AD NUMBER: AD0816073

LIMITATION CHANGES

TO:

Approved for public release; distribution is unlimited.

FROM:

This document is subject to special export controls; 01 Apr 1967, and each transmittal to foreign governments or foreign nationals may be made only with prior approval of the U.S. Naval Ordnance Test Station, China Lake, CA 93555.

AUTHORITY

ST-A USNWC LTR, 24 MAR 1972

NOTS TP 4267

STRAPPED-DOWN INERTIAL GUIDANCE

THE COORDINATE TRANSFORMATION UPDATING PROBLEM

by

William F. Ball
Weapons Development Department

ABSTRACT. This report provides a detailed introduction to the problem of updating the rotational transformation of a strapped-down inertial guidance system. Analyses are specifically devoted to problems of updating the direction cosine transformation. Various digital computer algorithms for updating the direction cosines are displayed and discussed. Methods of performing error analyses on these algorithms are reviewed and results of IBM 7094 analyses are shown. The primary analytical results are representations of the truncation error resulting from the particular numerical forms studied.

U.S. NAVAL ORDNANCE TEST STATION
China Lake, California
April 1967

U. S. NAVAL ORDNANCE TEST STATION

AN ACTIVITY OF THE NAVAL MATERIAL COMMAND

G. H. Lowe, Capt., USN, Commander
Wm. B. McLean, Ph.D., Technical Director

FOREWORD

In the analysis and development of strapped-down inertial systems at the U. S. Naval Ordnance Test Station (NOTS), China Lake, California, considerable attention has been given to the evaluation of problems related to rotational transformation updating. The literature published during the past few years gives evidence of concern by other governmental agencies and private industry with this problem.

This report is concerned with approaching the problem from the specific standpoint of direction cosines. Drawing on the backlog of diverse published material available, a unified approach has been taken toward developing a capability in understanding and evaluating the various updating algorithm forms.

Digital computer solutions used in the analyses were developed by C. Messinger (NOTS) and by the author. Programming assistance on the MIT Gyro Model was given by Miss R. Hession and R. Bumstead of the MIT Instrumentation Lab., Cambridge, Massachusetts. Program descriptions and user information are available in Ref. 17 to this paper.

This work was performed under AirTask A36-533-205/216/1/F009-03-03.

Review for technical accuracy was given by Czerna Flanagan, Research Department. This report is released at the working level for information purposes only.

Released by
D. K. MOORE, Head,
Undersea Systems Div.
11 April 1967

Under authority of
F. H. KNEMEYER, Head,
Weapons Development Dept.

NOTS Technical Publication 4267

Published by . . . . . Weapons Development Department
Manuscript 40/66-99
Collation . . . . . Cover, 61 leaves, DD Form 1473, abstract cards
First printing . . . . . 145 unnumbered copies
Security classification . . . . . UNCLASSIFIED

CONTENTS

ACKNOWLEDGMENT

Particular acknowledgment is due Czerna Flanagan who reviewed the manuscript and offered many valuable suggestions for the improvements as regards consistency, mathematical correctness, and general organization.

SYMBOLS

$\mathcal{H}$	Gyro wheel angular momentum
$\underline{1}_i$	Unit vector with direction along i^{th} body axis
C_k	Direction cosine matrix at k^{th} iteration inverval
$[C]$	Direction cosine matrix
$[C']$	Reference direction cosine matrix in digital computer programs
$\{C(o)\}$, $\lvert C_o \rvert$	$[C(t)]$ for $t=0$: initial condition on $[C]$
$[\epsilon]$	Difference in true and algorithm updated direction cosine matrix
$[H]$	A matrix that "updates" a direction cosine matrix
$\lvert M_{\cdot\cdot}^{-1} \rvert$, $\lvert M_{\cdot\cdot} \rvert$	Bumstead-Hession "forward" and "reverse" algorithms
$[\Phi]$	Orthogonal rotational transformation updating matrix, the "exact" updating or "reference" updating
$[\Phi']$	Reference updating used in digital computer programs
$[\Omega]$	Skew symmetric angular velocity matrix
$\lvert \Omega_k \rvert$	$[\Omega]$ at $t = t_k$
C	Viscous friction
I	Gyro float moment of inertia
K_1	Pulse rebalance mode/ternary
M	Rebalance torque = $\mathcal{H}\,\omega_{max}$
R_A	Shift register containing the word A
$\mathfrak{I}$	Rotational transformation operation
T_G	Gyro clock period multiple - a "grouping interval"

$\underline{a}$ Vector inertial acceleration

$\underline{a}_B$ Inertial acceleration vector measured in body frame

$\underline{a}_C$ Inertial acceleration vector measured in computational frame

a_i, a_j Inertial acceleration component along i^{th} or j^{th} axis

C_{ij} Components of direction cosine matrix for

$k+a$, $k+b$ Intermediate updating steps (subiteration)

kT_s k^{th} clock period, T_s is the clock period

r' Coefficient used in setting gyro threshold

$\Delta\underline{V}_B$ Velocity increment measured in body axes

$\Delta\underline{V}_C$ Velocity increment measured in computation axes

Δt_a Accelerometer-pulse time increment

Δt_g Gyro-pulse time increment

$\Delta\underline{\theta}$ Vector angular increment

$\Delta\theta_i$, k_i Gyro angular increment (i^{th} gyro)

θ_{float} Total float angle

θ_i, $\theta_{i,float}$ Float angle of i^{th} gyro

$\left.\begin{matrix}\theta_o, \omega_{oi} \\ \theta_{o,f}, \omega_{o,f}\end{matrix}\right\}$ Gyro float initial conditions

θ_T Gyro threshold angle

θ_ϵ Gyro storage error

τ_{ij} Components of 3rd term in matricant

λ Eigenvalues

τ Gyro float time constant

ψ Root sum square of $\int_0^t \omega_i dt$; $i = 1, 2, 3$

ω_f Gyro float rate

ω_{IA} Gyro input rate

$\omega_i(t_k)$ Input seen by i^{th} gyro at $t = \delta_k$

(ω_{max}) $\Delta\theta \cdot T_s = hT_s$: maximum input rate capability of gyro

$\angle\psi_\phi$ Angle with arc tangent ψ_ϕ

C Closed contour

$Z[f(t)]$ Z-transform of a function $f(t)$

$Z^{-1}[F(z)]$ Inverse Z-transform

INTRODUCTION

Inertial navigation or inertial guidance systems are systems that depend solely upon on-board measurements to accomplish their functions. Conventional inertial systems employ inertial sensors (gyros and accelerometers) mounted on a platform isolated from vehicle motion by a system of gimbals. An alternate configuration for inertial navigation and guidance systems is a system employing inertial instruments fixed to the vehicle. This type of inertial system is called gimballess or strapped-down, for obvious reasons.

An important element of a strapped-down inertial system is the transformation computer. The transformation computer converts the vehicle-fixed inertial sensor information to a frame of reference that is generally isolated from vehicle motion. Observing that both the gimballed and gimballess systems can produce inertial sensor information in an isolated frame of reference, it is possible to draw an analogy between the physical gimbal set and the "mathematical gimbals" (or coordinate transformation) of the strapped-down system. Thus, except for actual configuration, the gimballed and gimballess systems are equivalent.

A considerable amount of unclassified literature has appeared in recent years on the subject of strapped-down inertial systems. Many of these reports have dealt with the problems of coordinate transformations. The purpose of this report is to address the coordinate transformation problem using direction cosines. Computational methods, algorithms, and error analysis methods appearing in some of the publications are gathered together with a unified notation and organization to serve as an introduction to the subject, to provide a handbook of methods, and to assist the engineer and analyst in making comparisons and evaluations. A complete list of references is given.

To provide a basis for the study of the coordinate transformation problem, it will be helpful to consider the following discussion. Assume that three orthogonal components of inertial acceleration, a_1, a_2, a_3, are measured in a body frame of reference. These three components are represented by the vector

$$\underline{a}_B = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}_B , \qquad (1)$$

with B denoting "Body frame." The vector $\underline{a}_B$ may be related to the vector $\underline{a}_C$, acceleration measured in the preferred computational frame, by the expression

$$\underline{a}_C = \mathfrak{Z}\left(\underline{a}_B\right) \quad . \tag{2}$$

The symbol $\mathfrak{Z}$ represents some rotational transformation operation. The body-fixed axes are, in general, changing their relative orientation with respect to the computational reference frame continuously as a function of the body rates about these axes. It is necessary, therefore, to continuously update the separate elements of the transformation $\mathfrak{Z}$. To update $\mathfrak{Z}$, the body rate information available from the strapped-down gyros is employed. The problem as studied herein, involves the computer mechanization of schemes that will use the gyro-sensed body angular rates in performing the $\mathfrak{Z}$-updating task.

The report is organized in the following manner—the first section, "Preliminary Considerations," reviews some of the basic aspects of strapped-down system mechanization. Included in the mechanization discussion is a summary dealing with the use of pulse-controlled, strapped-down gyros and accelerometers. Also, in this first section is the development of two basic gyro models. Finally, the topic of computer arithmetic is discussed. Those readers familiar with strapped-down mechanization can skip this section. Similarly, those readers acquainted with the method of direction cosines can skip the second section, "Notations, Definitions, and Problem Statement." Readers interested only in the mathematics of updating direction cosines may proceed directly to the section, "Algorithms for Updating the Direction Cosine Matrix." In that section, the basic forms of the more common direction cosine algorithms are given. Following this, in "Algorithm Form Variations," some particular variations on the basic algorithm forms are discussed. These are variations that enhance some particular performance characteristic of the given algorithm. Having introduced the algorithms, the section, "Algorithm Analysis Methods" deals with the various analytical tools available for their study. Finally, the last section is devoted to investigations of the truncation error response of some of the algorithms to various inputs. Noncommutativity and roundoff errors will not be studied as such.

PRELIMINARY CONSIDERATIONS

The material presented in this section is intended to serve as background material for those readers unfamiliar with strapped-down inertial systems. The mechanization of a strapped-down inertial system will be described along with the instrumentation of the pulse-controlled, strapped-down sensors. Further, since sensor performance influences the system computational accuracy, two basic gyro models are developed and discussed. The sensor models are also a basic element in the analysis of the transformation updating scheme because they determine the form of the input to the computer. Finally, a brief discussion of elementary computer arithmetic is given.

STRAPPED-DOWN SYSTEM MECHANIZATION (Ref. 15 and 19)

Because of their contribution to system errors (e.g., quantitization), it is important to have an understanding of the operation of the pulse-controlled gyros and accelerometers. Also, a description of the strapped-down mechanization concept is given to aid in the understanding of the relationship of the sensors to the remainder of the system.

For purposes of this study, the gyros and accelerometers are assumed to be conventional single-degree-of-freedom floated instruments. Each gyro and each accelerometer is operated independently as a sampled-data control system, the torque generator of each instrument being pulse-controlled by the output of its signal generator. The pulse-control is effected by constant pulses appropriately applied by the control loops to perform the pulse-rebalancing task. Figure 1 is a schematic representation of a pulse-controlled gyro.

The pulse-torquing of these instruments is desirable from two standpoints: The rebalance torque can be more accurately generated with pulse-torquing equipment than with analog equipment, and the output of each pulse-rebalanced instrument is in the proper form for use in a digital computer.

In the pulse-torqued mode, the sensors are operated so that the angle between float (for the gyro) or pendulum (for the accelerometer) and sensor case is restrained to be very small. The float of the gyro, then, has essentially the same vector angular velocity as the vehicle-mounted gyro case; the pendulum of the accelerometer, likewise, has essentially the same acceleration as the vehicle.

For the pulse-controlled components described, the rate at which pulses are applied to the accelerometer torque motor gives a measure of the linear acceleration along the accelerometer sensitive axis. Also, the measure of angular rate is supplied by the pulse-rate required to rebalance the gyro float.

The pulsed gyro output will differ from the true gyro output by the amount of storage error in the gyro integrating element (see page 9). This *storage error* results, essentially, from the combined effects of the gyro float time-constant and the attempt to quantize the analog signal. Similarly, the pulse-rebalance accelerometer output differs from the true output over a given period of time by the storage error of the integrating element.

For a given interval of time Δt_a, a rebalance pulse to the accelerometer represents the incremental quantity

$$\Delta V \cong a \cdot \Delta t_a \quad , \tag{3}$$

where a is the constant or average acceleration over the interval Δt_a. This approximates the more general case, where a is a function of time and the incremental quantity is ideally expressed as the integral

$$\Delta V = \int_{t_1}^{t_2} a(t)\, dt \quad , \tag{4}$$

for $\Delta t_a = t_2 - t_1$.

A rebalance pulse to the gyro torquer represents the increment

$$\Delta\theta = \omega_{IA} \cdot \Delta t_g \tag{5}$$

where ω_{IA} is the constant or average gyro input rate over the interval Δt_g. As with the accelerometer, this is an approximation to the general case where $\omega_{IA} = \omega_{IA}(t)$ and

