests", W. H. Fincke, DRC Report E-1540U, 27 September 1968.

12. "Self Alignment Techniques for Strapdown Inertial Navigation Systems with Aircraft Application", K. R. Britting and T. Palsson, EAL-MIT RE-33, November 1968.

REFERENCES (cont.)

13. "Alignment Processing Techniques for Strapdown Inertial Guidance Systems", E. Farrell, AIAA Paper 68-829, August 1968.

14. "Launch Pad Alignment of a Strapdown Navigator by the Kalman Filter", J. F. Bellantoni, and E. J. Koenke, AIAA Paper 68-831, August 1968.

15. "Optimum Alignment of an Inertial Autonavigator", F. D. Jurenka, and C. T. Leondes, IEEE Vol. AES-3, No. 6, November 1967.

16. "Alignment of Inertial Systems on a Moving Base", A. H. Lipton, NASA TN D-4110, September 1967.

17. "Optimum Alignment of Inertial Navigation Systems", L. E. Hutchinson, and E. A. DeSousa, Project Themis, November 1968.

18. <u>Applied Optimal Control</u>, A. E. Bryson, Jr., and Y. C. Ho, Blaisdell Co., Waltham, Mass., 1969.

19. "Direction Cosine Computational Error", J. W. Jordan, NASA TR R-304, March 1969.

20. "New Results in Linear Filtering and Prediction Theory", R. E. Kalman, and R. S. Bucy, Trans. ASME, Journal of Basic Engineering, series D, Vol. 33, pp. 95-107, March 1961.

21. "Sensitivity Analysis", H. J. Rome, memorandum, DRC M-91U, March 18, 1968.

22. "Kalman Filter Incorporating Both Continuous and Discrete Measurements", H. J. Rome, memorandum, DRC M-104U, July 12, 1968.

23. "The Alignment of a Strapdown System", J. R. Hatfield, Interoffice Memo, TRW, Jan. 15, 1969.

24. "Operational Alignment and Calibration of A Strapdown Inertial Guidance System", D. O. Benson, W. H. Fincke, H. J. Rome, and David E. Lev, Interim Scientific Report, DRC E-1863U, Aug. 1969.

APPENDIX A

DESCRIPTION OF THE SIMULATION

The simulation is derived from the very general block diagram below.

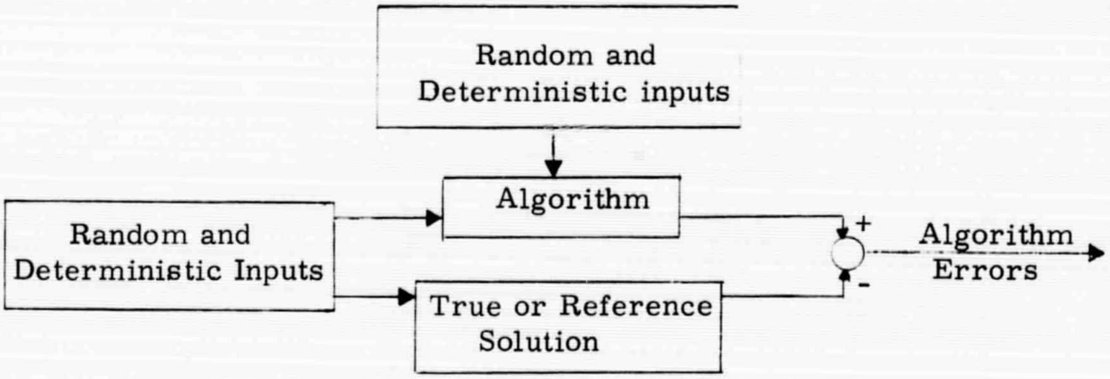

Random and Deterministic inputs
Algorithm
Random and Deterministic Inputs
True or Reference Solution
+
-
Algorithm Errors

Figure A-1 General Block Diagram

For the alignment study at hand, this specializes into

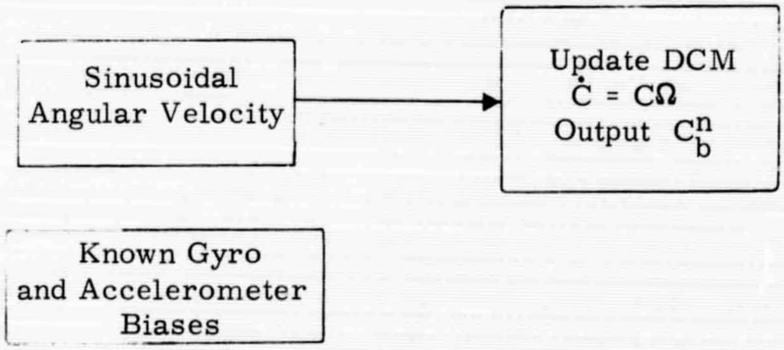

Sinusoidal Angular Velocity
Update DCM
$\dot{C} = C\Omega$
Output C_b^n
Known Gyro and Accelerometer Biases

Figure A-2 True Reference Solution

Initial Estimate of $C_b^n = C_b^{n^-}$

Figure A-3 Initialization of Alignment

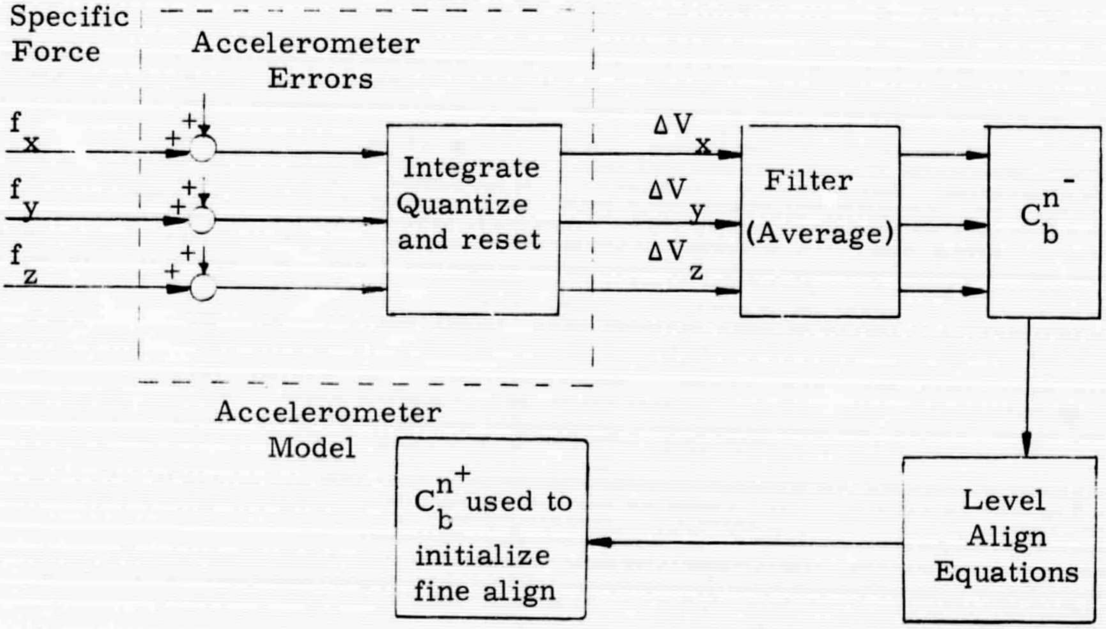


Figure A-4 Level Align

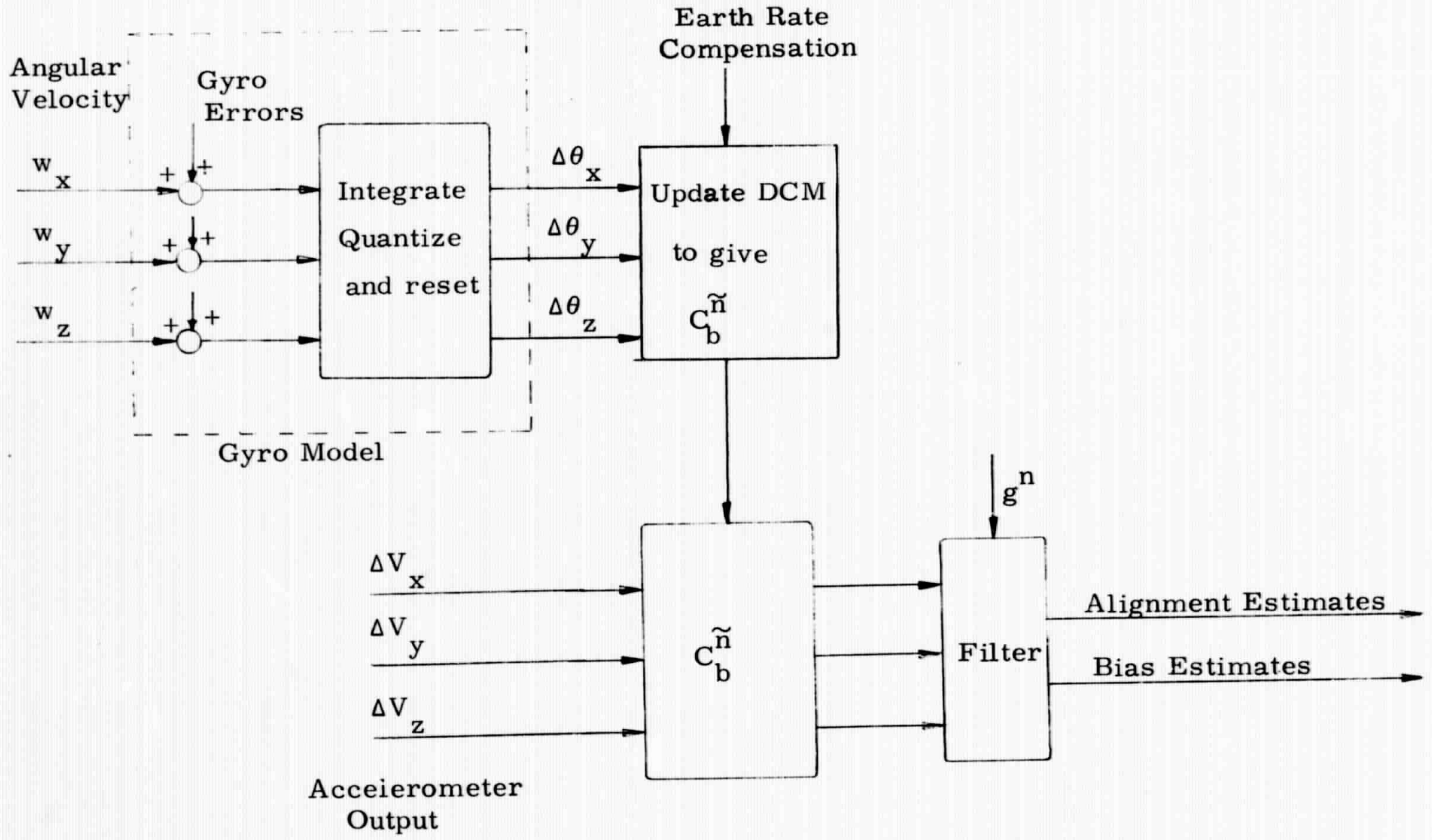


Figure A-5 Fine Align

SIMULATION INPUTS:

g^n	Gravity
L	Latitude
δV_g	Gyro biases
δV_a	Accelerometer biases
QG	Gyro quantization level
QA	Accelerometer quantization
T_1	Time for level align
T_2	Time for Stage I fine align
T_3	Time for Stage II fine align
NA_1	Samples in Stage I - fine align
NA_2	Samples in Stage II - fine align
$C_b^n(0)$	True initial DCM
C_b^{n-}	Initial estimate of $C_b^n(0)$
ΔT	Update time
w	Sinusoidal frequency
A_x A_y A_z	Magnitude of translational vibration
a_x a_y a_z	Magnitude of rotational vibration

SIMULATION OUTPUTS:

C_b^n	Reference or true DCM
$C_b^{\tilde{n}}$	Nominal solution
ZV	Indicated acceleration
$\hat{\epsilon}$	Misalignment estimates
$\hat{R}_N$	North component of gyro bias estimate
$\hat{A}_z$	Vertical component of accelerometer bias estimate

