APOLLO

R-531

WHOLE NUMBER STRAPDOWN COMPUTATIONS

by

J. C. Pennypacker

February 1966

MIT

CAMBRIDGE 39, MASSACHUSETTS

APOLLO

GUIDANCE AND NAVIGATION

Approved: Eldon C Hall Date: 2/21/66
ELDON C. HALL, DIRECTOR, DDG
INSTRUMENTATION LABORATORY

Approved: David G. Hoag Date: 21 Feb 66
DAVID G. HOAG, DIRECTOR
APOLLO GUIDANCE AND NAVIGATION PROGRAM

Approved: Roger B Woodbury Date: 24 Feb 66
ROGER B. WOODBURY, DEPUTY DIRECTOR
INSTRUMENTATION LABORATORY

R-531

WHOLE NUMBER STRAPDOWN COMPUTATIONS

by

J.C. Pennypacker

February 1966

MIT INSTRUMENTATION LABORATORY

CAMBRIDGE 39, MASSACHUSETTS

COPY # 114

ACKNOWLEDGMENT

The report was prepared under DSR Project 55-362, sponsored by the Manned Spacecraft Center of the National Aeronautics and Space Administration through Contract NAS 9-4065.

There are two persons who, although not authors of this report, have nevertheless contributed substantially to the study reported on here. The computer simulations described herein are based entirely upon an algorithm which was originally suggested by Albert Hopkins. Ramon Alonso directed the scope of the study and provided the initial interpretation of the computer simulations. These two individuals jointly provided the motivation required to carry this study to its logical conclusion.

TABLE OF CONTENTS

WHOLE NUMBER STRAPDOWN COMPUTATIONS

ABSTRACT

An inertial navigation system employing a gimballess inertial measurement unit requires an analytical transformation of the vehicle co-ordinate system into the inertial co-ordinate system. An algorithm is developed for maintaining an up-to-date transformation matrix in a general purpose whole number computer. A method of implementing the algorithm in the Apollo Guidance Computer (AGC) is described. The performance of the algorithm, the effects of flight profile parameters upon the accuracy of the algorithm, and the effects of certain equipment constraints are detailed in the results of computer simulations. Extensive computer simulations were conducted to verify the validity of the algorithm; while conclusions about navigation computer design were drawn from the simulation results, raw simulation data is included for individual interpretation. For comparative purposes, the results of simulation of a digital differential analyzer (DDA) are included. It is concluded that for at least certain missions, general purpose computers can be built which will perform the strapdown computation with sufficient accuracy and which will not significantly detract from the other tasks required of the general purpose computer by doing these tasks fast enough.

by J. C. Pennypacker

February, 1966

I. INTRODUCTION

There is at the present time considerable interest among designers of inertial navigation systems in the utilization of inertial measurement units which are mounted directly to the vehicle frame; the resulting configuration is a gimballess inertial measurement unit (GIMU), commonly referred to as a "strapdown" system. Such a configuration requires that analytic methods rather than the conventional physical gimbals be employed to isolate the vehicle co-ordinate axes from the inertial co-ordinate system in which navigation and guidance of the vehicle are performed. There are at least two basic methods of implementing the required analytic functions: the more generally accepted approach is to use a digital differential analyzer (DDA), the other approach is to use a general purpose whole number computer. The desirability of the latter method becomes pronounced in those systems for which a general purpose computer is required to perform functions other than those required for navigation; in such a system, the hardware configuration need not include an extra processor - specifically the DDA - for navigation.

The primary question in using a general purpose computer centers around the algorithms used for updating the transformation matrix. For the general purpose computer approach to be practical, the computer must spend only a small fraction (less than 10%) of its time in the strapdown task otherwise performed by the DDA. The time spent by the general purpose computer is a function of both computer speed and the updating algorithm utilized. The DDA algorithms are ill suited for implementation in a general purpose computer and the question thus arises as to whether the whole number algorithms of the class proposed by A. Hopkins[1] will give sufficiently precise results without requiring excessive computation times. While the advantages, disadvantages and capabilities of the DDA are generally understood, such insight into the performance of a general purpose computer operating in conjunction with a strapdown navigation system is lacking.

---

(1) Albert Hopkins, Digital Development Report #5, Updating a Cosine Matrix in a Whole Number Computer, MIT Instrumentation Laboratory, August 12, 1964.

This paper presents one algorithm which could be used to perform navigation functions on a whole number general purpose digital computer; the results of extensive computer simulation of this algorithm are also included. Because of the current interests of the author, the study undertaken is oriented towards the Apollo mission; of specific interest is the feasibility of utilizing the Apollo Guidance Computer (AGC) to perform the navigation functions of the Lunar Excursion Module (LEM). The scope of this study is restricted to one portion only of the general navigation problem, that of maintaining an accurate and timely direction cosine matrix. The vehicle containing the strapdown system is assumed to be a spacecraft of the LEM type; this assumption is fundamental to the characteristics of the algorithm and simulations presented herein.

11. THE COSINE MATRIX

In order to perform the navigation and guidance computations in a fixed co-ordinate system, it is necessary first to resolve the accelerations measured by the accelerometers in the spacecraft (body) co-ordinate system into components in the fixed co-ordinate system. For a fixed co-ordinate system F and a body co-ordinate system B, the transformation of acceleration from the body system to the fixed system is given by the following equation:

$$\vec{A}_F = [C]\,\vec{A}_B \tag{1}$$

$\vec{A}_F$ = Acceleration vector resolved into the fixed co-ordinate system.

$[C]$ = Transformation matrix.

$\vec{A}_B$ = Acceleration vector resolved into the body co-ordinate system.

The transformation matrix is a matrix composed of the direction cosines of the angles between the axes of the two co-ordinate systems; thus, the elements of $[C]$ are given by the following:

$$C_{ij} = \vec{u}_{Fi} \cdot \vec{u}_{Bj} \tag{2}$$

$u_{(\ )}$ = a unit vector in the direction of the co-ordinate system indicated by the subscript,

The matrix [C], which is dependent only upon the attitude of the vehicle, must be precisely known at the time accelerations occur in order to determine the position in inertial space of the spacecraft. The analytical determination of the C matrix is the basic difficulty encountered in the strapdown configuration. As the vehicle rotates, the matrix [C] changes; thus in general the velocity of the spacecraft in the fixed co-ordinate system is given by:

$$\vec{V}_F(t) = \int_0^t [C(t)]\,\vec{A}_B\,(t)\,dt \tag{3}$$

The inertial position of the spacecraft is determined from a further integration of equation (3). In order to determine an expression for the change of $[C(t)]$ as the vehicle rotates, let the vehicle rotate with respect to the fixed co-ordinate system with an angular velocity $\vec{\Omega}_{FB}(t)$. Then from equation (2):

$$\frac{d}{dt} C_{ij}(t) = \dot{C}_{ij}(t) = \dot{\vec{u}}_{Fi} \cdot \vec{u}_{Bj} + \vec{u}_{Fi} \cdot \dot{\vec{u}}_{Bj} \tag{4a}$$

$$= \vec{u}_{Fi} \cdot (\vec{\Omega}_{FB}(t) \times \vec{u}_{Bj}) + 0$$

Evaluating the vector equation and writing in index form yields:

$$\dot{C}_{ij}(t) = \vec{u}_{Fi} \cdot [-\Omega_{FBk}(t)\vec{u}_{Bi} + \Omega_{FBi}(t)\vec{u}_{Bk}] \tag{4b}$$

From equation (2) this can be rewritten as:

$$\dot{C}_{ij}(t) = C_{i,j+1}(t)\ \Omega_{FB(j+2)}(t) - C_{i,j+2}(t)\ \Omega_{FB(j+1)}(t) \tag{4c}$$

Letting x, y and z represent the i, j and k^{th} axis of the spacecraft respectively, equation (4c) can be expressed as:

$$[\dot{C}(t)] = [C(t)][\Omega(t)] \tag{5}$$

where

$$[\Omega(t)] = \begin{bmatrix} 0 & -\omega_z(t) & \omega_y(t) \\ \omega_z(t) & 0 & -\omega_x(t) \\ -\omega_y(t) & \omega_x(t) & 0 \end{bmatrix}$$

III. THE BASIC ALGORITHM

3.1 Derivation

A. Hopkins[2] has described a method of approximating the solution to equation (5) utilizing a general purpose digital computer. Because this approximation provides the basis for the computer simulation, the remainder of this section presents the algorithm originally described by Hopkins.

Define a matrix [M(T)] which is a function of the ω's, of their derivatives, and of a sampling time interval T. The matrix [M(T)] relates the C matrix at the end of the sampling time interval T to its value at the beginning of the time interval. This relationship is defined thus:

$$[C(T)] = [C(0)][M(T)] \tag{6}$$

Knowledge of M(T) enables one to calculate the current value of [C(t)] by a recursive process. Owing to limitations of GIMU instruments, however, [M(T)] can only be approximated. Previous approaches have emphasized the use of digital differential analyzers (DDA's) in order to achieve maximum precision with a small computer. Utilization of a general purpose digital computer such as the Apollo Guidance Computer (AGC) requires a substantially different approach: a large time interval T with a sophisticated approximation to [M(T)] instead of the DDA's short interval and skeletal approximation to [M(T)]. The fundamental question associated with large intervals T centers around the uncertainties as to the order in which rotations occur, and the inaccuracies which result from these uncertainties.

The data from which [M(T)] can be approximated by a spacecraft navigation computer is a quantized representation of angle changes as detected by the body-mounted gyros. It is here assumed that these angle changes are known precisely; the effect of introducing imperfect gyros into the system is described in a later section. To express [M(T)] in terms of the spacecraft angle changes (denoted, θ_x, θ_y, θ_z), [C(t)] is expressed as a function of [C(0)]. The Taylor series expansion of element C_{ij} is:

---

(2) Ibid.

$$C_{ij}(t) = C_{ij}(0) + t\,\dot{C}_{ij}(0) + \frac{1}{2}t^2\,\ddot{C}_{ij}(0) + \frac{1}{6}t^3\,\dddot{C}_{ij}(0) + \ldots \qquad (7)$$

At this time it is convenient to rewrite equation (5) in index form:

$$\begin{aligned}
\dot{C}_{i1}(t) &= C_{i2}(t)\omega_z(t) - C_{i3}(t)\omega_y(t) \\
\dot{C}_{i2}(t) &= -C_{i1}(t)\omega_z(t) + C_{i3}(t)\omega_x(t) \qquad (8) \\
\dot{C}_{i3}(t) &= C_{i1}(t)\omega_y(t) - C_{i2}(t)\omega_x(t)
\end{aligned}$$

The expressions of equation (8) can be used to replace the $C_{ij}(t)$ term of equation (7) with undifferentiated terms. Equation 8 can also be differentiated to give $\ddot{C}_{ij}$'s in terms of $\dot{C}_{ij}$'s. Further substitutions utilizing equation (8) yield expressions for the $\dddot{C}_{ij}$'s in terms of the C_{ij}'s alone. These expressions may be substituted for the $\ddot{C}_{ij}$ term of equation (7). For example:

$$\begin{aligned}
\ddot{C}_{i1}(t) &= C_{i2}(t)\dot{\omega}_z(t) + \omega_z(t)\dot{C}_{i2}(t) - C_{i3}(t)\dot{\omega}_y(t) - \omega_y(t)\dot{C}_{i3}(t) \\
&= C_{i2}(t)\dot{\omega}_z(t) - \omega_z^2(t)C_{i1}(t) + \omega_x(t)\omega_z(t)C_{i3}(t) \\
&\quad - \dot{\omega}_y(t)C_{i3}(t) - \omega_y^2(t)C_{i1}(t) + \omega_x(t)\omega_y(t)C_{i2}(t) \\
&= [-\omega_y^2(t) - \omega_z^2(t)]\,C_{i1}(t) + [\omega_x(t)\omega_y(t) + \dot{\omega}_z(t)]\,C_{i2}(t) \\
&\quad + [\omega_x(t)\omega_z(t) - \dot{\omega}_y(t)]\,C_{i3}(t) \qquad (9)
\end{aligned}$$

Continuing in this manner, one can obtain expressions for the time derivatives of each C_{ij} in terms of all the C_{ij}'s. Since these expressions contain ω's and their derivatives, they will be of the general form:

$$\frac{d^k}{dt^k}C_{ij}(t) = f_{ijk1}[\omega(t)]C_{i1}(t) + f_{ijk2}[\omega(t)]C_{i2}(t) + f_{ijk3}[\omega(t)]C_{i3}(t) \qquad (10a)$$

Specifically, at time $t = 0$, equation (10a) becomes:

$$\frac{d^k}{dt^k} C_{ij}(0) = f_{ijk1}[\omega(0)]C_{i1}(0) + f_{ijk2}[\omega(0)]C_{i2}(0) + f_{ijk3}[\omega(0)]C_{i3}(0) \qquad (10b)$$

When equation (10b) is substituted for the differentiated terms of equation (7), one obtains:

$$C_{ij}(t) = \sum_{k=0}^{\infty} f_{ijk1}\,[\omega(0)]\,\frac{t^k}{k!}\,C_{i1}(0)$$

$$+ \sum_{0}^{w} f_{ijk2}\,[\omega(0)]\,\frac{t^k}{k!}\,C_{i2}(0)$$

$$+ \sum_{k=0}^{\infty} f_{ijk3}\,[\omega(0)]\,\frac{t^k}{k!}\,C_{i3}(0) \qquad (11)$$

Comparison with equation (6) shows that at time $t = T$, the elements of $[M(T)]$ are given by the infinite series in equation (11). Elimination of the redundant subscripts in equation (11) leads to the expression:

$$M_{ij}(T) = \sum_{k=O}^{w} f_{ijk}\,[\omega(0)]\,\frac{T^k}{k!} \qquad (12)$$

which is recognized as the Taylor series expansion of $M_{ij}(T)$ where f_{ijk} is the k^{th} derivative of $M_{ij}(0)$.

It has been shown that the elements of $[M(T)]$ can be approximated by a Taylor series whose terms are obtained from differentiation of equation (8); a list of these terms is given in Table 1. There remains to be shown how these terms can be expressed in terms of the spacecraft angle changes during the time interval T.

Letting the change of the spacecraft angle about the i^{th} axis be denoted by Θ_i, the first step is to use the Taylor series expansion to relate the Θ's to the respective ω's. According to the definition of $\Theta_i(T)$:

$$\theta_i(T) = \int_0^T \omega_i(t)dt$$

$$= \int_0^T [\omega_i(0) + t\dot{\omega}_i(0) + \frac{t^2}{2}\ddot{\omega}_i(0) + \ldots]dt$$

$$= T\omega_i(0) + \frac{T^2}{2}\dot{\omega}_i(0) + \frac{T^3}{6}\ddot{\omega}_i(0) + \ldots \quad (13)$$

It is evident that terms of equation (13) appear also in the Taylor expansion of some of the M_{ij}'s. For example, from equation (13) and from Table 1:

Order	$M_{11}(T)$	$M_{21}(T)$	$M_{31}(T)$
0	1	0	0
1	0	$T\omega_z$	$-T\omega_y$
2	$-\frac{1}{2}T^2(\omega_y^2 + \omega_z^2)$	$\frac{1}{2}T^2(\omega_x\omega_y + \dot{\omega}_z)$	$\frac{1}{2}T^2(\omega_x\omega_z - \dot{\omega}_y)$
m	$-\frac{1}{2}T^3(\omega_y\dot{\omega}_y + \omega_z\dot{\omega}_z)$	$\frac{1}{6}T^3(-\omega_z\omega^2 + \ddot{\omega}_z + \omega_y\dot{\omega}_x + 2\omega_x\dot{\omega}_y)$	$\frac{1}{6}T^3(\omega_y\omega^2 - \ddot{\omega}_y + \omega_z\dot{\omega}_x + 2\omega_x\dot{\omega}_z)$
	$\frac{1}{24}T^4[\omega_z\ddot{\omega}_z - 4\omega_y\ddot{\omega}_y$	$\frac{1}{24}T^4[3\dot{\omega}_x\dot{\omega}_y + \omega_y\ddot{\omega}_x$	$\frac{1}{24}T^4[3\dot{\omega}_x\dot{\omega}_z + \omega_z\ddot{\omega}_x + 3\omega_x\ddot{\omega}_z$
	$+ \omega^2(\omega_z^2 + \omega_y^2)$	$+ 3\omega_x\ddot{\omega}_y - \omega^2\omega_x\omega_y$	$- \omega^2\omega_x\omega_z + 3\omega_x\dot{\omega}_x\omega_y$
	$+ 2\omega_x(\omega_y\dot{\omega}_z - \omega_z\dot{\omega}_y)$	$- 3\omega_x\dot{\omega}_x\omega_z - 5\omega_y\dot{\omega}_y\omega_z$	$+ 5\omega_y\dot{\omega}_z\omega_z + 6\omega_y^2\dot{\omega}_y$
	$- 8(\dot{\omega}_y^2 + \dot{\omega}_z^2)]$	$- 6\omega_z^2\dot{\omega}_z - 8\omega_x^2\dot{\omega}_z$	$+ 3\omega_x^2\dot{\omega}_y + \omega_z^2\dot{\omega}_y - \dddot{\omega}_y]$
		$- \omega_y^2\dot{\omega}_z + \dddot{\omega}_z]$	

NOTE: $\omega^2 = \omega_x^2 + \omega_y^2 + \omega_z^2$

All ω s are evaluated at time 0, the beginning of the sampling interval T.

TABLE 1 ELEMENTS OF THE MATRIX [M(T)]

Order	$M_{12}(T)$	$M_{22}(T)$	$M_{32}(T)$
0	0	1	0
1	$-T\omega_z$	0	$T\omega_x$
2	$\frac{1}{2}T^2(\omega_x\omega_y - \dot{\omega}_z)$	$-\frac{1}{2}T^2(\omega_x^2 + \omega_N^2)$	$\frac{1}{2}T^2(\omega_y\omega_x + \dot{\omega}_x)$
3	$\frac{1}{6}T^3\omega_z\omega^2 - \ddot{\omega}_z + \omega_x\dot{\omega}_y + 2\dot{\omega}_y\omega_x)$	$-\frac{1}{6}T^3(\omega_z\dot{\omega}_N + \ddot{\omega}_x)$	$\frac{1}{6}T^3(-\omega_x\omega^2 + \ddot{\omega}_x + \omega_z\dot{\omega}_y + 2\omega_y\dot{\omega}_N)$
4	$\frac{1}{24}T^4[\dot{\omega}_x\dot{\omega}_y + \omega_x\ddot{\omega}_y + 3\ddot{\omega}_x\omega_y - \omega^2\omega_x\omega_y + 3\omega_y\dot{\omega}_y\omega_z + 5\omega_x\dot{\omega}_x\omega_z + 6\omega_z^2\dot{\omega}_z + 3\omega_y^2\dot{\omega}_z + \omega_x^2\dot{\omega}_N - \dddot{\omega}_N]$	$\frac{1}{24}T^4[\dot{\omega}_x\ddot{\omega}_x - 4\omega_z\ddot{\omega}_N + \omega^2(\omega_x^2 + \omega_N^2) + 2\omega_y(\dot{\omega}_x\omega_z - \ddot{\omega}_z) - 3(\dot{\omega}_N^2 + \dot{\omega}_z^2)]$	$\frac{1}{24}T^4[3\dot{\omega}_y\omega_z + \omega_z\ddot{\omega}_y + 3\omega_y\ddot{\omega}_z - \omega^2\omega_y\omega_z - 3\omega_x\dot{\omega}_y\dot{\omega} - 5\omega_x\dot{\omega}_z\omega_z - 6\dot{\omega}_x^2\omega_x - 3\dot{\omega}_x\omega_y^2 - 3\dot{\omega}_z^2\omega_x + \dddot{\omega}_x]$

TABLE 1 (cont

Order	$M_{13}(T)$	$M_{23}(T)$	$M_{33}(T)$
0	*0*	*0*	**1**
1	$T\omega_Y$	$-Tu!_x$	*0*
2	$\frac{1}{2} T^2(\omega_x\omega_z + \omega_Y)$	$\frac{1}{2} T^2(\omega_y\omega_z - \dot{\omega}_x)$	$-\frac{1}{2} T^2(\omega_x^2 + \omega_y^2)$
3	$\frac{1}{6} T^3(-\omega_Y\omega^2 + \ddot{\omega}_Y + \omega_x\omega_z + 2\omega_x\omega_z)$	$\frac{1}{6} T^3(\omega_x\omega^2 - \ddot{\omega}_x + \omega_y\dot{\omega}_z + 2\omega_z\dot{\omega}_y)$	$\frac{1}{2} T^3(\omega_x\dot{\omega}_x + \omega_Y\omega_Y)$
4	$\frac{1}{24} T^4[3\dot{\omega}_x\dot{\omega}_z + \omega_x\ddot{\omega}_z + 3\ddot{\omega}_x\omega_z - \omega^2\omega_x\omega_z - 3\omega_y\omega_z\dot{\omega}_z - 5\omega_x\dot{\omega}_x\omega_y - 6\omega^2\dot{\omega}_y - 3\omega_y\omega_z^2 - \omega_x^2\dot{\omega}_y + \dddot{\omega}_y]$	$\frac{1}{24} T^4[3\dot{\omega}_y\dot{\omega}_z + \omega_y\ddot{\omega}_z + 3\omega_z\ddot{\omega}_y - \omega^2\omega_y\omega_z + 3\omega_x\dot{\omega}_z\dot{\omega}_z + 5\omega_x\dot{\omega}_y\dot{\omega}_y + 6\omega^2\dot{\omega}_x\omega_x + 3\omega_x\omega_z^2 + \omega_y^2\dot{\omega}_x - \dddot{\omega}_x]$	$\frac{1}{24} T^4[-4\omega_x\ddot{\omega}_x - 4\omega_Y\omega_Y + \omega^2(\omega_x^2 + \omega_y^2) + 2\omega_z(\omega_x\omega_y - \omega_y\omega_x) - 3(\dot{\omega}_x^2 + \dot{\omega}_y^2)]$

TABLE 1 (cont.)

$$\theta_Y(T) = T\omega_Y(0) + \frac{T^2}{2}\dot{\omega}_y(0) + \frac{T^3}{6}\ddot{\omega}_y(0) + \ldots$$

and

$$M_{13}(T) = T\omega_y(0) + \frac{T^2}{2}[\omega_x(0)\omega_z(0) + \dot{\omega}_y(0)]$$

$$+ \frac{T^3}{6}[\ddot{\omega}_y(0) + \omega_x(0)\dot{\omega}_z(0) + \dot{\omega}_x(0)\omega_z(0) + 2\omega_x(0)\omega_z(0) - \omega^2(0)\omega_y(0)] + \ldots$$

Comparison shows that $\theta_Y(T)$ approximates $M_{13}(T)$ with an error function whose leading terms (for $T < 1$) are:

$$\frac{T^2}{2}\omega_x(0)\omega_z(0) + \frac{T^3}{6}[\dot{\omega}_x(0)\omega_z(0) + 2\omega_x(0)\omega_z(0) - \omega^2(0)\omega_y(0)] \qquad (14)$$

An improved approximation to $M_{13}(T)$ is obtained by expressing the first error term of expression *(14)* using the product $\theta_x(T)\theta_z(T)$. From equation (13):

$$\theta_x(T)_z(T) = T^2\omega_x(0)\omega_z(0) + \frac{T^3}{2}[\dot{\omega}_x(0)\omega_z(0) + \omega_x(0)\dot{\omega}_z(0)] + \ldots \qquad (15)$$

Utilizing Table 1, equation (13) and expression (14), one can approximate $M_{13}(T)$ by $\theta_y(T) + \frac{\theta_x(T)\theta_z(T)}{2}$ with an error function whose leading terms are now:

$$-\frac{1}{12}[\omega_z(0)\omega_x(0) - \omega_x(0)\omega_z(0) - 2\omega^2(0)\omega_y(0)]$$

Table 2 gives a number of functions of θ_i which are used in the formulation of yet better approximations to the $M_{ij}(T)$.

$$\theta_i = T\omega_i + \frac{T^2}{2}\dot{\omega}_i + \frac{T^3}{6}\ddot{\omega}_i + \frac{T^4}{24}\dddot{\omega}_i + \cdots$$

$$\theta_i^2 = T^2\omega_i^2 + T^3\omega_i\dot{\omega}_i + T^4\left(\frac{1}{4}\dot{\omega}_i^2 + \frac{1}{3}\omega_i\ddot{\omega}_i\right) + \cdots$$

$$\theta_i\theta_j = T^2\omega_i\omega_j + \frac{T^3}{2}\left(\omega_i\dot{\omega}_j + \dot{\omega}_j\omega_i\right) + T^4\left(\frac{1}{4}\dot{\omega}_i\dot{\omega}_j + \frac{1}{6}\omega_i\ddot{\omega}_j + \frac{1}{6}\ddot{\omega}_j\omega_i\right) + \cdots$$

$$\theta_{-i} = \int_0^{-T}\omega_i(t)dt = -T\omega_i + \frac{1}{2}T^2\dot{\omega}_i - \frac{1}{6}T^3\ddot{\omega}_i + \frac{T^4}{24}\dddot{\omega}_i + \cdots$$

$$\theta_i\theta_{-j} = -T^2\omega_i\omega_j + \frac{T^3}{2}\left(\omega_i\dot{\omega}_j - \dot{\omega}_j\omega_i\right) + T^4\left(\frac{1}{4}\dot{\omega}_i\dot{\omega}_j - \frac{1}{6}\omega_i\ddot{\omega}_j - \frac{1}{6}\ddot{\omega}_j\omega_i\right) + \cdots$$

$$\theta_i\theta_{-j} - \theta_j\theta_{-i} = T^3\left(\dot{\omega}_j\omega_i - \omega_j\dot{\omega}_i\right) + \cdots$$

$$\theta_i\theta^2 = T^3\omega^2\omega_i + T^4\left(\frac{1}{2}\omega^2\dot{\omega}_i + \omega_i^2\dot{\omega}_i + \omega_i\omega_{i+1}\dot{\omega}_{i+1} + \omega_i\omega_{i+2}\dot{\omega}_{i+2}\right) + \cdots$$

NOTE: All ω's are evaluated at time 0, the beginning of the sampling interval T.

$\omega^3 = \omega_x^2 + \omega_y^2 + \omega_z^3$.

θ_{-i} is the negative of the angle change in the preceding interval.

TABLE 2. AUXILIARY FUNCTIONS OF e

It is now convenient to define a matrix $[N(\Theta)]$ which approximates $[M(T)]$ with an error $[E(T)]$; that is:

$$[M(T)] = [N(\Theta)] + [E(T)] \tag{16}$$

The number *of* possible forms of $[N(\Theta)]$ is of course unlimited and no procedure is given here for deriving optimized approximations. Table *3* shows three N matrices .which are equivalent to Taylor expansions of $[M(T)]$ to the first, second and third order terms respectively. The N ,matrices are written in terms of body angles; the leading terms of the corresponding error matrices $[E(T)]$ are expressed as functions of the ω's and their derivatives. The process of updating the N matrices of Table *3* at regular sampling time intervals T constitutes the basic algorithm; modifications to this basic algorithm are discussed later.

3.2 Interpretation of Algorithm

The algorithm presented in the preceding section was developed from a purely mathematical basis with no physical interpretation of the algorithm included. The transformation matrix can be visualized as a vector originating at the center of the unit sphere and terminating on the surface of the unit sphere. Rotation of the vehicle employing the strapdown system corresponds to tracing a path on the surface of the unit sphere. The N matrix vector, which approximates the true transformation vector, is updated only at discrete time intervals. Because the vector addition of small angle changes is an ordered process, the N matrix vector which is updated based upon the algebraic sum of angle changes during a sampling time interval is accurate only within some cone of error. To reduce the size of this cone of error, the position of the N matrix vector is extrapolated not only on the basis of historical velocity, but also on the basis of changes in velocity. This extrapolatidn is evidenced by the inclusion in the third order update formula of angle changes over two successive sampling time intervals. The physical assumption which is thus being made in the basic algorithm is that changes in angular positions during successive sampling intervals caused by acceleration are small compared to changes in angular position caused by current rotation.

1. $$N_1(T) = \begin{bmatrix} 1 & -\theta_z & \theta_y \\ \theta_z & 1 & -\theta_x \\ \theta_y & \theta_x & 1 \end{bmatrix}$$

$$E_1(T) = M(T) - N_1(T) = \frac{T^2}{2} \begin{bmatrix} -(\omega_y^2 + \omega_z^2) & \omega_x\omega_y & \omega_x\omega_z \\ \omega_x\omega_y & -(\omega_x^2 + \omega_z^2) & \omega_y\omega_z \\ \omega_x\omega_z & \omega_y\omega_z & (\omega_x^2 + \omega_y^2) \end{bmatrix} + O(T^3)$$

NOTE: All ω's are evaluated at time 0, the beginning of the sampling interval T.