$$\Delta\theta = \int_{t_1}^{t_2} \omega_{IA}(t)\, dt \quad . \tag{6}$$

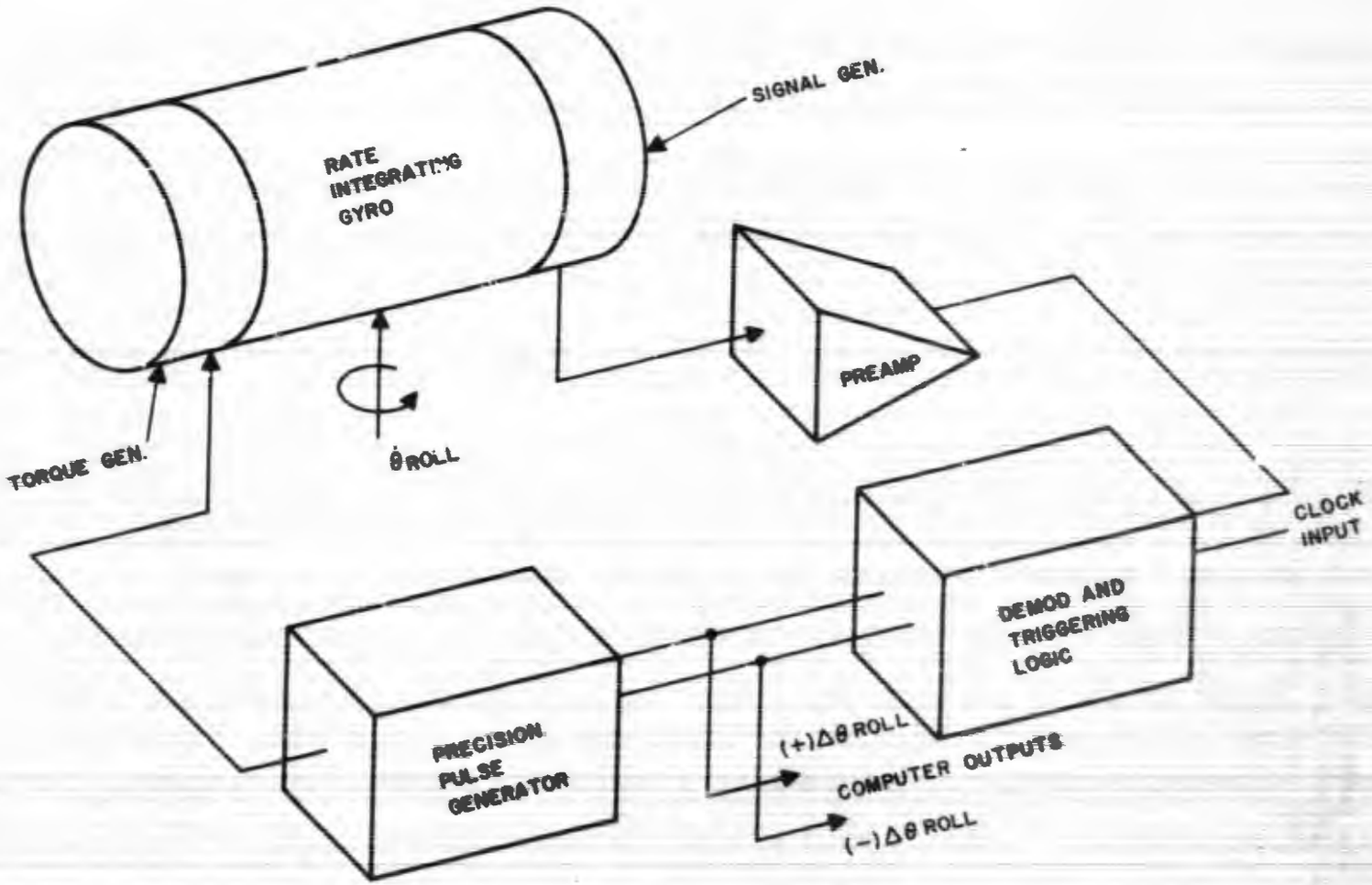


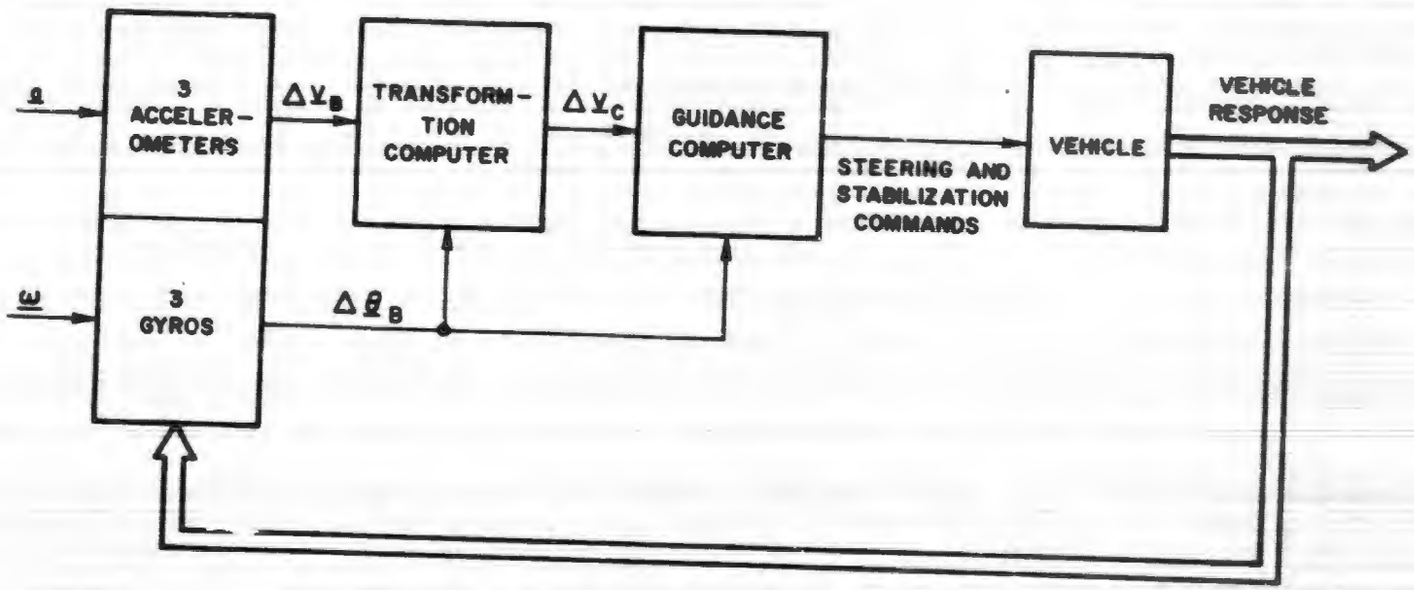
FIG. 1. Pulse-Rebalanced Gyro Schematic.

NOTS TP 4267

Now the general strapped-down mechanization concept using pulse-controlled sensors is described. The mechanization employed in this study utilizes three, single-axis accelerometers to measure components of acceleration along three orthogonal, body-fixed axes. Three orthogonally mounted single-axis gyros are used to measure body angular rates about the body-fixed axes defined by the accelerometer triad. This angular rate information is then used in a transformation computer to keep the transformation matrix current. The transformation matrix relates the body-fixed inertial information to a coordinate frame chosen as suitable for guidance or navigation computations, generally an inertial frame. With vehicle motion, the orientation of the body-fixed sensors' input axis are changing with respect to the preferred computational coordinate. It is necessary, therefore, to update (recompute) continually the individual elements of the transformation matrix by using the body angular rate information from the gyros (the information is actually available from the pulse-controlled gyros in the form of small angular increments). The output of the transformation computer, the transformed acceleration vector (actually available as velocity increments because the accelerometer is pulse-controlled), is analogous to the acceleration output of conventional gimbal isolated accelerometers. The navigation or guidance computations are then identical to those performed in a gimballed system. A block diagram of this mechanization is given in Fig. 2. In Fig. 2, the computations are performed in the inertial frame of reference on the transformed velocity increments, $\Delta \underline{V}_C$.

TWO GYRO MODELS (Ref. 1)

In the study of errors that arise in using a digital machine to compute the updating of the transformation matrix, consideration must be given to choosing not only representative inputs but realistic ones. The representative inputs are determined from considerations of the performance parameters, such as velocity and turning rate of the vehicle carrying the strapped-down system. For a representative gyro input, the turning rate might be specified as three smooth functions of time, one for each axis. The gyro output is an angular increment that becomes the input to the computer. This computer input is a discrete function (ideally the integral of the gyro inputs at discrete points in time); it is not, in practice, however, the integral of the smooth input functions. For one thing, the digital input is obtained by quantizing the gyro output. The result is a quantitization error. In addition, there are other errors that arise from the gyro system, the gyro storage error for example. Below, we consider two gyro models that were used in this study. The more basic of these models is simply a non-zero reset integrator, which implies that no gyro dynamics are simulated. The second model utilizes a programming method introduced by the MIT Instrumentation Laboratory (Ref. 1). The MIT model includes gyro dynamics that are absent in the first model. Both models will be discussed together with relevant equations and block diagrams.



$\underline{a}$ = VECTOR INERTIAL ACCELERATION

$\Delta \underline{V}_B$ = VELOCITY INCREMENT MEASURED IN BODY AXES

$\Delta \underline{V}_C$ = VELOCITY INCREMENT MEASURED IN COMPUTATION AXES

$\Delta \underline{\theta}_B$ = VECTOR ANGULAR INCREMENT MEASURED IN BODY AXES

$\underline{\omega}$ = VECTOR ANGULAR RATE

FIG. 2. Block Diagram of the Transformed Body Information Scheme.

NOTS TP 4267

SIMPLE GYRO MODEL

The gyro model is shown in block diagram form in Fig. 3. This model functions essentially as a clock-regulated, non-zero reset integrator. The input angular rates are integrated separately during the clock periods, kT_s. At each clock pulse, kT_s, the integrated rates are compared to threshold values $\Delta\theta_i = h_i$. If

$$\theta_i \geq \Delta\theta_i \quad , \tag{7}$$

the i^{th} integrator output is reset such that

$$\theta_\varepsilon = \theta_i - \Delta\theta_i \quad . \tag{8}$$

This is analogous to the i^{th} gyro float angle, having exceeded the given threshold, being rebalanced by a quantum of torque (with zero float time-constant). If $\theta_i < \Delta\theta_i$, the i^{th} integration is allowed to continue without alteration through the next clock interval, $(k + 1)T_s$, following which the procedure stated by Eq. 7 is repeated. For non-zero values, θ_ε is referred to as gyro storage error. This results from not being able to return the gyro float exactly to zero. This is demonstrated in Fig. 4. At $t = 3T_s$ in Fig. 4, Eq. 7 is finally true. At this point,

$$\theta_{float} = \Delta\theta + \theta_\varepsilon \quad . \tag{9}$$

$|\theta_{float}|$ has exceeded the threshold $\Delta\theta$ by the amount θ_ε during the period $(2T_s, 3T_s)$. Upon applying a quantum of torque to the gyro torquer, the gyro float moves to an angular error in float position lying between zero and $1\Delta\theta$.

$$\theta_{float} - \Delta\theta = \theta_\varepsilon \quad .$$

Further inspection of Fig. 4 reveals that if T_s were reduced in length, for the input rate shown, there is a particular T_s (denoted T_s' in the figure) that would allow quick recovery of the storage error. For the T_s shown, the storage error is zero at $3T_s'$.

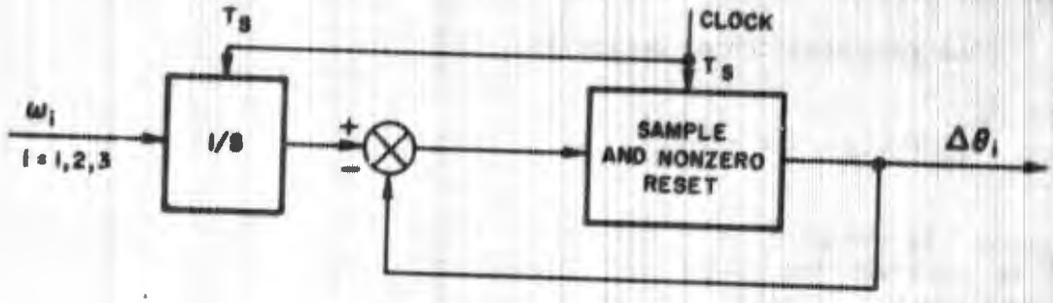


FIG. 3. Simple Gyro Model.

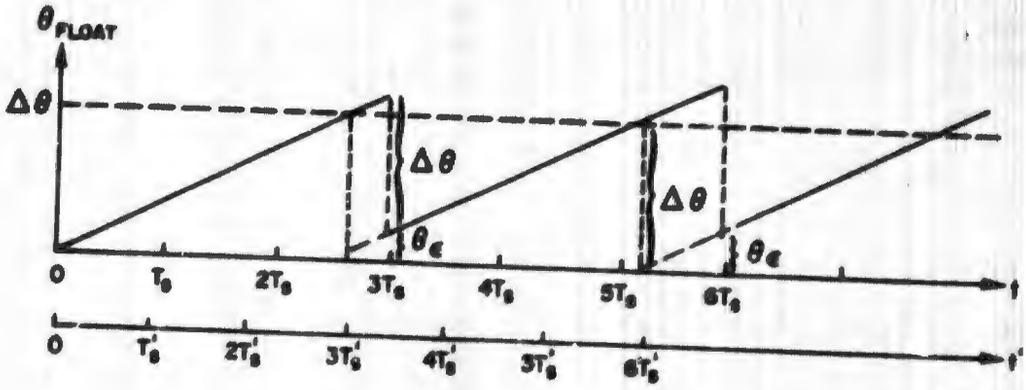


FIG. 4. Gyro Float-Storage Angle Example.

MIT GYRO MODEL

Inclusion of gyro dynamics, as in the MIT model, yields a model less artificial than the simple scheme previously described. The method described here is for a ternary pulse-rebalanced rate-integrating gyro.

The gyro equation of motion is

$$I\ddot{\theta}_f + C\dot{\theta}_f = K_1 M + H\omega_{IA} \quad , \tag{10}$$

where I is the gyro float moment of inertia, θ_f is the gyro float angle, M is the rebalance torque ($M = H\omega_{max}$), H is the gyro wheel angular momentum, ω_{IA} is gyro input rate, C is viscous friction, and

$$K_1 = 0 \quad \text{if} \quad -\theta_T < \theta_f < \theta_T \quad ,$$
$$K_1 = -1 \quad \text{if} \quad \theta_f \leq -\theta_T \quad ,$$
$$K_1 = 1 \quad \text{if} \quad \theta_f \geq \theta_T \quad ,$$

for θ_T, the gyro threshold angle.

In block diagram form, this gyro model appears as in Fig. 5. Figure 5 shows that the rebalancing torque, $K_1 M = 0, \pm M$, is clock-regulated. The torque quanta are therefore applied only at the clock interval. The threshold, θ_T, is variable and is determined by Eq. 11.

$$\theta_T = r' \frac{H}{C} \Delta\theta' \tag{11}$$

r' is a calibration constant dependent on the gyro. Depending on the state of the threshold detecting device, the precision pulse generator sends a pulse to the direction cosine computer scaled at

$$\Delta\theta' = 0, \pm 2^{-n} \text{ (radians)} \quad .$$

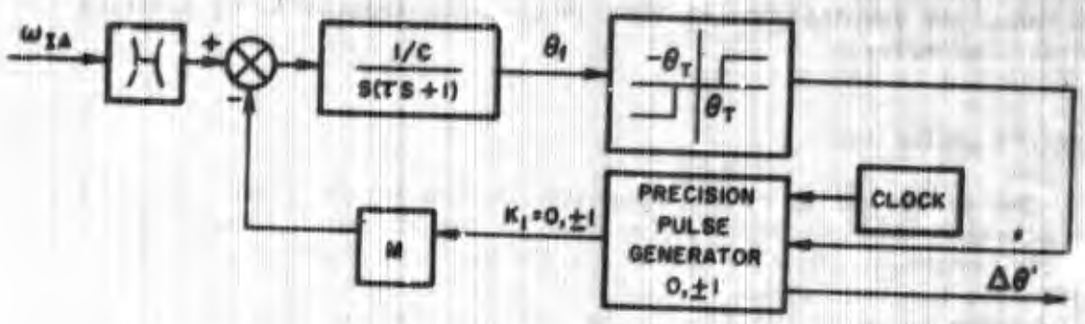


FIG. 5. Ternary Pulse-Rebalanced Gyro Block Diagram.

Integrating Eq. 10,

$$\omega_f(t) = \omega_{o,f} e^{-t/\tau} + R\left(1 - e^{-t/\tau}\right) \quad , \tag{12}$$

where

$$R = \frac{M}{C} \left|\omega_{IA} - K_1 \omega_{max}\right| \quad , \quad \tau = I/C \quad .$$

Writing R in this manner assumes that

$$M = K\omega_{max} \quad .$$

Since

$$\frac{d\theta_f}{dt} = \omega_f(t) \quad ,$$

integration of Eq. 12 gives

$$\theta_f = \theta_{o,f} + \int \omega_f \, dt = \theta_{o,f} + \tau\omega_o\left(1 - e^{-t/\tau}\right) + Rt - R\tau\left(1 - e^{-t/\tau}\right) \quad . \tag{13}$$

Note that the form of Eq. 10 is identical to that of the pulse-balanced pendulum: accelerometer. If appropriate changes of symbols are made, the developments of this section are applicable to modeling the accelerometer.