APPENDIX B

LISTING OF THE SIMULATION PROGRAMS

```
      C     THE ALIGNMENT SIMULATION PROGRAM
      C     P, MCKINNON FOR D, BENSON                                 SEPT, 1969
      C     NRM = TOTAL NUMBER OF RUNS
      C     NRDS= THE NO, OF UNIFORMLY DISTRIBUTED RANDOM NUMBERS THAT ARE
      C           DISCARDED AT THE BEGINNING OF A GROUP OF RUNS
      C     IF ID1 = 0 OR 1, THE TABLES ARE CONSTANT ( GENERATED ONCE)
      C     IF ID1=2, THE TABLES ARE VARIABLE (REGENERATED EVERY NTS)
      C     DT1 = CHANGE IN TIME (IN SECS ) FOR COARSE ALIGNMENT
      C     TF1 = AVERAGING TIME (OR FINAL TIME OF COARSE ALIGNMENT DATA)
      C              (IN SECS )
      C     NF1= NUMBER OF ITERATIONS IN COARSE ALIGNMENT
      C     QA = ACCELEROMETER QUANTIZATION LEVEL
      C     QG = GYRO QUANTIZATION LEVEL
      C     STH(3) = STANDARD DEVIATION OF THETA X, THETA Y, THETA Z RESPECTIV
      C              ELY (IN DEGREES)
      C     SVA(3) = STANDARD DEVIATION OF ACCELEROMETER BIAS X,Y AND Z RESPEC
      C              TIVELY (IN MU G)
      C     SVG(3) = STANDARD DEVIATION OF GYRO BIAS X,Y AND Z RESPECTIVELY
      C              (IN RADIANS / SECOND)
      C     DT2 = CHANGE IN TIME FOR TRUE SOLUTION AND FINE ALIGNMENT (SECS )
      C     TFS1= FINAL TIME FOR STAGE 1 OF TRUE SOLUTION AND FINE ALIGNMENT
      C           (IN SECS)
      C     NFS1= NUMBER OF ITERATIONS FROM THE BEGINNING OF THE RUN TO THE
      C           END OF STAGE 1 OF THE TRUE SOLUTION AND FINE ALIGNMENT
      C     TFS2= FINAL TIME FOR STAGE 2 OF TRUE SOLUTION AND FINE ALIGNMENT
      C           (IN SECS)
      C     NFS2= NUMBER OF ITERATIONS FROM THE BEGINNING OF THE RUN TO THE
      C           END OF STAGE 2 OF THE TRUE SOLUTION AND FINE ALIGNMENT
      C     NTS= TABLE SIZE
      C     NB1= NUMBER OF ITERATIONS/ GROUP IN STAGE 1
      C     NB2= NUMBER OF ITERATIONS / GROUP IN STAGE 2
      C     SL = SHIPS LATITUDE            ( IN DEGREES)
             COMMON AAA(5),Q(3),SIG(3,3),CBNO(3,3),CBNP(3,3),CBNT(3,3),DV(3),DV
            1O(3),DVN(3),FN(3),SDV(3),DTH(3),PHE1(3,3),PHE2(3,3),CBN(3,3),Y(5,5
            2),TDTH(3,50),TF(3,50),DTHB(3),GA(3,3),PHIG(3),PHIA(3),DTHA(3),
            3DTHBO(3),TW(3,50),E(3,5,30),D(3,4,30),GGII(6,6),GTMP(5,5),DVI(3),
            4ZV(3,30),GTZ(5),TDTH2(3,50),U2(50),U(50),SUDU(50),CU(50),OMCU(50),
            5 EST(3,3)
             DIMENSION STH(3),SVA(3),SVG(3),ZZ(5),II(5),JJ(5),UNT(3),GXYZ(3),
            1AXYZ(3),X(3,3),GGI(5,5)
             EQUIVALENCE (Q(1),QA),(Q(2),QG),(SIG(1,1),STH(1)),(SIG(1,2),SVA(1)
            1),(SIG(1,3),SVG(1)),(FN,UNT),(GA(1,1),GXYZ(1)),(GA(1,2),AXYZ(1)),
            2(Y,X),(GGII,GGI)
             NDRS = 3
             NCD = 3
             NDRCM = 3
             NRCM = NCD
             NCGA = 2
             NDRE = 3
             NDCE = 5
             NRE = 3
             NCE = 5
             NRD = NRE
             NCDD= NCE - 1
      REWIND 4
      REWIND 5
C     ZERO OUT INPUT MATRIX AREAS
      DO 14 I=1,NCD
      Q(I) = 0.0
      STH(I) = 0.0
      SVA(I) = 0.0
      SVG(I) = 0.0
      GXYZ(I) = 0.0
   14 AXYZ(I) = 0.0
      DO 87 I=1,NRCM
      DO 87 J=1,NRCM
 87   CBNO(I,J)=0.0
C     SET CONSTANTS
      PI = 3.1415927
      TPI = 2.0*PI
      TORAD = PI/180.0
      WE = 15.04196*TORAD / 3600.0
C     READ CONTROL DATA
      READ 10, NRM,NRDS,ID1
 10   FORMAT(3I10)
      IF(ID1 .EQ. 0), ID1= ID1 + 1
C     DISCARD RANDOM NOS.
      DO 22 I=1,NRDS
   22 RNX = RNDM(BBBB)
      DO 90 NR=1,NRM
      READ 12, DT1,DT2,NF1,NFS1 ,NFS2,NTS,NB1,NB2,
     1SL,CW,CG
 12   FORMAT(2F10.0,6I10/3F10.0)
      CW= CW*TPI
      TF1= DT1*NF1
      TFS1= DT2* NFS1
      TFS2= DT2 * NFS2
      NA1= (NFS1 - NF1)/NB1
      NA2= (NFS2- NFS1)/NB2
   17 READ 13, IT,(II(I),JJ(I),ZZ(I), I=1,5)
   13 FORMAT (I1,1X,5(2I2,E11.0))
      IF (IT .EQ. 9), GO TO 15
      DO 16 I=1,5
      IF (II(I) .EQ. 0), GO TO 17
      I1 = II(I)
      J1 = JJ(I)
      GO TO (18,19,20,40), IT
   18 Q(I1) = ZZ(I)
      GO TO 16
   19 SIG(I1,J1) = ZZ(I)
      GO TO 16
   20 CBNO(I1,J1) = ZZ(I)
      GO TO 16
   40 GA(I1,J1) = ZZ(I)
   16 CONTINUE
      GO TO 17
   15 CONTINUE
      WRITE (4) DT1,DT2,NF1,NFS1,NFS2,NTS,NB1,NB2,SL,CW,CG,TF1,TFS1,TFS
     1,NA1,NA2
      WRITE (4) ((Q(I), SIG(I,J),CBNO(I,J),GA(I,J),I=1,NCD),J=1,NCD)
   90 CONTINUE
      REWIND 4
      GO TO 91
   91 DO 11 NR=1,NRM
      READ (4) DT1,DT2,NF1,NFS1,NFS2,NTS,NB1,NB2,SL,CW,CG,TF1,TFS1,TFS2,
     1NA1, NA2
      READ (4) (( Q(I), SIG(I,J), CBNO(I,J),GA(I,J),I=1,NCD),J=1,NCD)
C     PRINT LABEL AND INPUT
      PRINT 23, NR,NRM
   23 FORMAT (21H1ALIGNMENT SIMULATION// 5H RUN ,I4, 4H OF ,I4///)
      PRINT 24, QA,QG
   24 FORMAT (35H ACCELEROMETER QUANTIZATION LEVEL= ,E15.8//35H GYRO QUA
     1NTIZATION LEVEL            = ,E15.8//)
      PRINT 26,DT1,TF1
   26 FORMAT ( 35H DELTA T FOR COARSE ALIGNMENT      = ,E15.8//35H AVERAGI
     1NG T FOR COARSE ALIGNMENT= ,E15.8 /)
      PRINT 37, DT2,TFS1,TFS2,SL
   37 FORMAT ( 47H DELTA T FOR TRUE SOLUTION AND FINE ALIGNMENT= ,E15.8/
     1/58H FINAL T FOR STAGE 1 OF TRUE SOLUTION AND FINE ALIGNMENT= ,
     2E15.8// 58H FINAL T FOR STAGE 2 OF TRUE SOLUTION AND FINE ALIGNMEN
     3T= ,E15.8// 18H SHIPS LATITUDE = , E15.8///)
      PRINT 25, ((SIG(I,J), J=1,NCD),I=1,NCD)
   25 FORMAT (5X,5HSIGMA,5X,5X,5HTHETA,5X,7X, 2HVA,6X,7X,2HVG/ 15X,4X,
     17HDEGREES,4X,6X,4HMU G,5X,4X,7HRAD/SEC//7X,1HX,7X,3E15.8//7X,1HY,
     27X, 3E15.8//7X,1HZ,7X,3E15.8///)
      PRINT 43
      CALL PRVL3(2HGA,GA,NCD,NCGA,1,NDRS,0,0,1)
   43 FORMAT (   36H1ROTATIONAL AND TRANSLATIONAL MOTION)
      CALL PRVL3 (4HCBNO,CBNO,NRCM,NRCM,1,NDRCM,0,0,1)
C     CONVERT SIGMA THETAIS TO RADS
      DO 21 I=1,NCD
   21 STH(I) = STH(I)*TORAD
C     GENERATE TABLES
C     GET PHIG AND PHIA MATRICES FROM A UNIFORM DISTRIBUTION
      DO 44 I=1,NCD
      PHIG(I) = TPI*RNDM(BBBB)
   44 PHIA(I) = TPI*RNDM(BBBB)
      SL = SL*TORAD
      COSL = COS(SL)
      SINL = SIN(SL)
      WCOSL = WE*COSL*DT2
      WSINL = WE*SINL*DT2
      CF1 = SVA(1)*DT1/QA
      CF2 = SVA(2)*DT1/QA
      CF3 =  SVA(3)     *DT1/QA
      CGDQ=-CG*DT1/QA
      QAW = QA*CW
      COSL2 = COSL**2
      SINL2 = SINL**2
      ISWE= 1
      TCWT2=2.0*COS(CW*DT2)
      TCWT1= 2.0*COS(CW*DT1)
```

```
          CTH1= CBNO(1,1)*WCOSL - CBNO(3,1)*WSINL
          CTH2= CBNO(1,2)*WCOSL - CBNO(3,2)*WSINL
          CTH3= CBNO(1,3)*WCOSL - CBNO(3,3)*WSINL
          GO TO (107,108),ID1
  107     NTS1= NTS
          GO TO 109
  108     NTS1=2
          JJ=3
  109     DO 47 J=1,NTS1
          WDT1 = CW*DT2*J
          WDT2 = CW*DT2*(J-1)
          WDT3 = CW*DT1*J
          WDT4 = CW*DT1*(J-1)
          DO 47 I=1,NCD
          TDTH(I,J) = -GXYZ(I)/CW*(COS(WDT1+PHIG(I))-COS(WDT2+
         1PHIG(I)))
       47 TF(I,J) = -AXYZ(I)/QAW*(COS(WDT3+PHIA(I))-COS(WDT4+PHIA(I)))
          GO TO (111,112), ID1
  112     IF(NTS .LE. 2), GO TO 111
          DO 110 J=JJ,NTS
          JJ1=NTS-(J-JJ)
          DO 110 I=1,NCD
          TDTH(I,JJ1)= TCWT2*TDTH(I,JJ1-1)- TDTH(I,JJ1-2)
  110     TF(I,JJ1)= TCWT1*TF(I,JJ1-1)- TF(I,JJ1-2)
  111     DO 46 J=1,NTS
          TDTH(1,J)= TDTH(1,J) + CTH1
          TDTH(2,J)= TDTH(2,J) + CTH2
          TDTH(3,J)=TDTH(3,J) + CTH3
          TF(1,J) = TF(1,J)+CF1
          TF(2,J) = TF(2,J)+CF2
          TF(3,J) = TF(3,J)+CF3
          U2(J)=0.0
          DO 61 I=1,NCD
          TW(I,J) = (TDTH(I,J)+SVG(I)*DT2)/QG
          TDTH2(I,J)= TDTH(I,J)* TDTH(I,J)
  61      U2(J)= U2(J) + TDTH2(I,J)
          U(J)= SQRT(U2(J))
          SUDU(J)= SIN(U(J))/U(J)
          CU(J)= COS(U(J))
          OMCU(J)= (1.0-CU(J))/U2(J)
       46 CONTINUE
          GO TO (113,52,58),ISWE
C         COMPUTE EIS AND DIS
  113     NA=NA1
          TI= (TFS1 - TF1)/NA1
          TID2 = TI/2.0
          DO 67 I=1,NCE
          DO 67 J=1,NCE
       67 GGI(I,J) = 0.0
          DO 65 K=1,NA
          T1 = K*TI
          T2 = TID2+(K-1)*TI
          ECOS = COS(WE*T2)
          ESIN = SIN(WE*T2)
```

```
      DCOS = COS(WE*T1)
      DSIN = SIN(WE*T1)
      E(1,1,K) = CG*SINL*ESIN
      E(1,2,K) = CG*ECOS
      E(1,3,K) = CG*COSL*ESIN
      E(1,4,K) = CG*(-SINL)*(1,0-ECOS)/WE
      E(1,5,K) = 0,0
      E(2,1,K)=CG*(-SINL2*ECOS - COSL2)
      E(2,2,K) = CG*SINL*ESIN
      E(2,3,K) = -CG*SINL*COSL*(ECOS-1,0)
      E(2,4,K)= CG*(SINL2*ESIN/WE + COSL2*T2)
      E(2,5,K) = 0,0
      E(3,1,K) = 0,0
      E(3,2,K) = 0.0
      E(3,3,K) = 0,0
      E(3,4,K) = 0,0
      E(3,5,K) = 1,0
      D(1,1,K) = SINL2*DCOS+COSL2
      D(1,2,K) = -SINL*DSIN
      D(1,3,K) = SINL*COSL*(DCOS-1,0)
      D(1,4,K) = -SINL2*DSIN/WE-COSL2*T1
      D(2,1,K) = -D(1,2,K)
      D(2,2,K) = DCOS
      D(2,3,K) = COSL*DSIN
      D(2,4,K) = -SINL*(1,0-DCOS)/WE
      D(3,1,K) = D(1,3,K)
      D(3,2,K) = -D(2,3,K)
      D(3,3,K) = COSL2*DCOS+SINL2
      D(3,4,K) = -SINL*COSL*(DSIN/WE-T1)
C     COMPUTE GGI(5,5)
      DO 68 I=1,NRE
      DO 68 J=1,NCE
   68 Y(J,I) = E(I,J,K)
      CALL GMP3 (Y,E,GTMP,NCE,NRE,1,NCE,K,NDCE,NDRE,NDRE,NDCE,NDCE)