TABLE 3. ELEMENTS OF [N(T)]

$$2 \quad N_2(T) = N_1(T) + \frac{1}{2}\begin{bmatrix} -(\theta_y^2 + \theta_z^2) & \theta_x\theta_y & \theta_x\theta_z \\ \theta_x\theta_y & -(\theta_x^2 + \theta_z^2) & \theta_y\theta_z \\ \theta_x\theta_z & \theta_y\theta_z & -(\theta_x^2 + \theta_y^2) \end{bmatrix}$$

$$E_2(T) = M(T) - N_2(T) =$$

$$\frac{T^3}{12}\begin{bmatrix} 0 & (2\omega^2\omega_z + \omega_y\dot{\omega}_x - \omega_x\dot{\omega}_y) & (-2\omega^2\omega_y + \omega_z\dot{\omega}_x - \omega_x\dot{\omega}_z) \\ (-2\omega^2\omega_z + \omega_y\dot{\omega}_x - \omega_x\dot{\omega}_y) & 0 & (2\omega^2\omega_x + \omega_z\dot{\omega}_y - \omega_y\dot{\omega}_z) \\ (2\omega^2\omega_y + \omega_x\dot{\omega}_z - \omega_x\dot{\omega}_z) & (-2\omega^2\omega_x + \omega_y\dot{\omega}_z - \omega_z\dot{\omega}_y) & 0 \end{bmatrix} + 0(T^4)$$

TABLE 3 (cont

3. $N_3(T) = N_2(T) + \frac{1}{12}\begin{bmatrix} 0 & (2\dot{\theta}^2\theta_z + \theta_x\theta_{-y} - \theta_y\theta_{-x}) & (-2\dot{\theta}^2\theta_y + \theta_z\theta_{-x} - \theta_x\theta_{-z}) \\ (-2\dot{\theta}^2\theta_z + \theta_x\theta_{-y} - \theta_y\theta_{-x}) & 0 & (2\dot{\theta}^2\theta_x + \theta_z\theta_{-y} - \theta_y\theta_{-z}) \\ (2\dot{\theta}^2\theta_y + \theta_x\theta_{-z} - \theta_z\theta_{-x}) & (2\dot{\theta}^2\theta_x + \theta_y\theta_{-z} - \theta_z\theta_{-y}) & 0 \end{bmatrix}$

$$E_m(T) = M(T) - N_3(H) = \frac{4}{H} \begin{bmatrix} e_{11} & e_{12} & e_{13} \\ e_{21} & e_{22} & e_{23} \\ e_{31} & e_{32} & e_{33} \end{bmatrix} + 0(T^5)$$

$$e_{11} = \omega^2(\omega_z^2 + \omega_y^2) + 2\omega_x(\omega_y\dot{\omega}_z - \omega_z\dot{\omega}_y)$$

$$e_{21} = -\omega^2\omega_x\omega_y + \dot{\omega}_z(\omega_y^2 - \omega_x^2) + \omega_z(\dot{\omega}_x\omega_x - \dot{\omega}_y\omega_y) + \ddot{\omega}_x\omega_y - \ddot{\omega}_y\omega_x \quad (3h)$$

$$e_{31} = -\omega^2\omega_x\omega_z + \dot{\omega}_y(\omega_x^2 - \omega_z^2) + \omega_y(\dot{\omega}_z\omega_z - \dot{\omega}_x\omega_x) + \ddot{\omega}_x\omega_z - \ddot{\omega}_z\omega_x$$

$$e_{12} = -\omega^2\omega_y\omega_x + \dot{\omega}_z(\omega_y^2 - \omega_x^2) + \omega_z(\dot{\omega}_x\omega_x - \dot{\omega}_y\omega_y) + \ddot{\omega}_y\omega_x - \ddot{\omega}_x\omega_y$$

$$e_{22} = \omega^2(\omega_x^2 + \omega_z^2) + 2\omega_y(\omega_z\dot{\omega}_x - \omega_x\dot{\omega}_z)$$

$$e_{32} = -\omega^2\omega_y\omega_z + \dot{\omega}_x(\omega_z^2 - \omega_y^2) + \omega_x(\dot{\omega}_y\omega_y - \dot{\omega}_z\omega_z) + \ddot{\omega}_y\omega_z - \ddot{\omega}_z\omega_y$$

$$e_{13} = -\omega^2\omega_z\omega_x + \dot{\omega}_y(\omega_x^2 - \omega_z^2) + \omega_y(\dot{\omega}_z\omega_z - \dot{\omega}_x\omega_x) + \ddot{\omega}_z\omega_x - \ddot{\omega}_x\omega_z$$

$$e_{23} = -\omega^2\omega_z\omega_y + \dot{\omega}_x(\omega_z^2 - \omega_y^2) + \omega_x(\dot{\omega}_y\omega_y - \dot{\omega}_z\omega_z) + \ddot{\omega}_z\omega_y - \ddot{\omega}_y\omega_z$$

$$e_{33} = \omega^2(\omega_y^2 + \omega_x^2) + 2\omega_z(\omega_x\dot{\omega}_y - \omega_y\dot{\omega}_x)$$

TABLE 3 (cont

3.3 Error Accumulation

The accumulation of attitude error is a complicated process, and no manageable analytic description has been developed. However, a crude upper bound of the accumulated error can be calculated by using the assumption that the absolute value of the error is the sum of the absolute values of the errors at each update calculation. As an example of this calculation, consider the error terms of the matrix $[N_3(T)]$. Since:

$$[C(T)] = [C(0)][M(T)] \tag{17}$$

and

$$[M(T)] = [N_3(T)] + [E_3(T)] \tag{18}$$

it follows that:

$$[C(T)] = [C(0)][N_3(T)] + [C(0)][E_3(T)] \tag{19}$$

where the second term is the error resulting from the approximation formula N_3. Let this error be denoted by $[D(T)]$, i.e.,

$$[D(T)] = [C(0)][E_3(T)] \tag{20}$$

Referring to Table 3 we can write:

$$D_{i1}(T) = C_{i1}\, e_{11}(T) + C_{i2}\, e_{21}(T) + C_{i3}\, e_{31}(T) + 0(T^5) \tag{21}$$

where the e's are the elements of $[E_3(T)]$. An upper bound to equation (21) can be obtained by substituting unity for each of the C_{ij}'s and by replacing the ω's in the expressions for the e's by the magnitude of w. This gives:

$$D_{i1}(T) < \frac{T^4}{24} [\omega^4 + 2\omega^2\dot{\omega} + \omega^4 + 4\omega^2\dot{\omega} + 2\omega\ddot{\omega} + \omega^4 + 4\omega^2\dot{\omega} + \omega\ddot{\omega}]$$

or

$$D_{i1}(T) < \frac{T^4}{24} [3\omega^4 + 10\omega^2\dot{\omega} + 4\omega\ddot{\omega}] \tag{22}$$

Similarly $D_{i2}(T)$ and $D_{i3}(T)$ have an upper bound identical to that of equation (22). This upper bound gives a means of assessing the update formulas in connection with a particular time interval T and mission profile, i.e., relationship between ω, $\dot{\omega}$, $\ddot{\omega}$ and time. The final error of the C matrix can in principle be evaluated by the integral:

$$\int_{0}^{t_{final}} [D(t,T)]dt$$

where t_{final} is the elapsed mission time. The uncertainty of the spacecraft attitude at tine t_{final} is in turn a function of the final errors in C.

Of much greater interest than the analytical error expressions derived above are the actually observed errors resulting from the computer simulations.

3.4 Timing Considerations

The rationale behind the utilization of a general purpose whole number computer to perform the navigation functions in a strapdown navigation system is that such a computer must necessarily be included in the spacecraft to perform other necessary on-board functions. The argument is made that if the percentage of computer time required to perform the navigation function is sufficiently small such that the other functions are not adversely affected, then the special purpose digital differential analyzer (DDA) which is normally associated with the strapdown system can be eliminated from the spacecraft. Assuming that a whole number algorithm of updating the cosine matrix is sufficiently accurate, the problem reduces to one of comparing an estimate of the amount of computer time required to perform the algorithm with the amount of excess time capacity of the guidance computer.

If the Block II AGC as it is presently conceived were required to perform the full third order update calculations at the rate of, say, 10 complete updates per second, then rough estimates indicate that the AGC would be saturated performing this task alone. However, it is estimated that the AGC could perform an economized version of the third order update formula, $N_3(T)$, in less than eight milliseconds. (For a complete description of the economized form, see Section 5.8.1 Computer Word Length, page .) It is estimated that such an economized form would require less than ten percent of the AGC's computing time. A rough estimate of the types and number of instructions required by the AGC to perform the third order update calculations is given in Table 4.

Number	Function Performed	Total Memory Cycle Times
3	Read θ_i	9
3	Shift Once	12
3	Square	18
9	Cross Multiplies	54
3	Sum of Squares	6
3	$\delta^2\theta_i$ Terms	18
3	Multiply by 2	6
30	Double Precision Adds	90
9	Adds	27
27	Multiplies	108
27	Double Precision Adds	108
18	Exchanges	36
APPROXIMATE TOTAL		600 MCT ≈ 7 msec.

TABLE 4. ROUGH ESTIMATE OF INSTRUCTIONS AND TIMES REQUIRED TO PERFORM THIRD-ORDER UPDATE CALCULATIONS IN AGC

IV. DIGITAL DIFFERENTIAL ANALYZER (DDA)

The basic approach to maintaining an updated cosine matrix using DDA techniques is to solve equation (4c). Using rectangular rule integration, the DDA updates the transformation matrix by solving the difference equations:

$$\Delta C_{ij} = C_{i,j+1}\,\Delta\theta_{B,j+2} - C_{i,j+2}\,\Delta\theta_{B,j+1} \qquad (23)$$

where $\Delta\theta_{B,j}$ is that angle change of the spacecraft about the j^{th} axis which results in one pulse of a pulse torqued gyro.

Solution of equation (23) requires that the update cycle time of the DDA be sufficiently short such that not more than one $\Delta\theta$ change is observed by an axis gyro during the update cycle; i.e., $\Delta\theta$ pulses about any given axis must be processed one at a time and in the order observed. Examination of equation (23) indicates that, since addition of angle changes about three different axes is non-commutativg the updated transformation matrix is dependent upon the order in which the individual elements of the matrix are updated. This order dependency of the updating procedure of the DDA introduces into the updated matrix an inherent inaccuracy which is a function of the updating procedure and of the particular flight profile.

An analysis of various updating procedures for a DDA and of the errors associated with each of these procedures has been conducted by R. M. Hession[3]; the principal results and recommendations resulting from Hession's analysis were utilized in this study as a basis for comparing the performance of a whole number updating algorithm with the performance of a DDA. Hession concludes that, considering the tradeoffs involved between required accuracy and machine speed, the optimum configuration of a DDA is one designated as Serial-Parallel (± separately; with reversal rule). Under this organization, a complete update of the transformation matrix consists of three partial updates; the DDA must thus operate at a cycle time sufficient for the three partial updates to be completed between successive $\Delta\theta$ changes about

---

(3) R. M. Hession, *R-481,* Analysis of a Transformation Computer Used with a Gimballess IMU, MIT Instrumentation Laboratory, January, 1965.

any axis. To minimize the error resulting from the order in which the updates are performed, the update order is reversed after an angular change of $\Delta\theta$ is observed about any body axis.

The difference equations to be updated by a Serial-Parallel organized DDA are shown below. To simplify the notation, h_i is used instead of $\Delta\theta_i$; $C_{ij}(K + n)$ refers to the value of the element C_{ij} after having been updated n times. The difference equations are:

$$C_{i1}(K+1) = C_{i1}(K)$$

$$C_{i2}(K+1) = C_{i2}(K) - C_{i1}(K)\, h_3$$

$$C_{i3}(K+1) = Ci3(K) + C_{i1}(K)\, h_2$$

$$C_{i1}(K+2) = C_{i1}(K+1) + C_{i2}(K+1)\, h_3$$

$$C_{i2}(K+2) = C_{i2}(K+1) \qquad (24a)$$

$$C_{i3}(K+2) = C_{i3}(K+1) - C_{i2}(K+1)\, h_1$$

$$C_{i1}(K+3) = C_{i1}(K+2) - C_{i3}(K+2)\, h_2$$

$$C_{i2}(K+3) = C_{i2}(K+2) + C_{i3}(K+2)\, h_1$$

$$C_{i3}(K+3) = C_{i3}(K+2)$$

Upon reversal, the equations become:

$$C_{i1}(K+4) = C_{i1}(K+3) - C_{i3}(K+3)\,h_2$$

$$C_{i2}(K+4) = C_{i2}(K+3) + C_{i3}(K+3)\,h_1$$

$$C_{i3}(K+4) = C_{i3}(K+3)$$

$$C_{i1}(K+5) = C_{i1}(K+4) + C_{i2}(K+4)\,h_3$$

$$C_{i2}(K+5) = C_{i2}(K+4) \qquad (24b)$$

$$C_{i3}(K+5) = C_{i3}(K+4) - C_{i2}(K+4)\,h_1$$

$$C_{i1}(K+6) = C_{i1}(K+5)$$

$$C_{i2}(K+6) = C_{i2}(K+5) - C_{i1}(K+5)\,h_3$$

$$C_{i3}(K+6) = C_{i3}(K+5) + C_{i1}(K+5)\,h_2$$

The set of equations (24) were used to describe a DDA subject to the following mechanization rules. Each of the elements of the transformation matrix consist of two finite length computer words, Y and R. Only the Y words were used in the multiplications with the products added into the appropriate R register. The lowest order "slot" of the Y word equals the magnitude of $1\Delta\theta$. (The terminology "slot" is introduced because the value of $\Delta\theta$ which was utilized, 1/4 milliradian, is not representable by a negative integral power of either 10 or of 2. Thus while in most DDA's $1\Delta\theta$ corresponds to the lowest order bit of a binary register, the value of $\Delta\theta$ chosen for the simulations prohibits the normal use of the word "bit" for the purposes of this study.) "he R register is restricted in magnitude to be less than $1\Delta\theta$; when the R register exceeds $1\Delta\theta$, an overflow of $1\Delta\theta$ is affected into the corresponding Y register. Under the above form of mechanization, a typical update equation from the set of equations (24) becomes:

$$R_{i2}\ (K+1) = R_{i2}\ (K) - Y_{i1}\ (K)\ h_3$$

If $R_{i2}\ (K+1)$ exceeds $1\Delta\theta$, then $Y_{i2}\ (K+1)$ is incremented by $1\Delta\theta$ and $R_{i2}\ (K+1)$ is decremented by $1\Delta\theta$.

In addition to the mechanization described above, a roundoff rule was employed. Before using a Y word in a multiplication, the corresponding R word was checked to determine the value in the R word. If R equalled or exceeded $1/2\ \Delta\theta$, then Y was incremented by $1\Delta\theta$ before being used in the multiplication; otherwise the value of Y was not altered. In neither case was the value of Y modified as it appeared in the Y register.

All of the DDA results obtained during the simulations resulted from the DDA as mechanized above where $1\Delta\theta = 1/4$ milliradian. At any given instant in time, the value of an element of the transformation matrix is equal to the algebraic sum of the contents of the corresponding Y and R registers.

V. COMPUTER SIMULATION

5.1 Goals of Simulation

Extensive simulation has been performed on a Honeywell 1800 computer to evaluate the algorithm developed in Section 3.1 and especially to obtain a more precise feel for the behavior of the accumulated error of the C matrix. The simulations were performed in floating point arithmetic with a mantissa of 10 decimal digits and an exponent of 2 digits.

The simulation programs were designed essentially to:

a. Simulate rotational velocities and accelerations incurred by a spacecraft for any specified flight profile.

b. Determine changes of spacecraft attitude angles (θ's) about each of the spacecraft's axis for consecutive sampling time intervals of length T for the duration of the flight profile.

c. Update the third order N matrix of Table 3 at time intervals *T*.

d. Determine the true C matrix as a function of time by utilizing knowledge of the flight profile to solve equation (5).

e. Determine the error matrix E(t) by comparing the matrices resulting from steps c and d.

It must be emphasized that the simulations were concerned only with determining the efficacy of the algorithms as an analytical msthod of maintaining the C matrix. No effort was made to solve the navigation and guidance equations of the simulated flight profiles. Thus this study at best represents an effort to investigate only one of the many problems associated with the strapdown configuration.

The final step of the study was to simulate the performance of the DDA as represented by the set of equations *(24)* for certain of the profiles and to determine the error matrix resulting from the DDA updating technique. These simulations permit a comparison of the performance of the DDA with the performance of the whole number algorithms. While results of the DDA

simulations are included, the remainder of this report is concerned principally with a discussion of the whole number algorithm.

5.2 Determination of True C Matrix

The principal uncertainty in the results of the many simulated flight profiles is the accuracy of the standard solution against which the results are compared. Theoretically, the solution to equation (5) would provide the precise standard which is desired; in practice, however, the approximations introduced by the computer differential equation subroutine make the accuracy of the solution questionable.

The differential equation subroutine utilized in this study permits the solution of any set of simultaneous differential equations of the form of equation (5) provided that the highest derivative is piecewise continuous and that the locations of the discontinuities are known in advance. To relate these restrictions to the problem at hand, it is noted that the LEM, and in fact the great majority of present day maneuverable spacecraft, is attitude controlled by the thrusting of reaction jets. Throughout the simulations, the attitude jets were assumed to be capable of existing in only one of two states, "on" or "off." When turned "on" the jets provide a thrust which results in a constant angular acceleration; when "off" the jets provide no angular acceleration, Because the transition between "on" and "off" is assumed to occur instantaneously, the angular accelerations measured by the spacecraft - and hence indirectly the elements of the matrix [C(t)] as updated by the computer - are discontinuous at the time the attitude jets are switched. In order to correctly evaluate equation (5) using the differential equation subroutine the times of such discontinuities must be known in advance.

It has been assumed in this study that the differential equation solution to equation (5) for each flight profile yields an accurate C matrix against which the results of the algorithm can be compared. The accuracy of the differential equation subroutine utilized depends upon the size of an incremental interval of time At, during which the dependent variable must be continuous. To determine the validity of the differential equation solution for a given flight profile, the subroutine was run several times with decreasing values of the increment At; the resulting solutions were

checked for convergence. As an example, at the termination of a particular 40 second mission, the convergence of the solutions corresponding to decreasing time increments At is depicted in Table 5.

The convergence indicated by Table 5 is typical of convergence observed for other flight profiles and suggests that a time increment of At = .0015625 second provides a sufficiently accurate solution to equation (5). However, it was discovered during the simulations that for flight profiles exceeding a duration of 100 seconds, the requirements imposed by the small value of Δt exceeded the single precision capabilities of the computer; it was also observed that solutions with a time increment of .0015625 second required an unreasonable amount of computer time in proportion to the scope of this study. Hence for all flight profiles the "true" C matrix was obtained by solving the differential equation (equation (5)) with an incremental time interval At = .003125. The resulting C matrix can be considered accurate to at least the sixth decimal place.

5.3 Characteristics of Simulated Flight Profiles

The flight profiles which were simulated in this study fall into two basic categories:

1. Missions which in the most general case consist of alternate polarity acceleration pulses applied independently to the attitude jets of each of the three spacecraft axes,.

2. Profiles which represent a typical LEM mission.

The following constraints were imposed upon the spacecraft maneuvers called for in the simulations:

1. All angular accelerations about each axis were of constant magnitude, 3/4 radian per second per second.

2. For profiles of the first category, rotational velocities about each axis were limited to magnitudes of 20" per second or less; maneuvers in the LEM missions were limited to rotational velocities whose magnitudes were 10" per second or less.

$$[C]_{\Delta t = .00625} - [C]_{\Delta t = .0125} =$$

$$\begin{bmatrix} 6.5 \times 10^{-6} & 2.1 \times & -2.7 \times 10^{-7} \\ -2.3 \times 10^{-6} & 6.4 \times & -3.2 \times 10^{-7} \\ -6.5 \times 10^{-7} & -1.4 \times 10^{-6} & 4.8 \times \end{bmatrix}$$

$$= .003125 - [C]_{\Delta t = .00625} =$$

$$\begin{bmatrix} -4.5 \times 10^{-7} & -1.3 \times 10^{-7} & 1.1 \times 10^{-7} \\ 1.7 \times & -4.4 \times & -8.3 \times 10^{-8} \\ 9.2 \times & 2.0 \times & -1.4 \times 10^{-7} \end{bmatrix}$$

$$[C]_{\Delta t = .0015625} - [C]_{\Delta t = .003125} =$$

$$\begin{bmatrix} 3.5 \times 10^{-8} & -1.1 \times 10^{-8} & -6.2 \times 10^{-9} \\ -1.2 \times 10^{-8} & 4.2 \times 10^{-8} & 1.5 \times 10^{-9} \\ 3.6 \times 10^{-9} & -1.7 \times 10^{-8} & 6.2 \times 10^{-9} \end{bmatrix}$$

TABLE 5. CONVERGENCE AT *40* SECONDS OF DIFFERENTIAL EQUATION SOLUTION OF COSINE MATRIX FOR DECREASING TIME INCREMENTS, At (At MEASURED IN SECONDS)

Characteristics of successive runs were dictated by the desire to isolate the effects of the various parameters which influence the capabilities of the basic algorithm.

5.4 Effect of Vehicle Rotation

The characteristics of the first few simulations were designed to isolate the effects of vehicle rotation. upon the matrix [E(T)] given in Table 3. A single acceleration pulse, commencing at time t = 0 and of a specified duration, was applied to the x axis attitude control jets; subsequent to the termination of acceleration, the spacecraft was allowed to rotate freely at a constant angular velocity for a duration of 200 seconds during which time the third order N matrix of Table 3 was updated at sampling time intervals of 0.1 second. Periodically during the 200 seconds, the updated N matrix was compared with the true transformation matrix [C(t)] (differential equation solution) and the resulting error matrix [E(t)] = [C(t)] - [N(t)] was determined. The magnitude of the elements of [E(t)] represents the degree to which the updated N matrix approximates the M matrix of equation (6).

The profile described above was simulated for the acceleration pulse lasting .025 second, 0.25 second and .465625 second. (The last value represents the approximate time which it would take a body under an angular acceleration of *3/4* radian per second per second to achieve a rotational velocity of 20" per second.) The behavior of one element, e_{22}, of the resulting error matrix for each of the three profiles is shown in Figure 1. It should be noted that there is nothing unique about the element e_{22}; its behavior is simply typical of - but not identical to - the other elements of the error matrix.

There are several interesting properties about the functions shown in Figure 1. According to the error matrix of Table 3, it was expected that the element e_{22} should be essentially a constant for the duration of the profile since e_{22} of Table 3 reduces in this case to $-\frac{T^4}{24}\omega_x^4$. It is apparent from Figure 1 that the truncated error matrix of Table 3 does not adequately represent the behavior of the basic algorithm; evidently

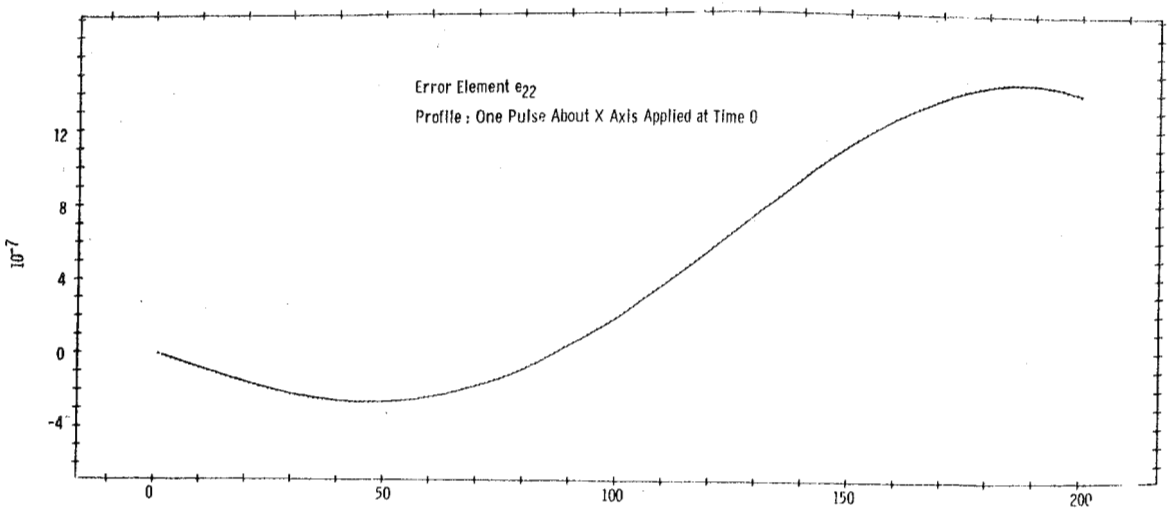


Fig. 1 A

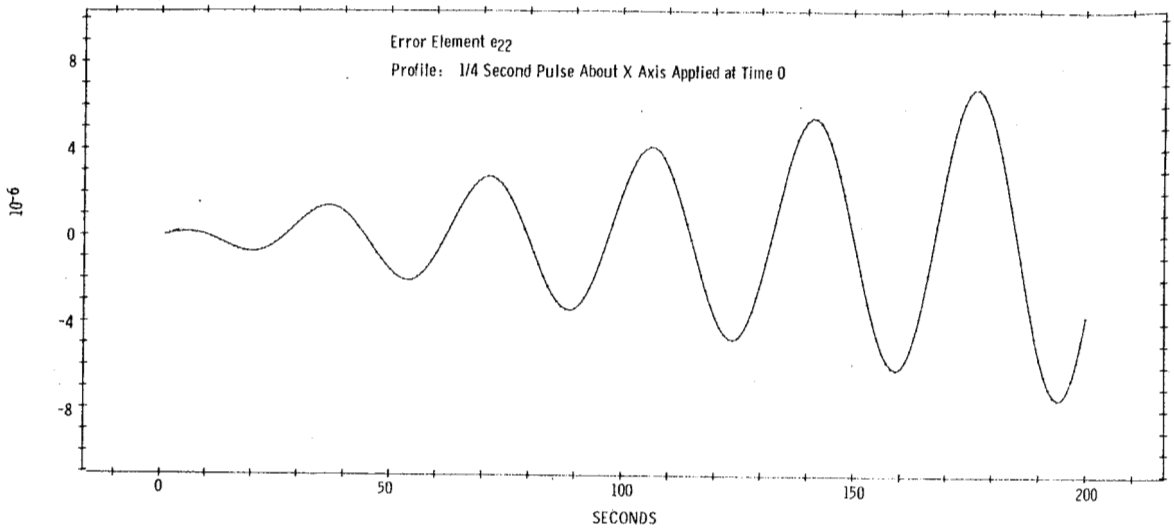


Fig. 1 B

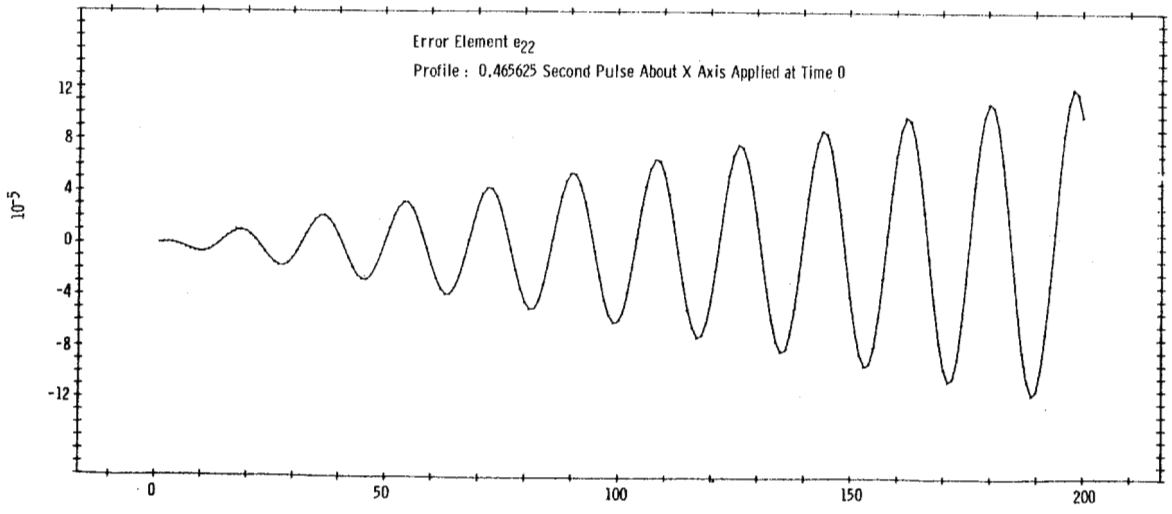


Fig. 1 C

higher order terms must be considered. The periods of oscillation of the error functions are identical to the time it takes for the spacecraft to rotate 360° while the peaks of the error function oscillations grow in a linear fashion which is apparently a function of the speed of rotation. The unfortunate result is that the error peaks appear to be monotonically increasing. From the three functions depicted, no simple relationship between the rotational velocity and the growth of the error peaks has been determined. One is tempted to conclude from Figure 1 that the errors resulting from the basic algorithm are functions of the spacecraft velocity and attitude. It should be remembered, however, that the differential equation solution was shown to converge only to the sixth decimal place for At = .003125; therefore, considering the magnitude of the error, one might question the accuracy of these initial conclusions.

To ensure that simultaneous rotation about each of the body axes does not adversely affect the performance of the basic algorithm, a flight profile similar to the above was simulated. This profile consisted of applying a .025 second acceleration pulse about each of the body axes at time t = 0 and then allowing the spacecraft to rotate freely for 200 seconds. The behavior of element e_{22} of the resulting error matrix is shown in Figure 2. While unquestionable conclusions cannot be drawn from one simulation, nevertheless comparison of Figure 2 with Figure la indicates that simultaneous rotation about the three body axes does not significantly affect the accuracy of the basic algorithm.

5.5 The Basic Profile

From the results observed for constant rotation of the spacecraft, it became apparent that more sophisticated maneuvers must be studied. A flight profile was designed which consisted essentially of limit cycle maneuvers performed about each of the body axes; because this profile was the basis of the great majority of the simulations, it will hereafter be referred to as the basic profile. The characteristics of the accelerations applied about each of the body axis are shown in Figure 3. The sequence of the pulses about the z axis permits acceleration to the maximum allowable rotational speed of 20° per second, free rotation at this velocity for a period of time, followed by deceleration to zero rotation about the

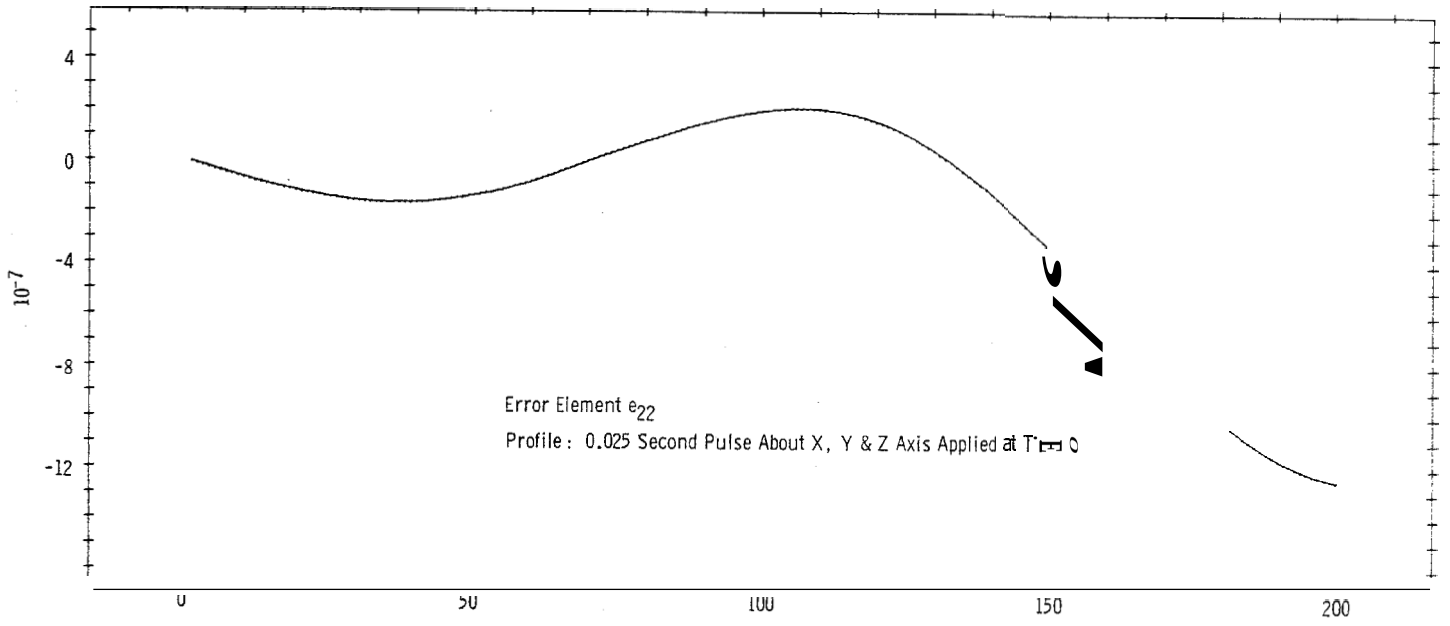
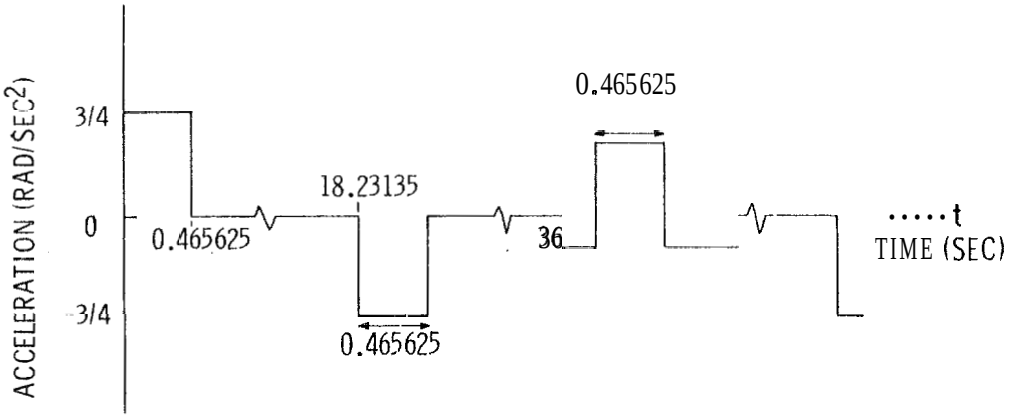
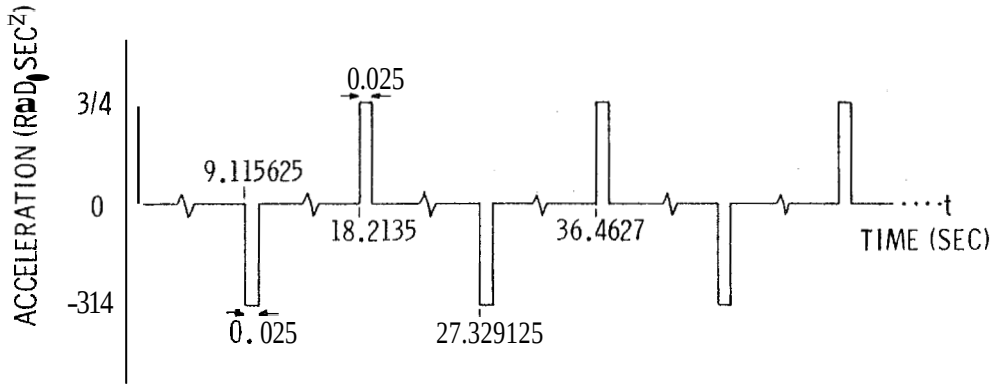

4
0
-4
-8
-12
10^{-7}
0
50
100
150
200
SECONDS
Error Element e_{22}
Profile : 0.025 Second Pulse About X, Y & Z Axis Applied at T = 0

Fig. 2



Z Axis Attitude Accelerations



X & Y Axes Attitude Accelerations

Fig. 3. Characteristics of Basic Profile

z axis. The vehicle is assumed to be rotating at time t = 0 about the x and the y axes with a velocity $\omega_x = \omega_y = -.009375$ radian per second; thus the x and y axis limit cycle maneuvers are centered about the respective axis.