COMPUTER ARITHMETIC

The mechanization of the computer carried in the vehicle to perform the transformation of coordinates affects the choice of transformation updating methods. In this report, a digital computer using binary arithmetic has been assumed. Multiplication of a number by any negative or positive power of two results in shifting the number to the left or to the right by the number of places determined by the power. For example, $5 \cdot 2^2 = 20$, in binary becomes $101 \cdot 100 = 10100$, or simply the binary representation of 5 shifted two places to the left. Similarly, $5 \cdot 2^{-2} = 5/4$ becomes binary 101 shifted right two places or 1.01.

If the characteristics of the vehicle are sufficiently well known so that the maximum angular rate on any axis, ω_{max}, can be determined, then the sampling interval Δt can be taken so that the maximum turning rate produces exactly one $\Delta\theta$ pulse each sampling time, i.e.,

$$\Delta\theta = \int_{t}^{t+\Delta t} \omega_{max}\, dt \quad .$$

The number $\Delta\theta$ is small and lies between two successive binary fractions

$$2^{-(n+1)} \leq \Delta\theta \leq 2^{-n} \quad .$$

If $\Delta\theta$ is approximated by 2^{-n} for $\Delta\theta > 2^{-(n+1)}$ and by $h = 2^{-n}$ for $\Delta\theta = 2^{-(n+1)}$, then

$$h = h_i \cong \Delta\theta_i \quad , \quad i = 1, 2, 3 \quad , \tag{14}$$

with $i = 1,2,3$ representing the three input axes numbered $1,2,3$. The pulsed output of the gyro is now a binary number; as a result, arithmetic

involving h can be performed by shifting and adding. For example, it may be required to perform

$$h_1 C_{i2} \tag{15}$$

where C_{i2} is some direction cosine. The resulting product would be binary C_{i2} shifted n binary places right. Furthermore, one form of an updating scheme may be preferred to another because it has a smaller count of shift and add operations, or in some way makes the binary computations easier.

NOTATIONS, DEFINITIONS, AND PROBLEM STATEMENT

A few general notations, symbols, and definitions particular to this paper are discussed. Others will arise and are discussed as they are used. A list of symbols is given at the beginning of the report.

Difference Equation Notations (Ref. 25, 22)

In this report, the problem of obtaining numerical solutions of differential equations will arise. To obtain these numerical solutions, we will have to draw for the most part on what are called linear ordinary *difference* equations. An ordinary difference equation of order r can be defined by

$$F\left(x_k, y_k, y_{k+1}, y_{k+2}, \ldots\ldots, y_{k+r}\right) = 0$$

$$(k = 0, \pm 1, \pm 2, \ldots\ldots) \tag{16}$$

Equation 16 relates values $y_k = y(x_k) = y(x_o + k\Delta x)$ of the function $y = y(x)$ defined on a discrete set of values $x = x_k = (x_o + k\Delta x)$, where Δx is a fixed increment. Examples of several difference equations follow:

$$(a) \quad y_{k+1} = a_1 y_k + a_2 x_k \tag{17}$$

$$(b) \quad y_{k+1} = y_k + y_{k-1} + x_k \tag{18}$$

$$(c) \quad \frac{d^2u(t)}{dt^2} - \frac{du(t-1)}{dt} + u(t) = 0 \quad . \tag{19}$$

The first two equations are ordinary difference equations with y_{k+1} determined as a function of discrete past values of y and x. Equation 19 is also a difference equation, but it contains derivatives and is referred to as a *differential-difference* equation.

Later in this report there is a section devoted to the determination of the solutions to difference equations. Further discussion of methods of solution will therefore be deferred until that section is reached.

Matrix Notations

This report deals primarily with sets of difference equations that may be written and operated on according to rules of ordinary matrix theory.

The first type of matrix notation discussed was introduced earlier in Eq. 1 and is the vector notation (actually a 3 x 1 matrix):

$$\begin{bmatrix} \delta_1 \\ \delta_2 \\ \delta_3 \end{bmatrix} = \underline{\delta} \quad . \tag{20}$$

In this equation, $\underline{\delta}$ is used to denote an arbitrary vector that may be projected onto an orthogonal right-handed triad with components δ_1, δ_2, δ_3. (Note that the underscore is used to indicate that $\underline{\delta}$ is a vector.) The length of the vector is given by

$$|\underline{\delta}| = \sqrt{\delta_1^2 + \delta_2^2 + \delta_3^2} \quad . \tag{21}$$

and the direction of the vector is specified by

$$\frac{\delta_1}{|\underline{\delta}|} = \cos\alpha; \quad \frac{\delta_2}{|\underline{\delta}|} = \cos\beta; \quad \frac{\delta_3}{|\underline{\delta}|} = \cos\gamma \quad . \tag{22}$$

Earlier, the expression relating two vectors, measured in different reference frames to each other, was written as $\underline{a}_C = \mathfrak{I}(\underline{a}_B)$. The transformation operation in this report, the matrix of direction cosines, is written as

$$\underline{a}_C = \mathfrak{I}(\underline{a}_B) = [C]\,\underline{a}_B \quad . \tag{23}$$

The matrix $[C]$ is the direction cosine matrix. In component form,

$$a_{i,C} = \sum_{j=1}^{3} C_{ij} \cdot a_{j,B} \quad , \qquad \text{for } i = 1, 2, 3 \quad . \tag{24}$$

C_{ij} is the element in the $i^{\underline{th}}$ row, $j^{\underline{th}}$ column of the matrix $[C]$, and denotes the direction cosine that relates the $i^{\underline{th}}$ axis of system C to the $j^{\underline{th}}$ axis of B. The notation $[C]$ in Eq. 24 implies the 3 x 3 array:

$$[C] = \begin{bmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{bmatrix} \quad .$$

When dealing with the difference equations, two notations are employed.

$$\left[C_{k+1}\right] = \left[C_k\right] \left[H\right] \tag{25}$$

$$\begin{bmatrix} C_{11} \\ C_{21} \\ C_{31} \end{bmatrix}_{k+1} = \begin{bmatrix} C_{11} \\ C_{21} \\ C_{31} \end{bmatrix}_k h_{11} + \begin{bmatrix} C_{12} \\ C_{22} \\ C_{32} \end{bmatrix}_k h_{21} + \begin{bmatrix} C_{13} \\ C_{23} \\ C_{33} \end{bmatrix}_k h_{31} \quad . \tag{26}$$

Equation 25 is a matrix equation representing the relationship between the direction cosine matrix at some arbitrary time k to the direction cosine matrix at time $k + 1$, given the matrix $[H]$. The subscripts of Eq. 25 denote the fixed time interval $(k, k + 1)$. The operation, Eq. 25, will therefore be referred to as *updating*, i.e., $[C_k]$ is *updated* to $\left[C_{k+1}\right]$ by post-multiplying by the matrix $[H]$. The notation of Eq. 26 represents the updating of a column of $[C]$; in this case, the first column. Notice that Eq. 26 is simply the operation that actually would be carried out given the components of $\left[C_k\right]$ and $[H]$ in Eq. 25.

A Rotation Defined in Terms of Direction Cosines

With the aid of Fig. 6 , the individual components of the direction cosine matrix are obtained by forming the dot product

$$C_{ij} = \underline{1}_{i,C} \bullet \underline{1}_{j,B} \text{ , for } i, j = 1, 2, 3 \quad . \tag{27}$$

$\underline{1}_{i,C}$ is the unit vector measured in the computational reference frame with direction along the $\underline{i^{th}}$ reference axis. $\underline{1}_{j,B}$ is a unit vector measured in the body frame with direction along the $\underline{j^{th}}$ body axis. The subscript C denotes computation frame and the subscript B denotes body frame.

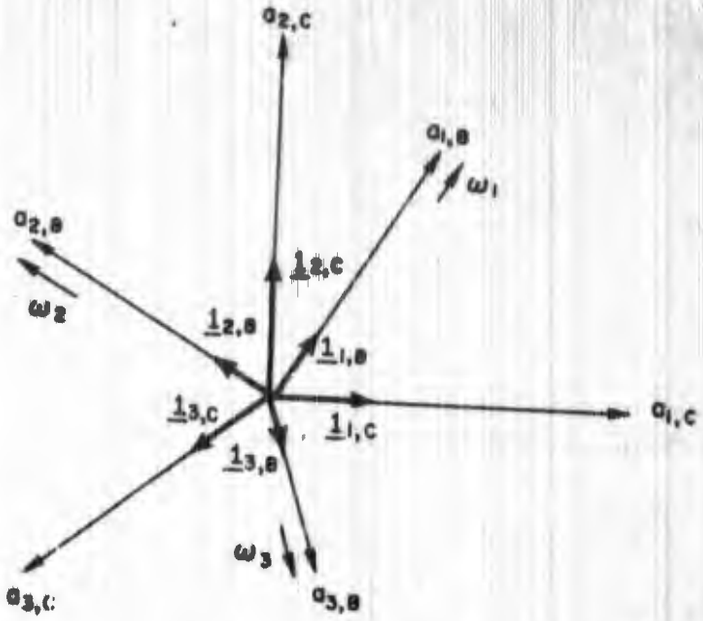


FIG. 6. Axis Sets.

Example 1

Consider as an example, the single-axis rotation described in Fig. 7. Equation 23, $\underline{a}_C = [C]\ \underline{a}_B$ states the relationship between the body-fixed components $a_{1,B}; a_{2,B}; a_{3,B}$ and the computational coordinate set $a_{1,C}; a_{2,C}; a_{3,C}$. The individual components of the matrix $[C]$ are defined by Eq. 27.

$$C_{11} = \underline{1}_{1,C} \cdot \underline{1}_{1,B} = \cos\theta$$

$$C_{12} = \underline{1}_{1,C} \cdot \underline{1}_{2,B} = -\sin\theta$$

$$C_{13} = \underline{1}_{1,C} \cdot \underline{1}_{3,B} = 0$$

$$C_{21} = \sin$$

$$C_{22} = \cos$$

$$C_{23} = 0$$

$$C_{31} = 0$$

$$C_{32} = 0$$

$$C_{33} = 1$$

Inserting these elements into Eq. 24, and writing the vectors $\underline{a}_C$ and $\underline{a}_B$ in component form, we have for the elements of $[C]$.

$$\begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}_C = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}_B$$

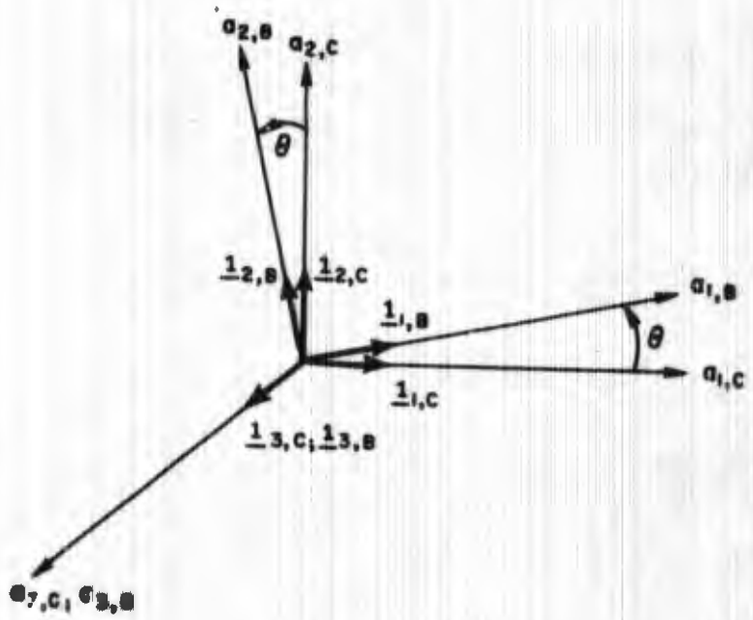


FIG. 7. Single-Axis Rotation.

Direction Cosine Differential Equation (Ref. 30)

The coordinate axes for the body system, B, are denoted $(1, B)$, $(2, B)$, $(3, B)$ and the computer system (preferred or inertial system), C, axes are denoted $(1, C)$, $(2, C)$, and $(3, C)$. Assume that the body system is rotating relative to the computer system with angular velocity $\underline{\omega}$. Writing θ_{11} for the angle measured from $(1, C)$ to $(1, B)$ (Fig. 8),

$$\frac{\dot{\underline{\theta}}_{11}}{\dot{\theta}_{11}} = \underline{1}_{1,C} \times \underline{1}_{1,B} / \sin \theta_{11} \quad , \tag{28}$$

where $\dot{\theta}_{11} \triangleq \frac{d}{dt} \theta_{11}$, and the underscore, _, indicates a vector quantity.

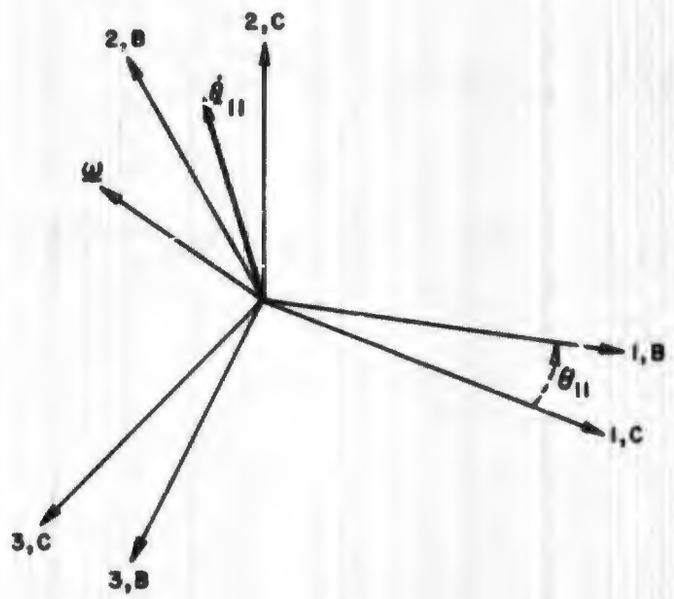


FIG. 8. Angular Velocity Component Caused by the Rate of Change of a Direction Cosine.