      DO 69 I=1,NCE
      DO 69 J=1,NCE
   69 GGI(I,J) = GGI(I,J)+GTMP(I,J)
   65 CONTINUE
      CALL XINV(GGI,GGII,NCE,NCE)
C     PART I--COARSE ALIGNMENT
C     PRINT THETA AND ACCELEROMETER AND GYRO BIAS
      PRINT 66
   66 FORMAT (40H1THETA AND ACCELEROMETER AND GYRO BIASES)
      CALL PRVL3 (1H ,SIG,NCD,NCD,1.NDRS,0,0,1)
C     FORM CBNT
      CTX = COS(STH(1))
      STX = SIN(STH(1))
      CTY = COS(STH(2))
      STY = SIN(STH(2))
      CTZ = COS(STH(3))
      STZ = SIN(STH(3))
      CBNT(1,1) = CTY*CTZ
      CBNT(2,1) = -CTX*STZ+STX*STY*CTZ
      CBNT(3,1) = STX*STZ+CTX*STY*CTZ
      CBNT(1,2) = CTY*STZ
      CBNT(2,2) = CTX*CTZ+STX*STY*STZ
      CBNT(3,2) = -STX*CTZ+CTX*STY*STZ
      CBNT(1,3)=-STY
      CBNT(2,3)=STX*CTY
      CBNT(3,3)=CTX*CTY
      CALL GMP3(CBNT,CBNO,CBNT,NCD,NCD,1,NCD,1,NDRS,0,NDRS,0,NDRS)
      ISWD=1
      GO TO 104
C     COMPUTE DELTA V'S
 105  ISWB=1
      DO 31 I=1,NCD
 31   SDV(I)=0.0
C     SET DVO(3) AT 0 HRS
      DO 48 I=1,NCD
 48   DVO(I)= 2.0*RNDM(BBBB) - 1.0
      ITB1=1
      DO 28 NN=1,NF1
      GO TO 106
C     COMPUTE DV(3)
 70   DO 50 I=1,NCD
      DVN(I)= AINT(DVO(I))
      DV(I) = TF(I,ITB1) + DVO(I) - DVN(I)           + CBN(3,I)* CGDO
 50   DVO(I) = DV(I)
      IF (ITB1 .EQ. NTS), GO TO 51
      ITB1= ITB1 + 1
      GO TO 52
 51   ITB1= 1
      ISWE=2
      GO TO (52,116),ID1
 116  NTSM1= NTS - 1
       DO 118 I=NTSM1,NTS
      TDTH(1,I)= TDTH(1,I)- CTH1
      TDTH(2,I)= TDTH(2,I) - CTH2
      TDTH(3,I)= TDTH(3,I) - CTH3
      TF(1,I)= TF(1,I) - CF1
      TF(2,I)= TF(2,I)- CF2
 118  TF(3,I)= TF(3,I) - CF3
      DO 117 I=1,NCD
      TDTH(I,1)= TCWT2*TDTH(I,NTS)- TDTH(I,NTS-1)
      TDTH(I,2)=TCWT2*TDTH(I,1) - TDTH(I,NTS)
      TF(I,1)= TCWT1*TF(I,NTS)- TF(I,NTS-1)
 117  TF(I,2)=TCWT1*TF(I,1) - TF(I,NTS)
      GO TO 112
 52   DO 29 I=1,NCD
      DVN(I) =(AINT(DV(I))  + DVN(I))/2.0
      GO TO (71,29),ISWB
 71   SDV(I) = SDV(I) +DVN(I)
 29   CONTINUE
      GO TO (28,72), ISWB
 28   CONTINUE
      DO 32 I=1,NCD
 32   SDV(I) = SDV(I)*GA/TF1
C     PRINT SDV'S
```

```
      PRINT 88
   88 FORMAT(///17H SUM OF DELTA V S)
      CALL PRVL3(1H ,SDV,1,NCD,1,1,0,NF1,1)
      CALL GMP3( CBNT,SDV,FN,NRCM,NRCM,1,1,1,NDRCM,NDRCM,NDRS,1,NDRS)
      A= SQRT(FN(1)**2 + FN(2)**2 + FN(3)**2)
      DO 33 I=1,NCD
   33 UNT(I) = FN(I)/A
      UND= UNT(1)* UNT(1) + UNT(2) * UNT(2)
      UNN= 1.0 + UNT(3)
C     COMPUTE CBNP
      CBNP(1,1)= 1.0 - UNN * UNT(1)**2/UND
      CBNP(1,2)= -UNN* UNT(1)* UNT(2)     /UND
      CBNP(1,3)= UNT(1)
      CBNP(2,1)= CBNP(1,2)
      CBNP(2,2)= 1.0 - UNN* UNT(2)**2/ UND
      CBNP(2,3)= UNT(2)
      CBNP(3,1)=  -UNT(1)
      CBNP(3,2)= -UNT(2)
      CBNP(3,3)= - UNT(3)
      CALL GMP3(CBNP, CBNT, CBNP,NRCM,NRCM,1,NRCM,1,NDRCM,NDRCM,
     1 NDRCM,NDRCM, NDRCM)
      CALL PRVL3(4HCBNE,CBNP,NRCM,NRCM,1,NDRCM,0,NF1,2)
      WRITE (5) NF1, ((CBNP(I,J), I=1,NRCM),J=1,NRCM)
      NNV= NF1 + 1
      ISWD=2
      GO TO 95
C     PART II -- TRUE SOLUTION AND FINE ALIGNMENT
C     PART II A -- TRUE SOLUTION
C     COMPUTE PHE1
  104 SINL2= SINL**2
      COSL2= COSL**2
       SINWDT= SIN(WE*DT2)
      COSWDT= COS(WE*DT2)
      PHE1(1,1)= COSL2 + SINL2 * COSWDT
      PHE1(1,2)= -SINL*SINWDT
      PHE1(1,3)= SINL*COSL*(COSWDT- 1.0)
      PHE1(2,1)= SINL* SINWDT
      PHE1(2,2)= COSWDT
      PHE1(2,3)= COSL*SINWDT
      PHE1(3,1)= SINL*COSL*(COSWDT-1.0)
      PHE1(3,2)= - COSL*SINWDT
      PHE1(3,3)= SINL2 + COSL2* COSWDT
      DO 39 I=1,NRCM
      DO 39 J=1,NRCM
   39 CBN(I,J)= CBNO(I,J)
C     COMPUTE DTHBO(3) AT 0 HOURS
      DO 49 I=1,NCD
   49 DTHBO(I) = 2.0 * RNDM(BBBB) - 1.0
      NA= NA1
      NF2= NFS1
      NNV= 1
      NB= NB1
      ISWC=1
      DO 73 J=1,NA
      DO 73 I=1,NCD
   73 ZV(I,J)=0.0
      ITB2=1
      NBC=1
      NAC=1
      NPC=2
      ISL=1
      NBS= NF1
      GO TO 105
   95 DO 38 NN= NNV, NF2
C     COMPUTE DTH(3)
  106 DO 53 I=1,NCD
   53 DTH(I) = TDTH(I,ITB2)
C     COMPUTE PHE2
      DTX2= TDTH2(1,ITB2)
      DTY2= TDTH2(2,ITB2)
      DTZ2= TDTH2(3,ITB2)
      U2P= U2(ITB2)
      UP= U(ITB2)
      SINUDU= SUDU(ITB2)
      COSU= CU(ITB2)
      OMCUP= OMCU(ITB2)
      PHE2(1,1) = (DTX2+(DTY2+DTZ2)*COSU)/U2P
      PHE2(1,2) = DTH(1)*DTH(2)*OMCUP
      PHE2(1,3) = DTH(1)*DTH(3)*OMCUP
      PHE2(2,1) = PHE2(1,2)
      PHE2(2,2) = (DTY2+(DTX2+DTZ2)*COSU)/U2P
      PHE2(2,3) = DTH(2)*DTH(3)*OMCUP
      PHE2(3,1) = PHE2(1,3)
      PHE2(3,2) = PHE2(2,3)
      PHE2(3,3) =(DTZ2+(DTX2+DTY2)*COSU)/U2P
      PCON = DTH(3)*SINUDU
      PHE2(1,2) = PHE2(1,2)-PCON
      PHE2(2,1) = PHE2(2,1)+PCON
      PCON = DTH(2)*SINUDU
      PHE2(1,3) = PHE2(1,3)+PCON
      PHE2(3,1) = PHE2(3,1)-PCON
      PCON = DTH(1)*SINUDU
      PHE2(2,3) =PHE2(2,3)-PCON
      PHE2(3,2) =PHE2(3,2)+PCON
C     COMPUTE CBN(NN) = PHE1*CBN(NN-1)*PHE2
   63 CALL GMP3 (PHE1,CBN,X,NRCM,NRCM,1,NRCM,1,NDRCM,NDRCM,NDRCM,NDRCM,
     1NDRCM)
      CALL GMP3 (X,PHE2,CBN,NRCM,NRCM,1,NRCM,1,NDRCM,NDRCM,NDRCM,NDRCM,
     1NDRCM)
  100 IF (NN .LT. NF1), GO TO 85
      IF (NN .NE. NBS), GO TO 86
      CALL PRVL3 (4HCBNT,CBN,NRCM,NRCM,1,NDRCM,0,NN,ISL)
      WRITE (5) NN, ((CBN(I,J),I=1,NRCM),J=1,NRCM)
      ISL = 1
      NPC = NPC+1
      IF (NN .EQ. NF1), NBS = NBS+NB
      IF (NPC .LE. 4), GO TO 86
      NPC = 1
      ISL = 2
   86 IF (NN .GT. NF1), GO TO 55
   85 IF (ITB2 .EQ. NTS), GO TO 54
      ITB2 = ITB2+1
      GO TO (70,38),ISWD
   54 ITB2 = 1
      GO TO (70,38), ISWD
C     COMPUTE DTHA(3)
   55 CONTINUE
C     COMPUTE DTHB(3)
      DO 56 I=1,NCD
      DTHB(I) = TW(I,ITB2)+DTHBO(I)-AINT(DTHBO(I))
   56 DTHBO(I) = DTHB(I)
      IF (ITB2 .EQ. NTS), GO TO 57
      ITB2 = ITB2+1
      GO TO 58
   57 ITB2 = 1
      ISWE=3
      GO TO (58,116),ID1
C     QUANTIZE DTHA(3)
   58 DO 59 I=1,NCD
      DTHB(I) = AINT(DTHB(I))
   59 DTH(I) = QG*DTHB(I)
C     COMPUTE PHE2
      DTX2= DTH(1) * DTH(1)
      DTY2= DTH(2)* DTH(2)
      DTZ2= DTH(3)* DTH(3)
      U2P=DTX2 + DTY2 + DTZ2
      UP= SQRT(U2P)
      ISWS=1
      IF(ABS(UP) .GE. QG), GO TO 101
      ISWS=2
      GO TO 64
  101 SINU= SIN(UP)
      SINUDU= SINU/UP
      COSU= COS(UP)
      OMCUP= (1.0- COSU) / U2P
      PHE2(1,1) = (DTX2+(DTY2+DTZ2)*COSU)/U2P
      PHE2(1,2) = DTH(1)*DTH(2)*OMCUP
      PHE2(1,3) = DTH(1)*DTH(3)*OMCUP
      PHE2(2,1) = PHE2(1,2)
      PHE2(2,2) = (DTY2+(DTX2+DTZ2)*COSU)/U2P
      PHE2(2,3) = DTH(2)*DTH(3)*OMCUP
      PHE2(3,1) = PHE2(1,3)
      PHE2(3,2) = PHE2(2,3)
      PHE2(3,3) =(DTZ2+(DTX2+DTY2)*COSU)/U2P
      PCON = DTH(3)*SINUDU
      PHE2(1,2) = PHE2(1,2)-PCON
      PHE2(2,1) = PHE2(2,1)+PCON
      PCON = DTH(2)*SINUDU
      PHE2(1,3) = PHE2(1,3)+PCON
      PHE2(3,1) = PHE2(3,1)-PCON
      PCON = DTH(1)*SINUDU
      PHE2(2,3) =PHE2(2,3)-PCON
      PHE2(3,2) =PHE2(3,2)+PCON
C     COMPUTE CBNP(NN) = PHE1*CBNP(NN-1)*PHE2
   64 CALL GMP3(PHE1,CBNP,X,NRCM,NRCM,1,NRCM,1,NDRCM,NDRCM,NDRCM,NDRCM,
     1NDRCM)
      GO TO (102,103), ISWS
  102 CALL GMP3 (X,PHE2,CBNP,NRCM,NRCM,1,NRCM,1,NDRCM,NDRCM,NDRCM,
     1NDRCM,NDRCM)
  103 ISWB = 2
      IF (NN .NE. NBS), GO TO 70
      CALL PRVL3 (4HCBNP,CBNP,NRCM,NRCM,1,NDRCM,0,NN,ISL)
      WRITE (5) NN, ((CBNP(I,J),I=1,NRCM),J=1,NRCM)
      ISL = 1
      NPC = NPC+1
      NBS = NBS+NB
      IF (NPC .LE. 4), GO TO 70
      NPC = 1
      ISL = 2
      GO TO 70
   72 CALL GMP3(CBNP,DVN,DVI,NRCM,NRCM,1,1,1,NDRCM,NDRCM,NDRS,1,NDRS)
      DO 74 I=1,3
   74 ZV(I,NAC) = ZV(I,NAC)+DVI(I)
      NBC = NBC+1
      IF (NBC .LE. NB), GO TO 38
      NBC = 1
      IF (NAC .GT. NA), GO TO 38
      DO 76 I=1,NCD
   76 ZV(I,NAC) = ZV(I,NAC)*GA/(NB*DT2)
      ZV(3,NAC) = CG+ZV(3,NAC)
      NAC = NAC+1
   38 CONTINUE
      PRINT 89
 89   FORMAT(1H1)
      CALL PRVL3(2HZV,ZV,NCD,NA,1,NDRS,0,NA,1)
      WRITE (5) NA, ((ZV(I,J), I=1,NCD), J=1,NA)
      GO TO (92,11),ISWC
 92   DO 79 I=1,NCE
 79   Y(I,1)=0.0
      DO 80 K=1,NA
      DO 81 J=1,NCE
      DO 81 I=1,NRE
 81   Y(J,1)= Y(J,1) + E(I,J,K)*ZV(I,K)
   80 CONTINUE
      CALL GMP3(GGI,Y,GTZ,NCE,NCE,1,1,1,NDCE,NDCE,NDCE,NDCE,NDCE)
      CALL PRVL3 (3HGTZ,GTZ,1,NCE,1,1,0,NA,2)
      WRITE (5) NA, (GTZ(I), I=1,NCE)
      DO 82 K=1,NA
      DO 83 I=1,NRD
      Y(I,1) = 0.0
      DO 83 J=1,NCDD
   83 Y(I,1) = Y(I,1)+D(I,J,K)*GTZ(J)
      CALL PRVL3 (5HD*GTZ,Y,1,NRD,1,1,0,K,1)
   82 CONTINUE
      WRITE (5) NA, (Y(I,1), I=1,NRD)
C     COMPUTE EST
```

```
      CDX= COS(Y(1,1))
      SDX= SIN(Y(1,1))
      CDY= COS(Y(2,1))
      SDY= SIN(Y(2,1))
      CDZ= COS(Y(3,1))
      SDZ= SIN(Y(3,1))
      EST(1,1)= CDY*CDZ
      EST(1,2)= -CDX*SDZ + SDX*SDY*CDZ
      EST(1,3)= SDX*SDZ + CDX*SDY*CDZ
      EST(2,1)= CDY*SDZ
      EST(2,2)= CDX*CDZ + SDX*SDY*SDZ
      EST(2,3)= -SDX*CDZ + CDX*SDY*SDZ
      EST(3,1)= -SDY
      EST(3,2)= SDX*CDY
      EST(3,3)= CDX*CDY
      CALL PRVL3(3HEST,EST,NRCM,NRCM,1,NDRCM,0,NA,2)
      CALL GMP3(EST,CBNP,X,NRCM,NRCM,1,NRCM,1,NDRCM,0,NDRCM,0,NDRCM)
      CALL PRVL3(4HCBNP,X,NRCM,NRCM,1,NDRCM,0,NFS1,1)
      WRITE (5) NFS1, (( X(I,J),I=1,NRCM),J=1,NRCM)