The basic algorithm was used to update the N matrix during simulations of the above described profile for elapsed mission times of 1000 seconds. Sampling time intervals of 0.1, 0.05, and 0.025 second were employed. A crude graph of the behavior of one element, e_{31}, of the error matrix $[E(t)] = [C(t)] - [N(t)]$ for each of the three values of the sampling time interval is shown in Figure 4. Figure 5 is a more detailed presentation of the behavior of e_{31} for the first 200 seconds of the mission with a sampling time interval T = 0.1 second.

Figure 4 substantiates the prediction that areduction in size of the sampling time interval results in a corresponding decrease in the magnitude of the errors. According to the error matrix of Table 3, a reduction in size of the sampling time interval by a factor of two should reduce the error by a factor of sixteen. While it is not immediately apparent from the functions of Figure 4, comparison of corresponding raw data points indicates that halving the sampling time interval results in a reduction of error magnitude by a factor of only five. This result tends to substantiate the earlier conclusion that at least some of the higher order effects which were omitted in the basic algorithm and the error matrix are not truly negligible.

There are several interesting observations which can be drawn from the function shown in Figure 5, which is an expansion of the first 200 seconds of the profile. The frequency of the sinusoidal type pulses, which evidently result from high speed rotation about the z axis, is identical to that observed in Figure 1c. Since the magnitude of rotatation is the same for both cases, this is not an unexpected result; however the magnitude of the error now indicates that the differential equation routine is not the cause of oscillatory behavior of the error function. Where the peaks of the error in Figure 1c grow linearly, such is not the case for the basic profile. In fact, according to Figure 4, the magnitude of the error peaks is well bounded. The most significant result which is apparent in Figure 5 is the difference in the order of magnitude between the errors observed for this profile and the error function of Figure 1c even though the magnitude of body rotation is the same for both profiles. Such a discrepancy can only be caused by one or more of the following:

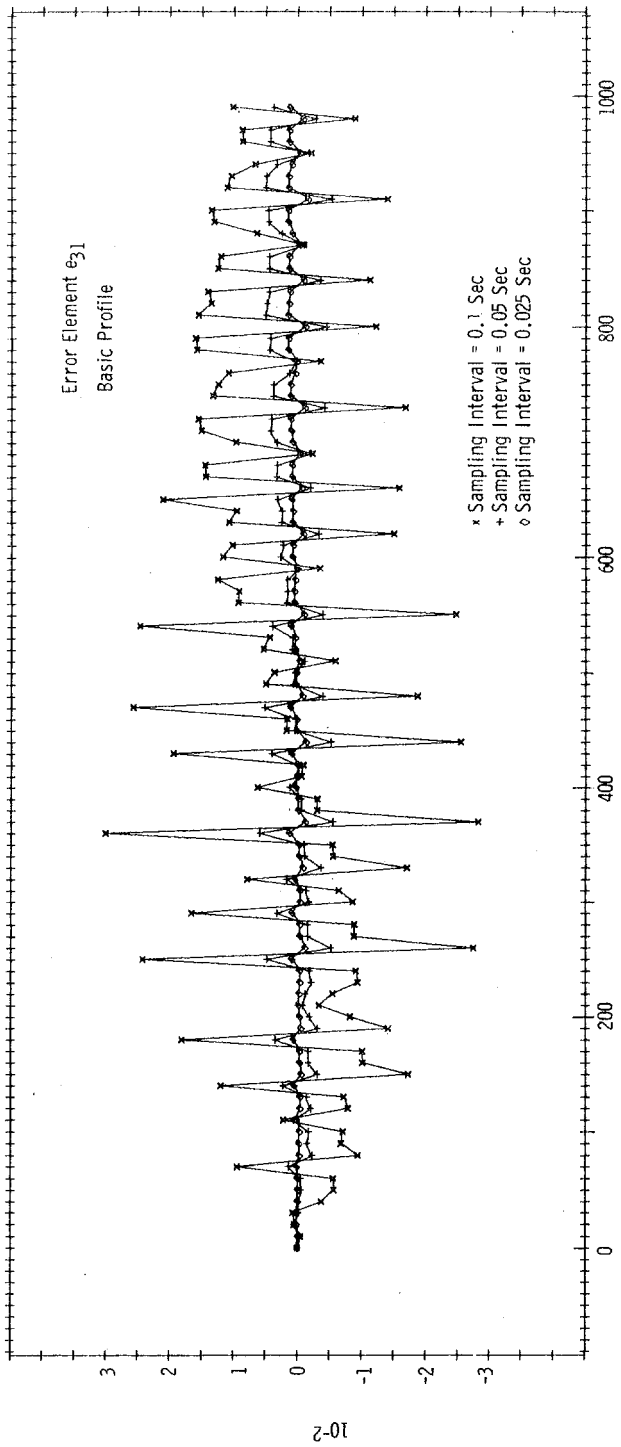

Error Element e_{31}
Basic Profile
× Sampling Interval = 0.1 Sec
+ Sampling Interval = 0.05 Sec
◊ Sampling Interval = 0.025 Sec
10-2
3
2
1
0
-1
-2
-3
0
200
400
600
800
1000
SECONDS

Fig. 4 -

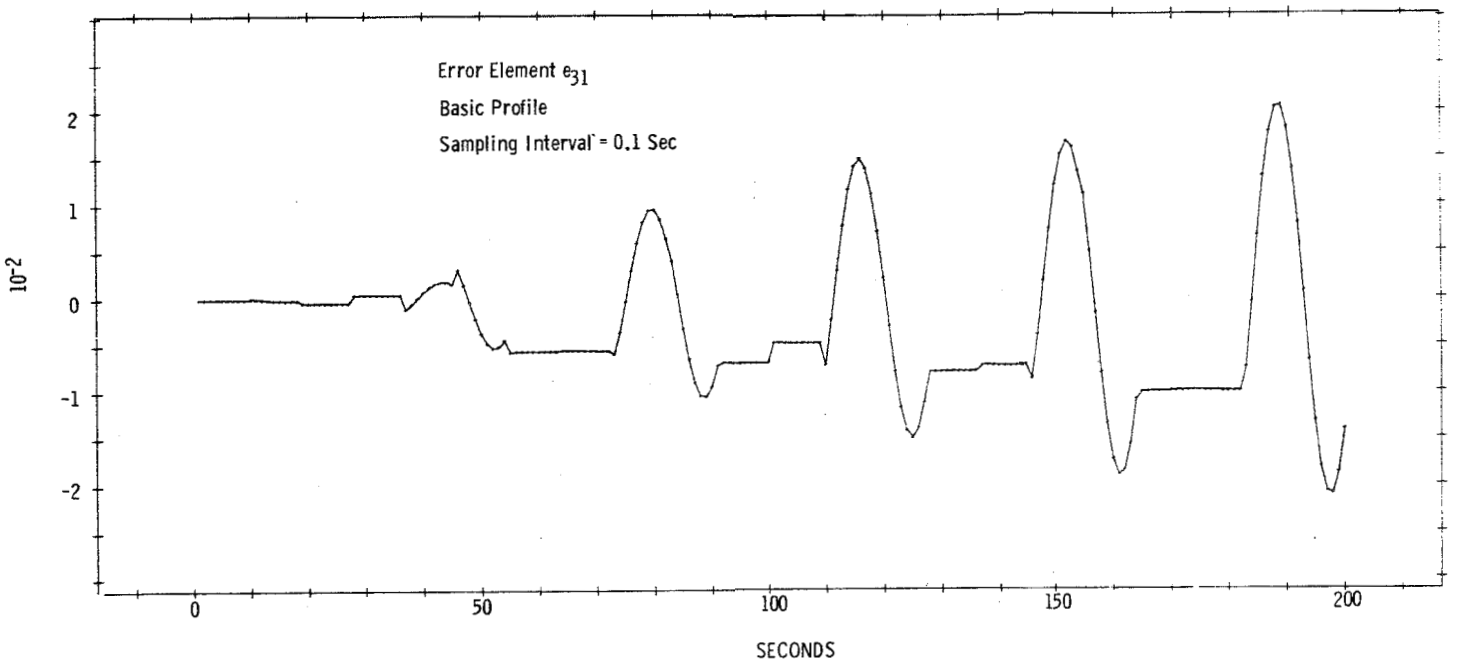

Error Element e_{31}
Basic Profile
Sampling Interval = 0.1 Sec
10^{-2}
2
1
0
-1
-2
0
50
100
150
200
SECONDS

Fig. 5

1. Simultaneous acceleration about one body axis and rotation about the other body axes; i.e., occurrrence of non-synchronous accelerations about the three body axes,

2. Occurrence of limit cycling or repetitive accelerations,

It should be remembered that simultaneous rotation of small magnitude about the three body axes did not previously appear to affect the performance of the basic algorithm.

To determine the effects of the limit cycle maneuver, a profile similar to the basic profile was simululated; in this profile, however, no accelerations were applied to the z axis. The error function $e_{31}(t)$ which results from limit cycle maneuvers about the x and z axes is shown in Figure 6. Although the speed of rotation about the two axes of the spacecraft was identical to that of the earlier profile, the magnitude of the errors is nevertheless considerably larger than that observed in Figure la. Because the x and y axis accelerations are synchronous, the error shown in Figure 6 can only be due to the repetition of accelerations. It appears that the errors are somewhat cumulative; however the periodicity of groups of three error pulses remains unexplained at this time.

Further evidence that the performance ,of the basic algorithm is influenced by the occurrence of repetitive accelerations is presented in Figure 7 which shows the error function $e_{31}(t)$ resulting from simulation of the z axis accelerations only of the basic profile. For the profile in which one .465625 second acceleration pulse was applied to the z axis attitude jets followed by a 200 second period of free rotation (profile for Figure 1c), the error function $e_{31}(t)$ was identically zero. Thus the existence of the error function shown in Figure 7 can be caused only by the repetitive accelerations.

5.6 Non-Synchronous Accelerations

The simulations considered thus far have consisted of applying accelerations simultaneously to various combinations of the spacecraft axes. To investigate the effect of asynchronous accelerations, two profiles were simulated. The profiles consisted of accelerations of alternating polarity applied to the x, y, and z axes at multiples of 4, 5, and 7 seconds respectively; in one profile the accelerations lasted for .025 second while in the other

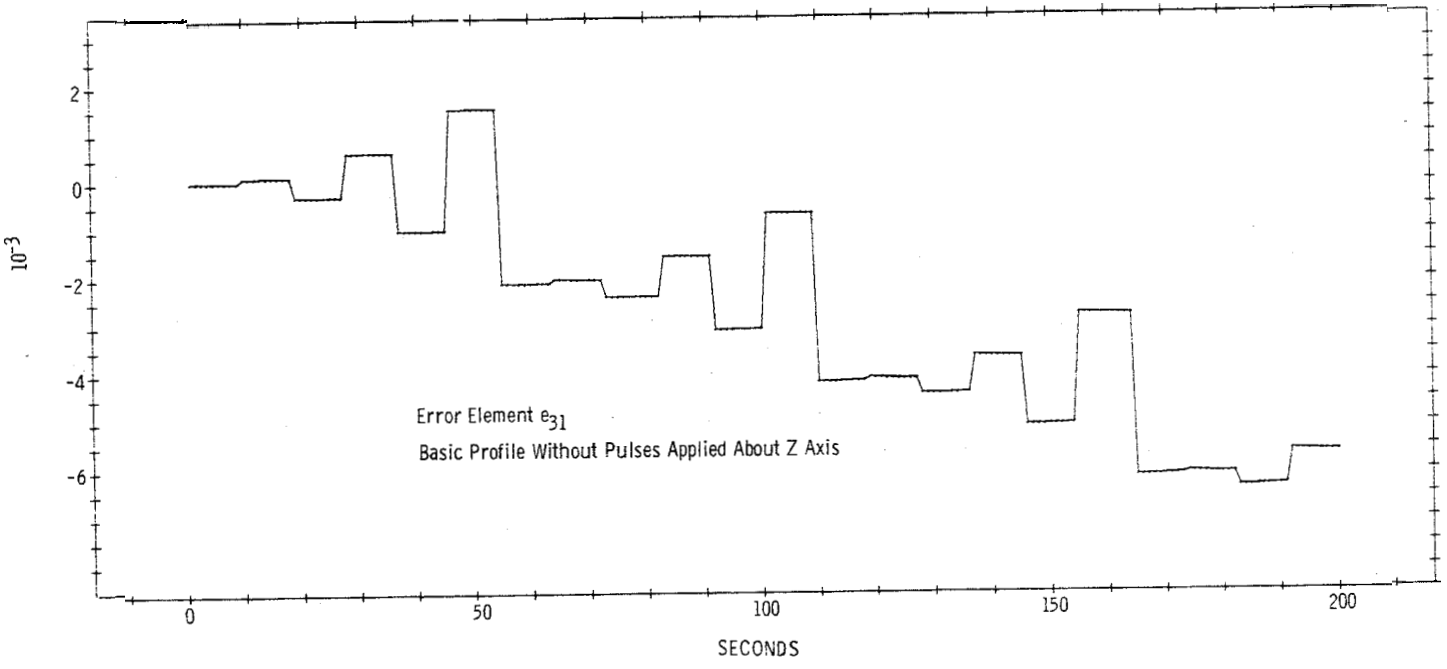

2
0
-2
-4
-6
10^{-3}
0
50
100
150
200
SECONDS
Error Element e_{31}
Basic Profile Without Pulses Applied About Z Axis

Fig. 6

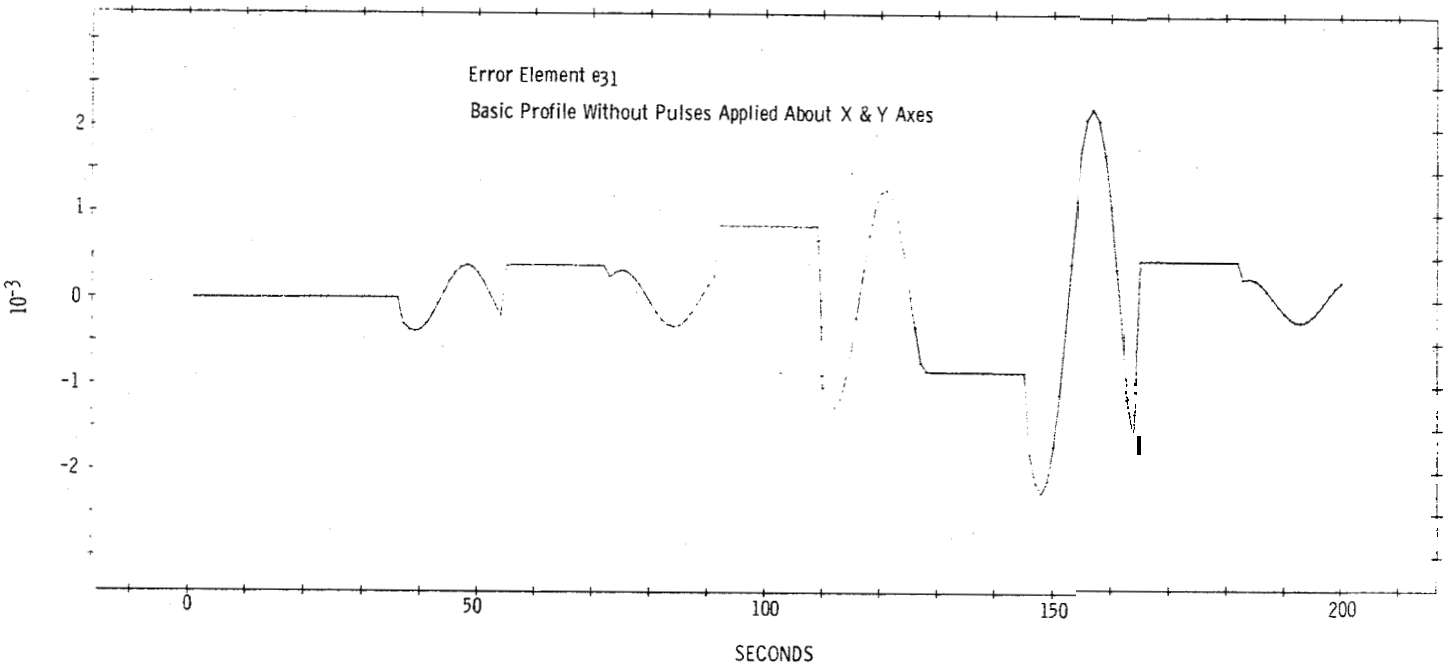

Error Element e_{31}
Basic Profile Without Pulses Applied About X & Y Axes
10^{-3}
2
1
0
-1
-2
0
50
100
150
200
SECONDS

Fig. 7

profile the pulses were of .465625 second duration. The error functions $e_{31}(t)$ for the .025 and the ,465625 second profiles are shown in Figures 8a and 8b respectively. The functions shown in Figure 8 substantiate the previous conclusion that the magnitude of error is dependent in part upon the speed of rotation of the spacecraft. It is significant to note, however, that the error shown in Figure 8b is of smaller magnitude than the corresponding error of the basic profile, even though the profile for Figure 8b calls for high speed rotation about each of the spacecraft axes while the basic profile calls for high rotation about only the z axis. Such a result was completely unexpected and at the time unexplained.

5.7 Extensions of the Basic Algorithm

Of particular concern to the design of the LEM navigation and guidance system is the magnitude of errors resulting from seemingly non-violent maneuvers. Since the updated N matrix is an approximation to a matrix of direction cosines, an error of 3×10^{-2} (see Figure 4) can represent a spacecraft attitude error of almost 2 degrees, an error which is unacceptable for the LEM mission. In at attempt to reduce the size of the errors while simultaneously gaining more insight into the characteristics of the basic algorithm, several extensions of the algorithm were developed and simulated. These modifications and the results of their simulation are described below.

5.7.1 Reduction of Sampling Time Interval

The error matrix of Table 3 predicts, and the error functions of Figure 4 verify, that a reduction of the sampling time interval results in a reduction of the error magnitude. However, if the sampling time interval is made small enough to ensure that the updated N matrix closely approximates the true solution, an unreasonable computational load is placed upon the navigation computer. An attempt was made to realize the reduction of error magnitude by sampling attitude angle changes at relatively short time intervals while performing update calculations only at longer time intervals. The update calculations are of course more complex than the elements of the N matrix shown in Table 3.

t)

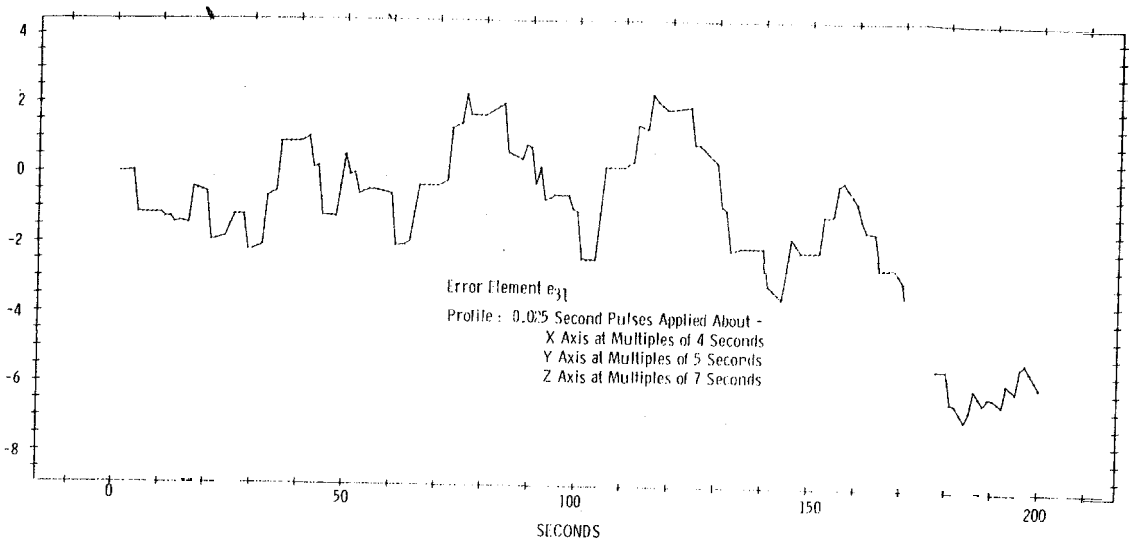


Fig. 8 A

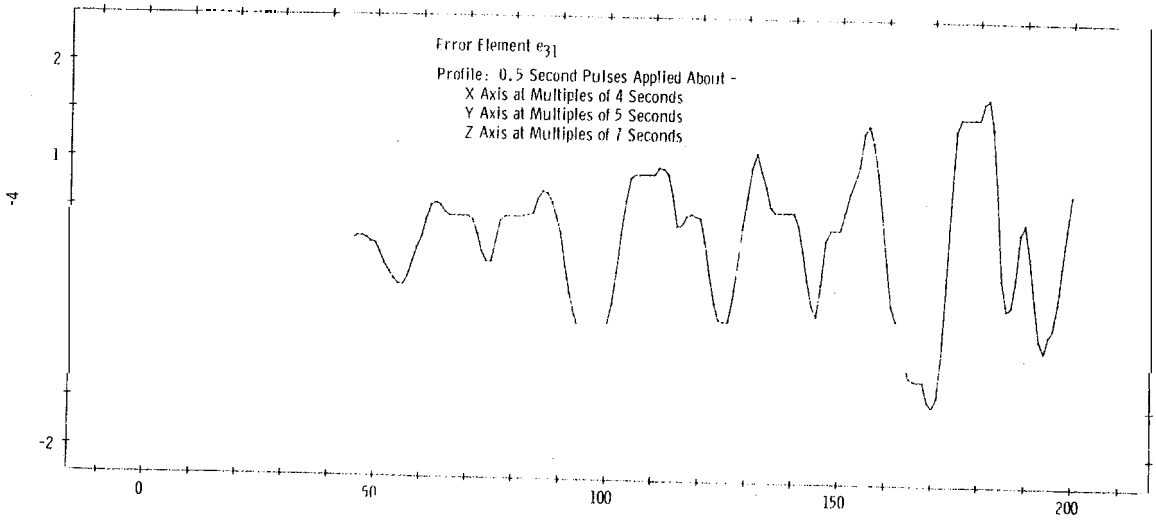


Fig. 8 B

Referring to Figure 9, let the long update time interval T be divided into two shorter time intervals, each of length T/2.

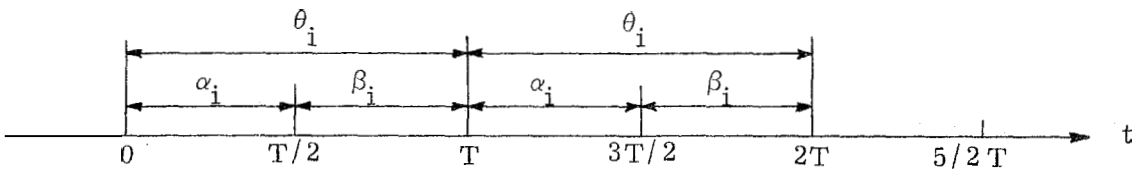


Fig. 9. Reduction of Sampling Time Interval

Within each interval T, denote the change of attitude angle about the i^{th} body axis during the first interval T/2 by α_i and the change of angle during the second interval T/2 by β_i; thus the total angle change, θ_i, equals $\alpha_i + \beta_i$. If the N matrix of Table 3 is updated at intervals T/2 rather than intervals T, the N matrix at time T is:

$$[N(\alpha, \beta, T)] = [N(\alpha, T/2)][N(\beta, T/2)] \tag{25}$$

where $[N(\alpha, T/2)]$ and $[N(\beta, T/2)]$ have the same elements as given in Table 3 for $[N(\theta, T)]$ except that θ becomes α and β respectively. By sampling the angles α and β at times T/2 the result given in equation (25) can be obtained by updating a new N matrix, [N'], periodically at time intervals T. The expression for [N'] is of course:

$$[N'(\alpha, \beta, T)] = [N(\alpha, T/2)][N(\beta, T/2)] \tag{26}$$

and includes terms of the sixth order rather than the third order as given in Table 3. (The expansion $[N(\alpha, T/2)][N(\beta, T/2)]$ is rather tedious and, since it is of no real significance, is not included in this report.) Thus the effect of updating the N matrix at shorter time intervals can be realized by measuring the angle changes at shorter time intervals T/2 while performing update calculations periodically at longer time intervals T.

The above procedure was simulated with the following exception: in order to keep the computational load on the navigation computer at a reasonable level, it was decided not to include in the elements of the N' matrix terms which were of the third order or greater. The results of the simulation of the N' matrix approximation are not plotted but in general the elements

of the error matrix resulting from this procedure are somewhat smaller than those resulting from the basic N matrix approximation; however, the reduction of the error matrix is less than one order of magnitude. Considering that the computational load is higher for the N' matrix than €or the N matrix, such results are not encouraging.

From the expansion of the product $[N(\alpha, T/2)][N(\beta, T/2)]$, it was observed that the leading terms of the difference $([N(\alpha, T/2)][N(\beta, T/2)]) - [N(\theta, T)]$ (remembering that $\theta = \alpha + \beta$) were terms typically of the form $(\alpha_i \beta_j - \alpha_j \beta_i)/2$. Since these terms do not represent an unreasonable amount of computation, a logical suggestion is to include these terms in the elements of the third order N matrix of Table *3* to determine whether or not they contribute significantly to the reduction of error observed in Figure 4 when angle changes are sampled at twice the update rate. The results obtained from the simulation of this amended N matrix approximation indicate that the elements of the resulting error matrix are slightly smaller than the error terms resulting from the N' matrix approximation. Again, however, the decrease in error magnitude is less than one order of magnitude.

5.7.2 Fourth Order N Matrix

The error matrix shown in Table *3* results from truncating the expansion of the N matrix elements of the third order. To verify that the updated third order N matrix is in fact a reasonable approximation to the cosine matrix, an N matrix was constructed whose elements include the fourth order terms necessary to eliminate the fourth order terms of the error matrix. The fourth order N matrix was then used in simulated profiles to approximate the cosine matrix. That it is unnecessary to include fourth order terms in the N matrix was demonstrated by the fact that the resulting errors were at least 90% as great as the errors observed for the third order N matrix.

The results obtained for the fourth order N .matrix in turn suggest that the necessity of including third order terms in the N matrix is questionable. Profiles were simulated wherein various combinations of the third order terms of the N matrix of Table 3 were not included in the elements of the update matrix; in one simulation, no third order terms were included. The results of these simulations showed that the omission of the third

order terms of the basic algorithm results in error functions which are one order of magnitude greater than the error functions resulting from a full third order N matrix. It thus appears that, at least for the flight profiles which were simulated, an updated full third order N matrix represents a reasonable approximation to the direction cosine matrix with a corresponding acceptable level of computational complexity.

5.7.3 Interrupted Sampling Time Interval

The derivation of the basic algorithm results from the Taylor series expansion of the changes of spacecraft attitude angles during a specified time interval as given by equation (13). For equation (13) to be a valid representation of Θ, the function $\Theta(t)$ and all of its time derivatives must be continuous throughout the time interval T. However, for the method described for implementing the basic algorithm, the requirement for continuous derivatives is not necessarily met. Permitting the spacecraft attitude control jets - and hence the body angular acceleration - to be in only one of two states, on or off, introduces a discontinuity in the angular accelerations at the time the control jets are switched. Unless these discontinuities occur at the initiation of a sampling time interval, the function $\Theta(t)$ has discontinuous derivatives and the expansion of equation (13) over the sampling interval becomes invalid.

A necessary condition to ensure that the function $\Theta(t)$ has no discontinuous derivatives during a sampling interval is that changes in spacecraft acceleration be made coincident in time with the beginning of a sampling time interval. For any realistic flight profile, it is impractical to predict the exact times when changes in acceleration will occur; it is therefore impossible to determine a priori a fixed value T for the sampling time interval which guarantees coincidence between the sampling interval and acceleration change for the entire mission. The obvious solution is to force, at the time of acceleration change, the current sampling interval to be terminated and the subsequent interval initiated.

Several practical methods of implementing the interruption of the sampling interval can be suggested. If the navigation computer is also performing the guidance functions of the spacecraft, knowledge of the attitude jet firing is implied; otherwise an interrupt signal from the

guidance equipment to the navigation computer is required. In either case, the navigation computer can terminate the current sampling time interval, perform update calculations and initiate the subsequent sampling time interval; thus no discontinuous derivatives are permitted to occur during a sampling time interval. On the other hand, in any realistic space mission, there are bound to be rotational accelerations of the spacecraft which will not be initiated or observed by the on-board computer. Fuel slosh, motion by the spacecraft occupants, etc., result in changes in rotational accelerations about which the guidance computer has no knowledge. Thus complete coincidence between sampling time intervals and changes in vehicle accelerations cannot be simply assured. It might be noted, however, that these sources of accelerations are not as sharp as the jets and presumably will not introduce large additional errors.

Although only one method of interruption was simulated, it is felt on the basis of earlier simulations that other methods would yield essentially the same results. Under the simulated procedure, the length of the sampling time interval is set at some constant value T and, in the absence of acceleration changes, update calculations are performed periodically at time intervals T as usual. When an acceleration change occurs at time t during the n^{th} sampling time interval (n = integer), the n^{th} interval is terminated at time t and the length of the $n + 1^{st}$ interval is set at nT - t. Until the next acceleration change occurs, all subsequent intervals are of length T. (The length of the $n + 1^{st}$ interval could have been set at T.) Changes in spacecraft attitude angles are noted and update calculations are performed periodically every nT seconds for the duration of the flight profile and in addition are performed at every change of acceleration.