$\dot{\theta}_{11}$ is the component of $\underline{\omega}$ along $(\underline{1}_{1,C} \times \underline{1}_{1,B})$; therefore,

$$\dot{\theta}_{11} = \omega \bullet \frac{\dot{\theta}_{11}}{\dot{\theta}_{11}} = \underline{\omega} \bullet \frac{\underline{1}_{1,C} \times \underline{1}_{1,B}}{\sin \theta_{11}} \tag{29}$$

where $\dot{\theta}_{11}/\dot{\theta}_{11}$ is a unit vector in the direction $(\underline{1}_{1,C} \times \underline{1}_{1,B})$. Thus

$$\dot{\theta}_{11} = \frac{\left(\omega_{1,B}\, \underline{1}_{1,B} + \omega_{2,B}\, \underline{1}_{2,B} + \omega_{3,B}\, \underline{1}_{3,B}\right) \bullet \left(\underline{1}_{1,C} \times \underline{1}_{1,B}\right)}{\sin \theta_{11}}$$

which, upon writing $\underline{1}_{1,C}$ in system B, becomes

$$\theta_{11} = \frac{\left[\left(\omega_{1,B}\, \underline{1}_{1,B} + \omega_{2,B}\, \underline{1}_{2,B} + \omega_{3,B}\, \underline{1}_{3,B}\right) \bullet \right.}{\sin \theta_{11}}$$

$$\left.\left(C_{11}\, \underline{1}_{1,B} + C_{12}\, \underline{1}_{2,B} + C_{13}\, \underline{1}_{3,B} \quad \underline{1}_{1,B}\right)\right]$$

Multiplying through by $\sin \theta_{11}$ and simplifying,

$$\sin \theta_{11} \cdot \dot{\theta}_{11} = \left(\omega_{2,B} + \omega_{3,B}\, \underline{1}_{3,B}\right) \cdot \left(C_{12}\left(-\underline{1}_{3,B}\right) + C_{13}\, \underline{1}_{2,B}\right) \quad . \tag{29a}$$

But

$$C_{11} = \cos \theta_{11}$$

$$\frac{d}{dt} C_{11} = -\sin \theta_{11}\, \dot{\theta}_{11} \quad . \tag{30}$$

Combining Eq. 29a and 30,

$$\frac{d}{dt} C_{11} = \left(\omega_{2,B}\, \underline{1}_{2,B} + \omega_{3,B}\, \underline{1}_{3,B}\right) \cdot \left(C_{13}\, \underline{1}_{2,B} - C_{12}\, \underline{1}_{3,B}\right)$$

$$= C_{12}\, \omega_{3,B} - C_{13}\, \omega_{2,B} \quad . \tag{30a}$$

Proceeding in a similar manner, we can obtain the derivatives of all nine direction cosines in terms of the components of ω in system B. Dropping the "B" notation, we have

$$\left.\begin{aligned} \frac{d}{dt} C_{i1} &= C_{i2}\, \omega_3 - C_{i3}\, \omega_2 \\ \frac{d}{dt} C_{i2} &= C_{i3}\, \omega_1 - C_{i1}\, \omega_3 \\ \frac{d}{dt} C_{i3} &= C_{i1}\, \omega_2 - C_{i2}\, \omega_1 \end{aligned}\right\} \text{ for } i = 1,2,3 \quad .$$

In matrix notation, we have for Eq. 30b

$$\frac{d}{dt}[C] = [C][\Omega] \quad . \tag{30c}$$

$[\Omega]$ is the skew symmetric matrix of body angular velocities

$$[\Omega] = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix} \quad . \tag{31}$$

PROBLEM STATEMENT

At this point, there is sufficient material on hand to make a more definite statement of the entire problem as it will be dealt with in the remainder of this report. The goal is to find a suitable means for integrating the equation set (Eq. 30b). At best, only a numerical approximation to this integration can be performed because (1) the angular rate information will be processed in a computer capable of operating only in a discrete numerical fashion, and (2) the digital transformation computer requires that the gyro-output angular-rate information be quantized. As a final problem statement, perform a numerical integration of Eq. 30b given quantized angular rate information. The solution (or integration) of Eq. 30b will then yield a transformation matrix that will correctly relate body to computation axes as a function of time (discrete function of time).

ALGORITHMS FOR UPDATING THE DIRECTION COSINE MATRIX

DEFINITION OF THE TERM ALGORITHM

Algorithm is a word derived from the word algorism, which means the act of computing with Arabic numerals. The term algorithm is now used to denote any method of computation consisting of comparatively small steps; the steps are taken in a preassigned order and usually involve iteration.

CONVENTIONAL NUMERICAL INTEGRATION SCHEMES

For the direction cosine differential equation (Eq. 17), simply say that

$$\frac{d[C]}{dt} = f(t, [C]) \quad . \tag{32}$$

The following symbols are used

$$\left|C_k\right| = \left|C(t_k)\right| ; \frac{d\left|C_k\right|}{dt} = \left.\frac{d[C(t)]}{dt}\right|_{t = t_k} \quad . \tag{33}$$

Where t_k is the value of t at the k^{th} step in the integration and $t_{k+1} - t_k = \Delta t$. For first-order equations represented by Eq. 32, there are basically two general types of integration algorithms:

(1) Forward integration formulas in which $\left|C_{n+1}\right|$ is expressed as a linear combination of $[C_k]$ and $d[C_k]/dt$ for $k \leq n$;

(2) Iterative formulas in which $[C_{n+1}]$ is expressed as a linear combination of $d[C_{n+1}]/dt$ and $[C_k]$ and $dC/dt\ k$ for $k \leq n$.

Almost any standard text on numerical analysis gives the more common integration algorithms (Ref. 21, 22, 24, and 28).

RECTANGULAR INTEGRATION

Rectangular integration is the simplest of forward integration formulas. Beginning with the differential equation

$$\frac{d}{dt}[C] = [C]\,[\Omega] \quad ,$$

We form the integral

$$\int_{t_o}^{t_o + \Delta t} d[C] = \int_{t_o}^{t_o + \Delta t} [C][\Omega]\, dt \quad , \tag{34}$$

where $\Delta t = t_1 - t_o$ and is small. From calculus, a first approximation to an integral is given by

$$\int_a^b f(\zeta)\, d\zeta \cong f(a)\,(b-a) \quad . \tag{35}$$

The reason for calling this approximation rectangular integration is obvious from Eq. 35. Suppose $f(\zeta)$ were some curve as shown in Fig. 9. The shaded rectangular area in the figure is a rectangular approximation to the area under the $f(\zeta)$-curve from $\zeta=a$ to $\zeta=b$. Thus for Eq. 34,

$$[C] - [C_o] = \int_{t_o}^{t_o + \Delta t} d[C]$$

$$= \int_{t_o}^{t_o + \Delta t} [C][\Omega]\, dt = [C_o][\Omega_o]\, \Delta t \quad . \tag{36}$$

So, given the $[C]$ matrix at time t_o and the angular rates at time t_o, Eq. 36 then determines $[C]$ at time t_1. And in general, for an interval (t_k, t_{k+1}),

$$[C_{k+1}] = [C_k][\Omega_k]\, \Delta t + [C_k] \quad . \tag{37}$$

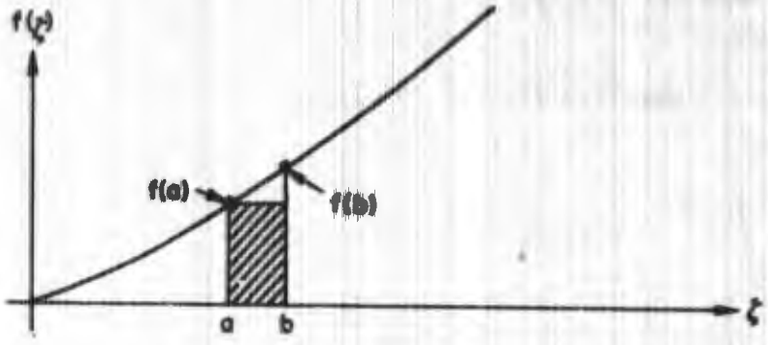


FIG. 9. $f(\zeta)$-Curve (Example Only).

The component equation of Eq. 37 may be written as

$$\begin{bmatrix} C_{11,k+1} & C_{12,k+1} & C_{13,k+1} \\ C_{21,k+1} & C_{22,k+1} & C_{23,k+1} \\ C_{31,k+1} & C_{32,k+1} & C_{33,k+1} \end{bmatrix} = \begin{bmatrix} C_{11,k} & C_{12,k} & C_{13,k} \\ C_{21,k} & C_{22,k} & C_{23,k} \\ C_{31,k} & C_{32,k} & C_{33,k} \end{bmatrix} \cdot \begin{bmatrix} 0 & -\omega_3(t_k) & \omega_2(t_k) \\ \omega_3(t_k) & 0 & -\omega_1(t_k) \\ -\omega_2(t_k) & \omega_1(t_k) & 0 \end{bmatrix} \Delta t + |C_k| \; . \tag{38}$$

Then, writing in columns,

$$\left.\begin{aligned} C_{i1,k+1} &= C_{i2,k}\,\omega_3(t_k)\,\Delta t - C_{i3,k}\,\omega_2(t_k)\,\Delta t + C_{i1,k} \\ C_{i2,k+1} &= C_{i3,k}\,\omega_1(t_k)\,\Delta t - C_{i1,k}\,\omega_3(t_k)\,\Delta t + C_{i2,k} \\ C_{i3,k+1} &= C_{i1,k}\,\omega_2(t_k)\,\Delta t - C_{i2,k}\,\omega_1(t_k)\,\Delta t + C_{i3,k} \end{aligned}\right\} \text{for } i = 1,\ 2,\ 3 \; . \tag{39}$$

In Eq. 39, note the recurrence of the terms

$$\omega_i(t_k)\ \Delta t \ , \ for\ i = 1, 2, 3 \quad .$$

Refer to Eq. 4, rewritten for the i^{th} gyro

$$\Delta\theta_i = \int_t^{t+\Delta t} \omega_i(t)dt \ , \ for\ i = 1, 2, 3 \quad .$$

If a rectangular integration of this integral is performed,

$$\Delta\theta_i \cong \omega_i(t_k)\ \Delta t \ , \ for\ i = 1, 2, 3 \quad . \tag{40}$$

For the case where $\omega_i\ (t_k) = \omega_i = constant$,

$$\Delta\theta_i = \omega_i\ \Delta t \ , \ for\ i = 1, 2, 3 \quad . \tag{41}$$

Substituting Eq. 40 or Eq. 41 into Eq. 39 gives the set

$$\left.\begin{aligned} C_{i1,k+1} &= C_{i2,k}\ \Delta\theta_3 - C_{i3,k}\ \Delta\theta_2 + C_{i1,k} \\ C_{i2,k+1} &= C_{i3,k}\ \Delta\theta_1 - C_{i1,k}\ \Delta\theta_3 + C_{i2,k} \\ C_{i3,k+1} &= C_{i1,k}\ \Delta\theta_2 - C_{i2,k}\ \Delta\theta_1 + C_{i3,k} \end{aligned}\right\} for\ i = 1, 2, 3 \quad . \tag{42}$$

Recall the material presented in the section on gyro models and study Fig. 5. The gyro output is mechanized such that it provides a fixed, clock-regulated pulse scaled at 2^{-n} radians whenever the gyro float position exceeded the threshold angle.

$$\Delta\theta' = 0, \pm 2^{-n} \quad .$$

The size of this fixed angular increment is related to a given maximum angular rate by the expression

$$\Delta\theta' = \omega_{max} T_g \quad , \tag{43}$$

wherein T_g is the gyro clock period. For specific $\Delta\theta'$ and T_g, ω_{max} is the maximum constant angular rate allowable over the interval T_g so that all the angular information is contained in the increment $?^{-n}$. For example, if $n=12$ and $T_g=(1/2000)$ sec, $\omega_{max}=0.48828125$ rad/sec. For any given system mechanization, an attempt is made to determine what the maximum input angular rate will be. Then an attempt is made to choose a quantitization level and clock period so that Eq. 43 is satisfied. For any larger constant rate to occur over that interval, because of a wrong ω_{max} prediction, some amount of angular information would be lost or stored in the gyro.

$$\Delta\theta' + \theta_\varepsilon = \left(\omega_{max} + \Delta\omega_\varepsilon\right) T_g \tag{44}$$

For a rate ω less than ω_{max}, Eq. 43 is replaced with

$$\Delta\theta' + \theta_\varepsilon = \omega \cdot m T_g \quad , \quad \omega < \omega_{max} \quad , \tag{45}$$

where m clock periods are required to make the gyro float position exceed the threshold. For the sum of clock periods less than m, the gyro output will be zero. Note also, the addition of the storage error term $\Delta\theta_\varepsilon$ in Eq. 44 and Eq. 45. For angular rates that are non-terminating decimal fractions of ω_{max}, θ_ε will be non-zero over all time. Terminating decimal fraction rates will permit $\theta_\varepsilon=0$ for multiples of that fraction.

Thus, the gyro is supplying $\Delta\theta$'s, particular fixed value $\Delta\theta'$s. Since these increments are fixed, denote

$$\Delta\theta'_i = h_i \quad , \quad i = 1, 2, 3 \quad . \tag{46}$$

If the rate is constant and maximum, then

$$h_i = \Delta\theta_i = \int_0^{T_g} (t)\, \omega_{i,\,max}\, dt = \pm 2^{-n} \quad , \tag{47}$$

and, for less than maximum rates,

$$h_i = \Delta\theta_i = \int_0^m T_s \omega_i \, dt - \theta_t$$

$$h_i \Delta o = \int_0^{T_s} \omega_i \, dt \quad , \quad \omega_i < \omega_{i,\,max} \quad . \tag{48}$$

Finally, if we mechanize the gyro output to be scaled at $h_i = \pm 2^{-n}, 0;\ i = 1, 2, 3$, we use these pulses as angular rate information inputs to a digital computer. In this computer we have mechanized the Eq. 42, which now appears as

$$\left.\begin{aligned} C_{i1,k+1} &= C_{i2,k}\, h_3 - C_{i3,k}\, h_2 + C_{i1,k} \\ C_{i2,k+1} &= C_{i3,k}\, h_1 - C_{i1,k}\, h_3 + C_{i2,k} \\ C_{i3,k+1} &= C_{i1,k}\, h_2 - C_{i2,k}\, h_1 + C_{i3,k} \end{aligned}\right\} \text{for } i = 1, 2, 3 \quad , \tag{49}$$

or in the matrix form

$$[C_{k+1}] = [C_k]\,[H] + [C_k] = [C_k]\,([I] + [H]) \quad , \tag{50}$$

where

$$[H] = \begin{bmatrix} 0 & -h_3 & h_2 \\ h_3 & 0 & -h_1 \\ -h_2 & h_1 & 0 \end{bmatrix}$$

and $[I]$ is the unit matrix.

Equation 50 is the final form of Eq. 37, where, in general,

$$[H] = [\Omega]\ \Delta t = [\Omega]\ T_s \quad , \tag{51}$$

and where $[\Omega]$ is constant over T_s and the elements of $[H]$ each take on one of 3 values: $\pm 2^{-n}, 0$. Equation 50 is a *rectangular integration algorithm*.

The arguments presented above are valid for any rate that is constant, at least over the clock interval, T_s. For nonconstant rates, if T_s is made sufficiently small, the assumption of constant rates is generally adequate, depending on overall accuracy requirements. In either case, we are faced with the problem of the discretization of a continuous signal and are forced to deal with quantitization errors. For the purposes of this paper, it shall be assumed that through judicious equipment design the assumption of constant rates over the clock interval is valid.

TRAPEZOIDAL INTEGRATION ALGORITHM

Beginning with the direction cosine differential equation $\frac{d}{dt}[C] = [C][\Omega]$, write

$$[C(t)] = [C(o)] + \int_o^t [C][\Omega]\ d\tau = [C(o)] + \int_o^t d[C(t)] \quad . \tag{52}$$

If Eq. 52 is true for the interval $(0,t)$ then, in general, it will hold for (t_k, t_{k+1}). Thus

$$\left|C\,(t_{k+1})\right| = \left|C_{k+1}\right| = \left|C_k\right| + \int_{t_k}^{t_{k+1}} d[C(\tau)] \quad . \tag{53}$$

In the trapezoidal integration of an integral, say $\int_{t_k}^{t_{k+1}} x(t)dt$, write for increment of area under the $x(t)$-curve

$$\frac{1}{2}\,(x_k + x_{k+1})\ \Delta t\ ,\ \Delta t = t_{k+1} - t_k \quad . \tag{54}$$

Applying Eq. 54 to the integration of Eq. 53, we have

$$|C_{k+1}| = |C_k| + \frac{1}{2}\left(|C_k| + |C_{k+1}|\right)[H] \quad , \tag{55}$$

where $[H]$ is as above, following Eq. 50.