      IF( NFS1 .EQ. NFS2), GO TO 11
      DO 93 I=1,NRCM
      DO 93 J=1, NRCM
   93 CBNP(I,J)= X(I,J)
      NA= NA2
      NF2= NFS2
      NNV= NFS1 + 1
      NB= NB2
      ISWC=2
      ISL= 2
      NBC=1
      NAC= 1
      NPC=1
      NBS= NFS1 + NB
      DO 94 J=1,NA
      DO 94 I=1,NCD
   94 ZV(I,J)= 0.0
      GO TO 95
   11 CONTINUE
      PRINT 84
   84 FORMAT (19H1END OF PROGRAM ASP)
      REWIND 4
      REWIND 5
      STOP
      END
```

```
            SUBROUTINE XINV (Y,X,NRX,NCX)
      C     SUBROUTINE TO INVERT A MATRIX AND STORE IT ON ITSELF
      C     P MORTELLITE                            9-9-65
      C     NOTE-- THE DIMENSION FOR ROWS AND COLUMNS MUST BE INCREASED BY 1
            DIMENSION X(6,6),Y(5,5)
      C     SPREAD OUT Y INTO X
            J1=NCX
            DO 19 J=2,NCX
            I1=NRX
            DO 20 I=1,NRX
            X(I1,J1)=Y(I1,J1)
       20   I1=I1-1
       19   J1=J1-1
            NR1=NRX + 1
            NC1=NCX+1
            DO 10 II=1,NRX
      C     ADD A UNIT COLUMN VECTOR
            X(1,NC1)=1.
            DO 11 I=2,NRX
       11   X(I,NC1)=0.0
      C     COMPUTE PIVOT ROW
            IF(X(1,1))16,17,16
       17   PRINT 18
       18   FORMAT(44H1ERROR,DIVISION BY 0 IN INVERSION ROUTINE S6)
       16   DO 12 J=1,NCX
       12   X(NR1,J)=X(1,J+1)/X(1,1)
      C     COMPUTE NEW VALUE FOR ALL OTHER ELEMENTS
            DO 13 I=2,NRX
            DO 14 J=1,NCX
       14   X(I-1,J)=X(I,J+1)-X(I,1)*X(NR1,J)
       13   CONTINUE
      C     MOVE PIVOT ROW UP TO NTH ROW
            DO 15  J=1,NCX
       15      X(NRX,J)=X(NR1,J)
       10   CONTINUE
      C     SHRINK X INTO Y
            DO 21 J=2,NCX
            DO 22 I=1,NRX
       22   Y(I,J)=X(I,J)
       21   CONTINUE
            RETURN
            END
```

```
              SUBROUTINE PRVL3(HOL,A,NRF,NCF,NDF,NDR,NDC,N,ISL)
       C      A SUBROUTINE TO PRINT A VARIABLE LENGTH MATRIX OF 1,2 OR 3 DIMENS
       C      THE PAGE IS SLEWED OR NOT ACCORDING TO ISL=2 OR 1, RESPECTIVELY
       C      P MORTELLITE                              4-12-66
              DIMENSION A(1)
              K= (NCF + 6)/7
              K1=1
              K2=7
              DO 10 I=1,K
              IF(NCF-K2)15,15,14
        15    K3=NCF
              GO TO 16
         14   K3=K2
        16    GO TO (20,18),ISL
        20    PRINT 21
        21    FORMAT(///1H )
              GO TO 17
        18    PRINT 19
        19    FORMAT(1H1)
        17    PRINT                 11,N,     HOL,( J ,J=K1,K3)
        11    FORMAT(15H SAMPLE NUMBER ,I7,
             1     //1X,A6,7(8X,I2,7X))
              LIM= K3 * NDR  + (NDF -1) * NDR * NDC
              DO 12 KV=1,NRF
              KV1= (K1-1)*NDR + KV  + (NDF -1)*NDR*NDC
              PRINT       13,        KV,(A(II),II=KV1,LIM,NDR)
           13 FORMAT (3X,I2,2X,7E17.10)
           12 CONTINUE
              K1=K1 + 7
           10 K2 = K2 + 7
              RETURN
              END
```

```
            SUBROUTINE GMP3(X,Y,Z,NRX ,NCX,NDX,NCY,NDY,NDXR,NDXC,NDYR,NDYC,
          1 NDZR)
      C     SUBROUTINE TO MULTIPLY X*Y AND STORE IN Z OR X
      C     P MCKINNON                                       OCT 1967
            DIMENSION X(1),Y(1),Z(1),A(1)
            COMMON A
            DO 12 K=1,NRX
            DO 10 I=1,NCY
            A(I)=0.0
            DO 11 J=1,NCX
            K1= K + (J-1)*NDXR   + (NDX - 1) * NDXR * NDXC
            J1= J + (I-1)*NDYR    + (NDY - 1) * NDYR*NDYC
      11    A(I)=A(I) + X(K1)*Y(J1)
      10    CONTINUE
            DO 13 I=1,NCY
            I1= K + (I-1)*NDZR
      13    Z(I1)=A(I)
      12    CONTINUE
            RETURN
            END
```

```
C       ASP DATA MANIPULATOR 2 -- ADM2
C       P MCKINNON FOR D BENSON                                  OCT 69
C       NRM= MAX NO, OF RUNS
C       DT1= CHANGE IN TIME BETWEEN GROUP COMPUTATIONS IN STAGE 1 OF FINE
C            ALIGN OF ASP
C       DT2= CHANGE IN TIME BETWEEN GROUP COMPUTATIONS IN STAGE 2 OF FINE
C            ALIGN  OF ASP
C       NA1= NO, OF COMPUTATIONS IN STAGE 1
C       NA2= NO, OF COMPUTATIONS IN STAGE 2
C       NOTE( IF NA2=0, ONLY COMPUTE STAGE 1
C       NAV= AVERAGING NUMBER
C       IF NAV=0, RUN 2 STAGES; DON;T AVERAGE
C       IF NAV IS GREATER THAN 0 FIRST RUN REGULAR RUN (1 OR 2 STAGES)
C               ;THEN RUN LAST STAGE AGAIN AVERAGING ACCORDING TO NAV
C       NOTE; TO CHANGE NAV IN THE SAME RUN PUT IN ANOTHER CARD WITH
C             DT1=DT2=0,0, THIS MAY BE DONE AS MANY TIMES AS NECESSARY,
C             HOWEVER EACH NEW PASS MUST BE COUNTED AS A RUN--ADD 1 TO NRM
C       IF ID1=0 OR 1, READ DATA FROM CARD READER ONLY
C       IF ID1=2, READ FROM CARDS  AND PUT ON TAPE FOR NIGHT RUNNING
C       IF ID2=0 OR 1, RUN ENTIRE PROGRAM
C       IF ID2=2, RUN U SECTION ONLY
C       IF ID2=3, RUN CALCULATIONS,LEAVE OUT U SECTION
C       IF ID3=0 OR 1, DIVIDE U;S BY 3600.0
C       IF ID3=2, DO NOT DIVIDE U;S BY 3600,0
        COMMON A(5),E(3,5,30), D(3,4,30), X(5,5),ZV(3,30), SETZ(5),
      1Y(6,6),CBNP(3,3), CBNPP(3,3), CBNT(3,3), THE(3),ER(3),SETE(5,5),
      2U(5,5),SV(5,5),EST(3,3),YV(5),CBNPPT(3,3)
        DIMENSION GTGI(5,5)
        EQUIVALENCE (Y,GTGI)
C       SET UP SIZES
        NRS=3
        NDRS=3
        NRE=3
        NCE=5
        NDCE=5
        NRD=3
        NCD=4
C       SET CONSTANTS
        CG=32,17
        PI= 3,1415927
        TORAD= PI/180,0
        WE= 15,04196*TORAD/3600,0
        READ 10,NRM,ID1,ID2,ID3
   10   FORMAT(4I10)
        IF(ID2 ,EQ, 0),ID2= ID2 + 1
        IF(ID1 ,EQ, 0), ID1= ID1 + 1
        IF(ID3 ,EQ, 0), ID3= ID3 + 1
        IF( ID1 ,EQ, 2), REWIND 4
        IF(ID2 ,NE, 2),REWIND 5
        REWIND 6
        GO TO (13,14),ID1
   14   DO 15 NR=1,NRM
   18   READ 16, DT1,DT2,NA1,NA2,SL,AX,BI,NAV
   16   FORMAT(2F10,0,2I10,3F10,0,I10)
```

```
      GO TO (17,15),ID1
   15 WRITE (4) DT1,DT2,NA1,NA2,SL,AX,BI,NAV
      REWIND 4
   13 DO 11 NR=1,NRM
      PRINT 12,NR,NRM
   12 FORMAT(31H1ASP DATA MANIPULATOR 2 -- ADM2//5H RUN ,I4,4H OF ,I4///
     1)
      GO TO (18,19),ID1
   19 READ (4) DT1,DT2,NA1,NA2,SL,AX,BI,NAV
   17 CONTINUE
      IF(ID2 .EQ. 2), GO TO 93
      IF((DT1 + DT2) .NE. 0.0), GO TO 93
      DT1=DT10
      DT2=DT20
      NA1=NA10
      NA2=NA20
      IF(NA2 .NE. 0), GO TO 94
      DO 95 I=1,4
   95 BACKSPACE 5
      GO TO 96
   94 BACKSPACE 5
   96 PRINT 20, DT1,NA1,DT2,NA2,AX,BI,SL,NAV
      GO TO 64
   93 REWIND 6
      ISWC=1
       PRINT 20, DT1,NA1,DT2,NA2,AX,BI,SL,NAV
   20 FORMAT(6H DT1= ,E15.8//6H NA1= ,I12//6H DT2= ,E15.8//6H NA2= ,I12/
     1/6H  AX= ,E15.8//6H  BI= ,E15.8//6H  SL= ,E15.8//6H  NA= ,I12///)
C     READ ZV FROM TAPE 5
C     ADVANCE TAPE 5 TO ZV
      NA1P1= NA1 + 1
      IF ( ID2 .EQ. 2), GO TO 71
      DO 36 I=1,NA1P1
      READ (5) DUM
   36 READ (5) DUM
      READ (5) NOP, ((ZV(I,J),I=1,NRS),J=1,NA1)
C     REPOSITION TAPE 5 TO BEGINNING OF RUN
      GO TO 61
   61 BACKSPACE 5
      DO 35 I=1,NA1
      BACKSPACE 5
   35 BACKSPACE 5
      PRINT 43
      CALL PRVL3(2HZV,ZV,NRS,NA1,1,NDRS,0,NA1,1)
   43 FORMAT(1H1)
      PRINT 43
   71 SL= SL * TORAD
      SINL= SIN(SL)
      COSL= COS(SL)
      SIL2= SINL* SINL
      COL2= COSL*COSL
      ISWA=1
      REWIND 6
      NS=1
```

```
            NSKL=1
            NSCP=3
      C     COMPUTE E AND D
            IF (NR .EQ, 1), GO TO 21
            IF(DT10 .NE, DT1), GO TO 21
            IF( NA10 .GE, NA1), GO TO 23
       21   DT= DT1
            NA= NA1
            GO TO 25
       23   NA= NA1
            DT= DT1
       26   READ(6) (((E(I,J,K),I=1,NRE),J=1,NCE),K=1,NA)
            READ (6) (((D(I,J,K),I=1,NRD),J=1,NCD),K=1,NA)
            ISWB=1
            GO TO 22
       25   DTD2= DT/2,0
            ISWB=2
            DO 24 I=1,NA
            T1= DTD2 + DT * (I-1)
            T2= I*DT
            SWT = SIN(WE*T1)
            CWT= COS(WE*T1)
            E(1,1,I)= CG* SINL* SWT
            E(1,2,I)= CG* CWT
            E(1,3,I)= CG* COSL*SWT
            E(1,4,I)= -CG*SINL*(1,0-CWT)/WE
            E(1,5,I)= 0,0
            E(2,1,I)= CG* (-SIL2*CWT - COL2)
            E(2,2,I)= CG*SINL*SWT
            E(2,3,I)= CG*(-SINL*COSL)*(CWT-1,0)
            E(2,4,I)= CG*(SIL2*SWT/WE + COL2*T1)
            E(2,5,I)=0,0
            E(3,1,I)=0,0
            E(3,2,I)=0,0
            E(3,3,I)=0,0
            E(3,4,I)=0,0
            E(3,5,I)=1.0
            IF (I .LT, 4), GO TO 24
            SWT= SIN(WE*T2)
            CWT = COS(WE*T2)
            D(1,1,I)= SIL2*CWT+COL2
            D(1,2,I)= -SINL*SWT
            D(1,3,I)= (SINL*COSL)*(CWT - 1.0)
            D(1,4,I)= -SIL2*SWT/WE - COL2*T2
            D(2,1,I)= -D(1,2,I)
            D(2,2,I)= CWT
            D(2,3,I)= COSL*SWT
            D(2,4,I)= -SINL*(1,0-CWT)/WE
            D(3,1,I)= D(1,3,I)
            D(3,2,I)= -D(2,3,I)
            D(3,3,I)= COL2*CWT + SIL2
            D(3,4,I)= -SINL*COSL*(SWT/WE - T2)
       24   CONTINUE
            WRITE (6) (((E(I,J,K),I=1,NRE),J=1,NCE),K=1,NA)