Although the interruption of the sampling time interval assures that the M matrix of equation (6) can in theory be approximated by an N matrix such as that given in Table 3, nevertheless the interruption introduces an additional error which results from the fact that all sampling intervals are not of the same length. To illustrate this inherent error, consider the case where the n^{th} sampling interval is of length T while the $n + 1^{st}$ interval is of length t. For the sake of notational convenience, let time 0 represent the termination of the n^{th} interval and the initiation of the $n + 1^{st}$ interval; in other words, at time t a change in acceleration occurs resulting

in two consecutive sampling intervals of different lengths. The time sequence is illustrated in Figure 10. It is convenient now to investigate some of the functions of Table 2 as they are calculated at the end of the $n + 1^{st}$ interval. For example:

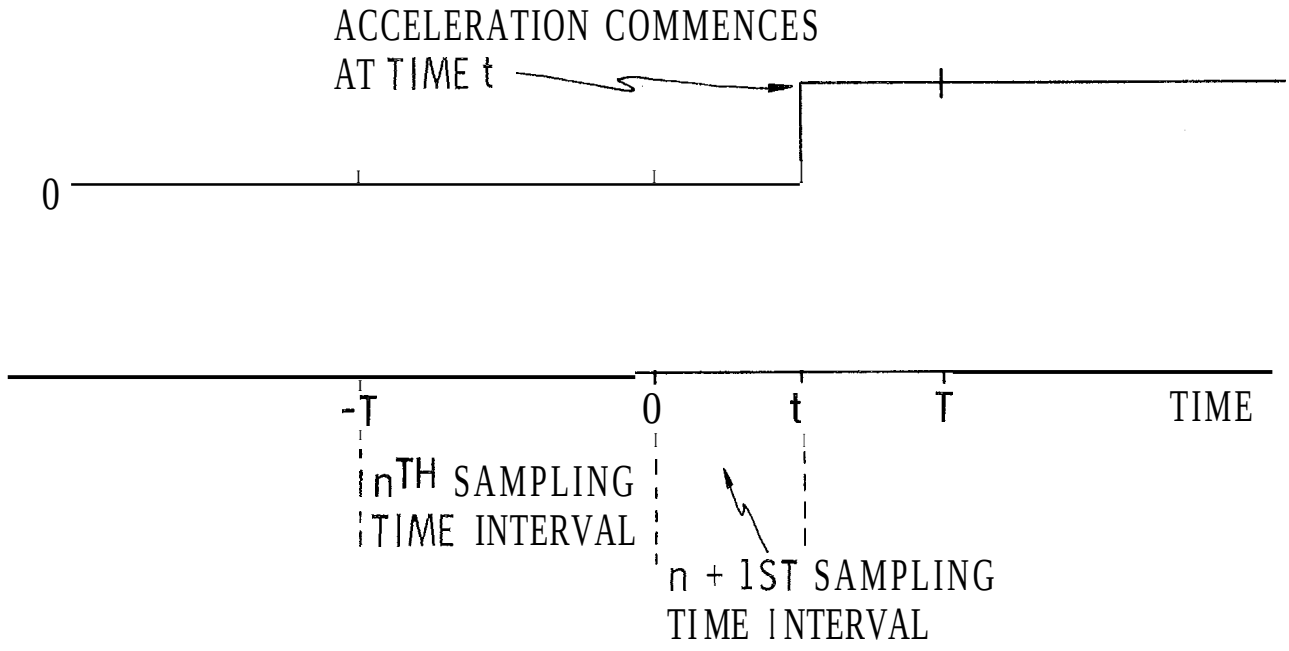
$$\theta_i = \int_0^t \omega_i(t)dt = \int_0^t \left[\omega_i(0) + \frac{t^2}{2}\ddot{\omega}_i(0) + \ldots\right] dt$$

$$= t\omega_i(0) + \frac{t^2}{2}\dot{\omega}_i(0) + \frac{t^3}{6}\dddot{\omega}_i(0) + \ldots \qquad (27a)$$

$$\theta_{-j} = \int_0^{-T} \omega_j(t)dt = \int_0^{-T} \left[\omega_j(0) + t\dot{\omega}_j(0) + \frac{t^2}{2}\ddot{\omega}_j(0) + \ldots\right] dt$$

$$= -T\omega_j(0) + \frac{T^2}{2}\dot{\omega}_j(0) - \frac{T^3}{6}\dddot{\omega}_j(0) + \ldots \qquad (27b)$$

$$\theta_i\theta_{-j} = -tT\omega_i(0)\omega_j(0) - T\frac{t^2}{2}\dot{\omega}_i(0)\omega_j(0) + t\frac{T^2}{2}\dot{\omega}_j(0)\omega_i(0) + \ldots \qquad (27c)$$



(27c)

Fig. 10 The Sequence of Interrupted Sampling Time Interval

$$\theta_i\theta_{-j} - \theta_{-i}\theta_j = \frac{Tt^2}{2}\left[\dot{\omega}_j(0)\omega_i(0) - \omega_i(0)\dot{\omega}_j(0)\right.$$

$$+ \frac{tT^2}{2}\left[\dot{\omega}_j(0)\omega_i(0) - \omega_j(0)\dot{\omega}_i(0)\right]$$

$$= \left[\frac{Tt^2}{2} + \frac{tT2}{2}\right]\left[\dot{\omega}_j(0)\omega_i(0) - \dot{\omega}_i(0)\omega_j(0)\right] \qquad (27d)$$

Inspection shows that equation (27d) is identical to the corresponding function given in Table 2 only when $t = T$, a condition which is not possible when an interrupt occurs. Thus the interrupt inherently introduces an error not shown in Table 3 which in this case is represented by the factor $T^3 - \frac{}{2}(Tt + tT^2)$. In general the error is a function of the difference in length between consecutive sampling time intervals.

Updating the full third order N matrix given in Table 3 with an interrupted sampling time interval was simulated using the basic profile. The normal sampling time interval was 0.1 second, the same value used in the earlier simulations. The error functions $e_{31}(t)$ resulting from the interrupted simulation is shown in Figure 11. The principle result to be noticed from a comparison of Figure 11 with Figure 5 is that the error has been reduced by three orders of magnitude. Such significant reduction is not limited to this profile; comparison of the interrupted update method with the basic algorithm for other profiles revealed similar results.

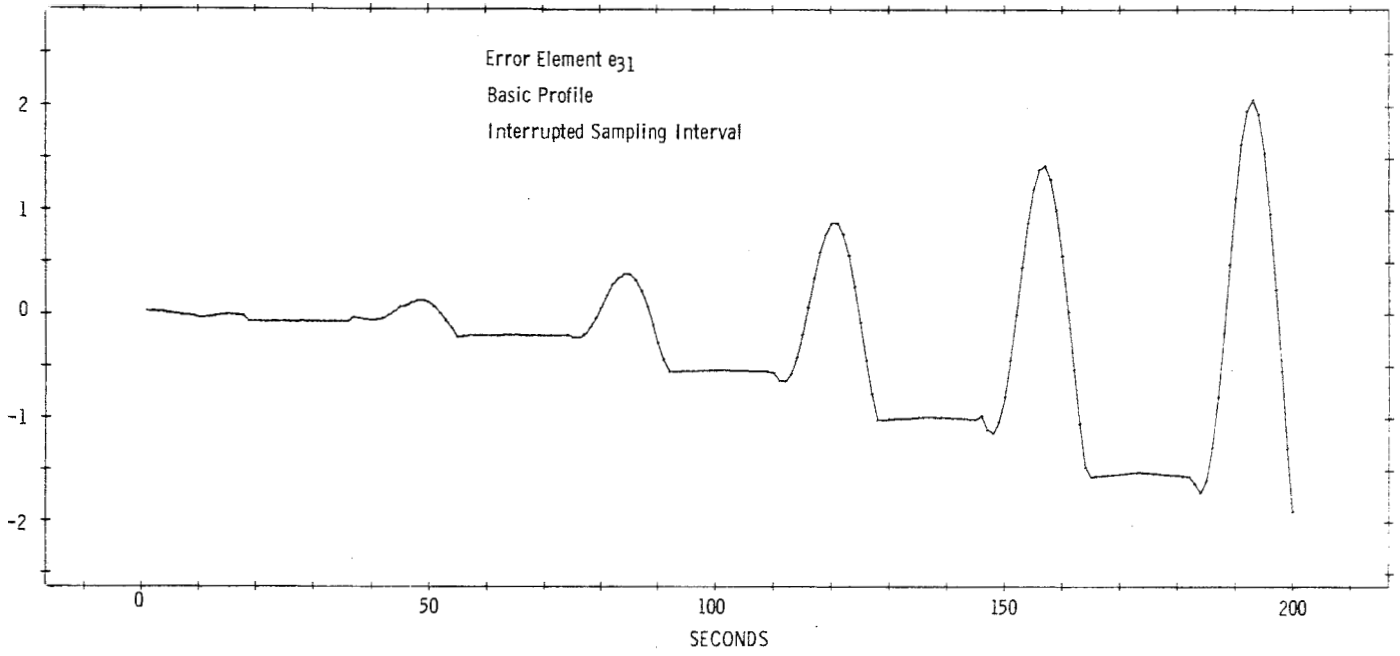

Error Element e_{31}
Basic Profile
Interrupted Sampling Interval
5-01
2
1
0
-1
-2
0
50
100
150
200
SECONDS

Fig. 11

5.8 Practical Considerations

5.8.1 Computer Word Length

A full third order update N matrix approximation evaluated ten times a second and at every change of acceleration results in an error matrix which corresponds to at least milliradian accuracy for the profiles which were simulated and in considerably better accuracy for the LEM profile which was simulated. To achieve such accuracy with the 15 bit word length of the Apollo Guidance Computer, calculations would have to be performed in double precision form. A rough estimate indicates that if the interpretive mode were employed, the computer would be fully occupied with this job alone. In order to reduce the time requirement levied against the AGC by the update formula, the calculations are expressed in a more convenient form.

The variables are first scaled such that the number of double precision additions is minimized. Terms of the N matrix in Θ occupy the higher component of the double precison word; terms in θ_i^2, $\theta_i\theta_j$ and $\theta^2\theta_i$ occupy the lower component. The maximum value of the θ_i is scaled by choosing the interval T in accordance with the maximum angular velocity of the spacecraft. These terms are accumulated to form the nine elements of the matrix [N-I] where [I] is the identity matrix. The matrix multiplication [C] x [N-I] is performed in single precision using the upper components of each matrix and the resulting double length product is added to [C]. The lower component of [N-I] is saved to be added to the next sampling period's [N-I]. The process is like that of the DDA where the lower component of a double precision word is saved and accumulated at each step. Because this form of mechanizing the computational procedure yields results which are less accurate than those resulting from precision, it will hereafter be referred to as computation of 1 1/2 precision.

The 1 1/2 precision form of computation described above was implemented on the Honeywell 1800 with two slight variations.

1. The 15 bit word length of the AGC was simulated as a five digit decimal word length. Thus the accuracy of the simulated solutions should be less than the true solutions for the described procedure. (Because the N matrix is a matrix of cosines, the first digit of either the binary or decimal word must have a magnitude of either 0 or 1 leaving 4 decimal digits and 14 bits to provide the precision of the cosine.)

2. All terms of the N matrix were accumulated in the lower component of the double precision word but were allowed to overflow into the higher component of the double precision word. However, because of the floating point arithmetic employed by the simulation computer, the higher component of the simulated double precision word always contains five significant digits of the cosine approximation. Thus the simulated procedure is accurate to the extent that terms of θ_i^2 and $\theta^2\theta_i$ do not contribute to the first five significant digits of the cosine approximation. The effects of the 11/2 computational precision will be discussed with the results of the simulated LEM mission.

5.8.2 Gyro Limitations

The results presented to this point are predicated upon the assumption that the changes in the spacecraft attitude angles during each sampling time interval are known precisely, i.e., that the gyros by which the angle changes are measured provide a continuous readout of angle change data to the navigation computer. In practice, of course, this situation is not realized. In the case of the LEM mission, the gyros are pulse torqued gyros which require one output pulse from the computer for each change of attitude angle $\Delta\theta$, a positive pulse for a net positive angle change and a negative pulse for a net negative angle change. Because angle changes are algebraically accumulated, are measured with respect to a fixed reference, and can only be measured to the nearest $\Delta\theta$ through which the vehicle has rotated, it is possible for the spacecraft to rotate between the angles $+\Delta\theta$ and $-\Delta\theta$ with no pulses being applied to the gyros. This range of angles is known as the dead band of the gyros and introduces a memory type effect into the determination of angle changes. A dead band of 1/2 milliradian ($\Delta\theta = \pm$ 1/4 mr.) was introduced into the simulations of all profiles; the effect of introducing imperfect gyros was observed to be essentially independent of the mission profile and is therefore discussed only with the results of the simulated LEM mission.

5.9 LEM Profile

The fundamental purpose of the entire investigation was to determine the practicality of utilizing the N matrix given in Table 3 to maintain spacecraft attitude during the LEM mission. The LEM mission profile which was simulated is in fact a fairly simple approximation to an actual LEM profile, the simplicity being a result of the author's ignorance of the detailed flight profile. The approximation will, however, suffice for the purposes of evaluating the results of the update approximation.

The simulated LEM mission profile consisted of maneuvering the spacecraft about the pitch (x) axis while limit cycling the LEM about the roll and yaw (y and z) axes. The maneuvers performed about the pitch axis can best be described with reference to Figure 12a and consist of the following:

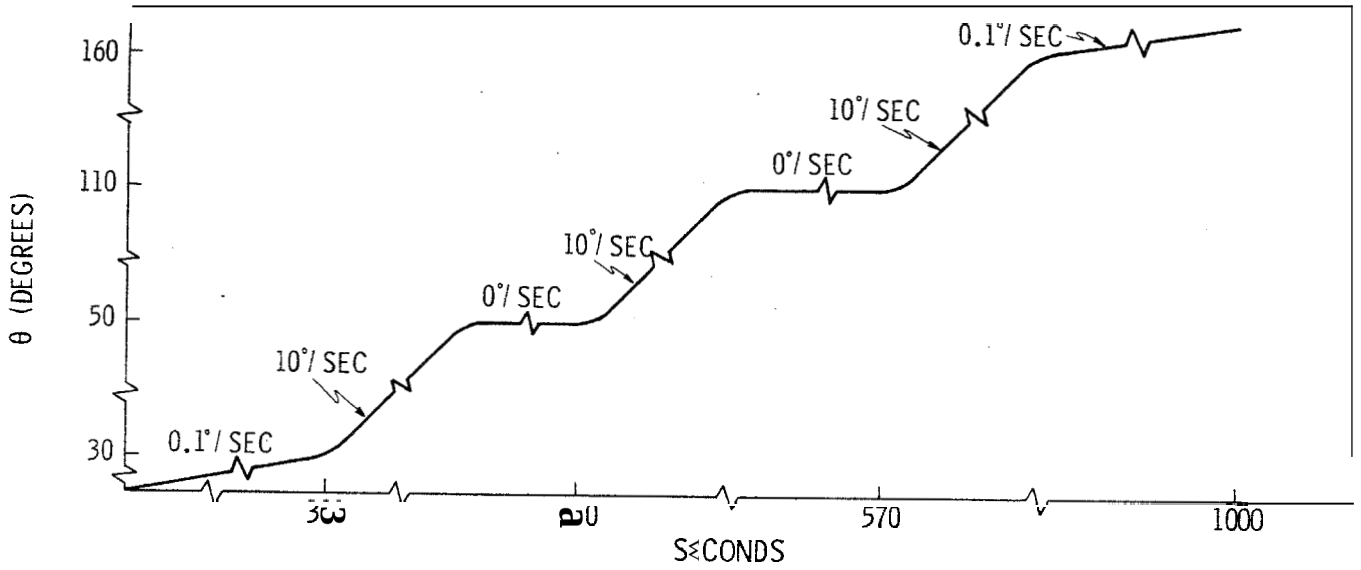
1. At time $t = 0$, the LEM leaves the orbiting platform with an inertial pitch rate of 0.1°/second, allowing the LEM to retain local orientation with the moon.

2. At $t = 300$ seconds, the LEM pitches 20" in preparation for approaching the moon; this attitude is held for 150 seconds while the LEM descends towards the surface. The pitch of 20" is made at the maximum allowable rotation of 10°/second and maximum accelerations of 3/4 radian/second^2.

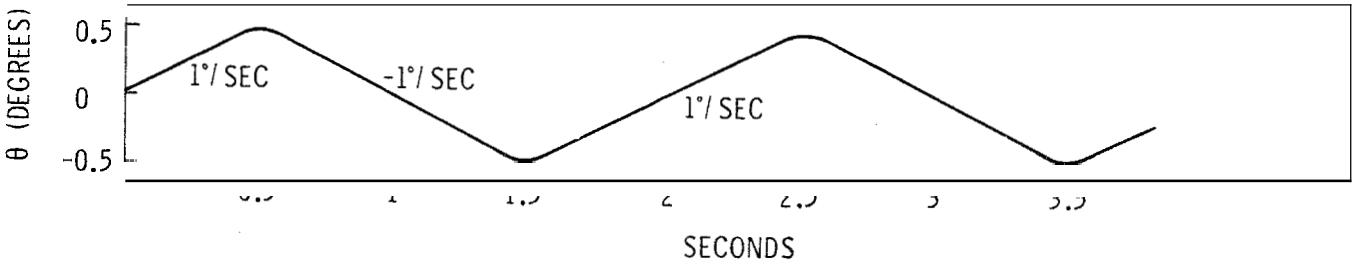
3. At 450 seconds, the LEM pitches 60°, at which point it is oriented with the local vertical of the moon.

4. For 120 seconds, the LEM hovers in a vertical attitude over the landing spot.

5. Failing to find an acceptable landing point, the LEM aborts a lunar touchdown, pitches 60° and lifts off to a rendezvous with the orbiting platform; during this period of the mission, a pitch rate of 0.1" is again employed to maintain local orientation.



(a) Pitch Maneuvers for Simulated LEM Profile



(b) Roll & Yaw Limit Cycle Maneuvers for Simulated LEM Profile

Fig. 12. Characteristics of LEM Profile

6. The mission terminates at t = 1000 seconds at which time rendezvous with the orbiting command module occurs.

7. Throughout the mission, limit cycling as shown in Figure 13b is occurring about the roll and yaw axes of the LEM.

As a result of knowledge gained from the earlier simulations, maintenance of the LEM attitude for LEM profile was simulated using the basic algorithm both with and without interrupted sampling time intervals. For each of these methods of updating the N matrix, the following combinations of computational precision and gyroscope performance were simulated:

1. full precision, ideal gyros
2. full precision, gyro readout quantized at 1/4 milliradian
3. 1 1/2 precision, ideal gyros
4. 1 1/2 precision, gyro readout quantized at 1/4 milliradian

It should be mentioned that the above combinations were also simulated for the profiles described earlier in this report and the results described below were observed for all profiles.

For the four simulations using the basic update formula, the error functions corresponding to the 9 elements of the error matrix are shown in Figure 13. Only very general and almost insignificant statements can be made about these error functions. The first such statement is that the errors resulting from computing in 1 1/2 precision are of the same magnitude as those resulting from the utilization of quantized gyros. One interesting result is that the errors resulting from the combination of computing 1 1/2 precision and of using quantized gyros are not significantly greater than the errors caused by either of these two factors separately. Furthermore, at least for the simulated LEM mission, the basic algorithm utilizing full computational precision and employing ideal gyros yields error functions which are of the same order of magnitude as the errors resulting from the utilization of 1 1/2 precision and quantized gyros.

Figure 14 shows the 9 error functions resulting from the simulation of the LEM mission using the third order N matrix with interrupted sampling intervals. It is significant to note that the errors resulting from the utilization of this method with full computational precision and perfect