SIMPSON'S ONE-THIRD RULE

Simpson's one-third rule integration of the direction cosine differential equation is given by

$$C_{k+2} = C_k + \frac{1}{3} C_k [H] + \frac{4}{3} C_{k+1} [H] + \frac{1}{3} C_{k+2} [H] \quad . \tag{56}$$

Inspection of Eq. 55 and 56 shows that these algorithms are iterative.

TAYLOR SERIES

Use of a Taylor series gives rise to algorithms that gain greater accuracy by using more information at a point. That is,

$$|C_{k+1}| = f\left(|C_k|, \frac{d}{dt}|C_k|, \frac{d^2}{dt^2}|C_k|, \frac{d^3}{dt^3}|C_k|, \ldots\right) \quad . \tag{57}$$

Write $|C_{k+1}| = |C(t_k + \Delta t)|$; $|C_k| = |C(t_k)|$. Expanding $|C_{k+1}|$ in a Taylor series, we have,

$$|C_{k+1}| = |C(t_k + \Delta t)| = |C_k| + \frac{d}{dt}|C_k| \Delta t + \frac{d^2}{dt^2}|C_k| \frac{\Delta t^2}{2} + \frac{d^3}{dt^3}|C_k| \frac{\Delta t^3}{6} + \ldots\ldots \quad . \tag{58}$$

Now, by use of the direction cosine equation, $\frac{d}{dt}[C] = [C][\Omega]$, write

$$\frac{d}{dt}[C] = [C][\Omega]$$

$$\frac{d^2}{dt^2}[C] = [C]\left([\Omega]^2 + \frac{d}{dt}[\Omega]\right)$$

$$\frac{d^3}{dt^3}[C] = [C]\left([\Omega]^3 + 3[\Omega]\frac{d}{dt}[\Omega] + \frac{d^2}{dt^2}[\Omega]\right)$$

$$\vdots$$

$$etc. \qquad (59)$$

By substitution, we then have

$$[C_{k+1}] = [C_k]\left([I] + [\Omega]\,\Delta t + [\Omega]^2 + \frac{d}{dt}[\Omega]\,\frac{\Delta t^2}{2} + \ldots\ldots\right). \qquad (60)$$

If we assume that $[\Omega]$ is constant over the interval Δt and that $[H] \cong [\Omega]\,\Delta t$, then

$$[C_{k+1}] = [C_k]\left([I] + [H] + \frac{[H]^2}{2} + \frac{[H]^3}{6} + \ldots\ldots\right). \qquad (61)$$

Now in Eq. 61 let us truncate terms of order two and higher,

$$[C_{k+1}] = [C_k]\left([I] + [H]\right), \qquad (62)$$

which is the same result as that obtained by rectangular integration. Equation 62 is referred to as a *first order algorithm*. If terms of order three and higher are ignored and all derivatives of $[\Omega]$ are negligible,

we obtain a *second order algorithm*, given by

$$[C_{k+1}] = [C_k]\left([I] + [H] + \tfrac{1}{2}[H]^2\right) \quad . \tag{63}$$

Both the first- and second-order algorithms give rise to difference equations of order one and are called *open* formulas.

OTHER SOLUTIONS OF THE DIRECTION COSINE DIFFERENTIAL EQUATION

The Matricant

We now consider several solutions of $\frac{d}{dt}[C] = [C][\Omega]$, which are of general interest. The first of these solutions is a series solution. This solution is effected by integration and successive substitutions. As in Eq. 52,

$$[C(t)] = [C(o)] + \int_o^t [C][\Omega]\, d\tau \quad .$$

But for $[C]$ under the integral, we may substitute the integrated expression itself:

$$[C(t)] = [C(o)] + \int_o^t \left([C(o)] + \int_o^\tau [C][\Omega]\, d\tau'\right)[\Omega]\, d\tau \quad . \tag{64}$$

Substituting again

$$[C(t)] = [C(o)] + \int_o^t \left\{ [C(o)] + \int_o^\tau \left([C(o)] + \int_o^{\tau'} [C][\Omega]\, d\tau'' \right) [\Omega]\, d\tau' \right\} [\Omega]\, d\tau \quad . \tag{65}$$

These successive substitutions may be continued indefinitely so that we have an infinite series, which, upon rearranging, has the form

$$[C(t)] = [C(o)] \left[[I] + \int_o^t [\Omega]\, d\tau + \int_o^t \left(\int_o^\tau [\Omega]\, d\tau' \right) [\Omega]\, d\tau + \ldots\ldots \right] . \tag{66}$$

Gantmacher calls this series a Matricant (Ref 18).

Before we proceed to make any use of Eq. 66, let us recall several specific points regarding the quantized gyro outputs, our source of angular rate information.

(1) For a given gyro clock rate and specified maximum angular rate, $h_i = \pm 2^{-n}$, 0 (RAD) is the scaled gyro pulse. $h_i = \int_o^{T_s} \omega_{i,\,max} dt$ (Eq. 47).

(2) For $\omega < \omega_{max}$, $h_i = -\theta_\epsilon + \int_o^{MT_s} \omega_i\, dt$, with θ_ϵ the storage error. Alternatively, $h_i \triangleq 0$ at the gyro output if $\int_o^{MT_s} \omega_i\, dt \neq \int_o^{T_s} \omega_{i,\,max} dt$; $h_i = \pm 2^{-n}$, if $\int_o^{MT_s} \omega_i\, dt = \int_o^{T_s} \omega_{i,\,max} dt$, with $\omega_i < \omega_{i,\,max}$.

Then in Eq. 66, if the elements [Ω] are constant over the interval $(0, T_s)$, we can write for the second term inside the large brackets

$$\int_o^t [\Omega]\, d\tau = [H] \quad ,$$

with each h_i in $[H]$ being subject to points (1) and (2) above. Of course, if $[\Omega]$ is constant over the interval, then

$$\int_o^t [\Omega]\, d\tau = [\Omega] \int_o^t d\tau = [\Omega]\, \tau \Big|_o^t \quad ,$$

as was stated in Eq. 51. For the third term of Eq. 66, we have, in component form,

$$\int_o^t \left(\int_o^t [\Omega]\, d\tau' \right) [\Omega]\, d\tau \triangleq \begin{bmatrix} \eta_{11} & \eta_{12} & \eta_{13} \\ \eta_{21} & \eta_{22} & \eta_{23} \\ \eta_{31} & \eta_{32} & \eta_{33} \end{bmatrix} \quad , \tag{67}$$

where the elements η_{ij} are

$$\eta_{11} = -\int_o^t \omega_3 \int_o^\tau \omega_3\, d\tau'\, d\tau - \int_o^t \omega_2 \int_o^\tau \omega_2\, d\tau'\, d\tau$$

$$\eta_{22} = -\int_o^t \omega_1 \int_o^\tau \omega_1\, d\tau'\, d\tau - \int_o^t \omega_3 \int_o^t \omega_3\, d\tau'\, d\tau$$

$$\eta_{33} = -\int_o^t \omega_2 \int_o^\tau \omega_2\, d\tau'\, d\tau - \int_o^t \omega_1 \int_o^\tau \omega_1\, d\tau'\, d\tau$$

$$\eta_{ij}\Big|_{i \neq j} = \int_o^t \omega_j \int_o^\tau \omega_i\, d\tau'\, d\tau \quad . \tag{68}$$

Making the assumption of constant $[\Omega]$, as before, allows

$$\int_0^t \left(\int_0^\tau [\Omega]\, d\tau' \right) [\Omega]\, d\tau = [\Omega]^2 \int_0^t \left[\int_0^\tau d\tau' \right] d\tau = \left. \frac{[\Omega]^2\, \tau^2}{2} \right|_0^t \quad . \tag{69}$$

And, as before, introducing the $[H]$ matrix,

$$\left. \frac{[\Omega]^2\, \tau^2}{2} \right|_0^t = [H]^2/2 \quad , \tag{70}$$

If we truncate the series (Eq. 66) after this second integration, we have

$$[C(t)] = [C(o)] \left([I] + [H] + [H]^2/2 \right) \quad . \tag{71}$$

For the general interval $(t_k,\ t_{k+1})$,

$$[C_{k+1}] = [C_k] \left([I] + [H] + [H]^2/2 \right) \quad , \tag{72}$$

which is the second-order algorithm of the Taylor series.

The Matricant for Nonconstant $[\Omega]$ (Ref. 5)

For an $[\Omega]$ matrix that is nonconstant, we are able to draw on some theorems of matrix theory and ordinary algebra that will make the matricant (Eq. 66), under some circumstances, more amenable to the simplified digital operations of SHIFT and ADD. In Eq. 66, denote

$$\int_0^t [\Omega]\, d\tau = [U(t)] = [U] \quad . \tag{73}$$

Consider the second-order term,

$$\int_0^t \left(\int_0^\tau [\Omega]\, d\tau' \right) [\Omega]\, d\tau \quad .$$

Assume that we have the special condition where ω_i, $i = 1, 2, 3$ is nonconstant, but that

$$\frac{\omega_3}{\omega_2} = k_{32} \; ; \; \frac{\omega_2}{\omega_1} = k_{21} \quad . \tag{74}$$

In the second-order term of Eq. 66, let us look at the elements along the main diagonal, such as

$$\int_0^t \omega_3 \int_0^\tau \omega_3\, d\tau'\, d\tau = f \quad . \tag{75}$$

As before, integration by parts yields

$$f = \frac{1}{2} \left(\int_0^t \omega_3\, d\tau \right)^2 \quad .$$

For the off-diagonal elements, we have, for example

$$\int_0^t \omega_2 \int_0^\tau \omega_1\, d\tau'\, d\tau = k_{21} \int_0^t \omega_1 \int_0^\tau \omega_1\, d\tau'\, d\tau = \frac{1}{2} k_{21} \left(\int_0^t \omega_1\, d\tau \right)^2$$

$$= \frac{1}{2} \left(\int_0^t \omega_2\, d\tau \right) \left(\int_0^t \omega_1\, d\tau \right) \quad . \tag{76}$$

Applying this scheme to all elements of the second-order term of Eq. 66 yields, under the assumption of angular velocity proportionality,

$$\int_0^t \left(\int_0^\tau [\Omega]\, d\tau' \right) [\Omega]\, d\tau = \frac{1}{2} \left(\int_0^t [\Omega]\, d\tau \right)^2 \tag{77}$$

Then, according to Eq. 73,

$$\frac{1}{2} \left(\int_0^t [\Omega]\, d\tau \right)^2 = \left[U^2 \right] / 2 \quad . \tag{78}$$

And, by induction

$$\left[\frac{U^3}{6} \right] = \int_0^t \left(\int_0^\tau \left(\int_0^{\tau'} [\Omega]\, d\tau'' \right) [\Omega]\, d\tau' \right) [\Omega]\, d\tau \quad , \tag{79}$$

$$\vdots$$

$$\left[\frac{U^n}{n!} \right] = \int_0^t \left(\int_0^\tau \left(\int_0^{\tau''} \left(\int_0^{\tau'''} \left(\cdots \left(\int_0^{\tau^{(n)}} [\Omega]\, d\tau^{(n+1)} \right) [\Omega]\, d\tau^{(n)} \right) \cdots \right) [\Omega]\, d\tau \right. \right. \tag{80}$$

Equation 66 can then be written as

$$[C(t)] = [C(o)] \cdot \left([I] + [U] + \left[U^2 \right] / 2 + \left[U^3 \right] / 6 + \cdots \right) \quad , \tag{81}$$

or

$$[C(t)] = [C(o)]\, e^{[U]} \tag{82}$$

where $[U]$ is the 3 x 3 square matrix

$$[U] = \begin{bmatrix} 0 & -\int_0^t \omega_3\, d\tau & \int_0^t \omega_2\, d\tau \\ \int_0^t \omega_3\, d\tau & 0 & -\int_0^t \omega_1\, d\tau \\ -\int_0^t \omega_2\, d\tau & \int_0^t \omega_1\, d\tau & 0 \end{bmatrix} \tag{83}$$

Proceeding, now write $e^{[U]}$ in the form

$$e^{[U]} = \left(A_0\,[I] + A_1\,[U] + A_2\,[U^2]\right) \quad . \tag{84}$$

where A_0, A_1, A_2 are to be determined. These coefficients can be determined by drawing on the Cayley-Hamilton theorem and a matrix generalization of the remainder theorem of ordinary algebra. The theory behind this method is detailed in Ref. 19 and will not be repeated here. According to the scheme in the reference, we determine the eigenvalues of $[U]$ to be

$$\lambda_1 = 0$$

$$\lambda_{2,\,3} = \pm j\psi \tag{85}$$

where

$$\psi = \sqrt{\left(\int_0^t \omega_1 \, d\tau\right)^2 + \left(\int_0^t \omega_2 \, d\tau\right)^2 + \left(\int_0^t \omega_3 \, d\tau\right)^2} \quad . \tag{86}$$

To determine the constants, we have the three equations

$$\begin{aligned} e^0 &= A_0 + A_1 \cdot 0 + A_2 \cdot 0 \\ e^{j\psi} &= A_0 + A_1 \cdot j\psi + A_2 (j\psi)^2 \\ e^{-j\psi} &= A_0 + A_1 \cdot (-j\psi) + A_2 (-j\psi)^2 \end{aligned} \quad . \tag{87}$$

This yields,

$$\begin{aligned} A_0 &= 1 \\ A_1 &= \sin \psi/\psi \\ A_2 &= (1 - \cos \psi)/\psi^2 \end{aligned} \quad . \tag{88}$$

Then Eq. 82 can be written for the general interval (t_k, t_{k+1}),

$$|C_{k+1}| = |C_k| \left\{ [I] + \frac{\sin \psi}{\psi} \int_{t_k}^{t_{k+1}} [\Omega] \, dt + \frac{1 - \cos \psi}{\psi^2} \left(\int_{t_k}^{t_{k+1}} [\Omega] \, dt \right)^2 \right\} . \tag{89}$$

Note that no terms higher than second order appear. The transcendental properties of $e^{[\Omega]}$ are, however, preserved through the appearance of $\sin\psi$ and $\cos\psi$. Recalling again that

$$\int_{t_k}^{t_{k+1}} [\Omega]\, dt = [H]\,, \qquad \int_{t_k}^{t_{k+1}} \omega_i\, dt = h_i\,,$$

where the elements of $[H]$ are such that $h_i = \pm 2^{-n}, 0$ and $i = 1, 2, 3$. Then Eq. 89 may be written, expanding and collecting terms, as

$$[C_{k+1}] = [C_k]\begin{bmatrix} \left[\frac{h_1^2}{\psi^2} + \frac{\left(h_2^2 + h_3^2\right)}{\psi^2}\cos\psi\right] & \left[\frac{h_3}{\psi}\sin\psi + \frac{h_1h_2}{\psi^2}(1-\cos\psi)\right] & \left[\frac{h_2}{\psi}\sin\psi + \frac{h_1h_3}{\psi^2}(1-\cos\psi)\right] \\ \left[\frac{h_3}{\psi}\sin\psi + \frac{h_1h_2}{\psi^2}(1-\cos\psi)\right] & \left[\frac{h_2^2}{\psi^2} + \frac{\left(h_1^2 + h_3^2\right)}{\psi^2}\cos\psi\right] & \left[\frac{h_1}{\psi}\sin\psi + \frac{h_2h_3}{\psi^2}(1-\cos\psi)\right] \\ \left[\frac{h_2}{\psi}\sin\psi + \frac{h_1h_3}{\psi^2}(1-\cos\psi)\right] & \left[\frac{h_1}{\psi}\sin\psi + \frac{h_2h_3}{\psi^2}(1-\cos\psi)\right] & \left[\frac{h_3^2}{\psi^2} + \frac{\left(h_1^2 + h_2^2\right)}{\psi^2}\cos\psi\right] \end{bmatrix}. \tag{90}$$

Equation 90 is exactly the form that will be used later in the report as a "reference" algorithm. The development here has been with that goal in mind.