      WRITE (6) (((D(I,J,K),I=1,NRD),J=1,NCD),K=1,NA)
      DO 27 I=1,NCE
      DO 27 J=1,NCE
   27 SETE(I,J)=0,0
      DO 28 K=1,3
      DO 29 KR=1,NCE
      DO 29 J=1,NCE
      DO 29 I=1,NRE
   29 SETE(KR,J)= SETE(KR,J) + E(I,KR,K)*E(I,J,K)
   28 CONTINUE
   22 DO 38 I=1,NCE
   38 SETZ(I)= 0,0
      DO 40 K=1,3
      DO 39 KR=1,NCE
      DO 39 I=1,NRE
   39 SETZ(KR)= SETZ(KR) + E(I,KR,K)*ZV(I,K)
   40 CONTINUE
      IF(ID2 .EQ. 2), GO TO 70
C     ADVANCE TAPE 5 TO THE BEGINNING OF GROUP 4
      ICBC= NSCP
      DO 45 I=1,NSCP
      READ (5) DUM
   45 READ (5) DUM
   70 ISL=1
      IPC=1
      DO 30 K=4,NA
      GO TO (33,34),ISWB
   34 DO 31 KR=1,NCE
       DO 31 J=1,NCE
      DO 31 I=1,NRE
   31 SETE(KR,J)= SETE(KR,J) + E(I,KR,K)*E(I,J,K)
      DO 32 I=1,NCE
      DO 32 J=1,NCE
   32 GTGI(I,J)=SETE(I,J)
      CALL XINV(GTGI,Y,NCE,NCE)
      WRITE (6) ((GTGI(I,J),I=1,NCE),J=1,NCE)
      GO TO 37
   33 READ (6) ((GTGI(I,J),I=1,NCE),J=1,NCE)
   37 IF (ID2 .EQ. 2), GO TO 30
      DO 41 KR=1,NCE
      DO 41 I=1,NRE
   41 SETZ(KR)= SETZ(KR) + E(I,KR,K)*ZV(I,K)
      DO 42 I=1,NCE
      YV(I)=0,0
      DO 42 J=1,NCE
   42 YV(I)= YV(I) + GTGI(I,J)*SETZ(J)
      CALL PRVL3(1HY,YV,1,NCE,1,1,0,K,ISL)
      ISL=1
      DO 44 I=1,NRD
      THE(I)=0,0
      DO 44 J=1,NCD
   44 THE(I)= THE(I) + D(I,J,K)*YV(J)
      CALL PRVL3(5HTHETA,THE,1,NRD,1,1,0,K,ISL)
C     COMPUTE EST
```

```
      CTX= COS(THE(1))
      STX= SIN (THE(1))
      CTY= COS(THE(2))
      STY= SIN(THE(2))
      CTZ= COS(THE(3))
      STZ= SIN(THE(3))
      EST(1,1)= CTY*CTZ
      EST(1,2)= -CTX*STZ + STX*STY*CTZ
      EST(1,3)= STX*STZ + CTX*STY*CTZ
      EST(2,1)= CTY*STZ
      EST(2,2)= CTX*CTZ + STX*STY*STZ
      EST(2,3)= -STX*CTZ + CTX*STY*STZ
      EST(3,1)= -STY
      EST(3,2)= STX*CTY
      EST(3,3)= CTX*CTY
      DO 82 NSKP=NS,NSKL
      READ (5) NOP, ((CBNT(I,J),I=1,NRS),J=1,NRS)
      READ (5) NOP1, ((CBNP(I,J),I=1,NRS),J=1,NRS)
   82  CONTINUE
      ICRC= ICBC + NSKL
      CALL PRVL3(4HCBNT,CBNT,NRS,NRS,1,NDRS,0,ICRC,ISL)
      CALL PRVL3(4HCBNP,CBNP,NRS,NRS,1,NDRS,0,ICRC,ISL)
C     COMPUTE CBNPP= EST*CBNP
   83 CALL GMP3(EST,CBNP,CBNPP,NRS,NRS,1,NRS,1,NDRS,0,NDRS,0,NDRS)
      CALL PRVL3(5HCBNPP,CBNPP,NRS,NRS,1,NDRS,0,K,ISL)
C     COMPUTE ER(3)
      DO 74 I=1,NRS
      DO 74 J=1,NRS
   74 CBNPPT(I,J)= CBNPP(J,I)
      CALL GMP3(CBNT,CBNPPT,X,NRS,NRS,1,NRS,1,NDRS,0,NDRS,0,NDCE)
      ER(1) = -X(2,3)
      ER(2)= X(1,3)
      ER(3)= -X(1,2)
      CALL PRVL3(2HER,ER,1,NRS,1,1,0,K,ISL)
      IPC= IPC + 1
      IF (IPC .LE. 1), GO TO 46
      ISL=2
      IPC=1
   46 CONTINUE
   30 CONTINUE
      IF (ID2 .EQ. 3), GO TO 72
C     COMPUTE U MATRICES
      DO 48 I=1,NRD
      DO 48 J=1,NCD
   48 X(I,J)= D(I,J,NA)
      NCDP1= NCD + 1
      NRDP1= NRD +1
      NRDP2= NRD + 2
      DO 50 I=NRDP1,NRDP2
      DO 49 J=1,NCDP1
   49 X(I,J)=0.0
   50 X(I,I)=1.0
      DO 67 I=1,NRD
   67 X(I,NCDP1) =0.0
```

```
        CALL GMP3(X,GTGI,X,NCE,NCE,1,NCE,1,NDCE,0,NDCE,0,NDCE)
        DO 51 I=1,NCE
        DO 51 J=1,NCE
 51     SV(I,J)=0.0
        DO 66 I=1,NRE
 66     SV(I,I)=BI
        SV(4,4)=AX
        SV(5,5)=1.0
        CALL GMP3(SV,X,SV,NCE,NCE,1,NCE,1,NDCE,0,NDCE,0,NDCE)
        CALL PRVL3(5HC-D-G,SV,NCE,NCE,1,NDCE,0,NA,2)
        IPC=1
        ISL=2
        DO 52 K=1,NA
        DO 47 I=1,NRE
        DO 47 J=1,NCE
 47     X(J,I)= E(I,J,K)
        CALL GMP3(SV,X,U,NCE,NCE,1,NRE,1,NDCE,0,NDCE,0,NDCE)
        GO TO (88,89),ID3
 88     DTD=DT/3600.0
        DO 68 I=1,NCE
        DO 68 J=1,NRE
 68     U(I,J)= U(I,J)/ DTD
 89     CALL PRVL3(1HU,U,NCE,NRE,1,NDCE,0,K,ISL)
        ISL= 1
        IPC= IPC + 1
        IF(IPC .LE. 4), GO TO 53
        ISL=2
        IPC=1
C       PUNCH U
 53     NCEM1= NCE - 1
        DO 54 I=1,NCEM1
        PUNCH 55, (I,J,K,U(I,J),J=1,NRE)
 54     CONTINUE
 55     FORMAT(2X,3I2,E15.8,3I2,E15.8,3I2,E15.8)
        I=1
        J=1
        HOL=1HS
        PUNCH 56,  I,J,K,U(NCE,NRE),HOL
 56     FORMAT(2X,3I2,E15.8,56X,A1)
 52     CONTINUE
        I=9
        PUNCH 57,I
 57     FORMAT(I1)
 72     GO TO (63,64,77), ISWA
C       ADVANCE TAPE 5
 63     IF (ID2 .EQ. 2), GO TO 75
        GO TO 76
 75     IF (DT1 .NE. DT2), GO TO 73
        GO TO 11
 76     IF (NA2 .NE. 0), GO TO 90
        NAN=NA1
        DTN=DT1
        GO TO 64
 90     READ (5) DUM
```

```
      READ (5) DUM
      READ (5) DUM
      READ (5) DUM
      NAN= NA2
      DTN=DT2
      DO 59 I=1,NA2
      READ (5) DUM
   59 READ (5) DUM
      READ (5) NOP, ((ZV(I,J),I=1,NRS),J=1,NA2)
      GO TO 62
   62 BACKSPACE 5
      DO 60 I=1,NA2
      BACKSPACE 5
   60 BACKSPACE 5
      PRINT 43
      CALL PRVL3(2HZV,ZV,NRS,NA2,1,NDRS,0,NA2,1)
      PRINT 43
   73 DT= DT2
      ISWA= 2
      NA= NA2
      IF(NR .EQ. 1), GO TO 25
      IF(NA10 .NE. NA1), GO TO 25
      IF (DT20 .NE. DT2), GO TO 25
      IF(NA20 .GE. NA2), GO TO 26
      GO TO 25
   64 IF (NAV .EQ. 0), GO TO 77
      DO 81 I=1,NAN
      BACKSPACE 5
   81 BACKSPACE 5
      NSKL= NAV
      NS=1
      NSCP= NAV*3
      ISWA=3
      DT3=DTN*NAV
      NA3=NAN/NAV
      IF(NAN .GE.(NSCP + NAV)), GO TO 87
      DT3=0.0
      NA3=0
      GO TO 77
   87 DT= DT3
      NA= NA3
      K1 = 1
      K2=NAV
      K3=NAV
      DO 78 K=1,NA
      DO 80 I=1,NRS
      SM=0.0
      DO 79 J=K1,K3
   79 SM= SM + ZV(I,J)
   80 ZV(I,K)= SM/NAV
      K1= K3 + 1
   78 K3= K3 + K2
      PRINT 43
      CALL PRVL3(2HZV,ZV,NRS,NA,1,NDRS,0,NA,1)
      PRINT 43
      GO TO (92,25),ISWC
92    IF(NR .EQ, 1), GO TO 25
      IF (NA10 ,NE, NA1), GO TO 25
      IF (NA20 .NE, NA2), GO TO 25
      IF(DT30,NE, DT3), GO TO 25
      IF (NA30 .GE, NA3), GO TO 26
      GO TO 25
77    DT10= DT1
      NA10=NA1
      DT20= DT2
      NA20=NA2
      DT30= DT3
      NA30=NA3
      ISWC=2
      IF (ID2 .EQ, 2), GO TO 11
      NS= NAV*NA3 + 1
      IF(NS .GT, NAN), GO TO 86
      DO 85 I=NS,NAN
      READ (5) DUM
85    READ (5) DUM
86    READ (5) NOP,((ZV(I,J),I=1,NRS),J=1,NAN)
      IF (NA2 .NE, 0), GO TO 11
      READ (5) DUM
      READ (5) DUM
      READ (5) DUM
11    CONTINUE
      IF (ID1 .EQ, 2), REWIND 4
      IF(ID2 .NE, 2), REWIND 5
      PRINT 65
65    FORMAT(9H1END ADM2)
      STOP
      END
```

```
        SUBROUTINE XINV (Y,X,NRX,NCX)
C       SUBROUTINE TO INVERT A MATRIX AND STORE IT ON ITSELF
C       P MORTELLITE                              9-9-65
C       NOTE-- THE DIMENSION FOR ROWS AND COLUMNS MUST BE INCREASED BY 1
        DIMENSION X(6,6),Y(5,5)
C       SPREAD OUT Y INTO X
        J1=NCX
        DO 19 J=2,NCX
        I1=NRX
        DO 20 I=1,NRX
        X(I1,J1)=Y(I1,J1)
  20    I1=I1-1
  19    J1=J1-1
        NR1=NRX + 1
        NC1=NCX+1
        DO 10 II=1,NRX
C       ADD A UNIT COLUMN VECTOR
        X(1,NC1)=1.
        DO 11 I=2,NRX
  11    X(I,NC1)=0.0
C       COMPUTE PIVOT ROW
        IF(X(1,1))16,17,16
  17    PRINT 18
  18    FORMAT(44H1ERROR,DIVISION BY 0 IN INVERSION ROUTINE S6)
  16    DO 12 J=1,NCX
  12    X(NR1,J)=X(1,J+1)/X(1,1)
C       COMPUTE NEW VALUE FOR ALL OTHER ELEMENTS
        DO 13 I=2,NRX
        DO 14 J=1,NCX
  14    X(I-1,J)=X(I,J+1)-X(I,1)*X(NR1,J)
  13    CONTINUE
C       MOVE PIVOT ROW UP TO NTH ROW
        DO 15  J=1,NCX
  15       X(NRX,J)=X(NR1,J)
  10    CONTINUE
C       SHRINK X INTO Y
        DO 21 J=2,NCX
        DO 22 I=1,NRX
  22    Y(I,J)=X(I,J)
  21    CONTINUE
        RETURN
        END
        SUBROUTINE PRVL3(HOL,A,NRF,NCF,NDF,NDR,NDC,N,ISL)
C       A SUBROUTINE TO PRINT A VARIABLE LENGTH MATRIX OF 1,2 OR 3 DIMENS
C       THE PAGE IS SLEWED OR NOT ACCORDING TO ISL=2 OR 1, RESPECTIVELY
C       P MORTELLITE                              4-12-66
        DIMENSION A(1)
        K= (NCF + 6)/7
        K1=1
        K2=7
        DO 10 I=1,K
        IF(NCF-K2)15,15,14
   15   K3=NCF
        GO TO 16
    14  K3=K2
   16   GO TO (20,18),ISL
   20   PRINT 21
   21   FORMAT(///1H )
        GO TO 17
   18   PRINT 19
   19   FORMAT(1H1)
   17   PRINT               11,N,    HOL,( J ,J=K1,K3)
   11   FORMAT(15H SAMPLE NUMBER ,I7,
       1     //1X,A6,7(8X,I2,7X))
        LIM= K3 * NDR  + (NDF -1) * NDR * NDC
        DO 12 KV=1,NRF
        KV1= (K1-1)*NDR + KV  + (NDF -1)*NDR*NDC
        PRINT       13,        KV,(A(II),II=KV1,LIM,NDR)
      13 FORMAT (3X,I2,2X,7E17,10)
      12 CONTINUE
        K1=K1 + 7
      10 K2 = K2 + 7
        RETURN
        END
```

```
          SUBROUTINE GMP3(X,Y,Z,NRX ,NCX,NDX,NCY,NDY,NDXR,NDXC,NDYR,NDYC,
         1 NDZR)
    C     SUBROUTINE TO MULTIPLY X*Y AND STORE IN Z OR X
    C     P MCKINNON                                   OCT 1967
          DIMENSION X(1),Y(1),Z(1),A(1)
          COMMON A
          DO 12 K=1,NRX
          DO 10 I=1,NCY
          A(I)=0.0
          DO 11 J=1,NCX
          K1= K + (J-1)*NDXR  + (NDX - 1) * NDXR * NDXC
          J1= J + (I-1)*NDYR   + (NDY - 1) * NDYR*NDYC
    11    A(I)=A(I) + X(K1)*Y(J1)
    10    CONTINUE
          DO 13 I=1,NCY
          I1= K + (I-1)*NDZR
    13    Z(I1)=A(I)
    12    CONTINUE
          RETURN
          END
```