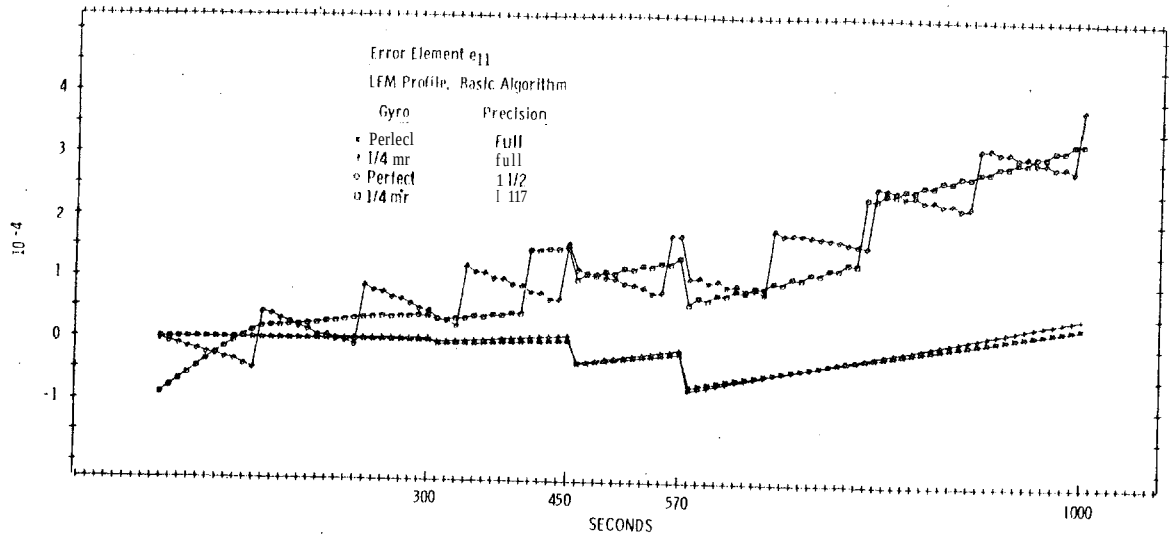


Fig. 13 A

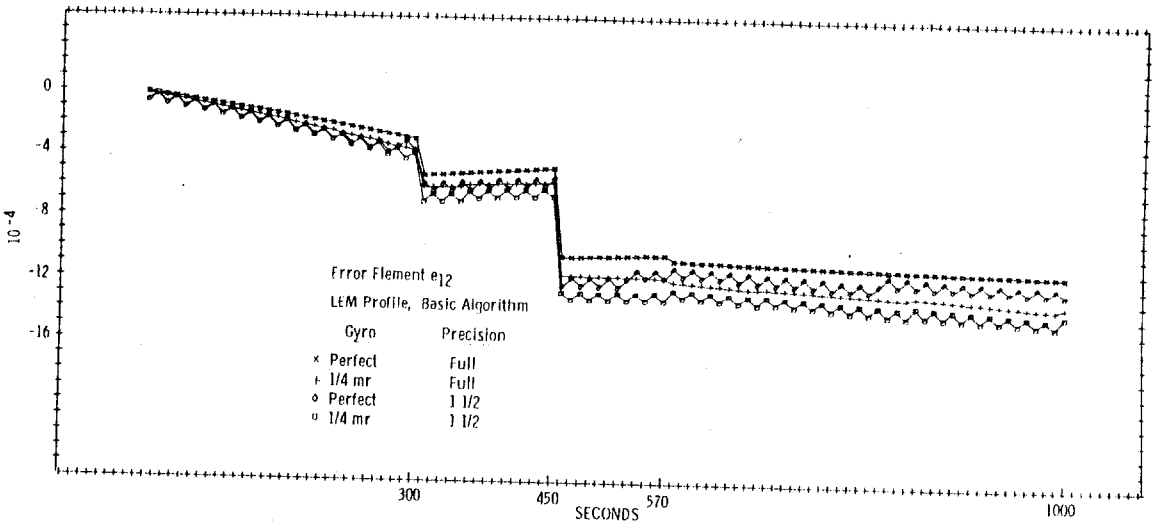


Fig. 13 B

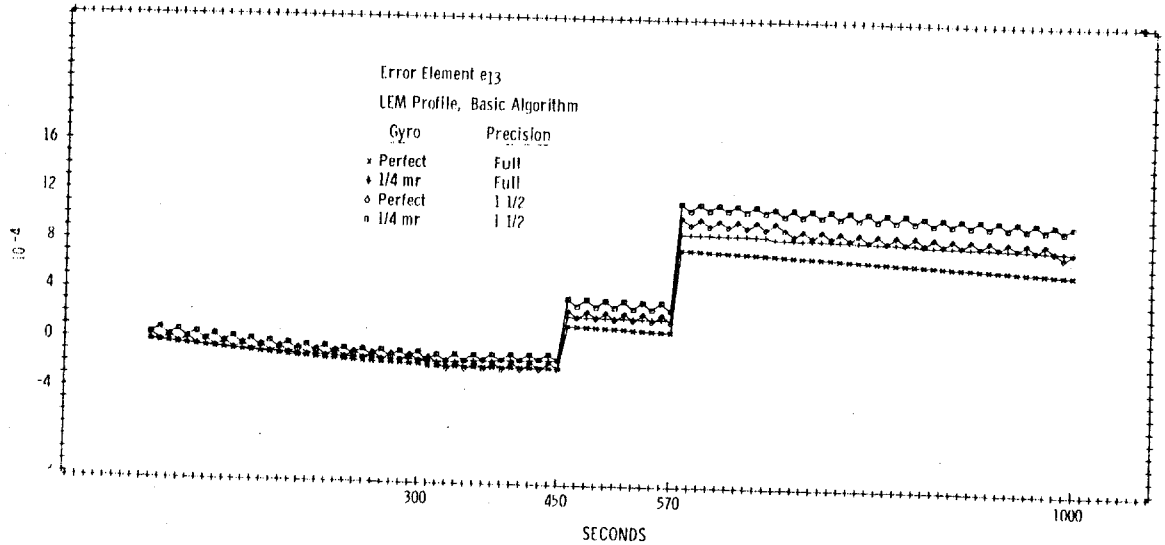


Fig. 13 C

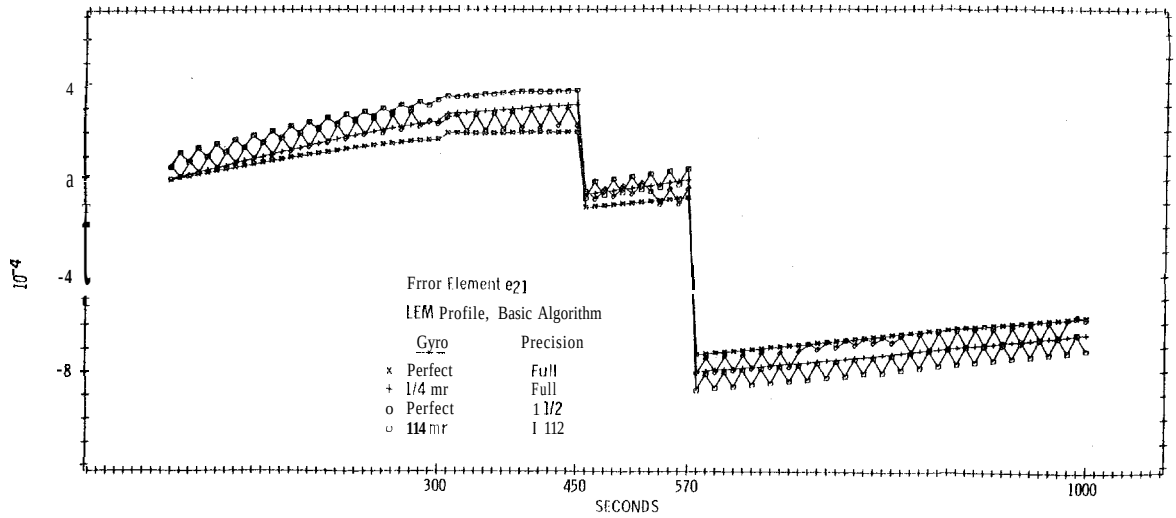


Fig. 13 D

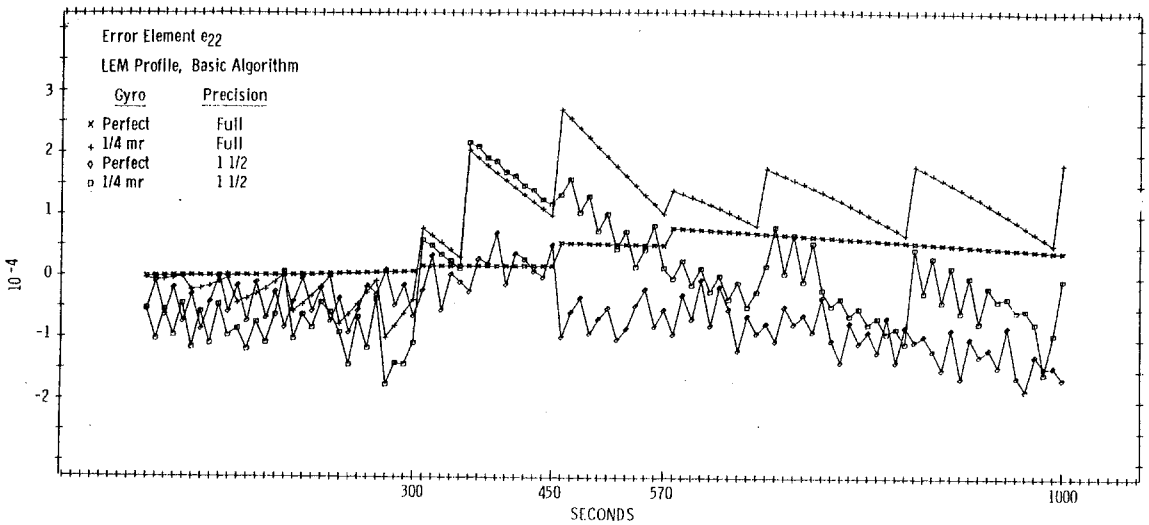


Fig. 13 E

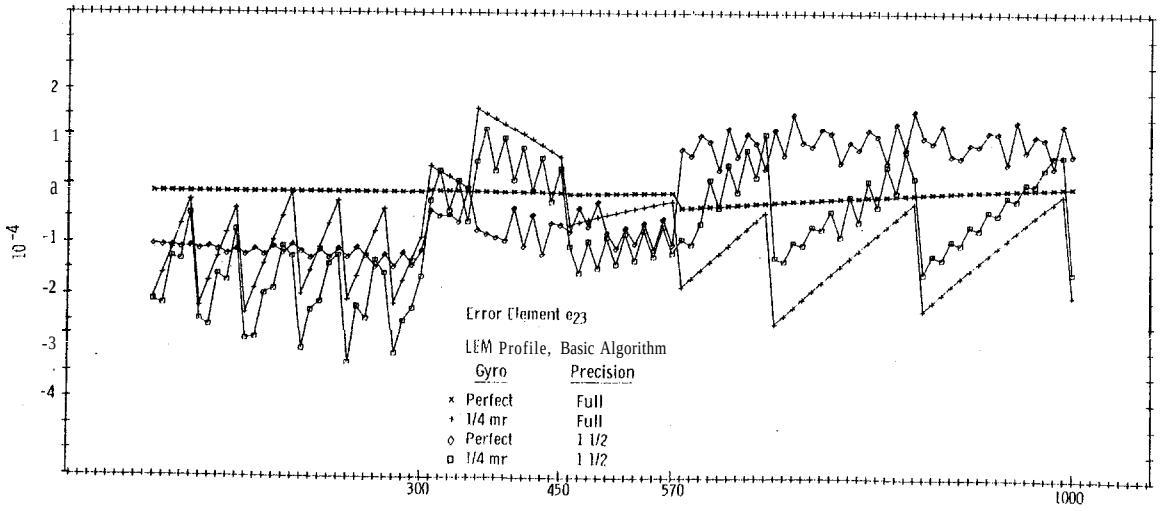


Fig. 13 F

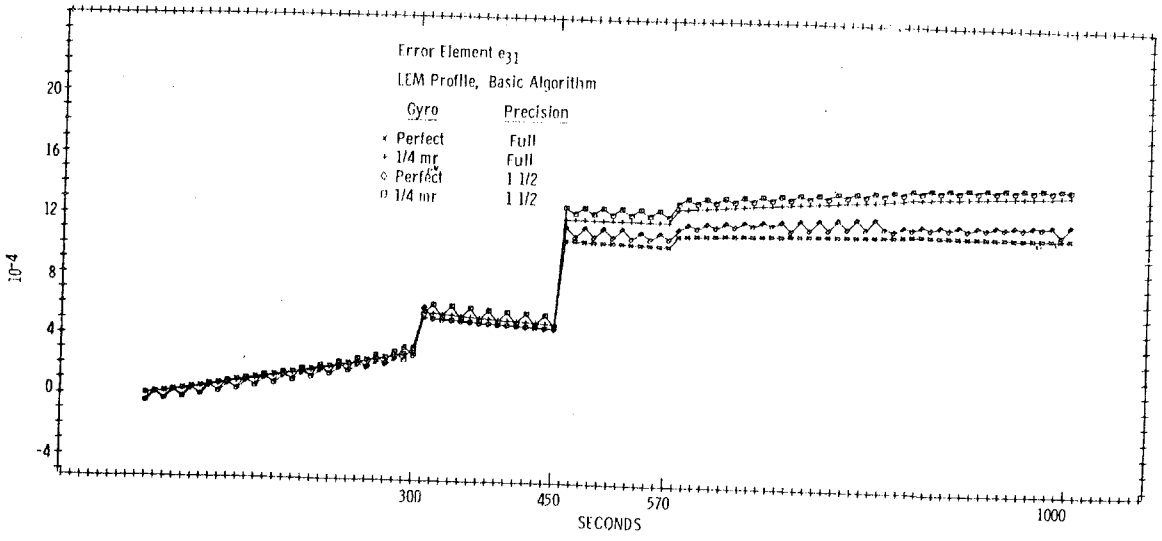


Fig. 13 G

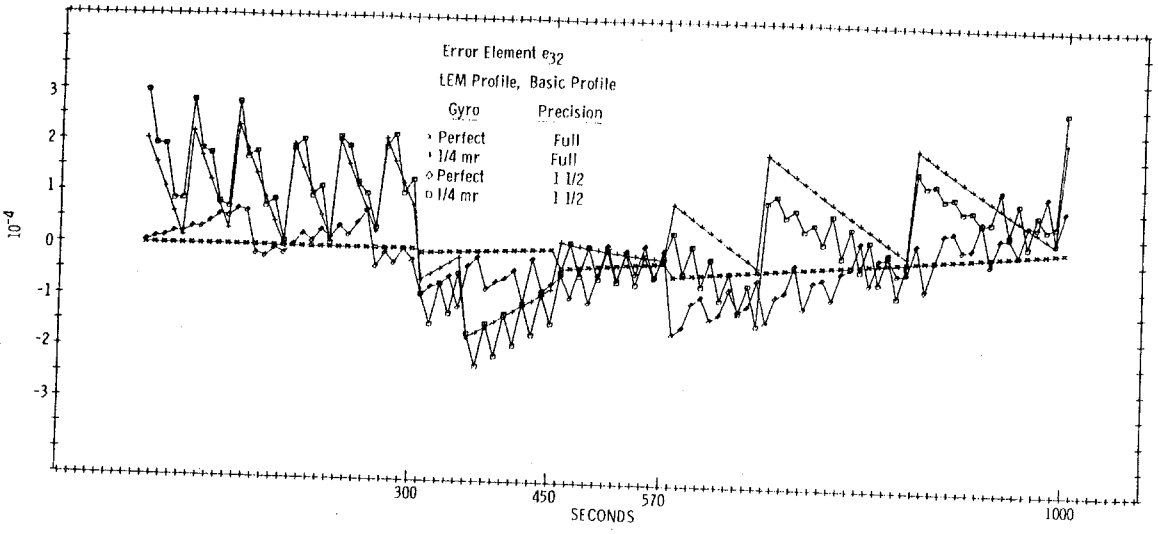


Fig. 13 H

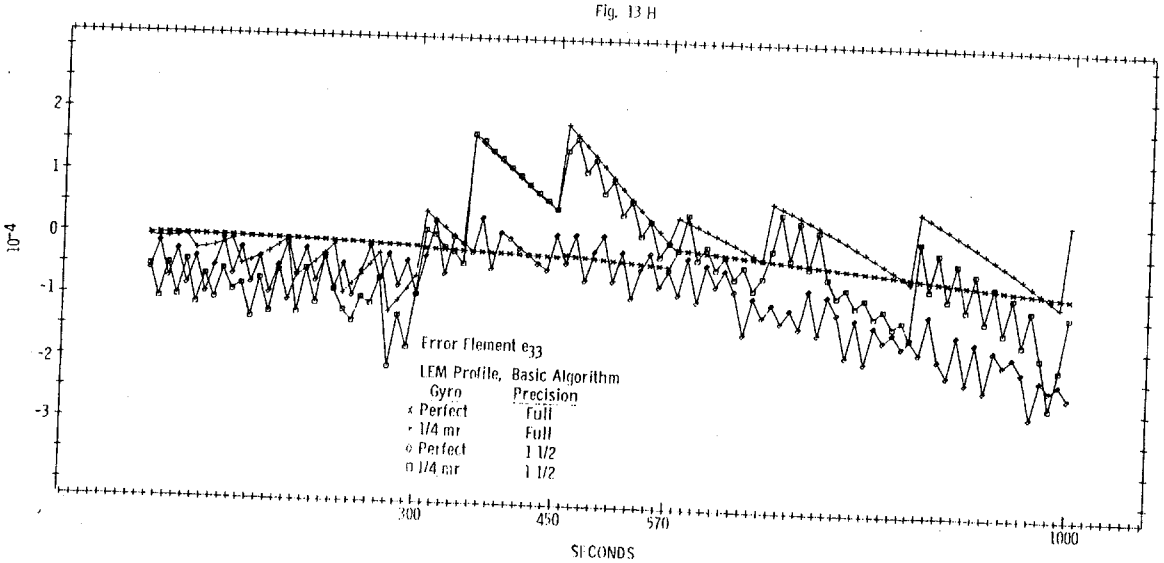


Fig. 13 I

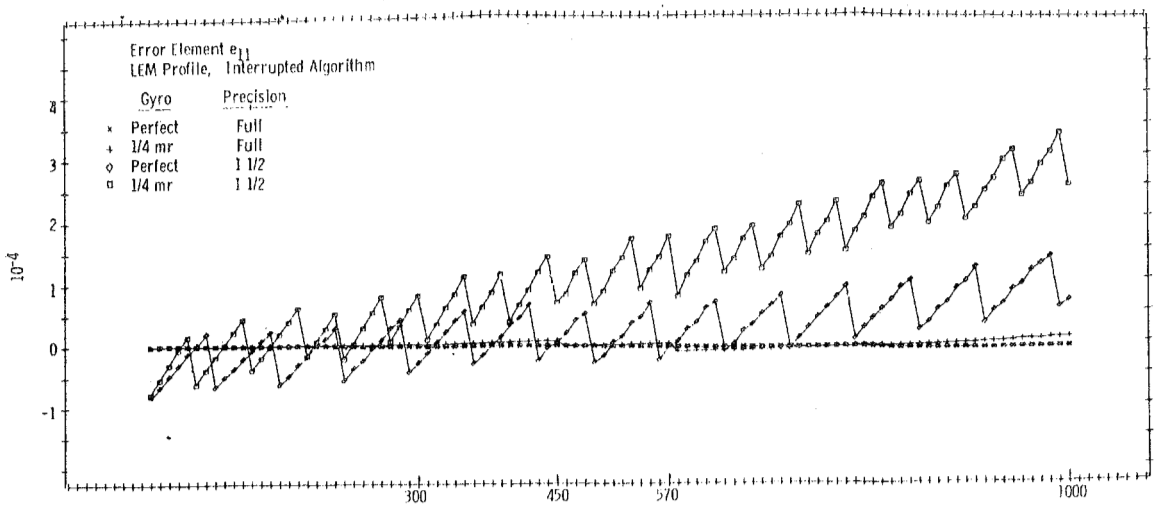


Fig. 14 A

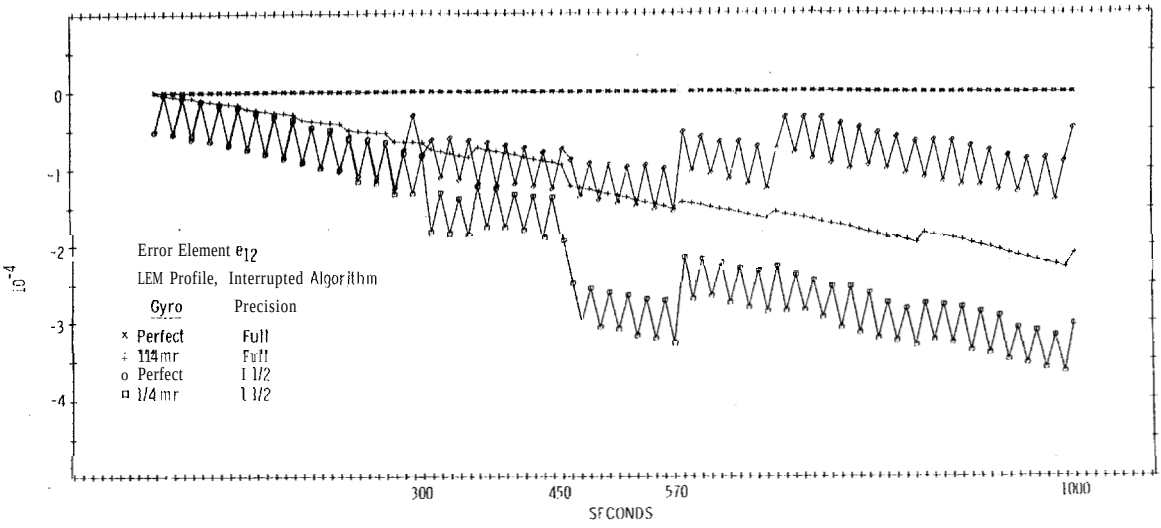


Fig. 14 B

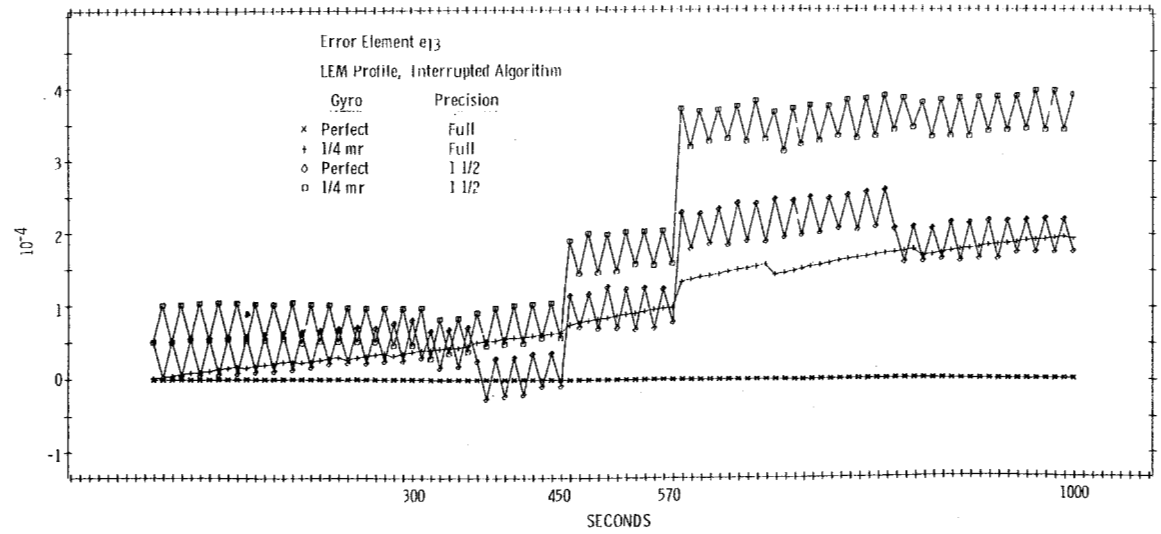


Fig. 14 C

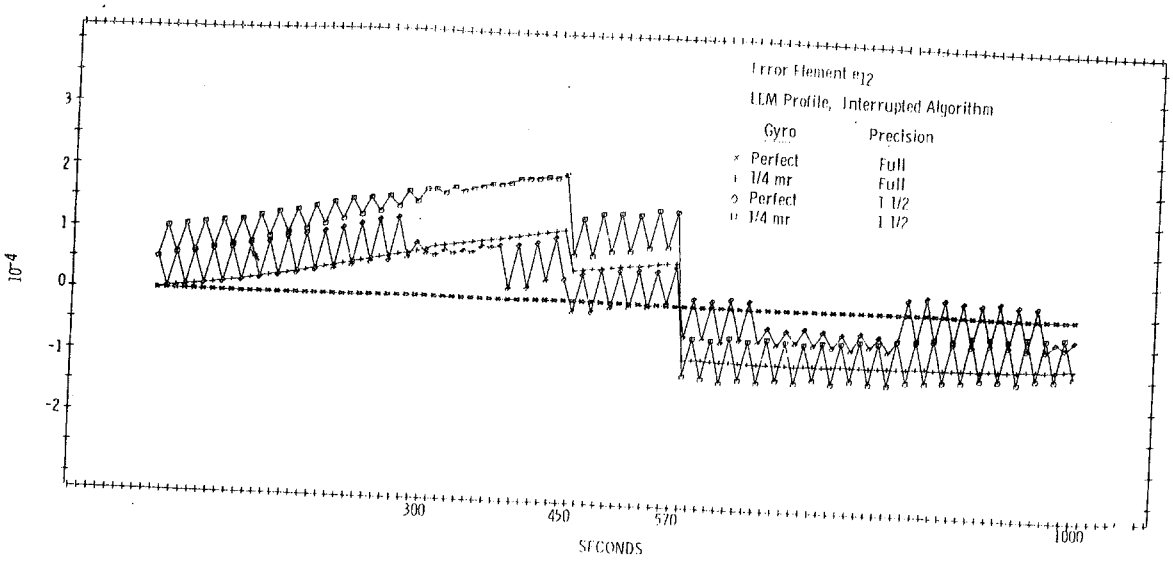


Fig. 14 D

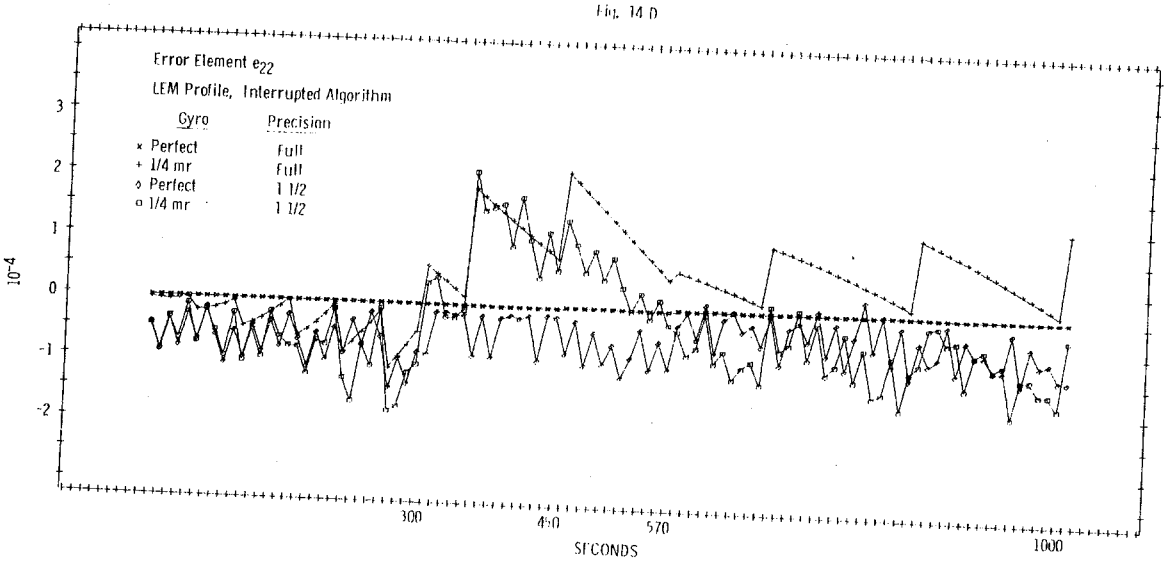


Fig. 14 E

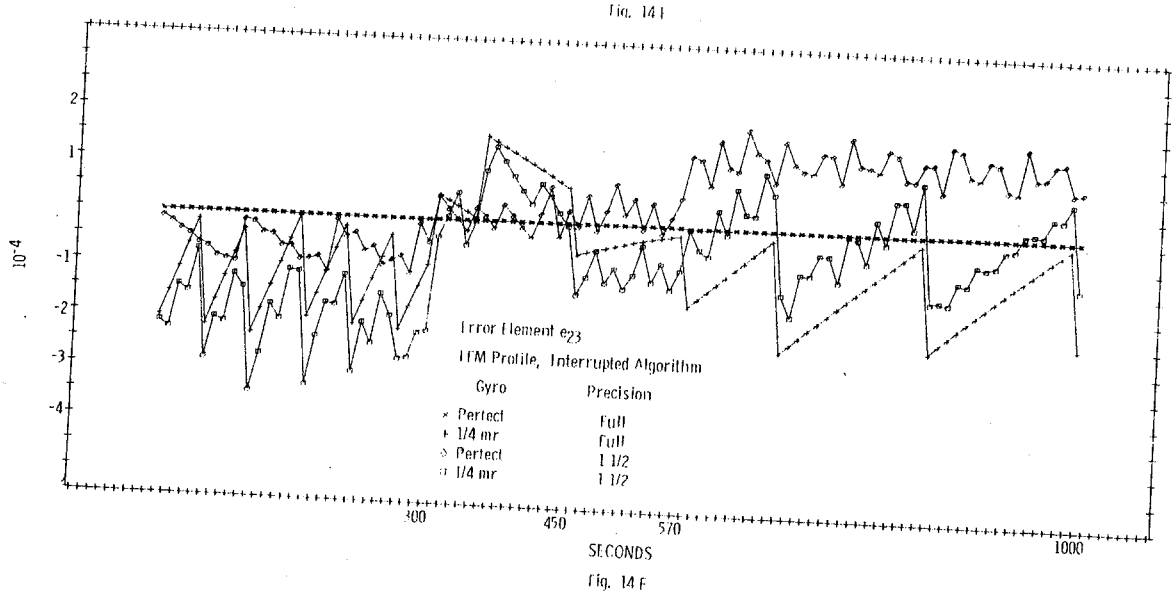
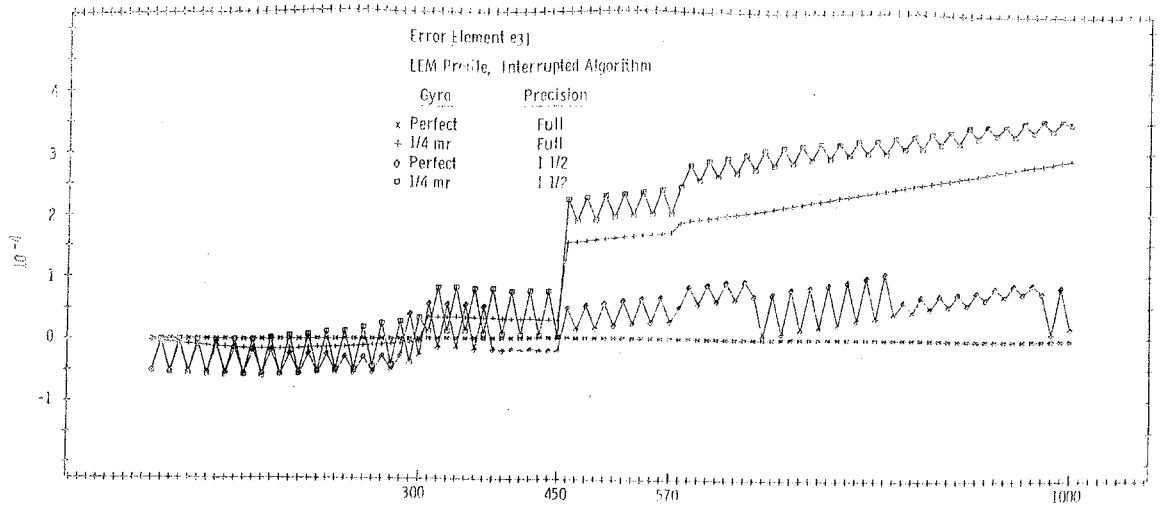
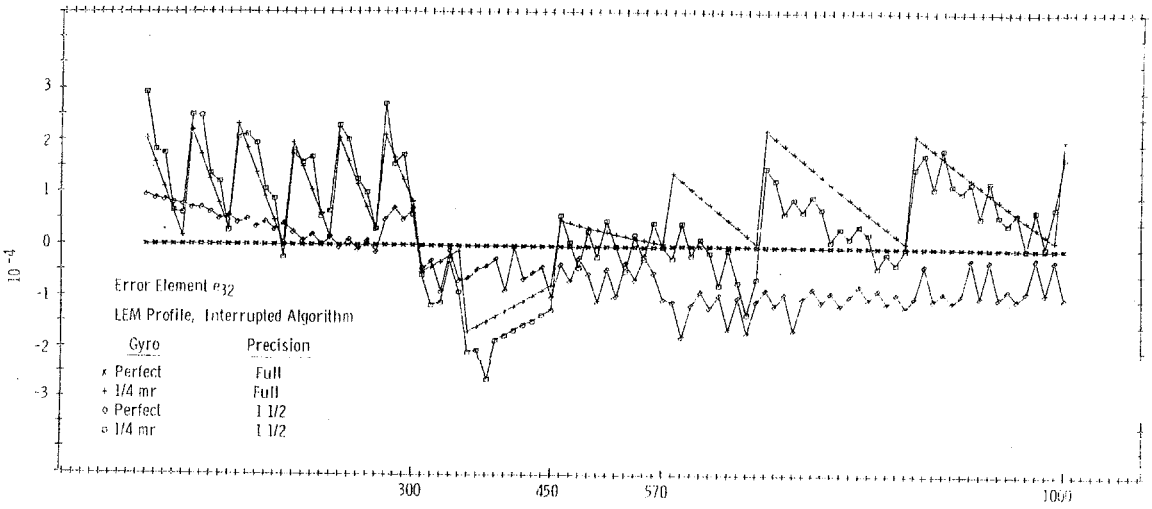


Fig. 14 F



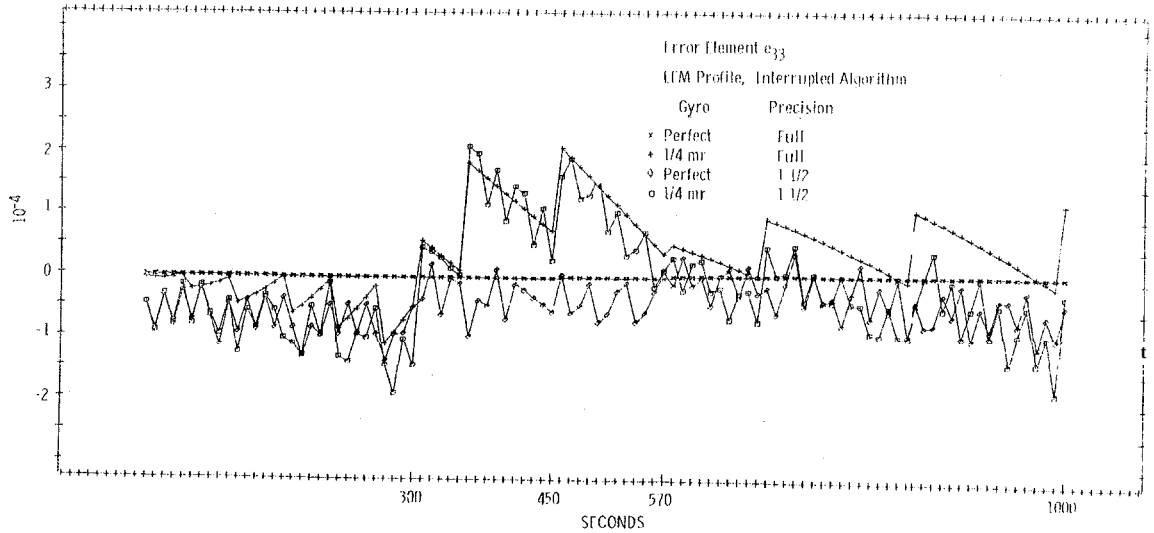
SECONDS

Fig. 14 G



SECONDS

Fig. 14 H



SECONDS

Fig. 14 I

gyros are orders of magnitude less than errors resulting from the other combinations of computational precision and gyro readout. Thus the limiting factor of the accuracy *of* the cosine matrix approximation for this profile is not the update method itself but rather can be attributed to external sources. It is again noted that the effect of 1 1/2 precision is essentially the same as the effect of quantized gyros and that these effected do not appear to be cumulative.

To more readily compare the results of the basic algorithm with those of the interrupted sampling interval update method, certain of the functions of Figures 13 and *14* are superimposed and presented in a common co-ordinate system in Figure 15. Figure 15 substantiates the major conclusion which was evidenced earlier: elimination of discontinuities in the angular accelerations (and the higher derivatives) results in significant improvement in the performance of the basic algorithm.

Figures 13, *14* and 15 reveal the performance of the basic algorithm for a mission which somewhat approximates one possible LEM mission. In order to delineate the relative performance between the whole number algorithm as herein implemented on a general purpose computer and the specialized techniques employed by what is felt to be a better-than-average configuration of a DDA, the performance of the DDA described in Section IV was simulated. The nine elements of the error matrix, the difference between the true transformation matrix and the transformation matrix as updated by the DDA for the LEM profile, are shown in Figure 16. Comparison of Figure 16 with Figures 13, *14* and 15 indicates that in general the elements of the transformation matrix as updated by the DDA are somewhat more accurate than those of the whole number algorithm as simulated.

5.10 Comparative Data for the Basic Profile

The error functions shown in Figures **13, 14,** 15 and **16** present a reasonably complete picture of the capabilities *of* the whole number algorithm for the simulated LEM mission. *To* forestall the possibility of doubt arising about the general validity of the information depicted because of the mild characteristics of the LEM mission profile, a complete set of error functions was obtained for the first 200 seconds of the basic profile. Of all the mission profiles which were simulated, this profile resulted in the largest