As a matter if interest, if we approximate $\cos\psi$ by $1 - \frac{\psi^2}{2}$ and $\sin\psi$ by ψ, Eq. 90 becomes

$$[C_{k+1}] = [C_k]\begin{bmatrix} 1 - \frac{h_2^2}{2} - \frac{h_3^2}{2} & -h_3 + \frac{h_1h_2}{2} & h_2 + \frac{h_1h_3}{2} \\ h_3 + \frac{h_1h_2}{2} & 1 - \frac{h_1^2}{2} - \frac{h_3^2}{2} & -h_1 + \frac{h_2h_3}{2} \\ -h_2 + \frac{h_1h_3}{2} & h_1 + \frac{h_2h_3}{2} & 1 - \frac{h_1^2}{2} - \frac{h_2^2}{2} \end{bmatrix} \tag{91}$$

or

$$[C_{k+1}] = [C_k]\left[[I] + [H] + \frac{[H]^2}{2}\right] , \tag{92}$$

which is the second-order algorithm developed earlier.

HIGHER-ORDER ALGORITHMS

Algorithms of order three and higher can be formed by including more terms in the series expansion that generates the algorithm. The wisdom of such higher-order refinements, however, is subject to question when overall system errors are considered. The higher-order forms are not studied as such in this paper.

DISCUSSION

This section has been devoted to the development of a set of simple direction cosine updating algorithms. Of the most interest are those forms that can be readily mechanized in a digital computer operating only on the principles of SHIFT and ADD. These are the first-order, or rectangular, algorithm and the second-order algorithm. When more sophisticated and complex digital computers are considered, other integration schemes become useful, even to the point of using numerical integration formulas such as fourth-order Runge-Kutta.

For the purposes of this report, we have assumed that the computer will not be of sufficient size or flexibility to allow programming of algorithms other than first- or second-order or simple variations thereof.

ALGORITHM FORM VARIATIONS

Now that the basic forms of desirable algorithms have been introduced, several variations on these forms will now be considered. To a certain extent, the form of the algorithm variation is directly related to the computer mechanization used.

PARALLEL-PROCESSING ALGORITHM

If we hypothesize that the direction cosine computer is capable of accepting the pulse information from the three gyros simultaneously, and can thereafter use this information to update all nine direction cosines simultaneously, the algorithm so employed is usually referred to as a parallel updating method. Under the assumptions given in the previous section, the updating of the direction cosine matrix can be represented by Eq. 49 as

$$C_{i1,\,k+1} = C_{i2,\,k}\,h_3 - C_{i3,\,k}\,h_2 + C_{i1,\,k}$$

$$C_{i2,\,k+1} = C_{i3,\,k}\,h_1 - C_{i1,\,k}\,h_3 + C_{i2,\,k}$$

$$C_{i3,\,k+1} = C_{i1,\,k}\,h_2 - C_{i2,\,k}\,h_1 + C_{i3,\,k}$$

for $i = 1, 2, 3$. This algorithm uses only the past, k^{th}, value of each direction cosine to update the direction cosine matrix to the $(k+1)^{st}$ value.

SERIAL-PROCESSING ALGORITHM

Referring again to Eq. 49, repeated in the preceding paragraph, if upon obtaining $C_{i1,\,k+1}$, we use it instead of $C_{i1,\,k}$ as part of the computation for $C_{i2,\,k+1}$, and subsequently use $C_{i1,\,k+1}$ together with $C_{i2,\,k+1}$ to obtain $C_{i3,\,k+1}$, we are using serial processing.

$$C_{i1,\,k+1} = C_{i2,\,k}\,h_3 - C_{i3,\,k}\,h_2 + C_{i1,\,k}$$

$$C_{i2,\,k+1} = C_{i3,\,k}\,h_1 - C_{i1,\,k+1}\,h_2 + C_{i2,\,k}$$

$$C_{i3,\,k+1} = C_{i1,\,k+1}\,h_2 - C_{i2,\,k+1}\,h_3 + C_{i3,\,k} \qquad (93)$$

for $i = 1, 2, 3$. With the updating equations written in this manner, it is implied that the gyro pulses, h_i, are accepted simultaneously by the computer, but that the actual updating thereafter is sequential.

BUMSTEAD ALGORITHM (Ref. 4 and 15)

The Bumstead Algorithm involves an updating order-reversal to eliminate problems associated with error buildup from vibration (limit-cycle) inputs. Limit-cycle inputs are characterized by pulse trains with the pulses alternating signs in a regular fashion. For example, a 1:1 limit cycle means the pulse train $(+h, -h, +h, -h, +h, \ldots\ldots, etc.)$. The method involves combinations of simple rectangular integration. For simplicity of discussion, we consider only a single-axis rotation here: $h_1 = h_2 = 0$. Suppose a first-order parallel updating of the form

$$\left.\begin{aligned} C_{i1,\,k+1} &= C_{i1,\,k} + h_3\, C_{i2,\,k} \\ C_{i2,\,k+1} &= C_{i2,\,k} - h_3\, C_{i1,\,k} \end{aligned}\right\} \begin{matrix} h_1 = 0 \\ h_2 = 0 \\ h_3 = h_3 \end{matrix} \qquad (94)$$

for $i = 1, 2$. Again, if we use $C_{i1,\,k+1}$ in place of $C_{i1,\,k}$, we have for sequential processing

$$\begin{aligned} C_{i1,\,k+1} &= C_{i1,\,k} + h_3\, C_{i2,\,k} \\ C_{i2,\,k+1} &= C_{i2,\,k} - h_3\, C_{i1,\,k+1} \quad , \end{aligned} \qquad (95)$$

for $i = 1, 2$. As shown in the next section, the <u>parallel</u> processing (Eq. 94) of a monotonic input has an error of approximately $h_3^2/2$ per step. For a limit-cycle condition, the same order of error occurs every computer iteration step. An example of the first-order parallel processing of a 1:1 limit cycle is shown in the following.

$$|C_{k+1}| = |C_k| \begin{bmatrix} 1 & -h_3 \\ h_3 & 1 \end{bmatrix} \longleftarrow \textit{first step, positive } \omega_3 \quad .$$

$$|C_{k+2}| = |C_{k+1}| \begin{bmatrix} 1 & h_3 \\ -h_3 & 1 \end{bmatrix} \longleftarrow \textit{second step, negative } \omega_3$$

$$= |C_k| \begin{bmatrix} 1 + h_3^2 & 0 \\ 0 & 1 + h_3^2 \end{bmatrix} . \tag{96}$$

For the monotonic input given, we assume

$$\int_0^{t_1} \omega_3 \, dt = +h_3$$

$$\int_{t_1}^{t_2} (-\omega_3) \, dt = -h_3 \quad . \tag{97}$$

If Eq. 97 is true, the coordinate system simply rotates about the 3-axis through a small angular increment, h_3, and then returns to its original position by moving through the increment $-h_3$. Under such a condition, Eq. 96 should have been

$$|C_{k+2}| = |C_k| \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} . \tag{98}$$

The error in processing the 1:1 limit cycle input over two steps is therefore

$$|\varepsilon| = \begin{bmatrix} 1 + h_3^2 & 0 \\ 0 & 1 + h_3^2 \end{bmatrix} - \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} h_3^2 & 0 \\ 0 & h_3^2 \end{bmatrix} . \tag{99}$$

For serial processing of a monotonic input, it can be shown that the error for each step is of modulus zero. However, for a 1:1 limit cycle, the error every other step is equal to that just demonstrated for parallel processing.

Let us now employ a technique by which the order of updating is reversed each time an input pulse from the 3-gyro is received. The difference equations corresponding to this process, for a monotonic input, are

$$\left.\begin{aligned} C_{i1,\,k+1} &= C_{i1,\,k} + h_3\, C_{i2,\,k} \\ C_{i2,\,k+1} &= C_{i2,\,k} - h_3\, C_{i1,\,k+1} \end{aligned}\right\} \text{Forward}$$

$$\left.\begin{aligned} C_{i2,\,k+2} &= C_{i2,\,k+1} - h_3\, C_{i1,\,k+1} \\ C_{i1,\,k+2} &= C_{i1,\,k+1} + h_3\, C_{i2,\,k+2} \end{aligned}\right\} \text{Reverse} \qquad (100)$$

In matrix form,

$$|C_{k+2}| = |C_k| \begin{bmatrix} 1 - 2h_3^2 & -2h_3 \\ 2h_3\left(1 - h_3^2\right) & 1 - 2h_3^2 \end{bmatrix} . \qquad (101)$$

For a 1:1 limit cycle

$$|C_{k+2}| = |C_k| \begin{bmatrix} 1 & -h_3 \\ h_3 & 1 - h_3^2 \end{bmatrix} \begin{bmatrix} 1 - h_3^2 & -h_3 \\ h_3 & 1 \end{bmatrix} = |C_k| \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} . \qquad (102)$$

Equation 102 then is the correct updating for the 1:1 limit-cycle input. It is simple to show that for other limit-cycle modings, 2:2, 3:3, etc., repetitive use of the serial forward-reverse rule (as shown in Eq. 100) will yield a direction cosine matrix without error at the step when the coordinate system has returned to the initial position.

For such a special case, this serial-reversal processing is obviously the preferred algorithm. This method is referred to as the Bumstead Algorithm.

BUMSTEAD-HESSION ALGORITHM (Ref. 1)

The organization described here is a serial-separate-reversal rule. The positive and negative gatings are performed separately as shown by the following equations.

$$\left.\begin{aligned} C_{i1,\,k+a} &= C_{i1,\,k} + h_3\, C_{i2,\,k} \\ C_{i2,\,k+a} &= C_{i2,\,k} + h_1\, C_{i3,\,k} \\ C_{i3,\,k+a} &= C_{i3,\,k} + h_2\, C_{i1,\,k+a} \end{aligned}\right\} \text{Forward Positive gatings only and serial use of } C_{i1}$$

$$\left.\begin{aligned} C_{i1,\,k+1} &= C_{i1,\,k+a} - h_2\, C_{i3,\,k+a} \\ C_{i2,\,k+1} &= C_{i2,\,k+a} - h_3\, C_{i1,\,k+1} \\ C_{i3,\,k+1} &= C_{i3,\,k+a} - h_1\, C_{i2,\,k+1} \end{aligned}\right\} \text{Negative gatings only and serial use of } C_{i1},\ C_{i2}. \qquad (103)$$

$$\left.\begin{aligned} C_{i3,\,k+b} &= C_{i3,\,k+1} - h_1\, C_{i2,\,k+1} \\ C_{i2,\,k+b} &= C_{i2,\,k+1} - h_3\, C_{i1,\,k+1} \\ C_{i1,\,k+b} &= C_{i1,\,k+1} - h_2\, C_{i3,\,k+b} \end{aligned}\right\} \text{Reverse Negative gatings only and serial use of } C_{i3}$$

$$\left.\begin{aligned} C_{i3,\,k+2} &= C_{i3,\,k+b} - h_2\, C_{i1,\,k+b} \\ C_{i2,\,k+2} &= C_{i2,\,k+b} - h_1\, C_{i3,\,k+2} \\ C_{i1,\,k+2} &= C_{i1,\,k+b} - h_3\, C_{i2,\,k+2} \end{aligned}\right\} \text{Positive gatings only and serial use of } C_{i2},\ C_{i3}. \qquad (104)$$

The subscripts $k + a$ and $k + b$ denote intermediate computation points between times (k and $k + 1$) and between times ($k + 1$ and $k + 2$), respectively.

A short-hand notation for Eq. 103 and Eq. 104 is

$$|C_{k+1}| = |C_k| \; |M_f|$$

$$|C_{k+2}| = |C_{k+1}| \; |M_r| \quad .$$

where the elements of $|M_f|$ and $|M_r|$ are indicated by Eq. 103 and Eq. 104 then

$$|C_{k+2}| = |C_k| \; |M_f| \; |M_r| = |C_k| \; [M] \quad .$$

For a 1:1 limit-cycle case, it turns out that

$$[M] = [I] \quad ,$$

which implies a zero error.

CROWDER-HESSION ALGORITHM (Ref. 1 and 8)

This rule is called a serial-parallel-reversal algorithm. Whereas the preceeding algorithm required one intermediate step to complete an updating, this form requires two. This algorithm uses each cosine at time k separately, e.g., $C_{i1,\,k}$ is used first, then $C_{i3,\,k}$.

$$\left.\begin{aligned} C_{i1,\,k+a} &= C_{i1,\,k} \\ C_{i2,\,k+a} &= C_{i2,\,k} - h_3 \, C_{i1,\,k} \\ C_{i3,\,k+a} &= C_{i3,\,k} + h_2 \, C_{i1,\,k} \end{aligned}\right\} \text{Forward using } C_{i1} \text{ only} \qquad (105)$$

Note that since $C_{i1,\ k}$ simply translates to $C_{i1,\ k+a}$, the computation for updating is parallel, likewise for the remaining partial updatings:

$$\left.\begin{aligned} C_{i1,\ k+b} &= C_{i1,\ k+a} + h_3\, C_{i2,\ k+a} \\ C_{i2,\ k+b} &= C_{i2,\ k+a} \\ C_{i3,\ k+b} &= C_{i3,\ k+a} - h_1\, C_{i2,\ k+a} \end{aligned}\right\} \text{Using } C_{i2} \text{ only}$$

$$\left.\begin{aligned} C_{i1,\ k+1} &= C_{i1,\ k+b} - h_2\, C_{i3,\ k+b} \\ C_{i2,\ k+1} &= C_{i2,\ k+b} + h_1\, C_{i3,\ k+b} \\ C_{i3,\ k+1} &= C_{i3,\ k+b} \end{aligned}\right\} \text{Using } C_{i3} \text{ only} \qquad (106)$$

Using C_{i1}, C_{i2}, and C_{i3}, separately, along with both positive and negative gatings, the direction cosine matrix is updated after the two intermediate steps. Applying a reversal rule, as before

$$\left.\begin{aligned} C_{i1,\ k+c} &= C_{i1,\ k+1} - h_2\, C_{i3,\ k+1} \\ C_{i2,\ k+c} &= C_{i2,\ k+1} + h_1\, C_{i3,\ k+1} \\ C_{i3,\ k+c} &= C_{i3,\ k+1} \end{aligned}\right\} \text{Reversal using } C_{i3} \text{ only}$$

$$\left.\begin{aligned} C_{i1,\ k+d} &= C_{i1,\ k+c} + h_3\, C_{i2,\ k+c} \\ C_{i2,\ k+d} &= C_{i2,\ k+c} \\ C_{i3,\ k+d} &= C_{i3,\ k+c} - h_1\, C_{i2,\ k+c} \end{aligned}\right\} \text{Using } C_{i2} \text{ only}$$

$$\left.\begin{aligned} C_{i1,\,k+2} &= C_{i1,\,k+b} \\ C_{i2,\,k+2} &= C_{i2,\,k+b} - h_3\, C_{i1,\,k+b} \\ C_{i3,\,k+2} &= C_{i3,\,k+b} + h_2\, C_{i1,\,k+b} \end{aligned}\right\} \text{Using } C_{i1} \text{ only} \qquad (107)$$

This rule also has zero error modulus under limit-cycle conditions.