APPENDIX C

THEORETICAL STUDIES INVOLVED IN ALIGNMENT AND CALIBRATION OF THE V/STOL STRAPDOWN INERTIAL SYSTEM

This appendix describes the various theoretical approaches taken in the alignment and calibration problem when the measurements were assumed doubly integrated accelerometer outputs. Indicative numerical results of these studies are given where applicable. The approaches discussed herein were discarded in favor of the measurement situation given in Chapter 5. However, they are documented so that future researchers working on similar problems will have the vicarious experience of the current researchers.

Discussion in the appendix can be divided into three sections. The first is optimal theoretical error analysis, which is determination of the minimum alignment error and calibration errors if a perfectly designed Kalman filter were used. Both continuous and discrete measurement Kalman filters are considered. The second part is definition and performance analysis of a two-stage alignment algorithm. Two approaches to the algorithm are given. One is based on the optimal Kalman filter, and the second is based on a least squares approach to the problem. A phenomenon called "P" sensitivity is observed. The third part is definition and performance analysis of a three-stage alignment algorithm. This algorithm eliminates the "P" sensitivity. Again the two algorithms based respectively on the Kalman filter and least squares are considered.

Comparisons and contrasts associated with each part of the discussions are made.

C.1 OPTIMAL ESTIMATION STUDIES

In order to establish minimum errors in alignment, optimal estimation studies have been carried out. These essentially involve analyzing the covariance of the outputs of a Kalman filter. The available measurements considered are doubly integrated effective north and east accelerometer outputs.

- THE SYSTEM

In the studies, the following model has been taken for the system measurement configuration (representing a system at LAT = 45°).

System:

$$
\begin{aligned}
\dot{\epsilon}_N &= -.1851\,\epsilon_E - R_{N1} - R_{N2} \\
\dot{\epsilon}_E &= .1851\,\epsilon_N + .1851\,\epsilon_z - R_{E1} - R_{E2} \\
\dot{\epsilon}_z &= -.1851\,\epsilon_E - R_{z1} - R_{z2} \\
\dot{R}_{N1} &= 0 \\
\dot{R}_{E1} &= 0 \\
\dot{R}_{z1} &= 0 \\
\dot{z}_{1N} &= g\,\epsilon_E + A_{N1} + A_{N2} \\
\dot{z}_{1E} &= -g\,\epsilon_N + A_{E1} + A_{E2} \\
\dot{z}_{2N} &= z_{1N} \\
\dot{z}_{2E} &= z_{1E} \\
\dot{A}_{N1} &= 0 \\
\dot{A}_{E1} &= 0
\end{aligned}
\tag{C-1}
$$

Measurements:

$$z_1 = z_{2N} + w_1$$
$$z_2 = z_{2E} + w_2 \qquad \text{(C-2)}$$

In the above $\epsilon_N, \epsilon_E, \epsilon_z$ are the misalignments of the analytic platform about the North, East and z axes, respectively.

R_{N1}, R_{E1}, R_{z1} are effective North, East and z gyro biases.

$\dot{z}_{1N}$, $\dot{z}_{1E}$ are, respectively, those components of the North and East accelerometer outputs caused by inertial system errors.

A_{N1}, A_{E1} are respectively the North and East accelerometer biases.

R_{N2}, R_{E2}, R_{z2} are respectively the random components of the North, East and z gyro drift rates. For simplicity these terms have been assumed, in some cases, as white noise. The variance of the white noise is chosen so that the RMS random drift after one hour is some specified amount.

A_{N2}, A_{E2} are respectively random components of the North and East accelerometer errors. For simplicity they are assumed white noise whose variance is chosen so that after one hour a specified velocity error is caused by the noise.

The terms w_1 and w_2 represent the linear excursions of the V/STOL from its base position. Because the vehicle is vibrating at a high frequency, this term can be assumed white. The variance of the terms are the rms excursions of the aircraft. In this case the vehicle vibrational acceleration spectrum is irrelevant because the system is 'told' that the aircraft is not going very far even though it is experiencing accelerations which are a large percentage of g.

- SOME INTERESTING RESULTS

The optimal results for the above measurement configuration were given in Tables 7.2-1 and 7.2-2 in [24]. In Table 7.2-1 the results assumed continuous measurement and an rms vibration level of one foot. Table 7.2-2 was the result of a gross error which implied an rms vibration level of 1000 feet. The second result was presented because it was found that alignment accuracy deteriorated by a surprisingly small amount for the absurd vibration environment. The conclusion one could then draw is that this particular alignment scheme is only slightly sensitive to the level of vehicle vibration.

Other studies which have not been previously documented have also been carried out. One study involved determining which states of the inertial system could be estimated. The following system was specified for study.

Initial Attitude Misalignment	.50°	
Nominal Gyro Bias	.01°/hr	
Nominal Random Gyro Drift Rate	.005°/hr (white)	(C-3)
Accelerometer Bias	10 μg	
Random Accelerometer Bias	.175 kts/hr	
Vibration Level	1 ft	
Sampling Interval	Continuous	

It was found that heading and tilts could be estimated to close to their theoretical limit. This is the same conclusion which was reached in Chapter 5. East gyro bias could not be estimated. The ability to estimate the North and z gyro biases depended on the amount of random noise on the gyro drift rates. Fig. C-1 shows normalized curves of percentage estimation of the bias values[(rms bias value - rms error in estimation of bias)/rms bias] versus percentage random noise on the gyro drift rate [(deg/hr, random)/(deg/hr bias)]. Biases of .01 and .02 deg/hr were considered in the construction of these curves.

Note that the smallest amount of random error on the z gyro drift rate effectively wiped out any estimation of that state. The North gyro bias could be estimated with small amounts of random error on the drift rate. One could therefore come to the same conclusions which were listed in Section 5.1., i.e., the states $\epsilon_N, \epsilon_E, \epsilon_z, R_N$ can be estimated effectively.

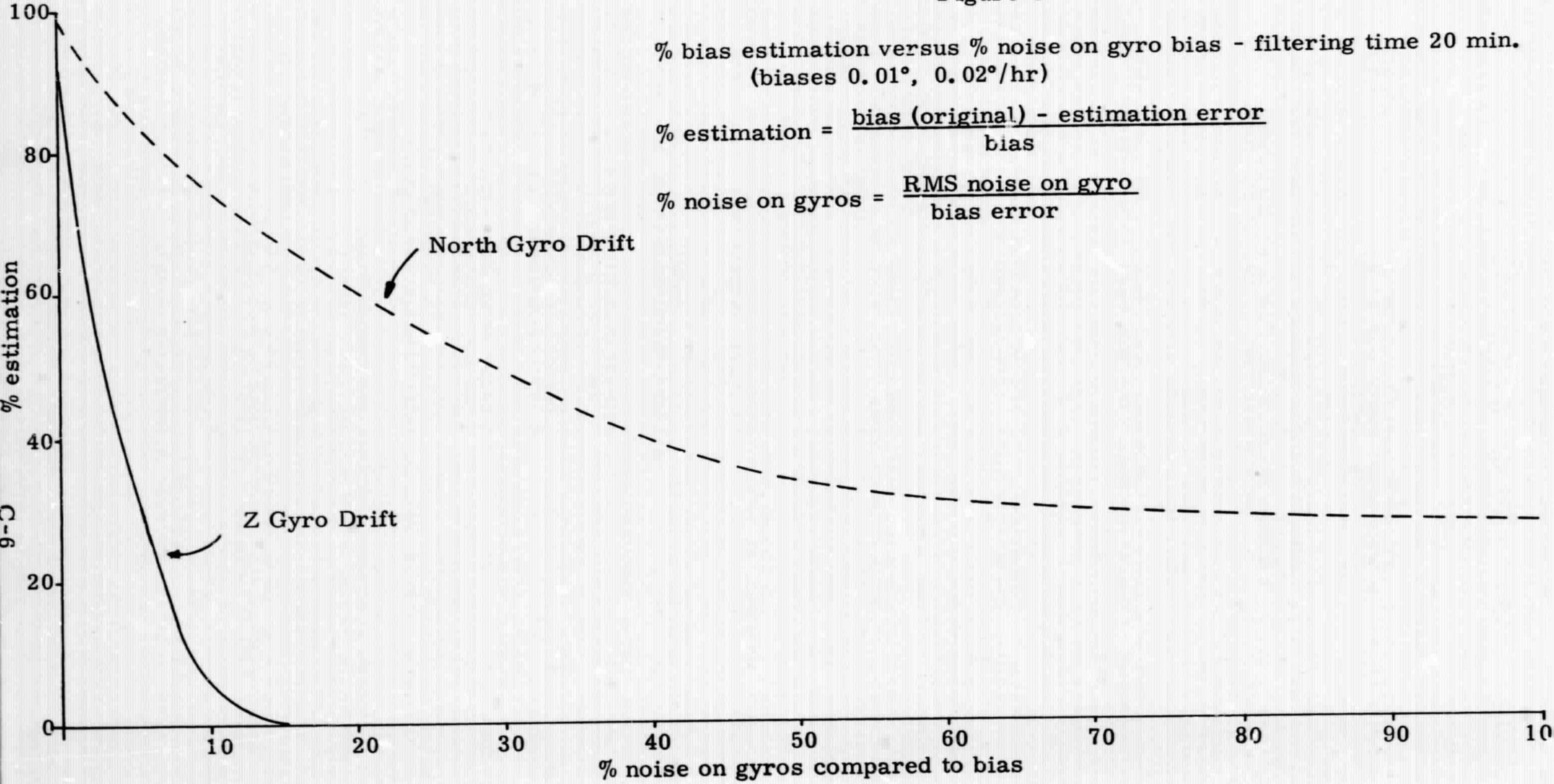
Another study also was carried out to determine the effect of sampling rate on estimation errors. The above nominal statistics were used. Fig. C-2 shows the results. The sampling rate was varied from continuous to two minutes. The total time of filtering was 20 minutes. Note that the misalignment errors did not deteriorate significantly as the sampling interval was increased. This result was in marked contrast with some of the results given in Section 5.1. When the measurements were considered time averaged accelerometer outputs, and a random vibration environment was assumed, there was significant deterioration of the estimation error for increasing sampling intervals.

Figure C-1

% bias estimation versus % noise on gyro bias - filtering time 20 min.
(biases 0.01°, 0.02°/hr)

$$\% \text{ estimation} = \frac{\text{bias (original) - estimation error}}{\text{bias}}$$

$$\% \text{ noise on gyros} = \frac{\text{RMS noise on gyro}}{\text{bias error}}$$



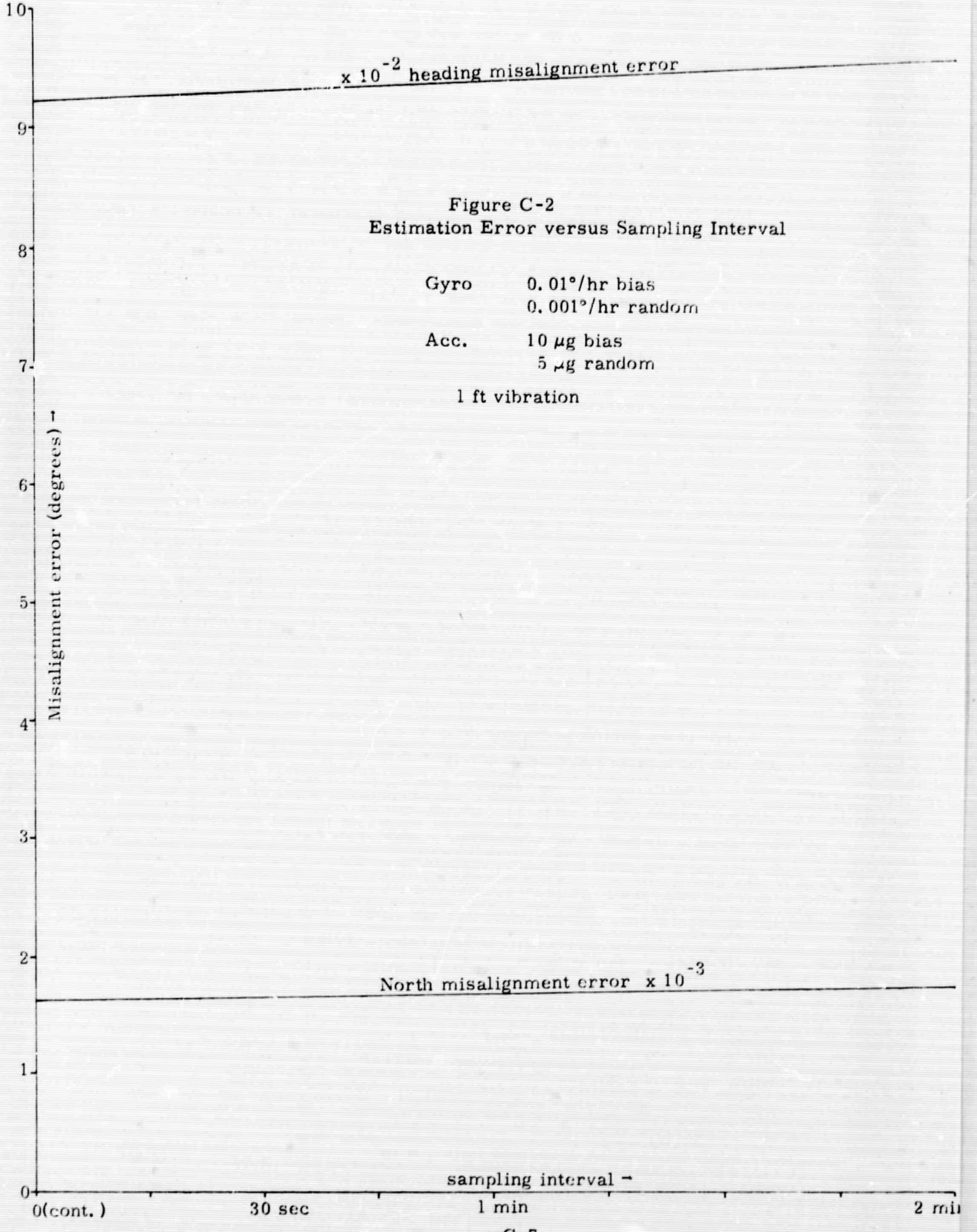


Figure C-2
Estimation Error versus Sampling Interval

Gyro	0.01°/hr bias 0.001°/hr random
Acc.	10 μg bias 5 μg random

1 ft vibration

Other optimal studies have also been carried out for various statistical situations. The statistics and the results are listed below.