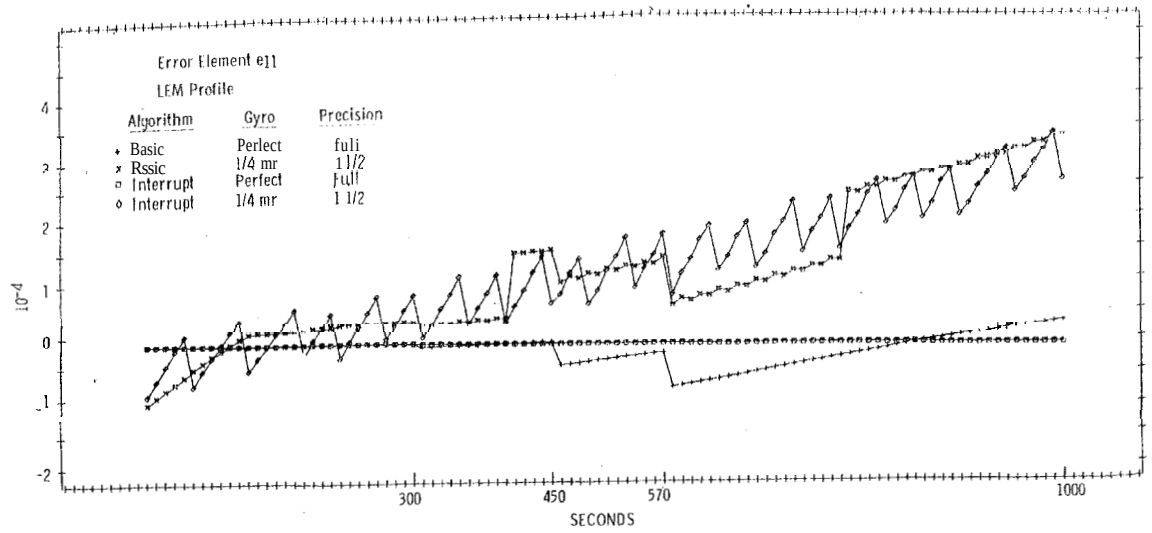


Fig. 15 A

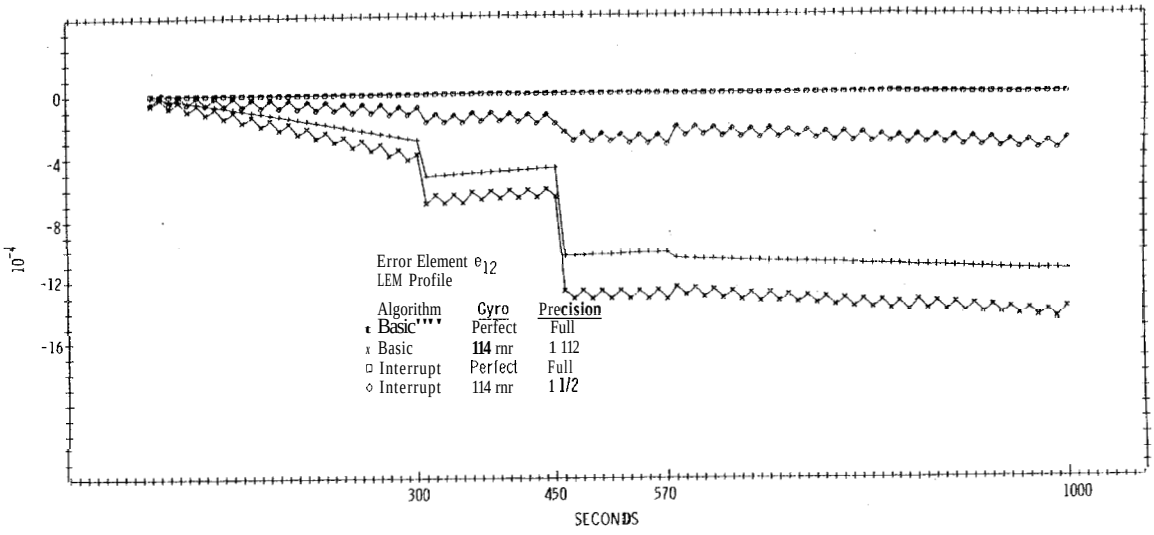


Fig. 15 B

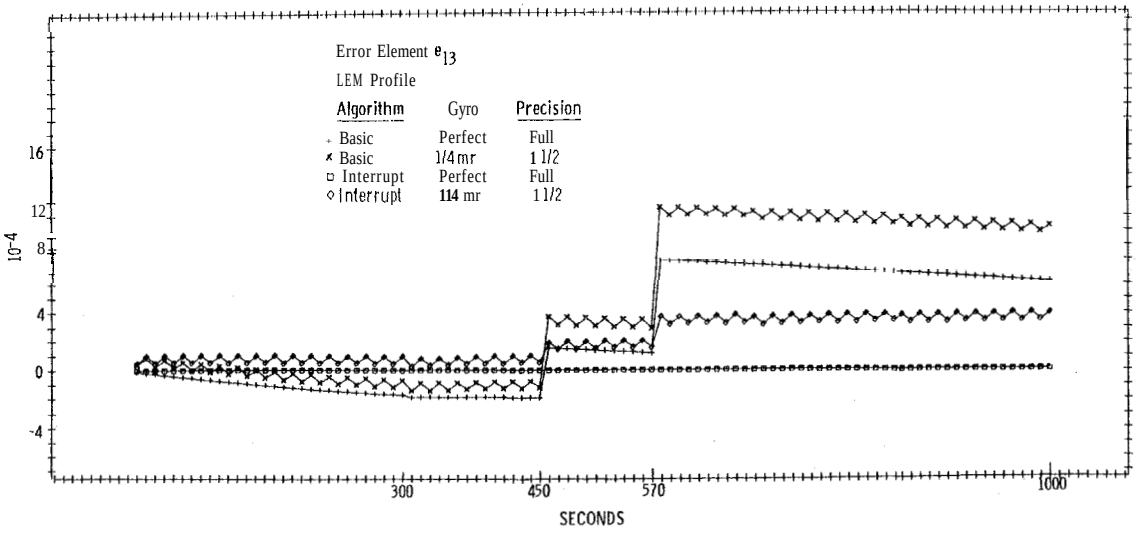


Fig. 15 C

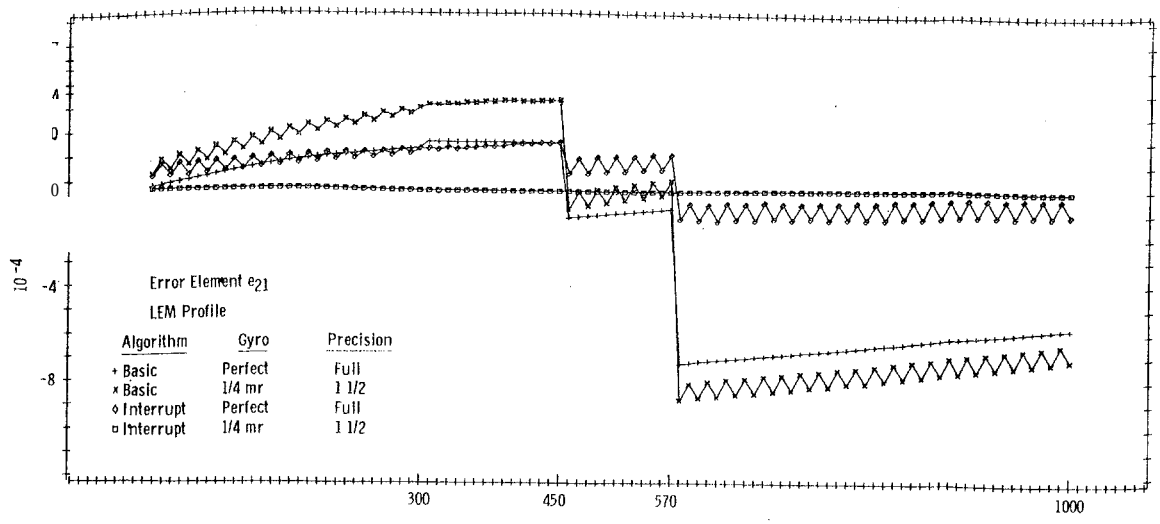


Fig. 15 D

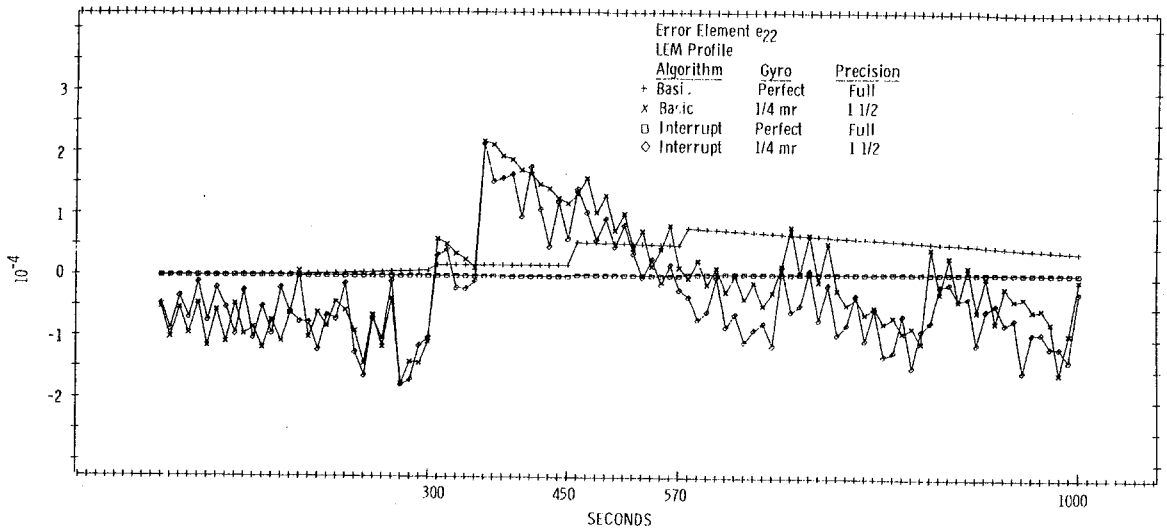


Fig. 15 E

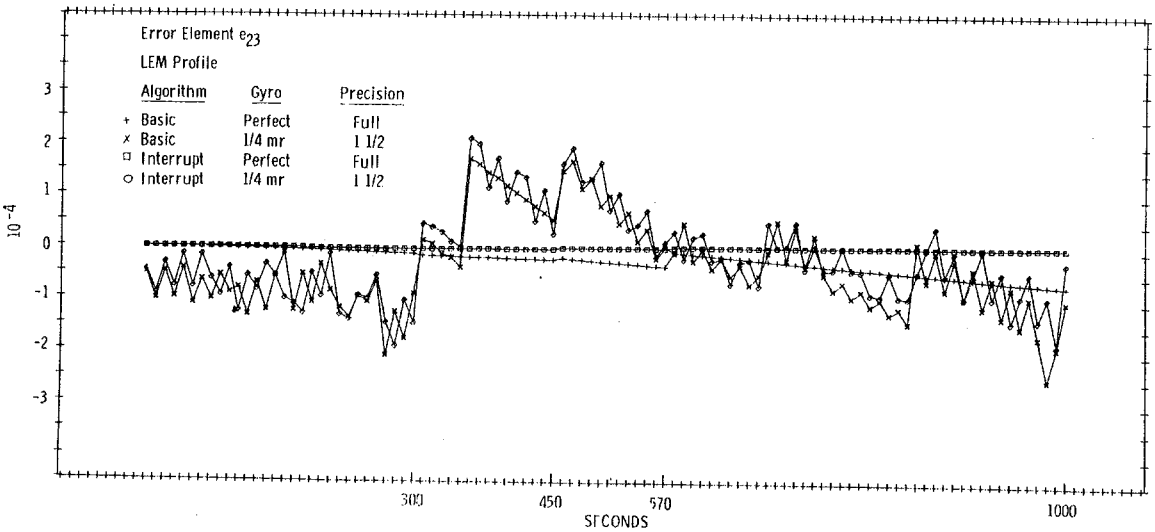


Fig. 15F

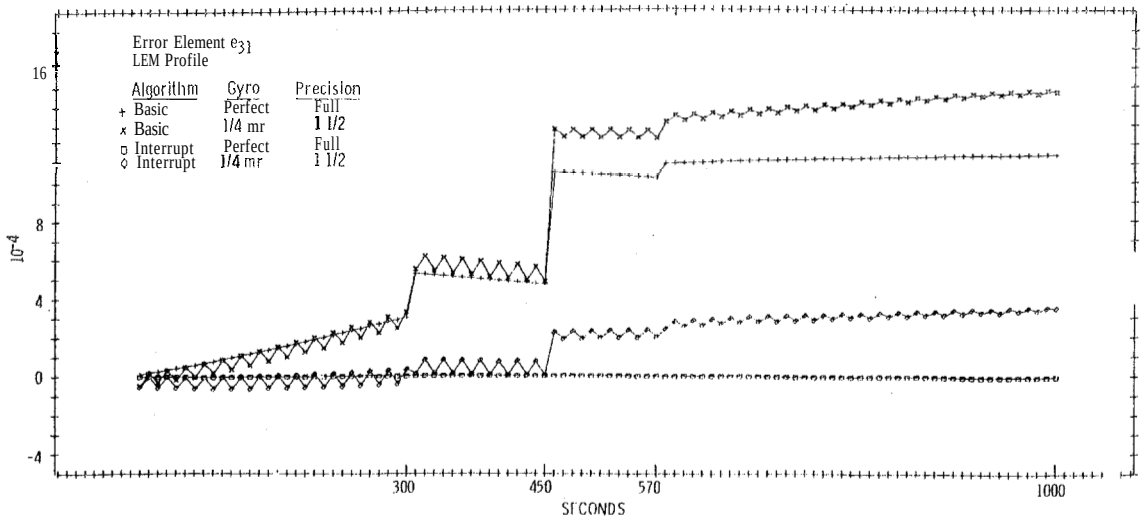


Fig. 15 G

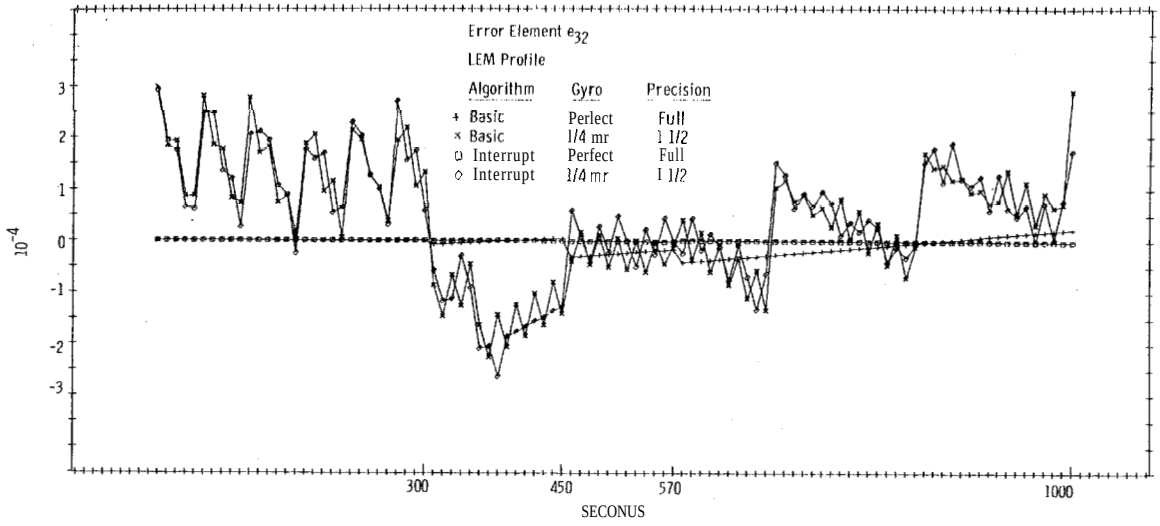


Fig. 15 H

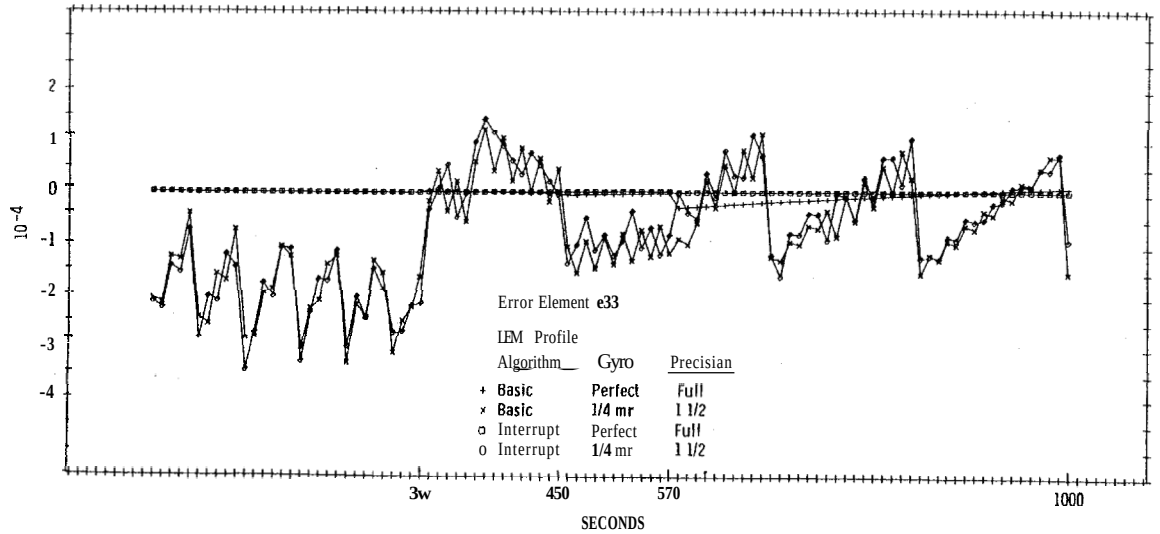


Fig. 15 I

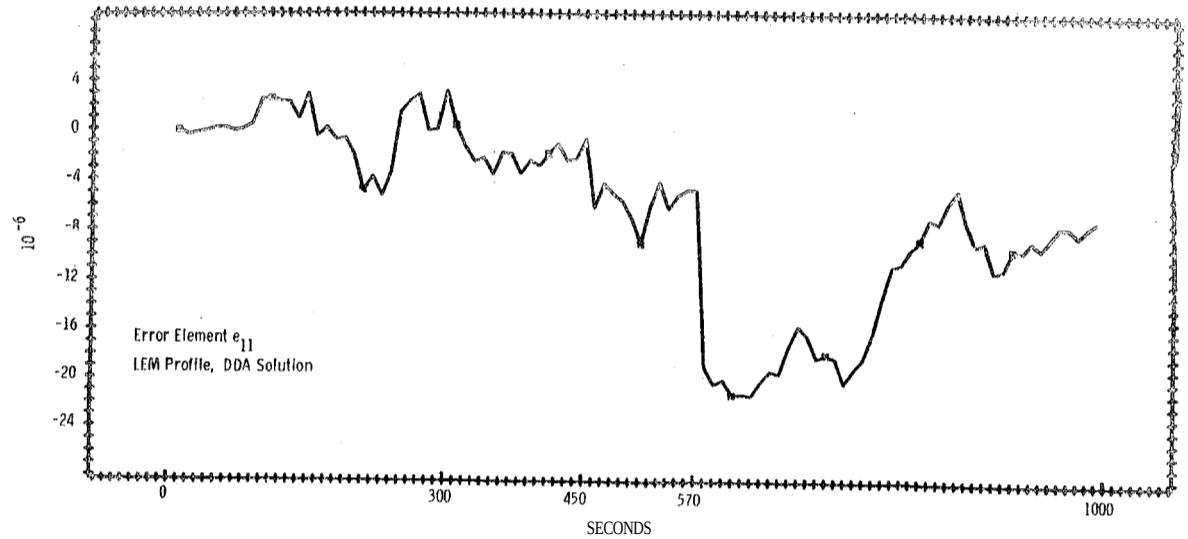


Fig. 16 A

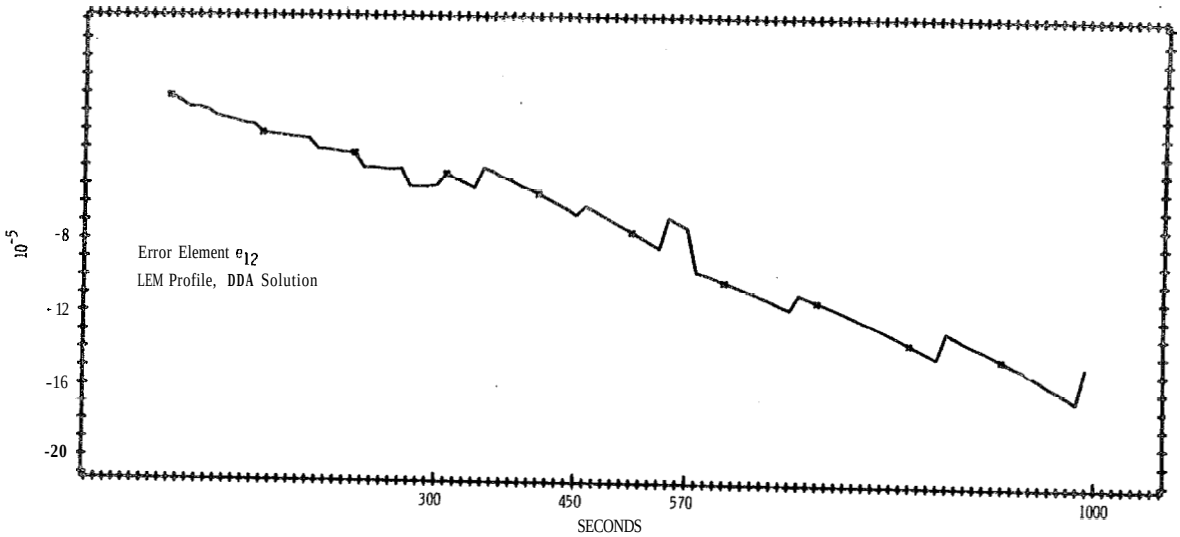


Fig. 16 B

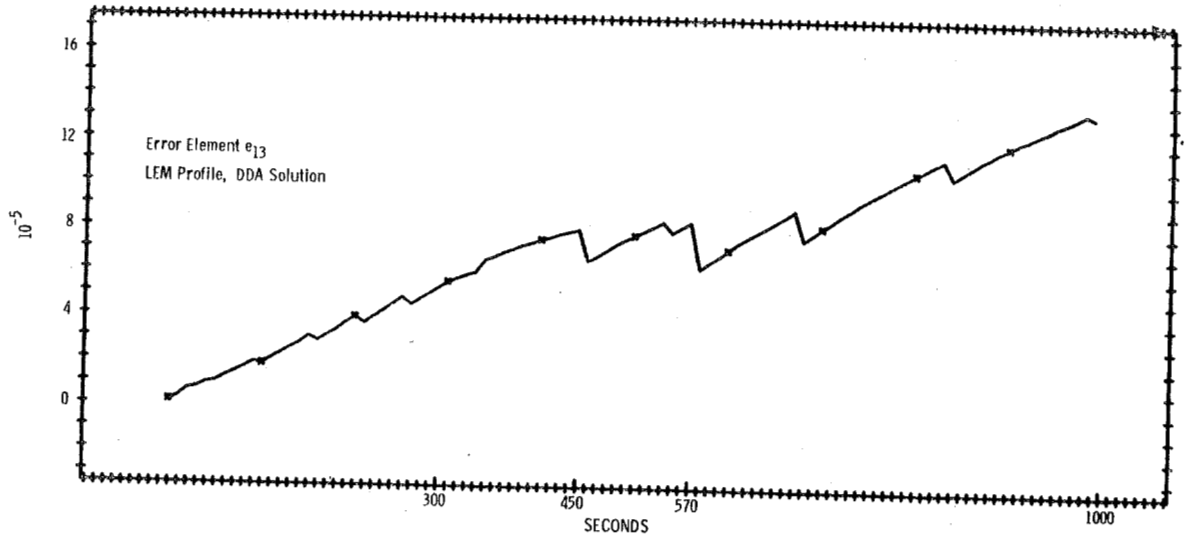


Fig. 16 C

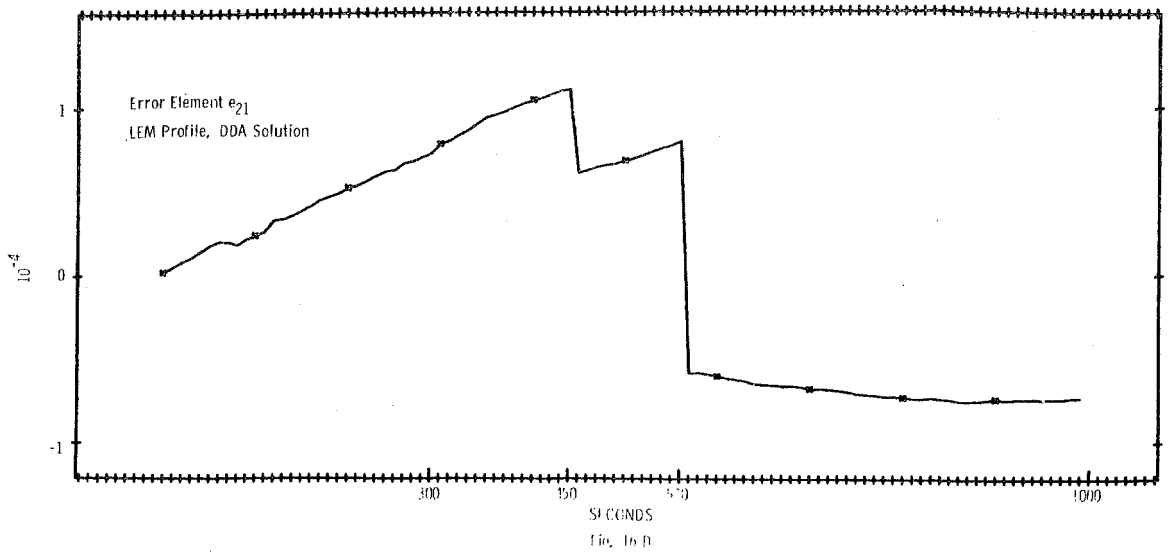


Fig. 16.D

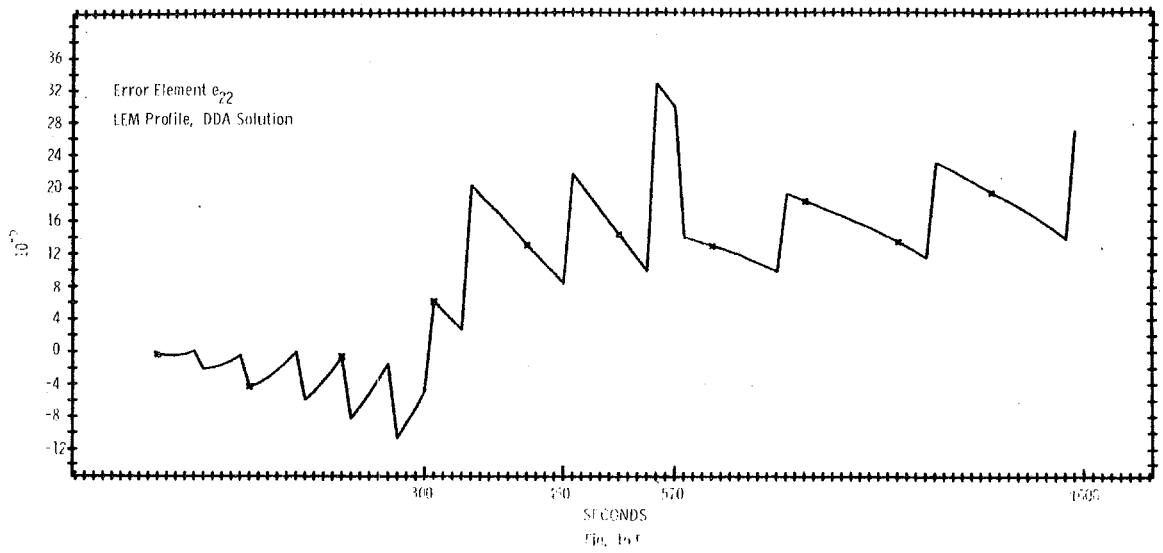


Fig. 16.E

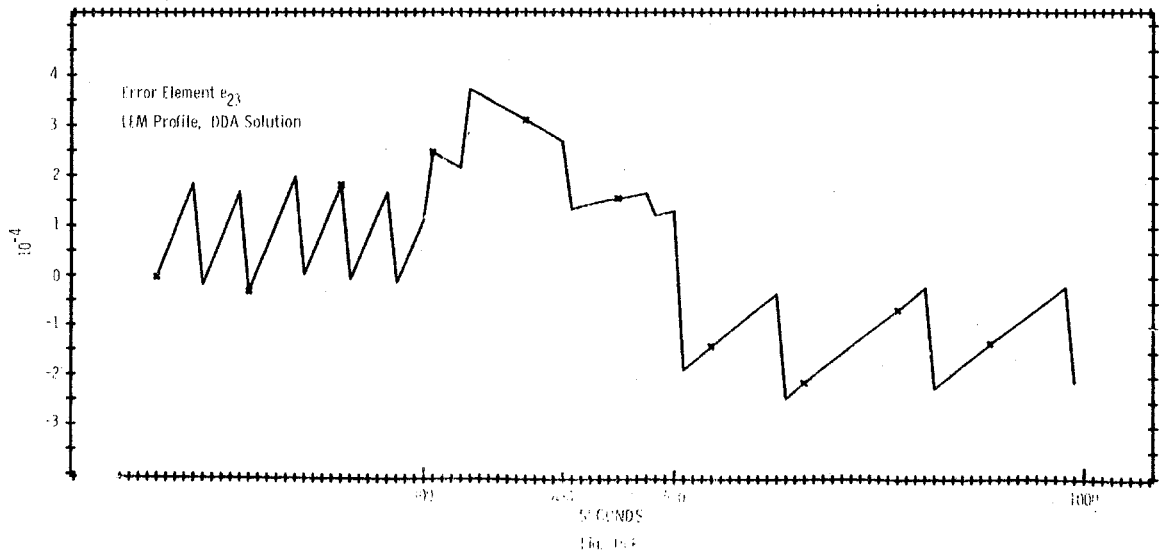


Fig. 16.F

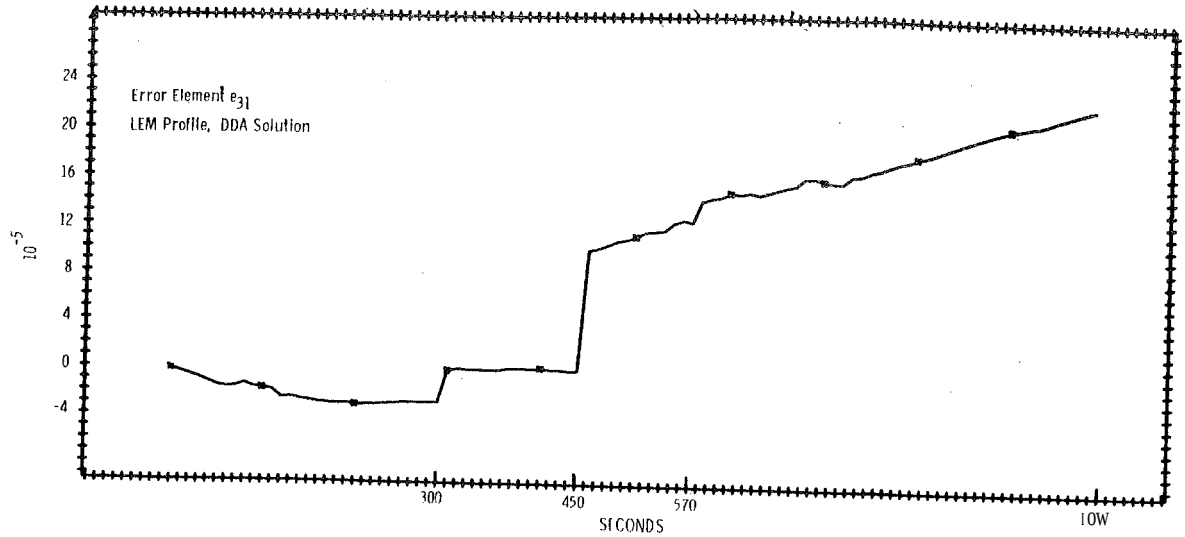


Fig. 16 G

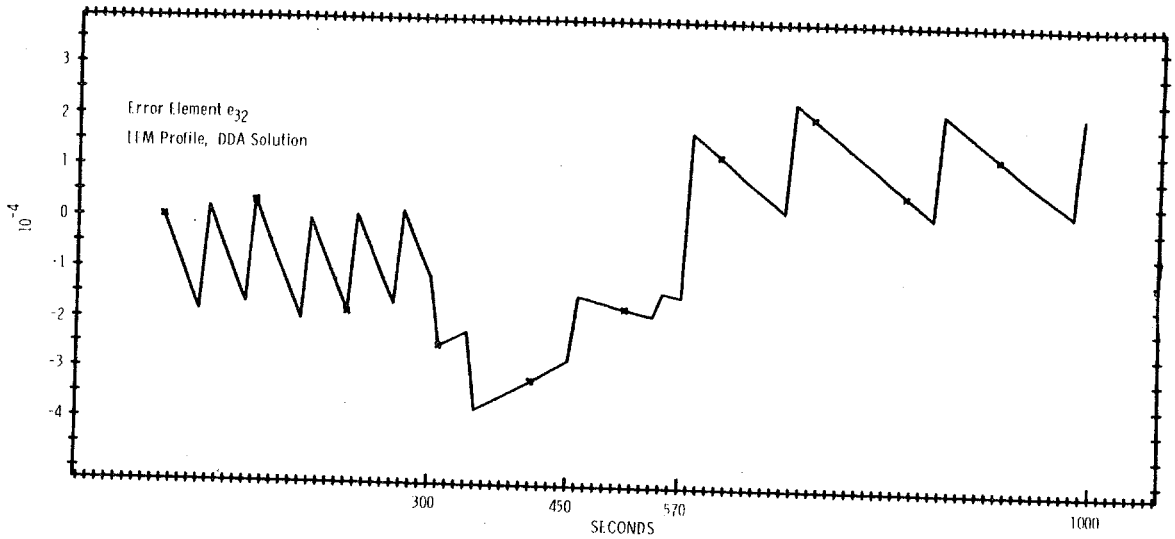


Fig. 16 H

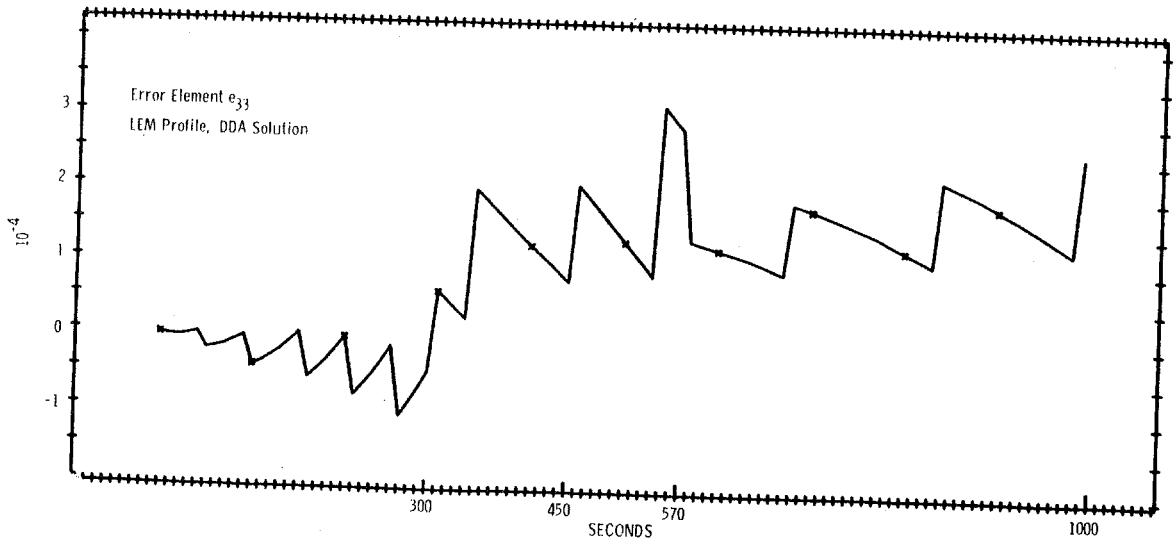


Fig. 16 I

magnitudes of the error functions. However, the maneuvers encountered in the basic profile are far more extreme and violent than one would anticipate for an actual spacecraft mission; therefore the large errors resulting from this profile should not be viewed with excessive alarm.