MARMON ALGORITHM (Ref. 7)

This algorithm is similar to the Bumstead-Hession algorithm, except that it does not employ serial processing. The equations are

$$C_{i1,\,k+a} = C_{i1,\,k} + h_3\, C_{i2,\,k}$$

$$C_{i2,\,k+a} = C_{i2,\,k} + h_1\, C_{i3,\,k}$$

$$C_{i3,\,k+a} = C_{i3,\,k} + h_2\, C_{i1,\,k}$$

Forward

$$C_{i1,\,k+1} = C_{i1,\,k+a} - h_2\, C_{i3,\,k+a}$$

$$C_{i2,\,k+1} = C_{i2,\,k+a} - h_3\, C_{i1,\,k+a}$$

$$C_{i3,\,k+1} = C_{i3,\,k+a} - h_1\, C_{i2,\,k+a} \qquad (108)$$

A periodic reversal rule is applied as a function of real time rather than as a function of the directional sense of the incremental angles. This method permits a statistically random second-order error accumulation under limit cycle conditions in return for greater large-angle accuracy and a somewhat lower computational frequency requirement on a DDA mechanization.

SEQUENTIAL PULSE PROCESSING (Ref. 4)

In the preceeding algorithms and those of the previous section, it was assumed that the gyro pulses were processed simultaneously. Let us now consider a case where the pulses are processed separately as received, and when simultaneous pulses occur, they will be processed in the preassigned order 1, 2, 3.

At time $t_1, t_2, t_3, \ldots\ldots, t_m$, pulses are received from the 1-, 2-, and 3-axis gyros. Let us use a single-axis algorithm $[H_1]$, $[H_2]$, or $[H_3]$ whenever a pulse is received from the 1, 2, or 3 gyro.

If using a second-order Taylor expansion, then

$$[H_1] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 - \frac{1}{2} h_1^2 & -h_1 \\ 0 & h_1 & 1 - \frac{1}{2} h_1^2 \end{bmatrix}$$

$$[H_2] = \begin{bmatrix} 1 - \frac{1}{2} h_2^2 & 0 & h_2 \\ 0 & 1 & 0 \\ -h_2 & 0 & 1 - \frac{1}{2} h_2^2 \end{bmatrix}$$

$$[H_3] = \begin{bmatrix} 1 - \frac{1}{2} h_3^2 & -h_3 & 0 \\ h_3 & 1 - \frac{1}{2} h_3^2 & 0 \\ 0 & 0 & 1 \end{bmatrix} . \qquad (109)$$

To test this matrix formulation, consider a typical sequence of updatings represented by

$$[H_B] = [H_1 \; H_2 \; H_3] \quad . \qquad (110)$$

$[H_1]$, $[H_2]$, $[H_3]$ are defined in Eq. 107; thus $[H_B]$ represents the complete updating for an interval t_o, t_y. Multiplying the members of $[H_B]$,

$$[H_B] = \begin{bmatrix} 1 - \frac{1}{2}h_2^2 - \frac{1}{2}h_3^2 & -h_3 & h_2 \\ h_3 + h_1h_2 & 1 - \frac{1}{2}h_1^2 - \frac{1}{2}h_3^2 & -h_1 \\ -h_2 + h_1h_3 & h_1 + h_2h_3 & 1 - \frac{1}{2}h_1^2 - \frac{1}{2}h_2^2 \end{bmatrix} , \quad (111)$$

and

$$[C_{k+3}] = [C_k][H_B] .$$

Equation 111 is a correct second-order updating for pulses received sequentially.

Now, assume that the pulses occurred simultaneously, but that the computer processed them sequentially according to the preassigned ordering, 1, 2, 3. The result would be that given by Eq. 111. However, since the pulses occurred simultaneously, the correct updating is given by Eq. 91. Comparing Eq. 111 with Eq. 91, one sees the following:

a. Main-diagonal elements agree;

b. First-order off-diagonal elements agree;

c. Second-order off-diagonal elements *do not* agree; and

d. Sums of second-order off-diagonal elements agree.

The error (c) is a noncommutativity error resulting from the sequential processing. This noncommutativity problem arises from the angular increment, h_i, being finite, and finite rotations do not commute.

BROXMEYER GROUPING ALGORITHM

Broxmeyer (Ref. 4) has suggested a possible method for reducing the error in the off-diagonal terms shown in the sequential pulse processing section (above). In certain special cases, it is possible to reduce

these errors by using a matrix corresponding to a group of updatings. The special case in question here is that case where ω_1, ω_2, ω_3, the angular rate components of $[\Omega]$, are assumed proportional to one another over short time intervals.

Assume that in a time interval, T_G, the computer receives pulses from the gyros in the following quantities:

1-axis gyro ..... m-pulses

2-axis gyro ..... n-pulses

3-axis gyro ..... p-pulses

with m, n, p being arbitrary from interval to interval. Define

$$mh = \int_0^{T_G} \omega_1 \, dt \; ; \; nh = \int_0^{T_G} \omega_2 \, dt \; ; \; ph = \int_0^{T_G} \omega_3 \, dt \quad . \tag{112}$$

It is recognized, of course, that if the ω_i, $i = 1, 2, 3$, are nondeterminating decimal fractions of some ω_{max} for the system, then Eq. 112 will not hold. However, for our purposes here, it is assumed that the m-pulses counted are exact. In Eq. 112, mh, nh, and ph are multiples of the basic angular increment of magnitude, 2^{-n} Radians. If ω_1, ω_2, ω_3 are in constant proportion to each other, then we write

$$\omega_2 = k_{21} \, \omega_1 \; ; \quad \omega_3 = k_{31} \, \omega_1 \tag{113}$$

From these definitions, nh and ph in Eq. 112 become

$$nh = k_{21} \int_0^{T_G} \omega_1 \, dt \; ; \quad ph = k_{31} \int_0^{T_G} \omega_1 \, dt \quad . \tag{114}$$

Now, use Eq. 114 and Eq. 112 on Eq. 66, the matrisant. Truncate terms of order three and higher. For the first-order integral of Eq. 66, we have

$$\int_0^{T_G} [\Omega]\, d\tau = \begin{bmatrix} 0 & -\int_0^{T_G} \omega_3\, d\tau & \int_0^{T_G} \omega_2\, d\tau \\ \int_0^{T_G} \omega_3\, d\tau & 0 & -\int_0^{T_G} \omega_1\, d\tau \\ -\int_0^{T_G} \omega_2\, d\tau & \int_0^{T_G} \omega_1\, d\tau & 0 \end{bmatrix} = \begin{bmatrix} 0 & -ph & nh \\ ph & 0 & -mh \\ -nh & mh & 0 \end{bmatrix} . \tag{115}$$

The second-order term is

$$\int_0^{T_G} \left(\int_0^{\tau} [\Omega]\, d\tau' \right) [\Omega]\, d\tau \quad .$$

For the off-diagonal terms, we have, typically

$$\eta_{12} = \int_0^{T_G} \omega_2 \int_0^{\tau} \omega_1\, d\tau\, d\tau = k_{21} \int_0^{T_G} \omega_1 \int_0^{\tau} \omega_1\, d\tau'\, d\tau = \frac{1}{2} k_{21} \left(\int_0^{T_G} \omega_1\, dt \right)^2$$

$$= \frac{1}{2} nh \cdot mh = \frac{1}{2} mnh^2$$

$$\eta_{13} = \int_0^{T_G} \omega_3 \int_0^{\tau} \omega_1 \, d\tau' \, d\tau = \frac{1}{2} pmh^2$$

$$\eta_{23} = \int_0^{T_G} \omega_3 \int_0^{\tau} \omega_2 \, d\tau' \, d\tau = \frac{1}{2} pnh^2 \quad . \tag{116}$$

Proceeding in a similar manner, all terms of the second-order integral are written. Thus, our updating equation using a second-order algorithm appears in this case as,

$$[C_{k+1}] = [C_k] \begin{bmatrix} 1 - \frac{1}{2}(nh)^2 - \frac{1}{2}(ph)^2 & -ph + \frac{1}{2}mnh^2 & nh + \frac{1}{2}mph^2 \\ ph + \frac{1}{2}mnh^2 & 1 - \frac{1}{2}(mh)^2 - \frac{1}{2}(ph)^2 & -mh + \frac{1}{2}nph^2 \\ -nh + \frac{1}{2}mph^2 & mh + \frac{1}{2}pnh^2 & 1 - \frac{1}{2}(mh)^2 - \frac{1}{2}(nh)^2 \end{bmatrix} . \tag{117}$$

Notice the product terms, such as $(1/2)mnh^2$. When dealing with simple single-pulse intervals to perform multiplication such as $hC_{i1,k}$ or $1/2\ h^2\ C_{i2,\ k}$, we had only to shift $C_{i1,\ k}$ or $C_{i2,\ k}$, etc. Now we are forced with a multiple-pulse problem, and must form the products mn, np, and np.

Broxmeyer suggests a simple scheme that requires only monitoring the gyros and counting m, n, and p. Define U, V, W, X, Y, and Z as follows:

U is the sum of the number of times an h_1-pulse follows an h_2-pulse
V is the sum of the number of times an h_2-pulse follows an h_1-pulse
W is the sum of the number of times an h_1-pulse follows an h_3-pulse
X is the sum of the number of times an h_3-pulse follows an h_1-pulse
Y is the sum of the number of times an h_2-pulse follows an h_3-pulse
Z is the sum of the number of times an h_3-pulse follows an h_2-pulse

Therefore, for U, V, W, X, Y, and Z so defined,

$$U + V = mn$$

$$W + X = np$$

$$Y + Z = np \tag{118}$$

Example 2

Assume one receives over a time, T_G, the series of pulses $(h_1\ h_2\ h_3\ h_3\ h_2\ h_1)$.

$$U = 2 \quad W = 2 \quad Y = 2$$

$$V = 2 \quad X = 2 \quad Z = 2$$

$$m = 2 \quad U + V = 2 + 2 = mn = 2.2 = 4$$

$$n = 2 \quad W + X = np = 4$$

$$p = 2 \quad Y + Z = np = 4$$

For the m 1-gyro pulses received in an interval T_G, form the matrix

$$|A_1| = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \frac{1}{2}(mh)^2 & -mh + Yh^2 \\ 0 & mh + Zh^2 & \frac{1}{2}(mh)^2 \end{bmatrix} . \tag{119}$$

For the n 2-gyro pulses received in a same interval T_G, form

$$|A_2| = \begin{bmatrix} \frac{1}{2}(nh)^2 & 0 & nh + Wh^2 \\ 0 & 0 & 0 \\ -nh + Xh^2 & 0 & \frac{1}{2}(nh)^2 \end{bmatrix} . \tag{120}$$

Finally, for the p 3-gyro pulses received over T_C, form

$$[A_3] = \begin{bmatrix} -\frac{1}{2}(ph)^2 & -ph + Uh^2 & 0 \\ ph + Vh^2 & -\frac{1}{2}(ph)^2 & 0 \\ 0 & 0 & 0 \end{bmatrix} . \tag{121}$$

When the three formations are complete, add Eq. 119, Eq. 120, and Eq. 121 to the unit matrix $[I]$ so that the following is formed:

$$[A] = [I] + [A_1] + [A_2] + [A_3] . \tag{122}$$

Expanding $[A]$ in component form,

$$[A] = \begin{bmatrix} 1 - \frac{1}{2}(nh)^2 - \frac{1}{2}(ph)^2 & -ph + Uh^2 & nh + Yh^2 \\ ph + Vh^2 & 1 - \frac{1}{2}(mh)^2 - \frac{1}{2}(ph)^2 & -mh + Yh^2 \\ -nh + Xh^2 & mh + Zh^2 & 1 - \frac{1}{2}(mh)^2 - \frac{1}{2}(nh)^2 \end{bmatrix} . \tag{123}$$

Then, to the matrix $[A]$, add the correction matrix $[B]$

$$[B] = \begin{bmatrix} 0 & -\frac{1}{2}(U-V)h^2 & -\frac{1}{2}(W-Z)h^2 \\ \frac{1}{2}(U-V)h^2 & 0 & -\frac{1}{2}(Y-Z)h^2 \\ \frac{1}{2}(W-X)h^2 & \frac{1}{2}(Y-Z)h^2 & 0 \end{bmatrix} . \tag{124}$$

So, finally,

$$[H_c] = [A] + [B] \tag{125}$$

is the correct matrix for updating after receiving n 1-gyro pulses, m 2-gyro pulses, and p 3-gyro pulses.

To demonstrate the addition represented in Eq. 125, compute the term $H_{c_{12}}$

$$H_{c_{12}} = A_{12} + B_{12}$$

$$= -ph - Uh^2 - \frac{1}{2}Uh^2 + Vh^2$$

$$= -ph + \frac{1}{2}(U + V)h^2 \quad .$$

But, by Eq. 118, $U + V = mn$; so

$$H_{c_{12}} = -ph + \frac{1}{2}mnh^2 \quad ,$$

yielding the correct term compared with $H_{c_{12}}$ in Eq. 117. The remaining terms may be verified in a similar manner.

ALGORITHM ANALYSIS METHODS

The algorithms for updating the direction cosine matrix given in the previous sections, have varying quantities and errors associated with their use. In addition to the errors resulting from mechanizing the algorithms on a digital computer, errors inherent in each individual algorithm itself may be present.

In order to evaluate adequately the accuracy of given algorithm forms, error analyses on the algorithms must be performed. Additionally, if error analyses on the entire strapdown guidance system are to be made,

the determination of algorithm errors, or coordinate transformation errors, will comprise a vital portion of such an analysis program.

Three basic methods of algorithm error analysis are discussed here, only in the sense that they apply to the analysis of algorithms used in the digital solution of the direction cosine differential equation. Since the algorithms identified in the two preceding sections are essentially difference equations, perhaps the most obvious error analysis methods are those methods that result from solutions of the separate difference equations. The separate difference equations can be solved either in the classical manner or they can be solved by the discrete method of Z-transforms, which is a method equivalent to the analytic solution of continuous differential equations by Laplace transforms. The Z-transform method may also be applied to the matrix-direction cosine-differential equation, and an identical solution may be obtained. Another analysis method involves the use of matrix theory. With this matrix method, the deviation of the phase and magnitude of a unit vector in space from the ideal phase and magnitude gives an indication of algorithm performance. Finally, the last method discussed utilizes a digital computer program to study the algorithm error buildup.