Case 1

Statistical situation:

Acc. Bias:	20 μg
Random Acc. Error:	.175 kts/hr
Gyro Bias:	.05°/hr
Random Gyro Drift Rate:	.01°/hr (white)
Initial Attitude Error:	.5°
Vibration Level:	1 ft
Sampling Interval:	2 min

(C-4)

RMS ERROR RESULTS

Total Filtering Time	10 min	20 min
State:		
ϵ_N	8.1 $\widehat{\text{sec}}$	7.6 $\widehat{\text{sec}}$
ϵ_E	8.2 $\widehat{\text{sec}}$	7.6 $\widehat{\text{sec}}$
ϵ_z	.266°	.249°
R_N	.027°/hr	.0178°/hr

(C-5)

Case 2

Statistical situation:

Acc. Bias:	20μg	
Random Acc. Error:	.1 kts/hr	
Gyro Bias:	.05°/hr	
Random Gyro Drift Rate:	0.0	(C-6)
Initial Attitude Error:	.5°	
Vibration:	1 ft	
Sampling Interval:	2 min	

RMS ERROR RESULTS

Total Filtering Time	10 min	20 min
State:		
ϵ_N	4.56 $\widehat{\text{sec}}$	4.35 $\widehat{\text{sec}}$
ϵ_E	4.56 $\widehat{\text{sec}}$	4.35 $\widehat{\text{sec}}$
ϵ_z	.238°	.233°
R_N	.006°/hr	.0021°/hr

(C-7)

Case 3

Statistical situation:

Acc. Bias:	20 μg
Random Acc. Error:	.175 kts/hr
Gyro Bias:	.05°/hr
Random Gyro Drift Rate:	.005°/hr Markovian with a 2-hr correlation time
Initial Attitude Error:	.5°
Vibration:	1 ft
Sampling Time:	2 min

RMS ERROR RESULTS

Total Filtering Time	10 min	20 min
State:		
ϵ_N	5.6 $\widehat{\text{sec}}$	4.6 $\widehat{\text{sec}}$
ϵ_E	5.6 $\widehat{\text{sec}}$	4.6 $\widehat{\text{sec}}$
ϵ_z	.245°	.238°
R_N	.015°/hr	.006°/hr

(C-9)

From these results one can observe that:

1. The alignment accuracy depends in part on the random component of the accelerometer error. The random error has less effect for longer filtering times.
2. A 'white' gyro drift rate (which physically does not exist) has a much larger effect on the alignment accuracy than does a random gyro drift with a time constant of several hours.
3. Estimation of heading and North gyro drift is also affected by the characteristics of the random gyro drift.

C.2 Performance Analysis of the Two-Stage Alignment Scheme

The two-stage alignment scheme consisted of an "optical" crude align and one fine align. The following statistical situation was assumed.

Acc. Bias:	20 μg	
Random Acc. Error:	.175 kts/hr	
Gyro Bias:	.05°/hr	
Random Gyro Error:	.01°/hr (white)	(C-10)
Vibration Level:	1 ft	
Initial conditions:	After assumed crude align, all states initially uncorrelated (a diagonal P)	
Attitude Error:	.5°	
RMS Velocity Error:	1 kt	
RMS Position Error:	.1 nm	
Sampling Interval:	2 min	

The optimal filter using a sampling interval of 2 min was derived for the above situation. Total filtering time was 10 minutes. The filter was defined as a set of weighting coefficients to be applied to the 5 sets of measurements. The filter and the system were fed into the sensitivity analysis program described briefly in Appendix E. Results of the analysis are shown below. They should repr esent the optimal errors, and they do.

RMS Error (sensitivity) Results for Optimal Filter

State	ϵ_N	ϵ_E	ϵ_z	R_N
	8.1 $\widehat{\text{sec}}$	8.5 $\widehat{\text{sec}}$	.27°	.029°/hr

(C-11)

Another optimal filter was then designed using exactly the same error models except the initial covariance matrix was taken as the original diagonal covariance matrix propagated by two minutes in time. The matrix then had off-diagonal terms. Mathematically, the initial covariance used was

$$P_o(2\text{ min}) = \Phi(2\text{ min})\, P_o\, \Phi(2\text{ min})^T + Q(2\text{ min}) \qquad \text{(C-12)}$$

When the new filter was applied to the statistical system with the diagonal initial covariance matrix, the following result was observed.

RMS Error (sensitivity) Results for Suboptimal Filter

State	ϵ_N	ϵ_E	ϵ_z	R_N
	9.2 $\widehat{\text{sec}}$	10.6 $\widehat{\text{sec}}$	.36°	.051°/hr

(C-13)

Curiously, when the propagated P was used as the initial condition on this filter, the result was almost exactly the same as the optimal result (C-11) for the diagonal P case. What seems to have been observed in Eq. (C-13) is what is call P sensitivity, i.e., the filter is quite sensitive to the un-uncertainty in the knowledge of the initial covariance. This phenomenon baffles the intuition. The author has checked and double checked the numerics in obtaining the solution and has found nothing to indicate that numerical errors are responsible for this effect. One plausible, but heuristic, explanation is that the doubly integrated accelerometer outputs become very large when there are initial tilt angles of .5°. Although the filter does an excellent job in estimating the tilts in "one shot" - percentage wise - the actual value may vary considerably if there are only small imperfections in the filter parameters.

Another filter, based on least squares, was designed to estimate only the four states of interest; a performance analysis was carried out. Unfortunately, no exact algorithms existed in-house for synthesis of such a filter — when the measurements were doubly integrated accelerometer outputs. However, a least squares filter could be approximated by designing the appropriate Kalman filter based on a system undriven by noise, with very large I. C's. The resultant filter was called the approximate least squares filters. It was the Kalman filter designed for the following system.

Acc. Bias:	$A_N = 0$, $A_E = 0$	
Random Acc. Bias:	0.0	
Gyro Bias:	$R_N = 57°/hr \quad R_z, R_E = 0.0$	
Random Gyro Bias:	0.0	(C-14)
<u>Initial Conditions</u>:		
Attitude Error:	57°	
RMS Velocity Error	60 kts	
RMS Position Error	1 nm	

No Initial Correlation between states (a diagonal P)

Sampling Interval:	1 min

The optimal weights were derived for this filter and the sensitivity program was run using the true system model given in Eqs. (C-1), (C-2), (C-10). The results are given below:

Results, RMS error of Approximate Least Squares Filter
Designed for a Very Large Diagonal P Matrix

State	ϵ_N	ϵ_E	ϵ_z	R_N
Sens. Result	9.7 $\overset{\frown}{\text{sec}}$	9. $\overset{\frown}{\text{sec}}$	.327°	.0308°
Optimal Res.	7.2 $\overset{\frown}{\text{sec}}$	7.2$\overset{\frown}{\text{sec}}$	.27°	.027°

(C-15)

Another least squares filter was derived using the model given in Eq. (C-4), but the initial covariance was propagated forward by one minute; i.e., a very large P with off-diagonal elements was used to specify the filter. This filter was analyzed in the sensitivity program and the following results were obtained.

RMS Error Results for an Approximate Least Squares Filter
Specified by a Very Large Diagonal P Propagated by 1 Minute

State	ϵ_N	ϵ_E	ϵ_z	R_N
	18 $\overset{\frown}{\text{sec}}$	17 sec	.425°	.072°/hr

(C-16)

Note that the error has deteriorated considerably. Again this implies that a "P sensitivity" has occurred. Note that the estimate of the North gyro bias is actually worse than the original bias.

The existence of P sensitivity points out the need for a more complicated alignment scheme whose performance is reliable and is not dependent on knowledge of the initial covariance.

C.3 Three-Stage Alignment

In order to eliminate the P sensitivity, the three-stage alignment scheme was developed. The stages were as follows:

Stage I: Crude Align

Measure position for 18 sec, with sampling interval 6 sec.

Estimate only ϵ_N, ϵ_E

Reset C_b^n matrix with this information

Stage II: First Fine Align

Measure position output for 4.5 min, with sampling interval 30 sec.

Estimate ϵ_N, ϵ_E, ϵ_z, R_N

Reset the C_b^n matrix with above information

Stage III: Second Fine Align

Measure position output for 10 min or 20 min with sampling interval of 30 sec and/or 2 min.

Estimate ϵ_N, ϵ_E, ϵ_z, R_N

Reset C_b^n with this information.

A system with the following characteristics was considered.

Acc. Bias:	20μg
Random Acc. Error:	.1 kts/hr
Gyro Biases:	.05°/hr
Random Gyro Drift Rate:	0.0
Initial Condition on the system before the crude align:	
Attitude Error	5°
Velocity Errors	1 kt
Position Errors	.1 nm

(C-17)

Note that some of the elements of Eq. (C-17) have different values than those of Eq. (C-10). These changes represent an improved understanding of the inertial system environment.

Two estimation algorithms were considered in the three-stage alignment study. One was an algorithm based on a full state Kalman filter, and the other was based on an approximate least squares algorithm. The approximate least squares based algorithm was assumed to be the Kalman filter specified by Eq. (C-14). Only the sampling times were different. The Kalman filter algorithm was based on the statistics given in Eq. (C-17). The initial condition on the vertical tilts, however, were taken as 0.081° instead of 5°.* In addition, the third stage Kalman assumed a 0.5° initial heading error. The filter of course varied with the sampling times used. At the beginning of each stage the filter was told that it was starting filtering from time 0. In other words, each stage constituted a filter which was independent of the filter used on the previous stage.

Figure C-3 shows RMS error results using the three-stage alignment, for both approximate least squares and Kalman-based filters. In the first stage only the least squares based filter was used. Results of the Kalman based filter are shown only when there are discernable differences between the errors pertaining to the two algorithms. Note that P sensitivity does not seem to appear. Also note that the approximate least squares filter performs almost as well as the Kalman based filter. This implies that it may be worthwhile to consider only the least squares technique in the recommending an alignment scheme. It is conceptually

* The initial covariance actually specified was the diagonal covariance propagated by a sampling interval.

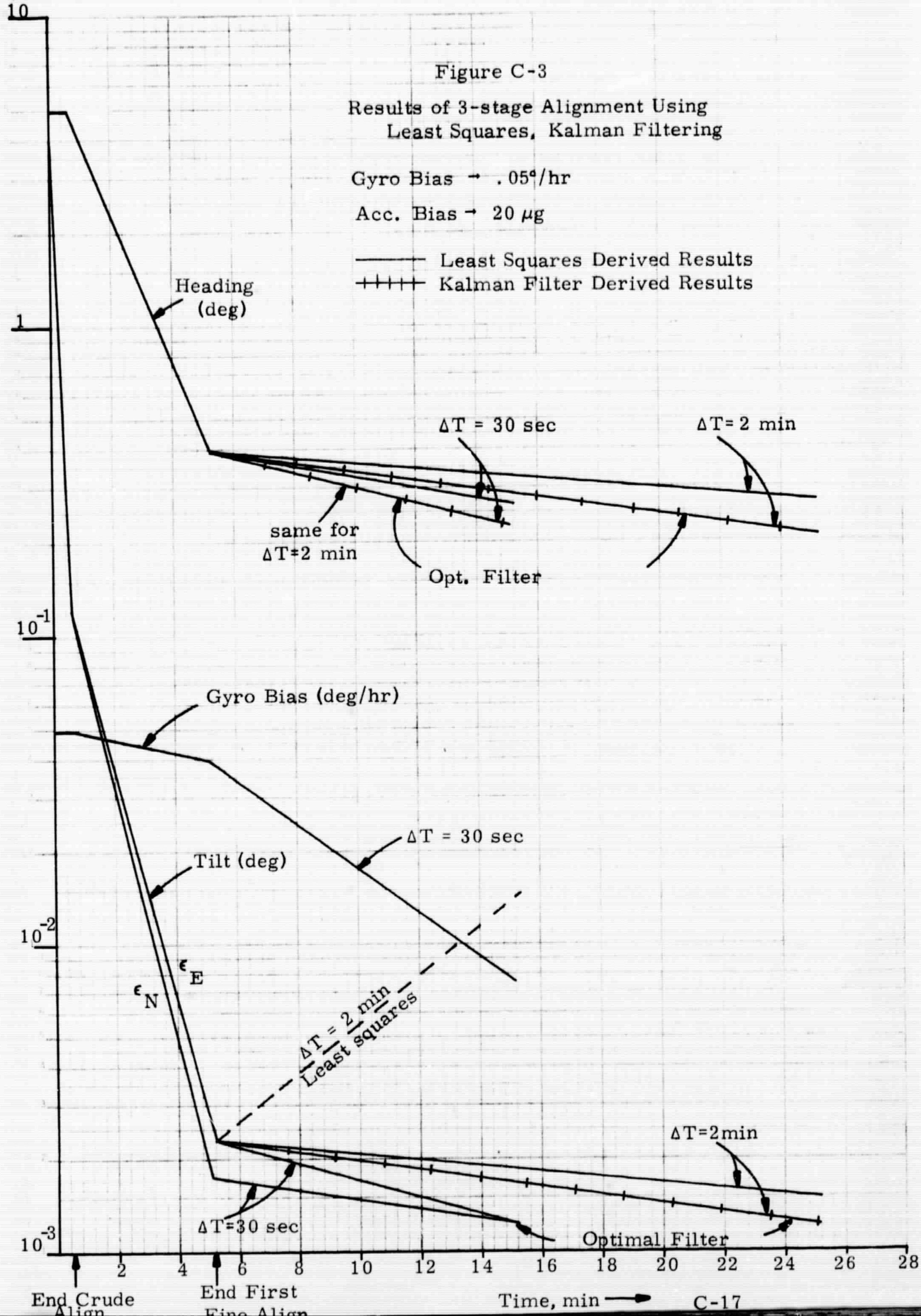


Figure C-3

Results of 3-stage Alignment Using Least Squares, Kalman Filtering

Gyro Bias → .05°/hr

Acc. Bias → 20 μg

simpler than the Kalman approach and does not significantly degrade accuracy.

There is one glaring inconsistency on this figure. It pertains to the result of the least squares filter on the third stage when the sampling interval is 2 minutes, third stage time 10 minutes. In this situation the estimation error goes up instead of down. Note that the inconsistency disappears when the third stage filtering time is 24 minutes. The problem may have been numerical errors in the computation of the estimation coefficients. Or it may have been the result of the same effect which caused P sensitivity. No additional investigation has been performed to find a rational explanation. However, the obvious conclusion which can be drawn from this anomaly is that one should design his algorithms with sampling intervals not greater than 30 seconds.