Figure 17 shows the 9 error functions of the basic profile resulting from three mechanization combinations of the noninterrupted update algorithm. Figure 18 shows the 9 error functions of the basic profile resulting from simulation of the interrupted update procedure.. Figure 19 shows the 9 error functions of the basic profile which result from simulation of the DDA techniques. The combined set of Figures 13 - 19 present a reasonably complete picture of the absolute and relative capabilities of the whole number algorithm derived in Section 111. It is felt that they, in conjunction with the results previously described, provide an objective basis from which designers of data processors associated with strapdown navigation systems are free to draw their own conclusions.

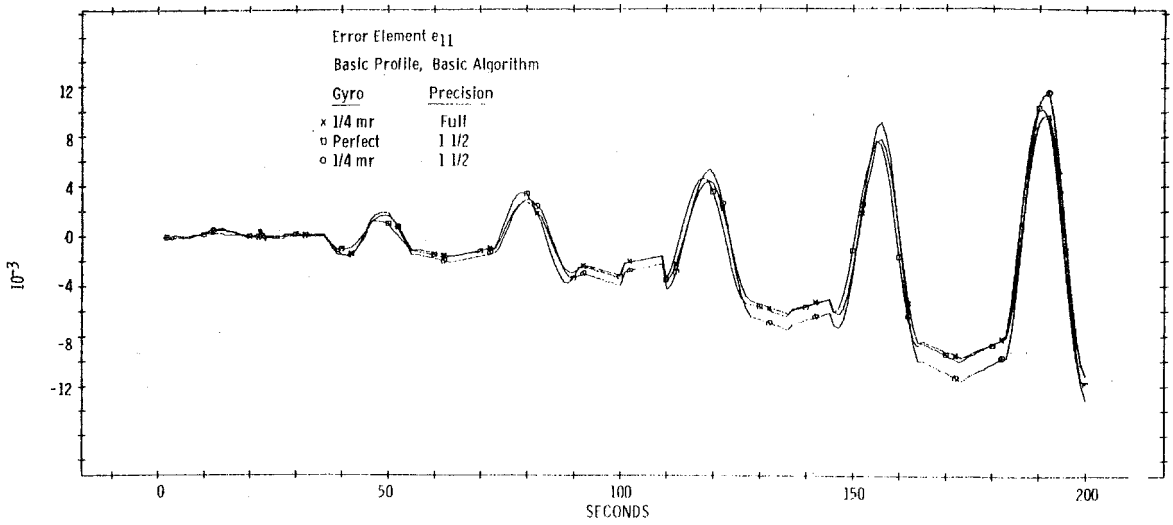


Fig. 17 A

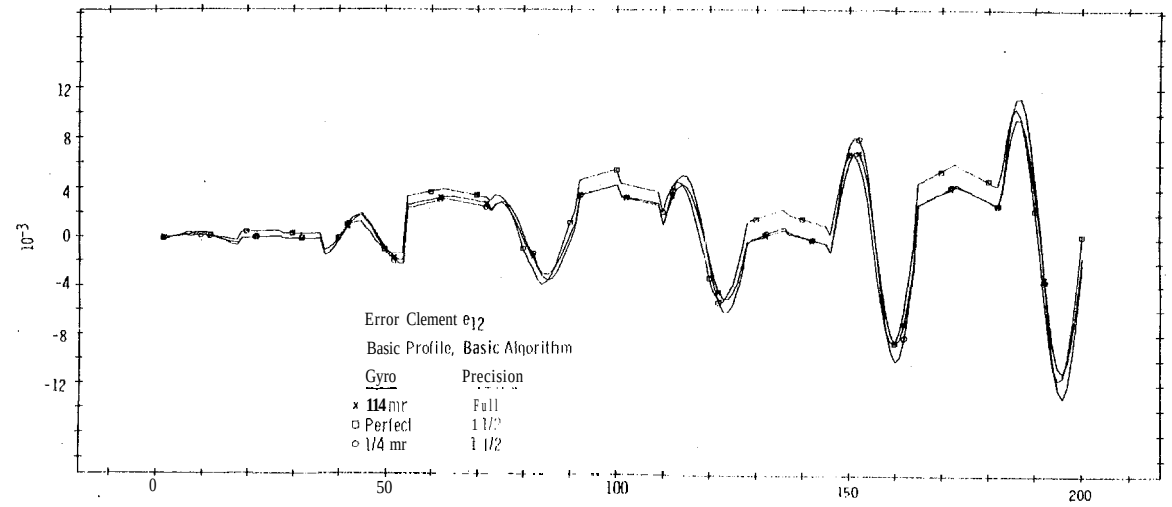


Fig. 17 B

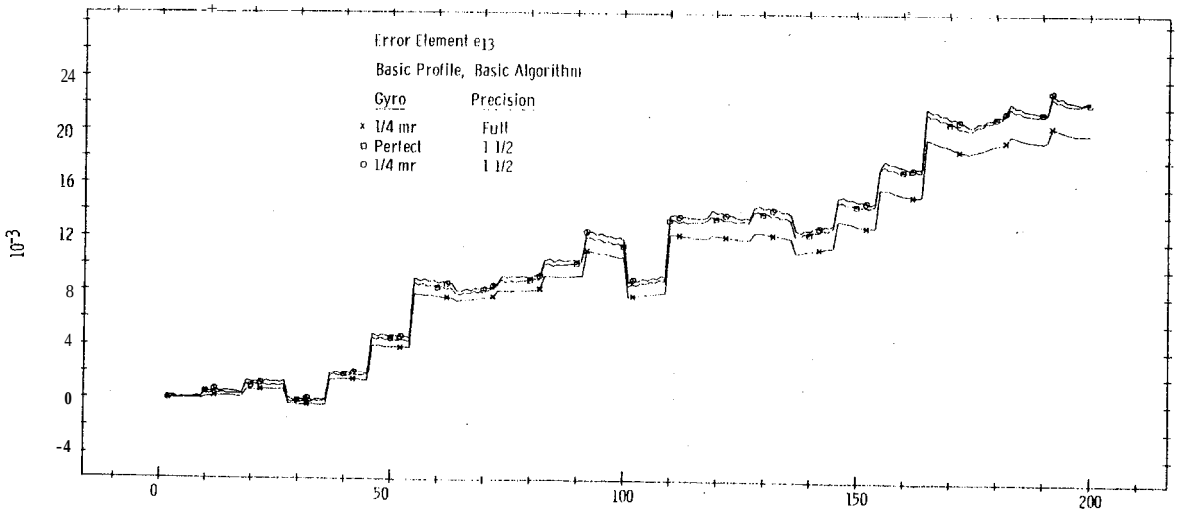


Fig. 17 C

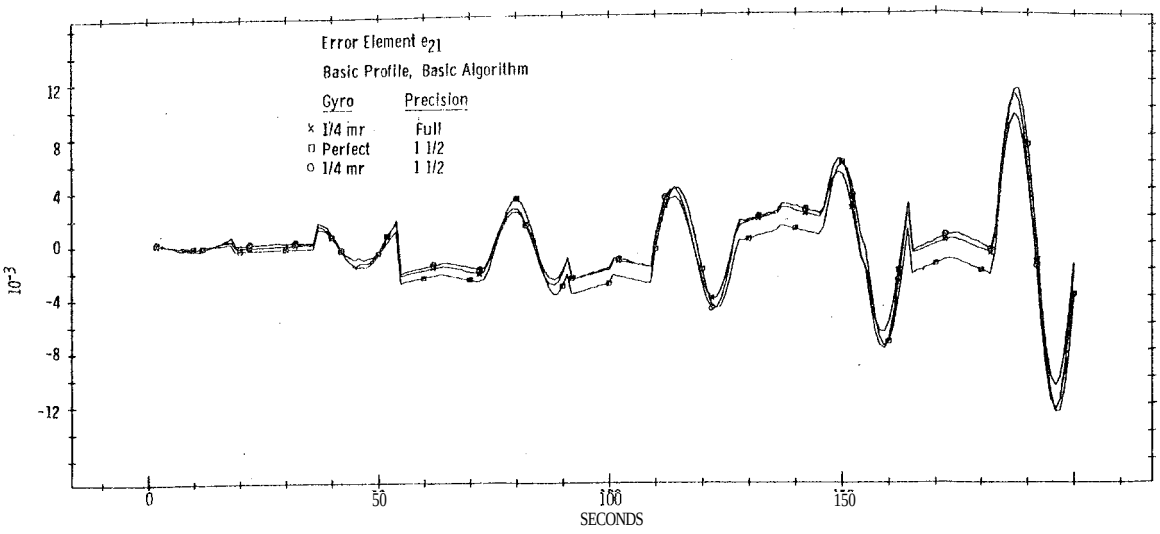
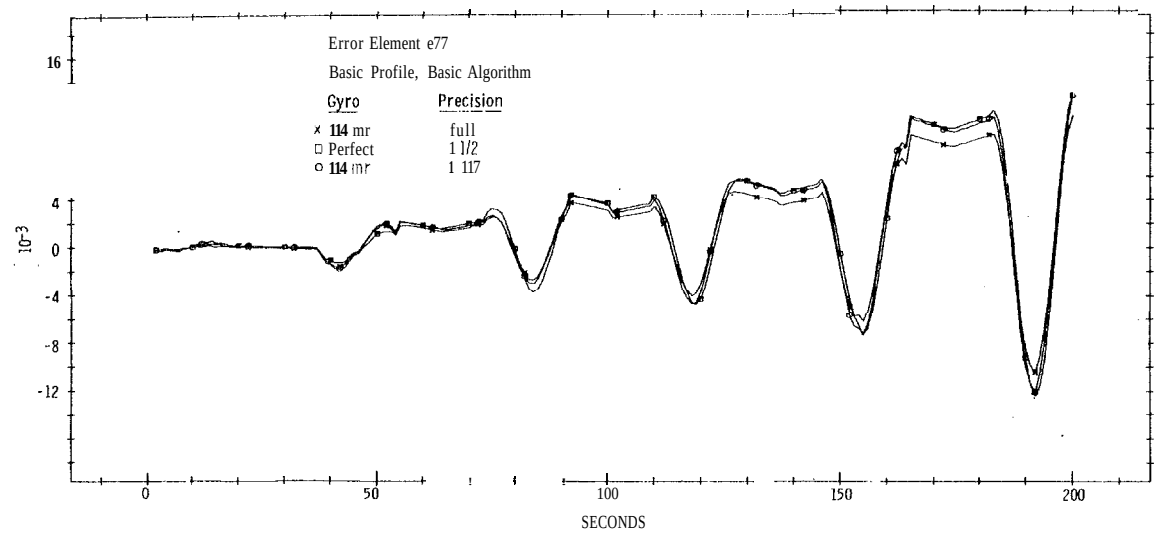