APPENDIX D

MODELLING THE QUANTIZATION ERROR IN A LINEAR SYSTEM

The inertial sensors on the VSTOL strapdown system are subject to quantization errors. These errors in turn affect the errors in the inertial system. In order to theoretically analyze the effects of these errors, one must be able to model the errors as some sort of linear system. In this appendix the quantization errors are cast into a form which can be modelled in the linear system used to analyze the inertial system.

Both gyros and accelerometers are pulse rebalanced integrating instruments. This implies that every time the acclerometer changes velocity by a specified amount, it emits a ± pulse to rebalance the instrument. The number of pulses is proportional to the total change in velocity. The incremental change in velocity associated with each pulse is called the quantization level. The quantized pulses can then be integrated or summed by the computer to determine the total change in velocity. The fact that the pulse represents a finite change in velocity implies that an error is induced by the quantization.

In a similar manner the gyro emits a rebalancing pulse each time its input axis changes through a finite incremental angle. The quantization level is the incremental angle represented by one pulse. Again quantization leads to an error in establishing the orientation of the platform

The integrating instrument is rebalanced by a quantization level with each pulse. However, it'remembers' the change in velocity (angle) which was observed and not rebalanced. It accounts for the quantization error the next time enough velocity (angle) change has occurred to generate another pulse. Then, however, a new quantization error is generated by a non-quantized level of the changed velocity (angle). This implies that the previous quantization error is wiped out with the generation of the next pulse. A mathematical description of the quantization error can then be written as follows:

$$q_i = -q_{i-1} + u_i \qquad \text{(D-1)}$$

Where q_i is the quantization error at time i. The term u_i is the effect of the new quantization. In general u_i will be a white process. This model applies to both accelerometers and gyros.

We must now incorporate this model into the system-measurement configuration used in alignment. Consider the accelerometer error first. The basic measurement in the alignment scheme is

$$Z_i = \frac{V_i - V_{i-1}}{T}$$

which can be written in terms of the incremental pulses as follows.

$$Z_i = \frac{\sum_{j=0}^{Ki} \Delta V_j - \sum_{j=0}^{K(i-1)} \Delta V_j}{T} \qquad \text{(D-2)}$$

K = number of possible pulses in one sampling interval

Expanding to include the quantization error, this becomes

$$Z = \frac{\sum_{j=1}^{Ki} \Delta V_{Tj} + q_j - \sum_{j=1}^{K(i-1)} \Delta V_{Tj} + q_j}{\Delta T} \qquad \text{(D-3)}$$

where ΔV_{Tj} is the true velocity change between j and j-1. Substituting Eq. (D-1) into (D-3), Eq. (D-3) reduces to

$$Z = \frac{V_{Ti} - V_{Ti-1} + u_i - u_{i-1}}{\Delta T} \qquad \text{(D-4)}$$

Thus the quantization error can be represented by a white noise at time i and that at time i-1. The rms value of the white noise, if assumed gaussian, is (Quantization level/ $\sqrt{12}$).

Inclusion of the gyro quantization error is slightly more complicated. A heuristic deviation follows. The C matrix updating algorithms can be written simply as

$$\tilde{C}^n_{b_{i+1}} = \tilde{C}^n_{b_i} \; [I + \Delta\theta + \delta\phi] \qquad \text{(D-5)}$$

where $\Delta\theta$ is matrix of the true incremental angle change assuming no gyro rate error during a C matrix update interval. $\delta\phi$ is the matrix of quantization errors where

$$\delta\phi_{i+1} = -\delta\phi_i + v_i \qquad \text{(D-1A)}$$

Assuming

$$\tilde{C}^n_b = C^n_b + \delta C \qquad \text{(D-6)}$$

one can write the above equation as:

$$C^n_{bi+1} = C^n_{bi} \; [I + \Delta\theta_{i+1}] \qquad \text{(D-7)}$$

$$\delta C_{i+1} = \delta C_i \, [I + \Delta\theta_{i+1}] + C_i [\delta\phi_{i+1}] \qquad \text{(D-8)}$$

Using Eqs. (D-1), (D-7) in Eq. (D-8), it can be shown that

$$\delta C_{i+2} = \delta C_{i+1} [I + \Delta\theta_{i+2}] + C_{i+1} [\delta\phi_{i+2}]$$

$$\delta C_{i+2} = \delta C_i \; [I + \Delta\theta_{i+1}] \, [I + \Delta\theta_{i+2}] + C_i \delta\phi_{i+1} + C_i [I + \Delta\theta_{i+1}] [\delta\phi_{i+2}] \qquad \text{(D-9)}$$

Assuming $\Delta\theta_i$, $\delta\phi_i$ are small and products are negligible, and substituting (D-1A) into Eq. (D-9), it reduces to

$$\delta C_{i+2} = \delta C_i [I + \Delta\theta_i + \Delta\theta_{i+1}] + C_{i+1} \, v_{i+2} \qquad \text{(D-10)}$$

Proceeding to i+3, it can be shown that

$$\delta C_{i+3} = \delta C_i \; [I + \Delta\theta_i + \Delta\theta_{i+1} + \Delta\theta_{i+2}] + C_{i+2} \; v_{i+3} \qquad \text{(D-11)}$$

It is now apparent that the misalignment error (ignoring gyro drift) is dependent only on the initial misalignment error, true angular rate and the current quantization error. This implies that the error equation governing the misalignment error can be written as a modification of Eq. (4. 3-19). That is

$$\begin{aligned} \dot{\epsilon}' &= -\Omega\epsilon' - R_b \\ \epsilon_i &= \epsilon_i' + C_i v_i \end{aligned} \qquad \text{(D-12)}$$

where v_i is the white noise vector of quantization errors at time i. ϵ' is the tilt vector assuming no quantization error.

When misalignment quantities are estimated, the partially predictable quantity ϵ' is actually estimated. The error covariance of the estimated true misalignment error is then

$$E[\tilde{\epsilon}' \tilde{\epsilon}'^T] + E[v\, v^T] \qquad \text{(D-13)}$$

The tilde represents error in the estimate.

In estimating ϵ', one must also include gyro quantization error. As before, measurements

$$\begin{aligned} Z_{1i} &= (V_{Ni} - V_{Ni-1})/\Delta T \\ Z_{2i} &= (V_{Ei} - V_{Ei-1})/\Delta T \end{aligned} \qquad \text{(D-14)}$$

are used. It can be shown that the quantities V_E, V_N* can be written as continuous differential equations as follows:

$$\begin{aligned} \dot{V}_N &= +g\epsilon_E' + a_{TN} + \Gamma_1 w_1 + A_N \\ \dot{V}_E &= -g\epsilon_N' + a_{TE} + \Gamma_2 w_2 + A_E \end{aligned} \qquad \text{(D-15)}$$

and this quantity can be modelled in the system.

* The integrated accelerometer outputs less accelerometer quantization.

In the above a_{TN}, a_{TE} is the true acceleration in the North and East directions, respectively. Γ_1, Γ_2 represent the effect of the gyro quantization error, where

$$\Gamma_1 = \Gamma_2 = \frac{g\phi}{\sqrt{12}} \sqrt{\Delta_C T}$$

$\Delta_C T$ is the C matrix update time; $g\phi$ is the quantization level times g; and

$$E[w_1(t)\, w_1(t + \tau)] = \delta(\tau)$$

$$E[w_2(t)\, w_2(t + \tau)] = \delta(\tau)$$

$$E[w_1(t)\, w_2(t = \tau)] = 0$$

In the covariance analysis one can then find the error covariance of estimating ϵ'. Then the error covariance of the true misalignment error can be written as Eq. (D-13).

APPENDIX E

FORMULATION OF EQUATIONS USED FOR STUDYING OPTIMAL ESTIMATION ERRORS AND SENSITIVITY ANALYSIS

Most of the system is modelled as a continuous system:

$$\dot{x} = F\,X + \Gamma u \qquad \text{(E-1)}$$

where

$$x = \begin{bmatrix} \epsilon_N \\ \epsilon_E \\ \epsilon_z \\ R_N \\ R_E \\ R_z \\ \delta V_N \\ \delta V_E \\ A_N \\ A_E \end{bmatrix} \quad \text{(E-2)} \qquad \Gamma = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ g\phi\sqrt{\dfrac{\Delta_C T}{12}} & 0 \\ 0 & g\phi\sqrt{\dfrac{\Delta_C T}{12}} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad \text{(E-3)}$$

$$u = \begin{bmatrix} u_1 \\ \\ u_2 \end{bmatrix} \Rightarrow \text{unit variance white noise}$$

$\Delta_C T$ is C matrix update time
$g\phi$ is gyro quantization level in radians xg (gravity)

Most of the terms in x are defined in the Glossary. $\delta V_N, \delta V_E$ are the inertial system velocity errors in the North and East axes, respectively, less accelerometer quantization error.

The system, F, is specified for a Latitude of 42°.

$$F = \begin{bmatrix}
0 & -.175612 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
.17561 & 0 & +.195097 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -.195097 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 68600. & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
-68600. & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix} \tag{E-4}$$

The above equations are written in radians - nautical miles - hours dimensions.

In order to model all pertinent processes, it is necessary to augment the system on the discrete level. When the acceleration is considered sinusoidal, the augmented difference equation can be written.

$$x'_{i+1} = \Phi' x'_i + \Gamma' u : \tag{E-5}$$

where

$$x'_i = \begin{bmatrix} x_i \\ \text{- - - - - - - -} \\ V_{TNi} \\ V_{TEi} \\ \delta V_{Ni-1} + V_{TNi-1} + qA_{Ni-1} \\ \delta V_{Ei-1} + V_{TEi-1} + qA_{Ei-1} \\ qA_{Ni} \\ qA_{Ei} \end{bmatrix} \text{ is a 16-state vector} \tag{E-6}$$

V_{TN} is true velocity North

V_{TE} is true velocity East

qA_N is North acceleration quantization error

qA_E is East acceleration quantization error.

$$\Phi' = \left[\begin{array}{c|cccccccccc|cccccc} & \multicolumn{10}{c|}{e^{(F)\,\Delta T}} & \multicolumn{6}{c}{0} \\ \hline & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 \\ \hline 11 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cos w\Delta T & \sin w\Delta T & 0 & 0 & 0 & 0 \\ 12 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\sin w\Delta T & \cos w\Delta T & 0 & 0 & 0 & 0 \\ 13 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 14 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 15 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 16 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right] \qquad \text{(E-7)}$$

and

$$\Gamma'\Gamma'^{T} = \left[\begin{array}{c|c|c|c} & & 11\ 12\ 13\ 14 & 15\ 16 \\ \hline & Q(\Delta T) & 0 & 0 \\ \hline \begin{array}{c}11\\12\\13\\14\end{array} & 0 & 0 & 0 \\ \hline \begin{array}{c}15\\16\end{array} & 0 & 0 & \begin{array}{cc}\overline{q^2_{AN}} & \\ & \overline{q^2_{AE}}\end{array} \end{array}\right] \qquad \text{(E-8)}$$

where w is the angular velocity of the sinusoidal acceleration and $Q(\Delta T)$ is the solution of

$$\dot{Q}(t) = FQ(t) + Q(t)F^{T} + \Gamma\Gamma^{T} \qquad \text{(E-9)}$$

$$\text{at } t = \Delta T$$

$$\text{with } Q(0) = 0$$

The velocity difference measurements can be written:

$$Z = H\, x' \tag{E-10}$$

where

$$Z = \begin{bmatrix} Z_1 \\ Z_2 \end{bmatrix}$$

$$H = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & 1 \end{bmatrix} \tag{E-11}$$

The initial covariance is $P = E[\, x'(0)\; x'(0)^T]$

The initial tilts of the platform are designated by setting the elements of the initial covariance elements P_{11}, P_{22}, P_{33}.

The gyro biases are specified by P_{44}, P_{55}, P_{66}.

The accelerometer biases are set by P_{99}, $P_{10,10}$.

The magnitude of the acceleration environment is found by setting

$$P_{11,11} = P_{12,12} = \left(\frac{\text{rms acc.}}{w} \right)^2 \tag{E-12}$$

Other diagonal elements of the P matrix are chosen to be reasonable numbers.

The quantities specified in Eqs. (E-7), (E-8), (E-11) and the initial covariance matrix are the inputs to the Kalman filter covariance programs which determine optimal estimation errors.

They are also inputs to the "sensitivity" program which determines the covariance of the estimation error. This covariance is computed in the following way. The error in the estimation can be written

$$\eta = \epsilon_s(N\Delta T) - \sum_{i=1}^{N} U_i Z(i\Delta T) \qquad \text{(E-13)}$$

The U's are the coefficients defined in Section 5.2 and $\underline{\epsilon_s}$ are the actual quantities to be estimated:

$$\epsilon_s = \begin{bmatrix} \epsilon_N \\ \epsilon_E \\ \epsilon_z \\ R_N \end{bmatrix}$$

The covariance of the error is then

$$E[\eta\eta^T] = E[\epsilon_s(N\Delta T)\epsilon_s^T(N\Delta T) - 2\sum_{i=1}^{N} \epsilon_s(N\Delta T)Z^T(i\Delta T)U_i^T + \sum_{ij}^{N}\sum U_i Z(i\Delta T)Z^T(j\Delta T)U_j^T]$$

(E-14)

A typical $\epsilon_s(N\Delta T)Z^T(i\Delta T)U_i^T$ term can be computed via

$$E[\epsilon_s(N\Delta T)Z^T(i\Delta T)] = D\,P_i\,\Phi'^{(N-i)T} H^T U_i^T \qquad \text{(E-15)}$$

and a typical term $U_i Z(i\Delta T)Z(j\Delta T)U_j^T$ can be computed via

$$U_i H \Phi'^{i-j} P_j H^T U_j^T, \text{ for } i > j \qquad \text{(E-16)}$$

$$U_i H P_i \Phi'^{j-i^T} H^T U_j^T \quad \text{for } j > i$$

and

$$P_i = \Phi' P_{i-1} \Phi'^T + \Gamma' \Gamma'^T \tag{E-17}$$

Also

$$D = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{E-18}$$