Fig. 17 D



Fir). 17 F

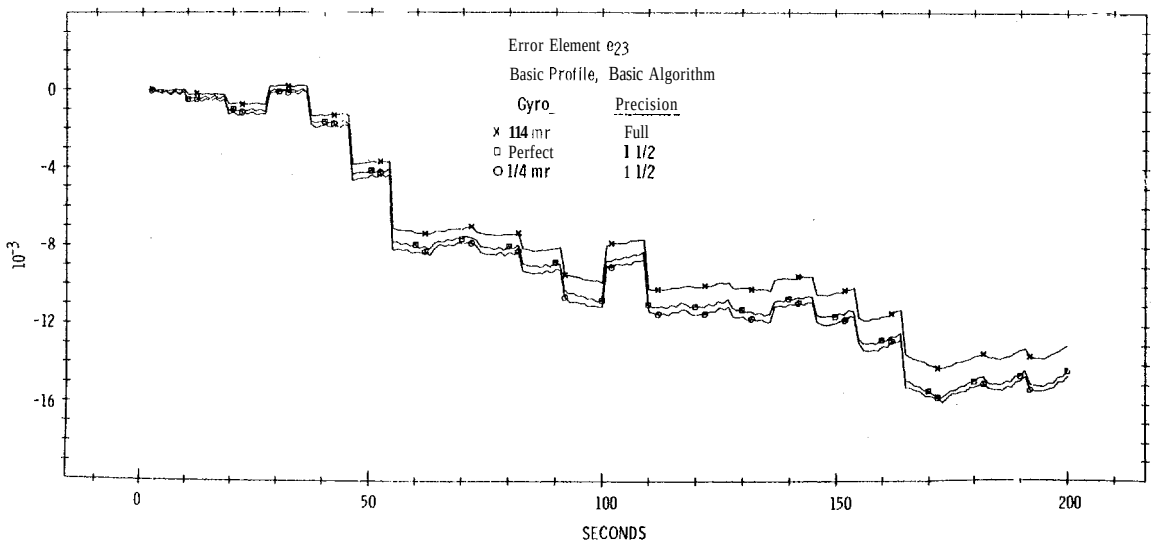


Fig. 17 F

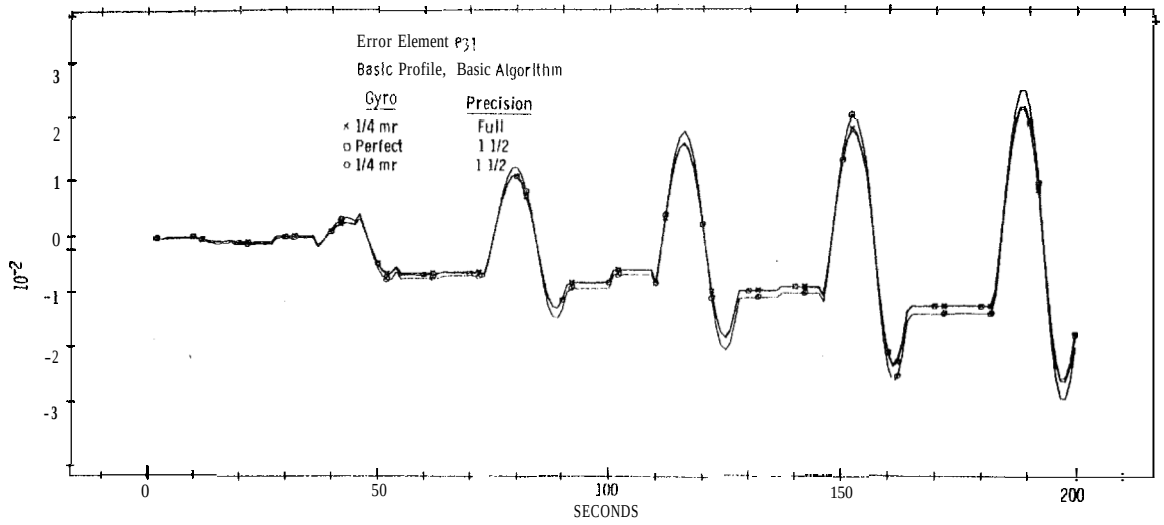


Fig. 17 G

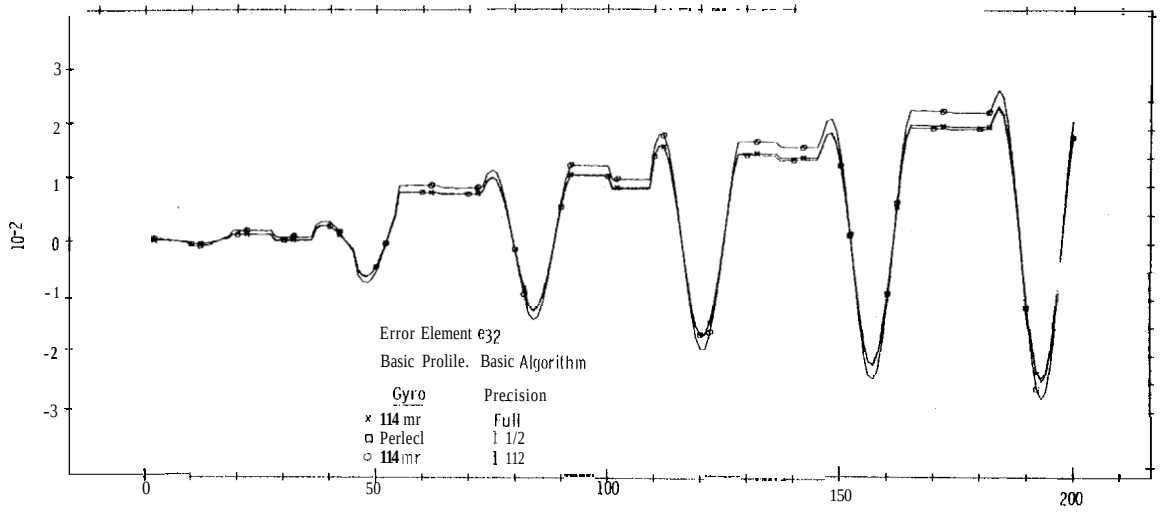


Fig. 17 H

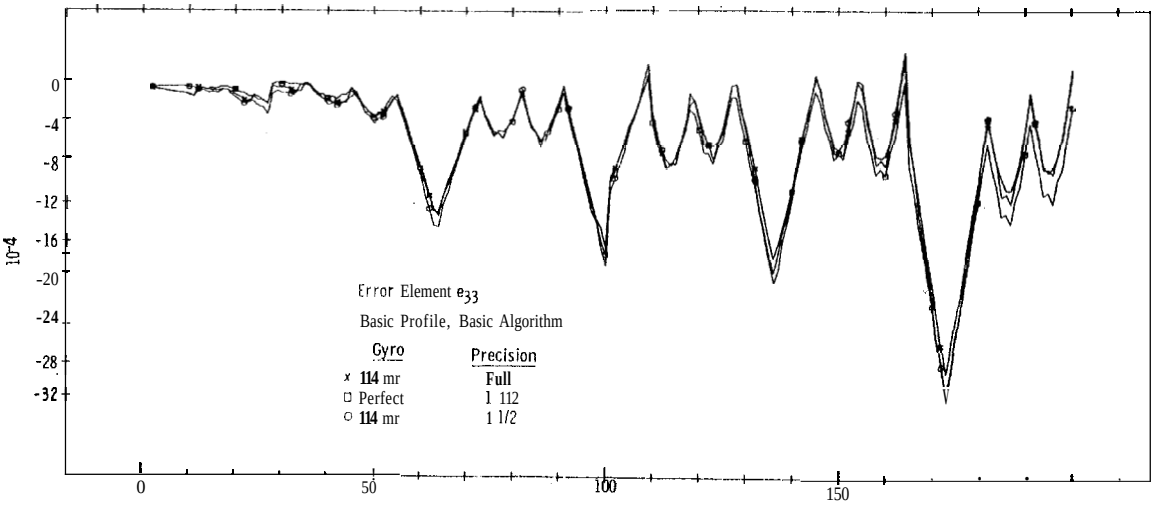


Fig. 17 I

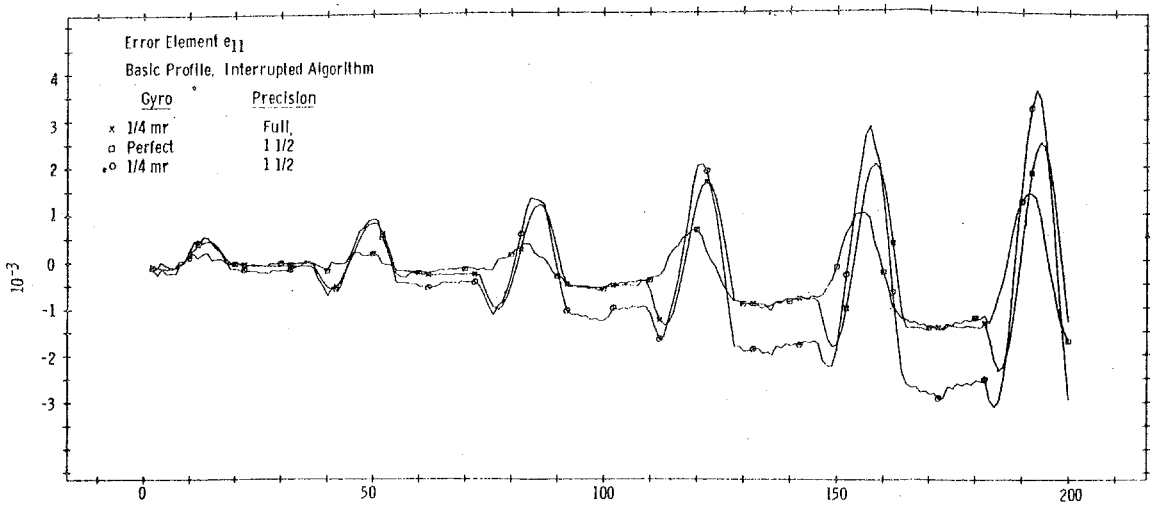


Fig. 18 A

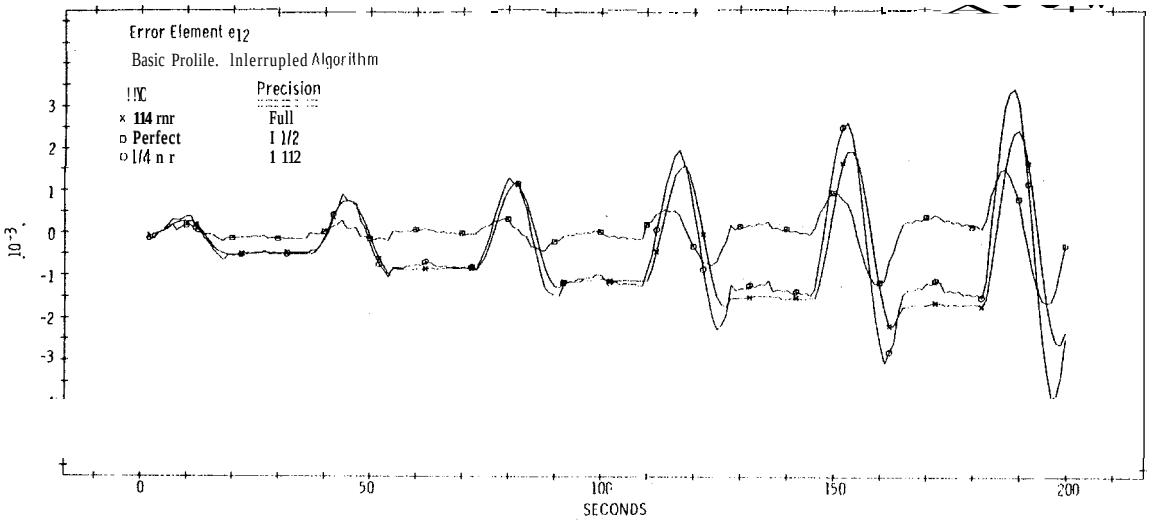


Fig. 18 B

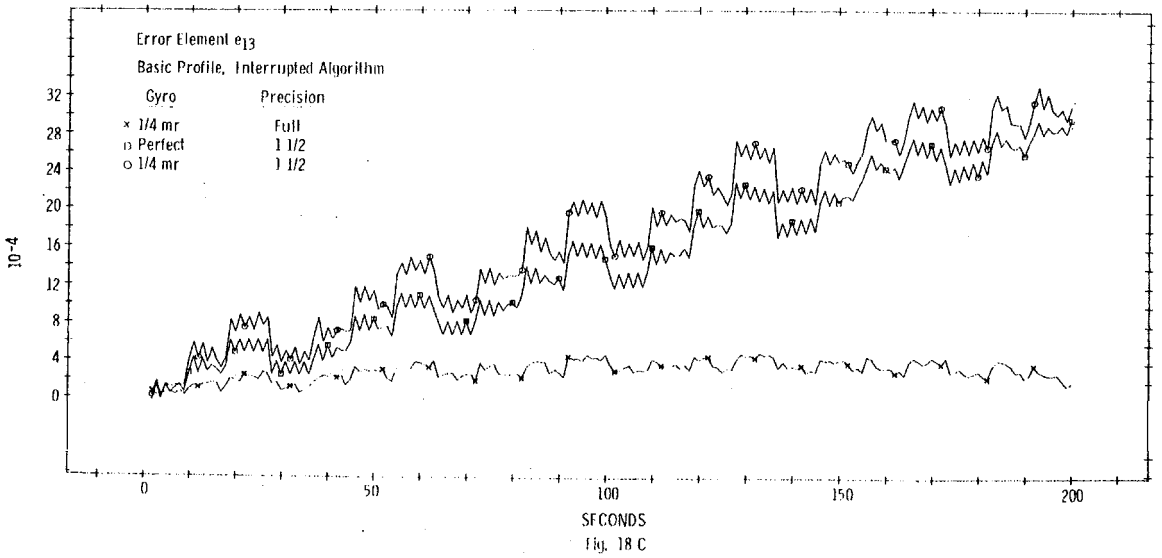


Fig. 18 C

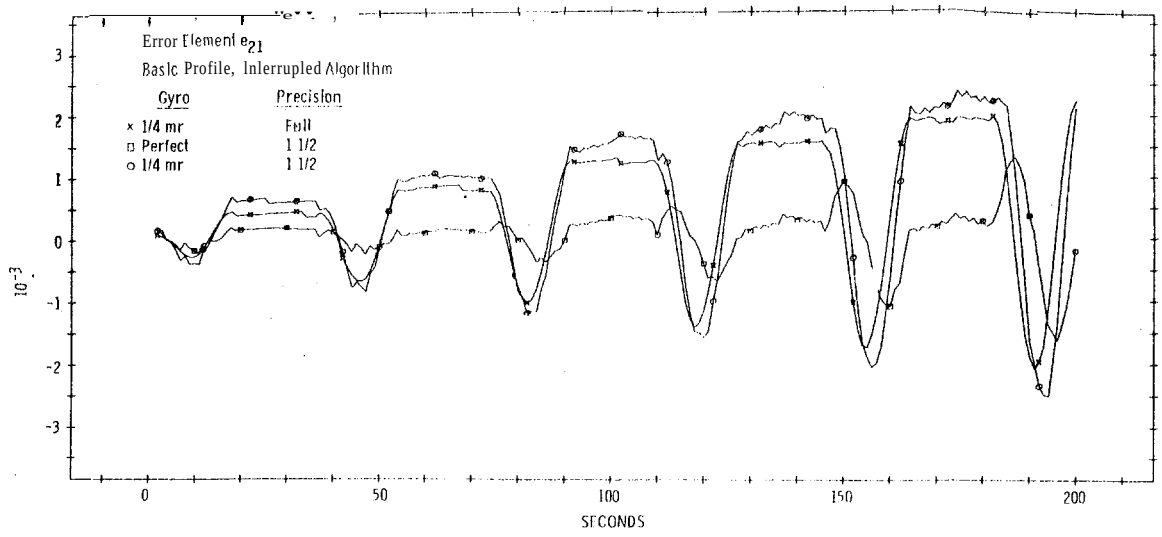


Fig. 18 D

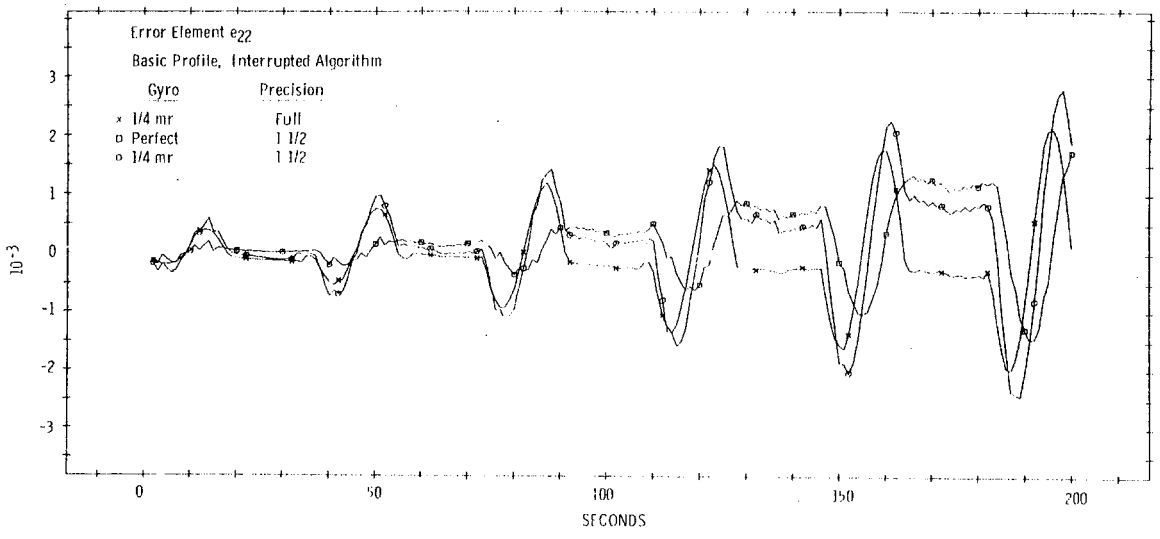


Fig. 18 E

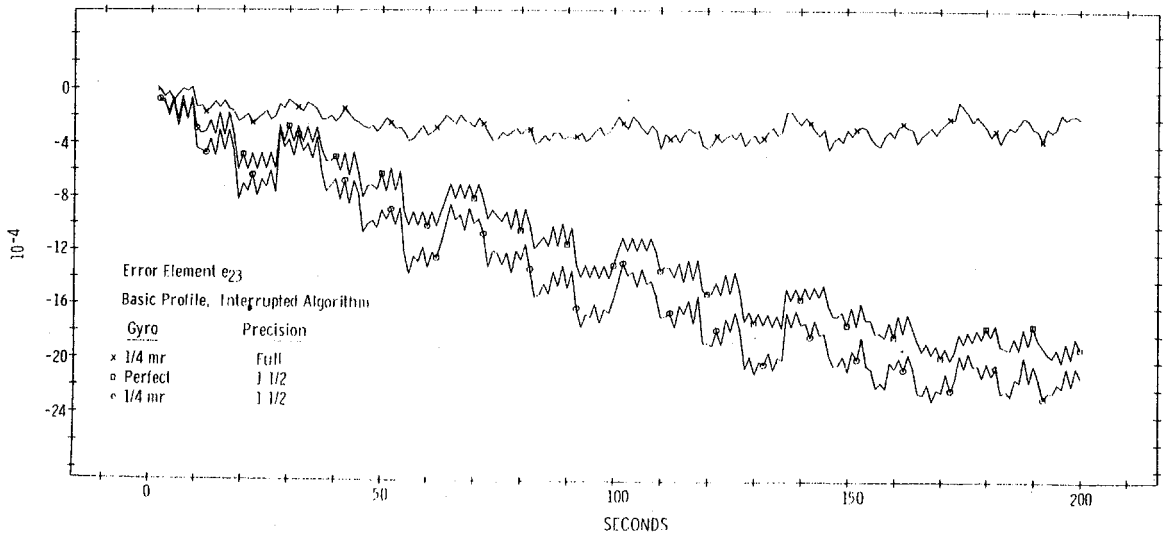


Fig. 18 F

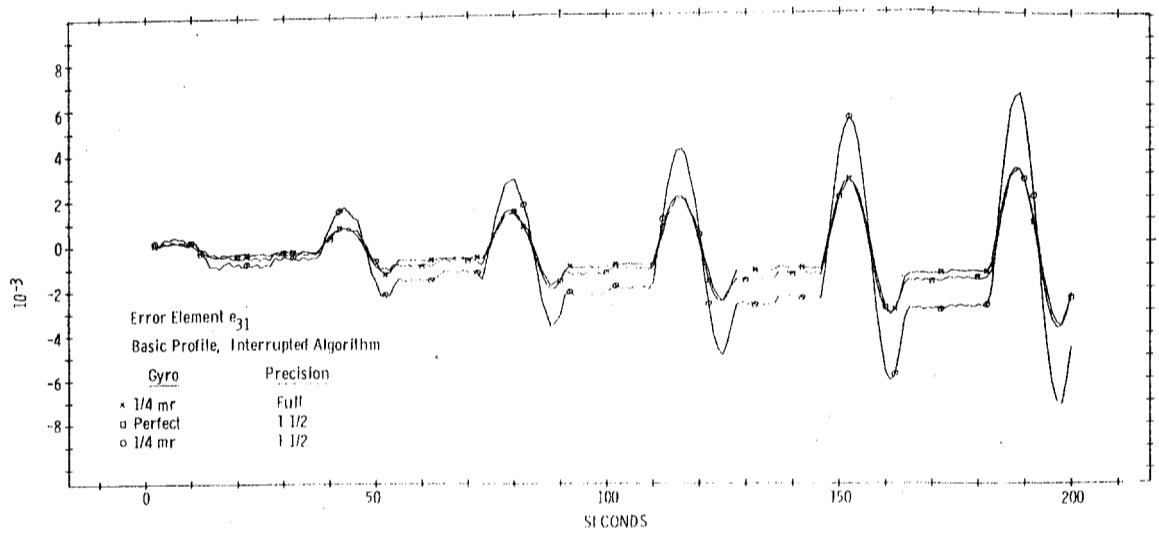


Fig. 18G

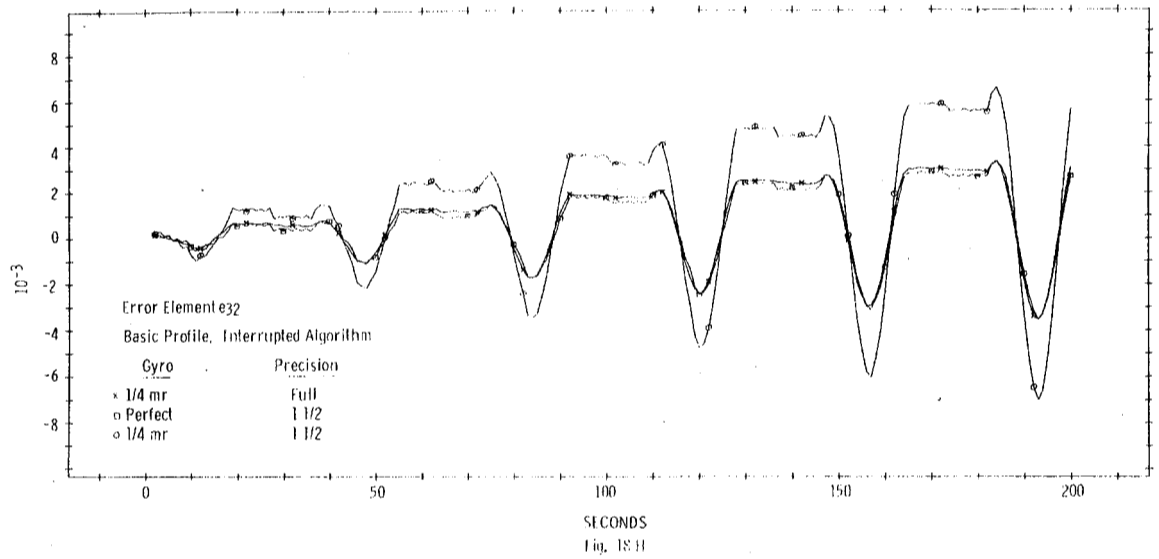


Fig. 18H

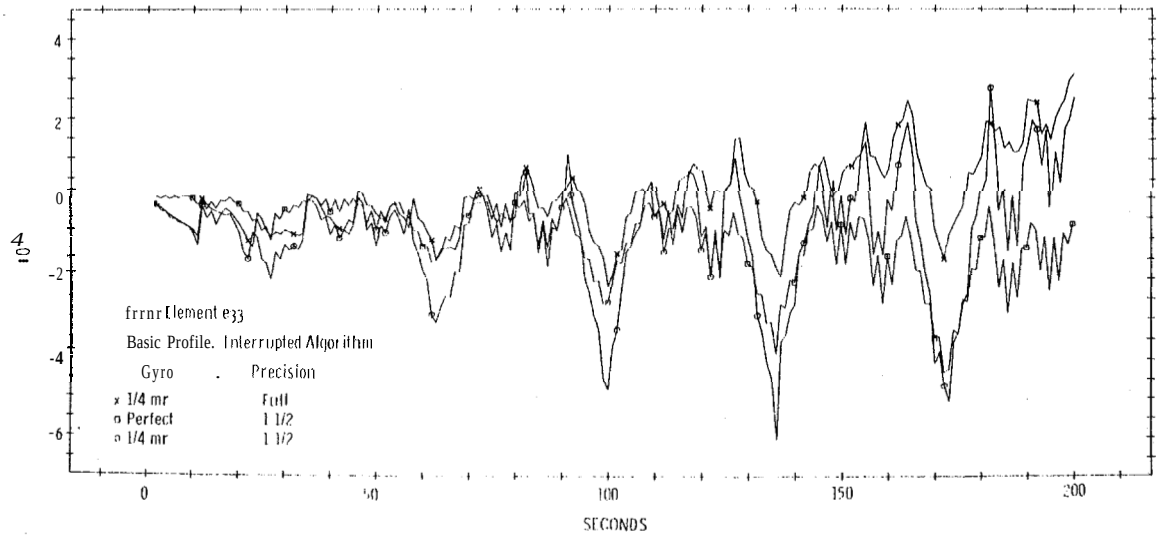


Fig. 18I

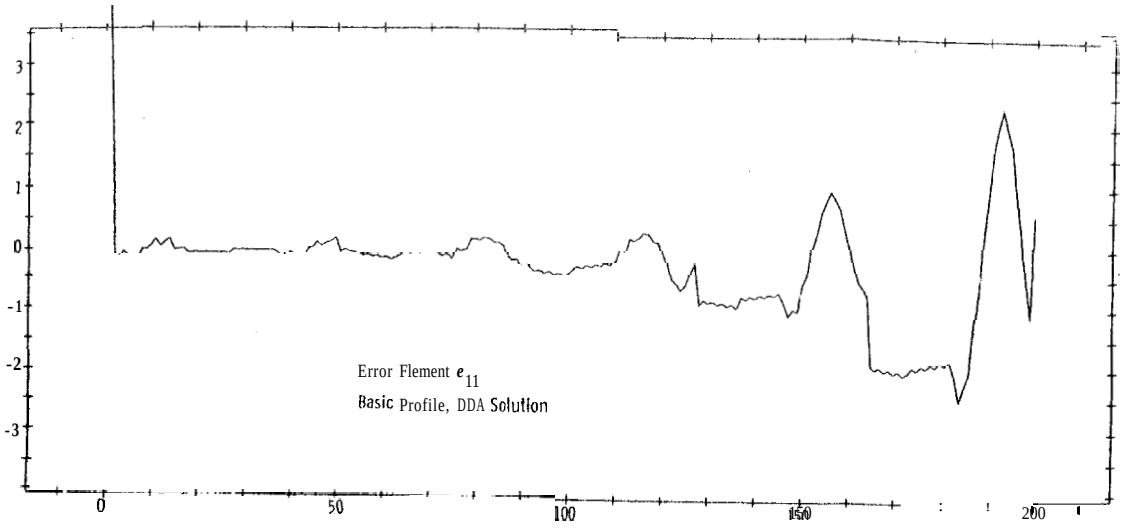


Fig. 19 A

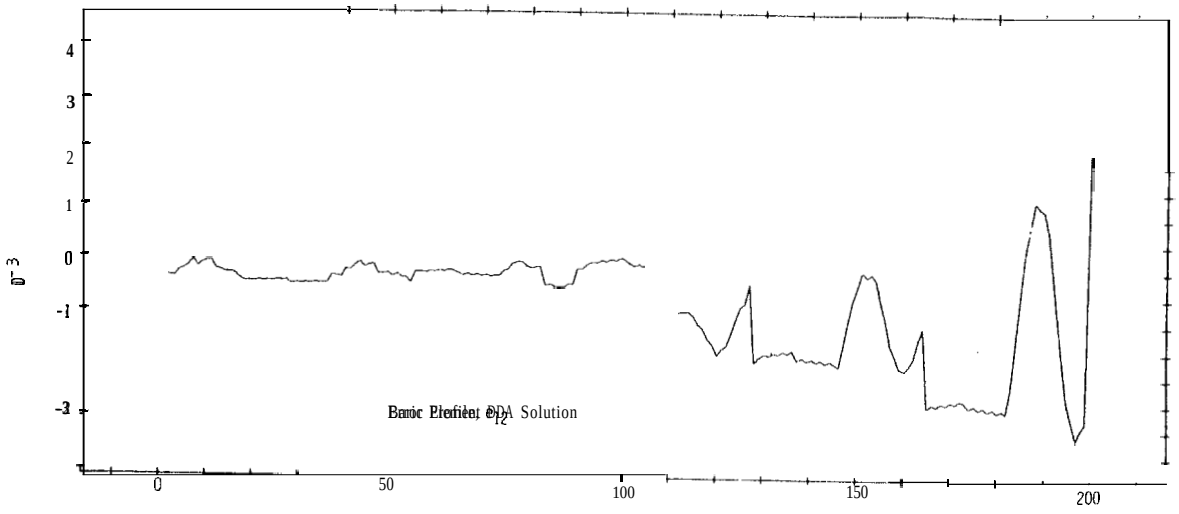


Fig. 19 B

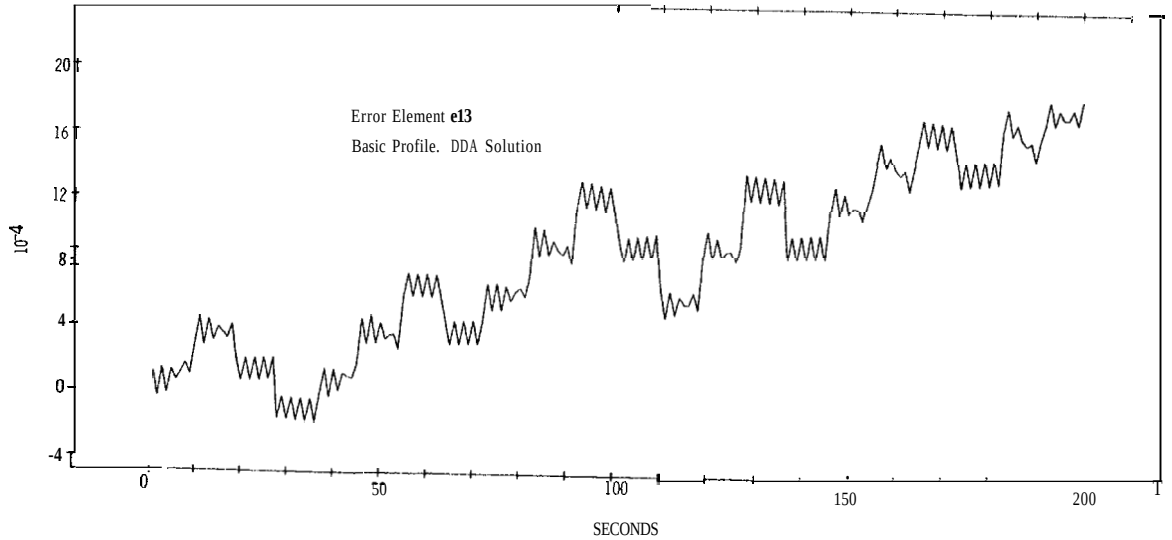


Fig. 19 C

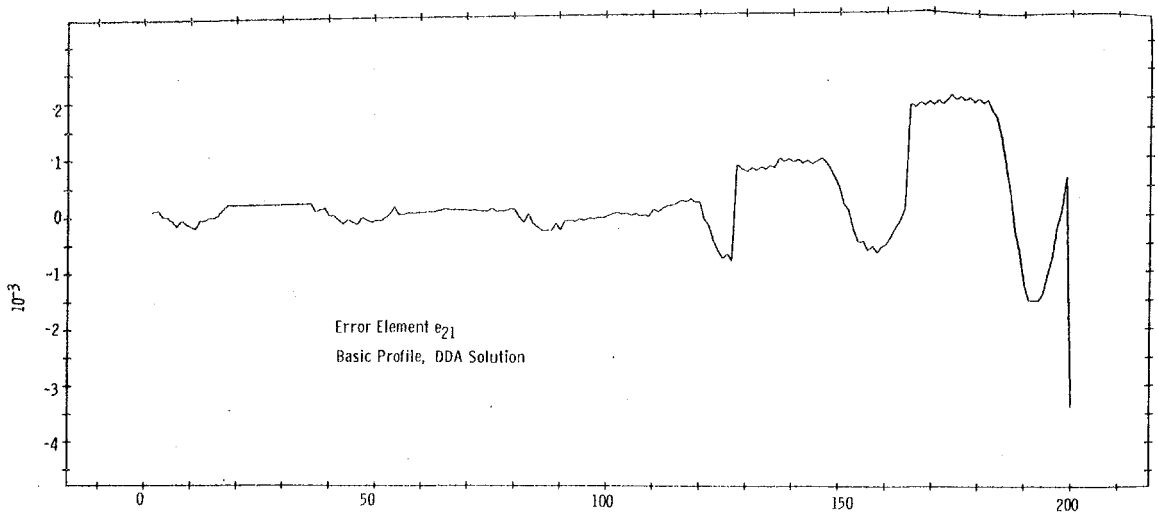


Fig. 19 D

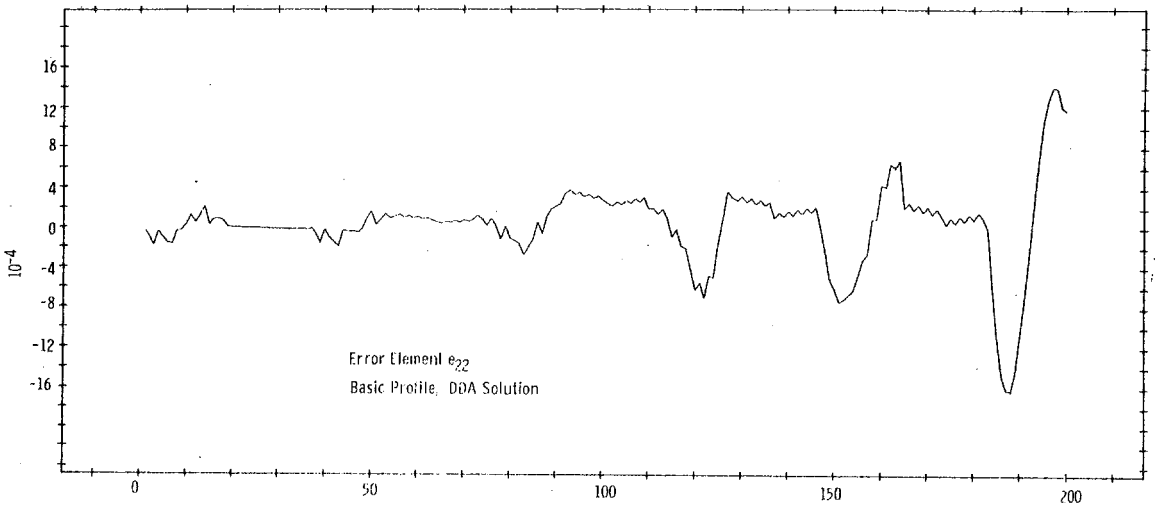


Fig. 19 E

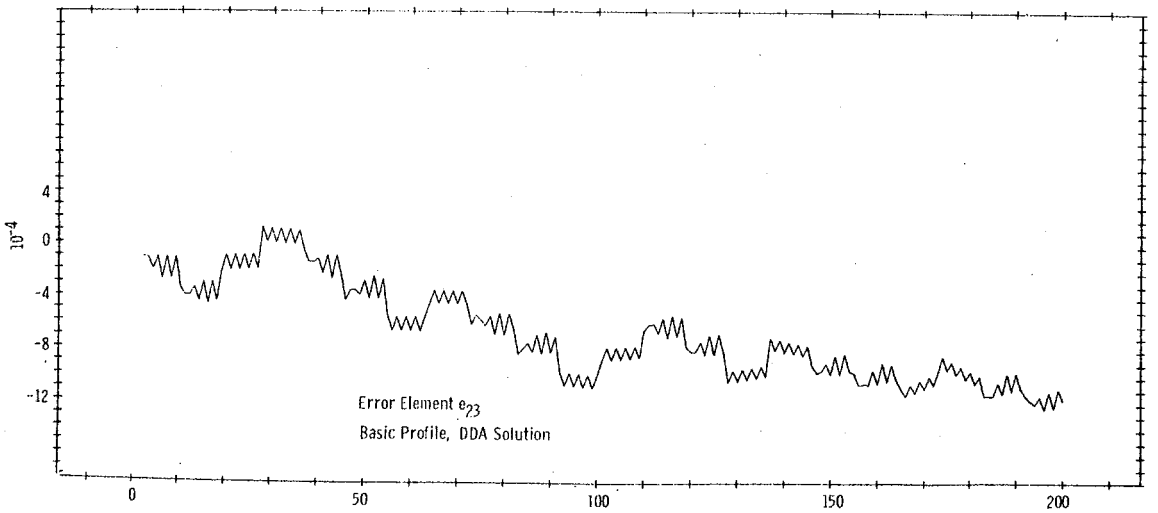


Fig. 19 F

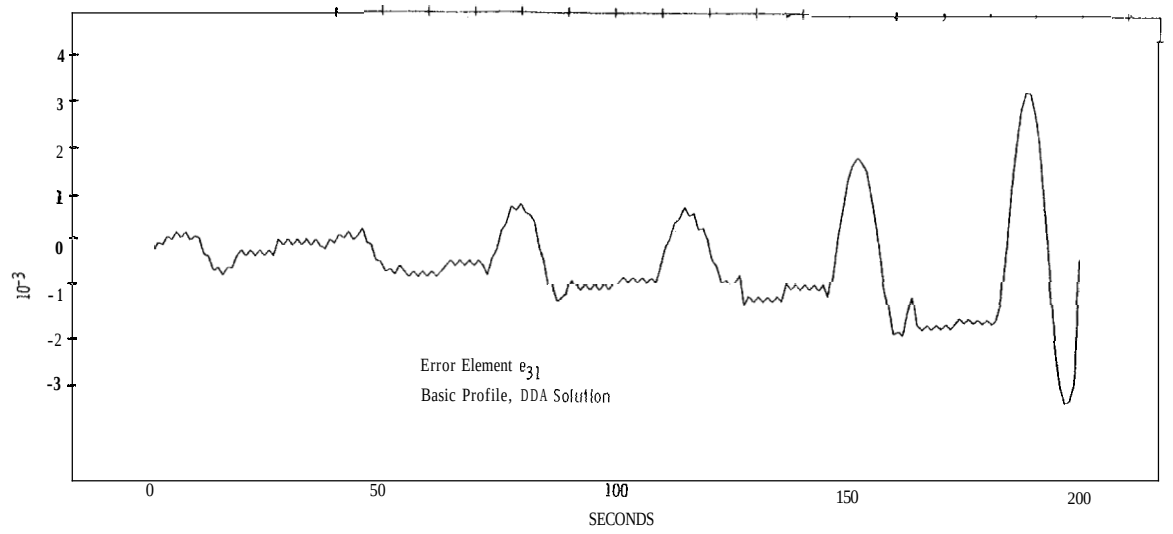


Fig. 19G

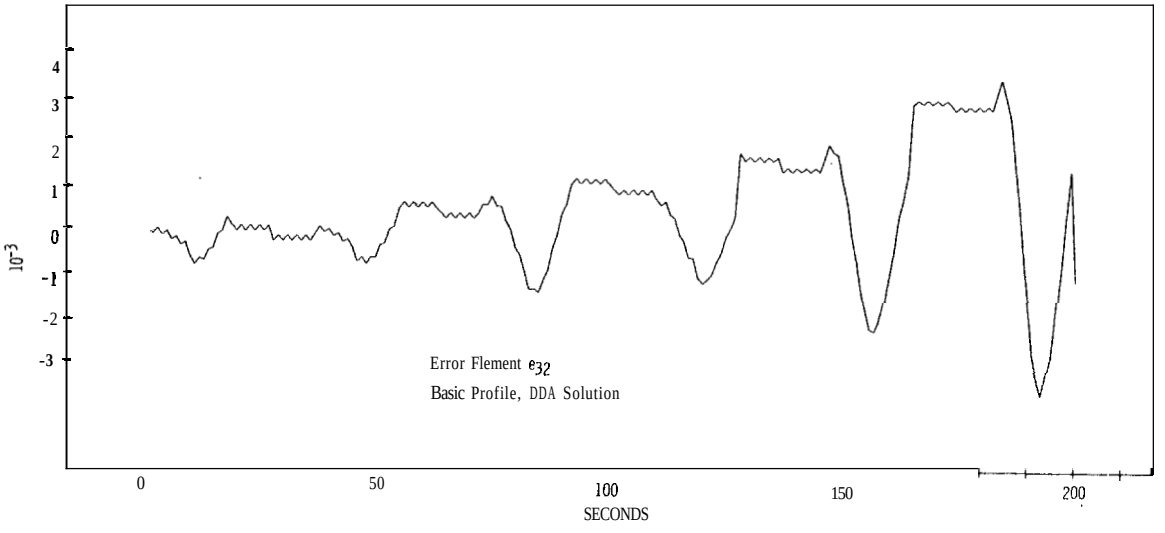


Fig. 19H

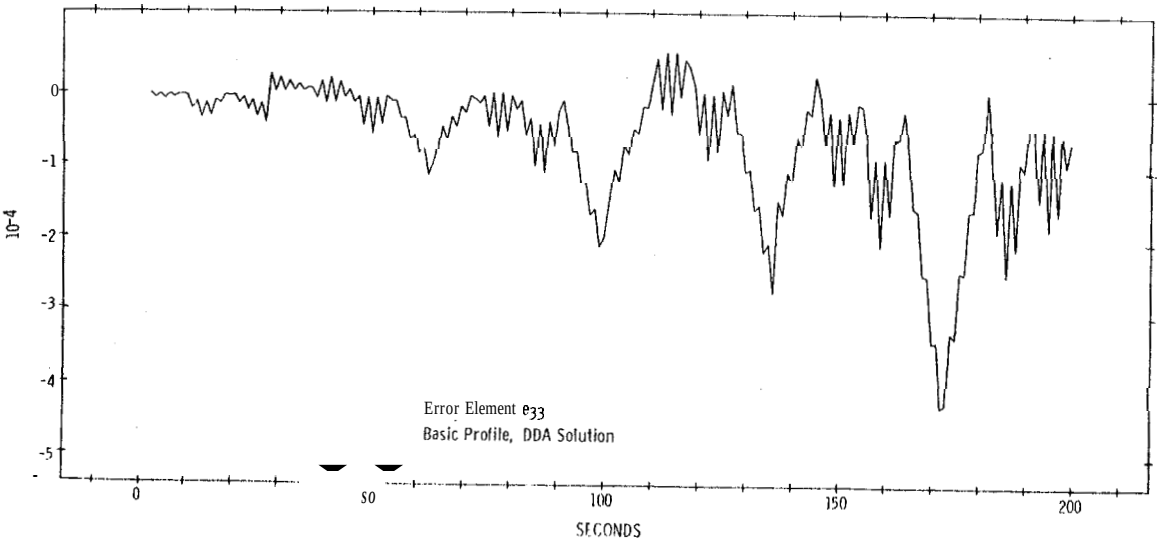


Fig. 19I

VI. MATRIX ORTHOGANALITY

The updated N matrix, regardless of how derived, is an approximation to the true cosine matrix C given in equation (1), a matrix whose elements are the direction cosines of the angles between the axes of the body *co*-ordinate system and the axes of the fixed or inertial coordinate system. Since the two coordinate systems are each orthogonal, it follows that the matrix C represents an orthorgonal transformation and is therefore orthogonal. Furthermore if the C matrix is orthogonal, it follows from equation (6) that the M matrix mast also be orthogonal. The update formula given in Table 3 yields an N matrix which is an approximation to the M matrix; therefore [N] should also be orthogonal.

Consider the matrix:

$$[Z] = [N][N]^T \tag{28}$$

where $[N]^T$ is the transpose of [N]. If [N] is orthogonal, [Z] is the identity matrix; the variation of [Z] from the identity matrix provides a measure of the degree of orthogonallty of [N].

It was initially suggested that the Z matrix might provide an evaluation of the update formula and might further be used to "correct" the N matrix on a real-time basis. In practice, however, it was found that the difference between [Z] and the identity matrix provides at best a crude indication of the effectiveness of the update methods. Examination of the Z matrices given in Table 6 reveals the difficulty in constructively utilizing the property of orthogonality to correct "the update formulas." For example, according to Figure 15, the use of interrupted sampling time intervals yields consistently more accurate results than those realized from the noninterrupted update technique; yet at the termination of the LEM mission, the Z matrix for the noninterrupted algorithm is closer to the identiy matrix than is the Z matrix for the interrupted updating. Similarly, according to the Z matrices d, e, and f of Table 6 the N matrix of the noninterrupted update method is, at the time of maximum error magnitude, less orthogonal than the N matrix of the interrupted N matrix at the same time. However, at the time of peak error magnitude, the N matrix for noninterrupted update formulas is more orthogonal than the N matrix of the interrupted formulas

at the time of its largest error functions. Such results indicate that the property of orthogonality cannot be constructively utilized,

(a)

Z Matrix at End of LEM Mission Resulting from Basic Update Formula, Perfect Gyros and Full Computational Precision.

1.0000119	- .00000012	.00000016
- .00000012	1.0000114	00000018
.00000016	.00000018	1.000011

(b)

Z Matrix at End of LEM Mission Resulting from Interrupted Update Formula, Perfect: Gyros and Full Computational Precision.

1.0000122	.00000044	.00000045
,00000044	1.0000133	.00000016
.00000045	.00000016	1.000013

(c)

Z Matrix at End of LEM Mission Resulting from Interrupted Update Formula, Gyros Quantized at 1/4 Milliradian, 1 1/2 Precision.

.99955	.000070	.000015
,000070	.99984	.000084
.000015	.000084	.99981

(d)

Z Matrix at t = 190 of Profile #1 Resulting from Basic Update Formula, Perfect Gyros and Full Computational Precision.

.9998908	.00001186	,00002645
.00001186	.9998949	,00002999
.00002645	.00002999	.99997939

TABLE 6. TYPICAL Z MATRICES FOR VARIOUS PROFILES

(e)

Z Matrix at t = 190 of Profile #1 Resulting from Interrupted Update Formula, Perfect Gyros and Full Computational Precision.

$$\begin{bmatrix} .9998918 & .00001155 & .00002580 \\ .00001155 & .9998965 & .00002995 \\ .00002580 & .00002995 & .9999797 \end{bmatrix}$$

(f)

Z Matrix at t = 193 of Profile #1 Resulting from Interrupted Update Formula, Perfect Gyros and **Full** Computational Precision.

$$\begin{bmatrix} .9998889 & .00001248 & .00002684 \\ .00001248 & .9998943 & .00003142 \\ .00002684 & .00003142 & .9999778 \end{bmatrix}$$

TABLE 6 (cont.)

R-531

DISTRIBUTION LIST

Internal

M. Adams (MIT/GAEC)
J. Alekshun
R. Battin
P. Bowditch/F. Siraco
R. Byers
G. Cherry
N. Cluett
E. Copps
R. Crisp
J. Dahlen
J. DeLisle
E. Duggan
J.B. Feldman
P. Felleman
S. Felix
J. Flanders (MIT/KSC)
F. Grant
Eldon Hall (50)
W. Keintz
T. Hemker (MIT/NAA)
E. Hickey
D. Hoag
F. Houston
L.B. Johnson
M. Johnston
A. Koso
A. Laats
D. Ladd
L. Larson
J, Lawrence (MIT/GAEC)
T.J. Lawton
T.M. Lawton (MIT/MSC)
D. Lickly
L. Martinage
G. Mayo
J. McNeil
James Miller
John Miller
J. Nevins
J. Nugent
E. Olsson
J. Pennypacker
M. Petersen
M. Richter (MIT/MSC)
R. Scholten
E, Schwarm
J. Sciegienny
N. Sears
J. Shillingford
W. Shotwell (MIT/AC)
J. Sitomer
W. Stameris
J. Suomala
W. Tanner
M. Trageser
R, Weatherbee
R. White
L. Wilk
M. Wolff
R. Woodbury
W. Wrigley
Apollo Library (2)
MIT/IL Library (6)

CONCLUSIONS

Extensive computer simulations have verified that the transformation matrix required for attitude reference in a strapdown inertial navigation system can for certain missions be updated at relatively long time intervals with sufficient accuracy by an on-board general purpose whole number computer. The accuracy of the update formulas is strongly dependent upon the characteristics of the particular flight profile, specifically upon the spacecraft rotational velocity, rotational acceleration, the number of times angular accelerations are encountered, and the times at which the accelerations occur. The third order update formulas offer a reasonable compromise between computational complexity and accuracy of the updated matrix; little improvement is realized by using fourth order formulas while significant degradation results from second order expressions. Sampling time intervals of the order of 0.1 second are sufficiently small to yield meaningfull results; smaller intervals will of course yield more accurate results but will also place an increasing computational load upon the navigation computer. Real time knowledge of the occurrence of discontinuities in the time derivatives of the angular velocities can be used to significantly improve the performance of the basic algorithm. A sufficient increase in computer word length such that computations can be performed in single precision without excessive loss in computational accuracy results in improvement in the accuracy of the whole number algorithm. Similarly, an increase in precision of readout from the strapped down gyros results in improved accuracy of the transformation matrix.

Relating the results of the study to the AGC Block II computer, it appears that the Block II computer yields results in terms of accuracy which are barely acceptable. Second generation airborne computers will, however, operate at speeds five to ten times faster than the Block II AGC. The increased accuracy resulting from shorter sampling intervals thus makes a strapdown navigation system employing a general purpose computer very attractive.

External:

W. Rhine (NASA/MSC)	(2)
NASA/RASPO	(1)
L. Holdridge (NAA/MIT)	(1)
T. Heuermann (GAEC/MIT)	(1)
AC Electronics	(3)
Kollsman	(2)
Raytheon	(2)
Major H. Wheeler (AFSC/MIT)	(1)

MSC: (25 + 1R)

National Aeronautics and Space Administration
Manned Spacecraft Center
Apollo Document Distribution Office (PA2)
Houston, Texas 77058

LRC: (2)

National Aeronautics and Space Administration
Langley Research Center
Hampton, Virginia
Attn: Mr. A. T. Mattson

GAEC: (3 + 1R)

Grumman Aircraft Engineering Corporation
Data Operations and Services, Plant 25
Bethpage, Long Island, New York
Attn: Mr. E. Stern

NAA: (18 + 1R)

North American Aviation, Inc.
Space and Information Systems Division
12214 Lakewood Boulevard
Downey, California
Attn: Apollo Data Requirements AE99
Dept. 41-096-704 (Bldg 6)

NAA RASPO: (1)

NASA Resident Apollo Spacecraft Program Office
North American Aviation, Inc.
Space and Information Systems Division
Downey, California 90241

ACSP RASPO: (1)

National Aeronautics and Space Administration
Resident Apollo Spacecraft Program Officer
Dept. 32-31
AC Electronics Division of General Motors
Milwaukee 1, Wisconsin
Attn: Mr. W. Swingle

ACSP KASPO:

Defense Contract Administration (1)
Service Office, R
Raytheon Company
Hartwell Road
Bedford, Massachusetts 01730

Mr. S. Schwartz (1)
DOD, DCASD, Garden City
605 Stewart Avenue
Garden City, L.I., New York
Attn: Quality Assurance

Mr, D. F. Kohls (1)
AFPRO (CMRKKA)
AC Electronics Division of General Motors
Milwaukee 1, Wisconsin 53201