SOLUTION BY METHOD OF UNDETERMINED COEFFICIENTS (Ref. 22)

This method is introduced with the following example. Consider the homogeneous difference equation

$$y_{n+1} + y_n = 0 \quad , \tag{126}$$

with the initial condition

$$y_0 = 1 \quad .$$

A rigorous solution of this difference equation may be obtained by letting in Eq. 126

$$y_n = Ax^n \quad , \tag{127}$$

the determination of algorithm errors, or coordinate transformation errors, will comprise a vital portion of such an analysis program.

Three basic methods of algorithm error analysis are discussed here, only in the sense that they apply to the analysis of algorithms used in the digital solution of the direction cosine differential equation. Since the algorithms identified in the two preceding sections are essentially difference equations, perhaps the most obvious error analysis methods are those methods that result from solutions of the separate difference equations. The separate difference equations can be solved either in the classical manner or they can be solved by the discrete method of Z-transforms, which is a method equivalent to the analytic solution of continuous differential equations by Laplace transforms. The Z-transform method may also be applied to the matrix-direction cosine-differential equation, and an identical solution may be obtained. Another analysis method involves the use of matrix theory. With this matrix method, the deviation of the phase and magnitude of a unit vector in space from the ideal phase and magnitude gives an indication of algorithm performance. Finally, the last method discussed utilizes a digital computer program to study the algorithm error buildup.

SOLUTION BY METHOD OF UNDETERMINED COEFFICIENTS (Ref. 22)

This method is introduced with the following example. Consider the homogeneous difference equation

$$y_{n+1} + y_n = 0 \quad , \tag{126}$$

with the initial condition

$$y_0 = 1 \quad .$$

A rigorous solution of this difference equation may be obtained by letting in Eq. 126

$$y_n = Ax^n \quad , \tag{127}$$

where x is an undetermined number. By substituting Eq. 127 into Eq. 126,

$$y_{n+1} + y_n = Ax^{n+1} + Ax^n = 0$$

$$Ax^n (x + 1) = 0 \quad . \tag{128}$$

If Eq. 127 is to be a solution of Eq. 126, then Eq. 128 must be satisfied for all values of n. So, x must be the single root:

$$x = -1 \quad .$$

Then Eq. 127 appears as

$$y_n = A(-1)^n \quad . \tag{129}$$

By the initial condition, $y_0 = 1$,

$$y_n\Big|_{n=0} = A(-1)^n\Big|_{n=0}$$

$$y_0 = 1 = A(-1)^0 = A \quad .$$

Therefore, the solution to Eq. 126 is

$$y_n = (-1)^n = \cos(n\pi) \quad , \tag{130}$$

which can be determined by inspecting Eq. 126.

Solution of a single equation, such as Eq. 126, does not generally require a great amount of effort. However, when this direct numerical method is applied to systems of coupled difference equations, simplicity of solution no longer holds.

As a further example of this method of solution, two coupled equations will be solved, showing the additional complexity associated with the multipole system. Consider a single-axis, first-order, direction-cosine-matrix updating for a monotonic input given by

$$\begin{bmatrix} C_{11,\,k+1} & C_{12,\,k+1} \\ C_{21,\,k+1} & C_{22,\,k+1} \end{bmatrix} = \begin{bmatrix} C_{11,\,k} & C_{12,\,k} \\ C_{21,\,k} & C_{22,\,k} \end{bmatrix} \begin{bmatrix} 1 & -h_s \\ h_s & 1 \end{bmatrix} , \tag{131}$$

or in component form

$$\left.\begin{aligned} C_{i1,\,k+1} &= C_{i1,\,k} + h_s\, C_{i2,\,k} \\ C_{i2,\,k+1} &= C_{i2,\,k} - h_s\, C_{i1,\,k} \end{aligned}\right\} \quad \text{for } i = 1,\ 2 \ . \tag{132}$$

Assume that $C_{i1,0}$ and $C_{i2,0}$ define the initial values of $C_{i1,k}$ and $C_{i2,k}$, and that k represents the number of incremental rotations that have taken place since $t = t_0$. If the angle of rotation is ϕ, then

$$\phi = \sum_{k'=0}^{k} h_{sk'} = kh_s \ , \tag{133}$$

where

$$h_s \triangleq \int_{t_k}^{t_{k+o}} \omega_{s,\,max}\, dt \ .$$

Proceeding as in Eq. 127, substitute

$$\begin{aligned} C_{i1,\,k} &= \alpha^k \\ C_{i2,\,k} &= \beta^k \end{aligned} \tag{134}$$

into Eq. 132, and obtaining

$$Ax^{k+1} = Ax^{k} + h_3\, Bx^{k}$$

$$Bx^{k+1} = Bx^{k} - h_3\, Ax^{k} \quad . \tag{135}$$

Eliminating A and B from Eq. 135,

$$x^2 - 2x + \left(1 + h_3^2\right) = 0 \quad . \tag{136}$$

Equation 136 has two roots,

$$x_{1,\,2} = 1 \pm j\, h_3 \quad . \tag{137}$$

Following the method used before (Eq. 129), the general solution to Eq. 132 is written as

$$C_{i1,\,k} = Dx_1^k + Ex_2^k$$

$$C_{i2,\,k} = Fx_2^k + Gx_2^k \quad . \tag{138}$$

With the roots (Eq. 137) substituted,

$$C_{i1,\,k} = D\,(1 + j\,h_3)^k + E\,(1 - j\,h_3)^k$$

$$C_{i2,\,k} = F\,(1 + j\,h_3)^k + G\,(1 - k\,j_3)^k \quad . \tag{139}$$

By Eq. 132, the four direction cosines can be represented by two equations, thus, it will be sufficient to consider the solution of a single set of two equations (say $i = 1$), rather than all four. Equation 132 cannot be solved in the general form with subscript i, since

$$C_{11,\,0} \neq C_{21,\,0} \; ; \; C_{12,\,0} \neq C_{22,\,0} \quad .$$

Assume for initial conditions, the unit matrix, or

$$C_{11,\,0} = 1 \; ; \; C_{12,\,0} = 0 \quad . \tag{140}$$

Applying Eq. 140 to Eq. 132 for $k = 0$,

$$C_{11,\,1} = C_{11,\,0} + h_3 \, C_{12,\,0} = 1$$

$$C_{12,\,1} = C_{12,\,0} - h_3 \, C_{11,\,0} = -h_3 \quad . \tag{141}$$

Applying these same initial conditions to Eq. 139 for $i = 1$ and $k = 0$, and $k = 1$,

$$C_{11,\,0} = D \left(1 + j \, h_3\right)^0 + E \left(1 - j \, h_3\right)^0 = D + E = 1$$

$$C_{11,\,0} = D \left(1 + j \, h_3\right) + E \left(1 - j \, h_3\right) = 1 \quad . \tag{142}$$

Therefore,

$$D = \frac{1}{2} \; ; \; E = \frac{1}{2} \quad .$$

Also in a similar manner

$$P = \frac{1}{2j} \; ; \; G = \frac{1}{2j} \quad . \tag{143}$$

By Eq. 142 and 143, Eq. 129 becomes

$$C_{11,\,k} = \frac{1}{2}\left(\left(1 + j\,h_3\right)^k + \left(1 - j\,h_3\right)^k\right)$$

$$C_{12,\,k} = \frac{1}{2j}\left(\left(1 - j\,h_3\right)^k - \left(1 + j\,h_3\right)^k\right) \quad . \tag{144}$$

Since terms such as $\left(1 + j\,h_3\right)^k$ may be written in polar form as

$$\left(1 + j\,h_3\right)^k = \left(1 + h_3^2\right)^{k/2} e^{jk\theta} \quad ,$$

where $\theta = \tan^{-1} h_3$, then

$$C_{11,\,k} = \left(1 + h_3^2\right)^{k/2} \cos k\theta$$

$$C_{12,\,k} = -\left(1 + h_3^2\right)^{k/2} \sin k\theta \quad . \tag{145}$$

The terms in Eq. 145 indicate the behavior of the direction cosines during k incremental rotations for a monotonic input.

For motion about one axis, which is what Eq. 131 represents, for k increments, the true values should be

$$C'_{11,\,k} = \cos k\theta$$

$$C'_{12,\,k} = -\sin k\theta \quad . \tag{146}$$

From Eq. 145 and 146, error equations can be formed so that

$$c_{11,\,k} = C_{11,\,k} - C_{11,\,k} = \cos k\theta \left[1 - \left(1 + h_3^2\right)^{k/2}\right]$$

$$c_{12,\,k} = -\sin k\theta \left[1 - \left(1 + h_3^2\right)^{k/2}\right] \quad . \tag{147}$$

For sufficiently small values of h_3,

$$\theta = \tan^{-1} h_3 \cong h_3 \quad ,$$

so that

$$c_{11,\,k} = \cos k\, h_3 \left[1 - \left(1 + h_3^2\right)^{k/2}\right]$$

$$c_{12,\,k} = -\sin k\, h_3 \left[1 - \left(1 + h_3^2\right)^{k/2}\right] \quad . \tag{148}$$

Looking at $c_{11,\,k}$,

$$c_{11,\,k} = \cos k\, h_3 \left[1 - \left(1 + h_3^2\right)^{k/2}\right]$$

$$\cong -\cos k\, h_3 \left[\frac{k\, h_3^2}{2}\right] \quad .$$

But $kh_3 = \phi$, so

$$\epsilon_{11,k} = -\frac{1}{2} h_3 \phi \cos\phi$$

$$\epsilon_{12,k} = \frac{1}{2} h_3 \phi \sin\phi \qquad (149)$$

It is obvious from review of the method of solution required to progress from Eq. 132 to the solution at Eq. 145 that, as the order of coupling in the system goes up (say 3 x 3), the complexity of solution compounds accordingly.

SOLUTION OF DIFFERENCE EQUATIONS BY THE Z-TRANSFORM METHOD (Ref. 10, 25, 31)

The Z-transform method, a modification of the Laplace transformation, is one of the transform methods that can be applied to the solution of linear difference equations. It reduces the solutions of such nonalgebraic equations to those of algebraic form. The Laplace transformation has been well developed for the solution of differential equations and discussed extensively in the literature. The Z-transform modification has extended the usefulness of the Laplace transforms to discrete systems.

The Z-transform denoted by $Z\,[f(t)]$ is defined by

$$Z\,[f(t)] = F(z) = \sum_{k=0}^{\infty} f(kT)z^{-k} \quad , \text{ with } k \text{ integer,} \qquad (150)$$

for $|z| > R = 1/\mu$, where μ is the radius of convergence of the series (Ref. 25). Since only discrete values of $f(t)$ are used, Eq. 150 can be written

$$F(z) = \sum_{k=0}^{\infty} f_k \cdot z^{-k} \qquad (151)$$

It can be shown that the Z-transform is related to the Laplace transform through the expression

$$e^{-kTs} = z^{-k} \tag{152}$$

This arises from the Laplace transformation of a sampled time function $f^*(t)$ being written as

$$Z\,[f^*(t)] = F^*(s) = \sum_{k=0}^{\infty} f(kT)e^{-kTs} \quad . \tag{153}$$

Exposition of this method will again by by example. The homogeneous equation $[C] = [C]\,[\Omega]$, for constant Ω, has the solution

$$[C(t)] = [C(o)]\, e^{\int_{o}^{t} [\Omega]\, d\tau} \quad . \tag{154}$$

Equation 154 holds for any interval t_o to t or t_k to t_{k+1} so that, for equally spaced time intervals, $t_{k+1} = t_{k+T}$

$$\left|C_{k+1}\right| = \left|C_k\right| e^{[\Omega]\,T} \,,\; T = t_{k+1} - t_k \quad . \tag{155}$$

For small sampling intervals, it is assumed that the ω_i are constants during the intervals. If ω_i is constant, $\omega_i T$ represents a constant angular increment with respect to the interval. Then

$$[\Omega]\ T \triangleq \begin{bmatrix} 0 & -h_3 & h_2 \\ h_3 & 0 & -h_1 \\ -h_2 & h_1 & 0 \end{bmatrix} \tag{156}$$

If $e^{[\Omega]\ T}$ is expanded in a series and truncated after the first-order term, $e^{[\Omega]\ T} \cong [I] + [\Omega]\ T$ with error of the order $([\Omega]\ T)^2/2$, (that is, the error in truncations of the series is at the next order higher than the last term retained). Hence, for

$$[\Phi] = e^{[\Omega]\ T} , \tag{157}$$

$$[\Phi] \cong [I] + [\Omega]\ T = \begin{bmatrix} 1 & -h_3 & h_2 \\ h_3 & 1 & -h_1 \\ -h_2 & h_1 & 1 \end{bmatrix} . \tag{158}$$

The Z-transform of the difference equation (Eq. 155) is taken by multiplying by z^{-k} and summing from $k = 0$ to $k = \infty$.

$$\left[C_{k+1}\right] = \left[C_k\right][\theta]$$

$$\sum_{k=0}^{\infty} \left[C_{k+1}\right] z^{-k} = \sum_{k=0}^{\infty} \left[C_k\right][\theta]\, z^{-k} \quad . \tag{159}$$

Using the definition of the Z-transform as given in Eq. 151, Eq. 159 can be rewritten as shown in Eq. 160 and 161.

$$[C(z)] \triangleq \sum_{k=0}^{\infty} \left[C_k\right] z^{-k} \quad , \tag{160}$$

$$\sum_{k=0}^{\infty} \left[C_{k+1}\right] z^{-k} = [C(z)][\theta] \quad . \tag{161}$$

Let $n = k + 1$ in Eq. 161. Thus,

$$\sum_{n=1}^{\infty} \left[C_n\right] z^{-n+1} = [C(z)]\,[\theta] \quad , \tag{162}$$

or

$$\sum_{n=0}^{\infty} \left[C_n\right] z^{-n+1} - [C(o)]\, z = [C(z)]\,[\theta] \quad . \tag{163}$$

Finally,

$$[C(z)]\, z - [C(o)]\, z = [C(z)]\,[\theta] \quad . \tag{164}$$

Solving for $C(z)$

$$[C(z)] = [C(o)]\,[z\,[I] - [\Phi]]^{-1}\,z \quad . \tag{165}$$

Substituting $[\Phi]$ as approximated in Eq. 158,

$$[C(z)] = [C(o)] \begin{bmatrix} (z-1) & -h_3 & -h_2 \\ -h_3 & (z-1) & -h_1 \\ -h_2 & h_1 & (z-1) \end{bmatrix}^{-1} z \quad .$$

which becomes, after performing the operation at the right of the equation,

$$[C(z)] = [C(o)] \begin{bmatrix} \dfrac{z\left[(z-1)^2 + h_1^2\right]}{D} & \dfrac{z\left[h_1h_2 - (z-1)h_3\right]}{D} & \dfrac{z\left[h_1h_3 + (z-1)h_2\right]}{D} \\ \dfrac{z\left[h_1h_2 + (z-1)h_3\right]}{D} & \dfrac{z\left[(z-1)^2 + h_2^2\right]}{D} & \dfrac{z\left[h_2h_3 - (z-1)h_1\right]}{D} \\ \dfrac{z\left[h_1h_3 - (z-1)h_2\right]}{D} & \dfrac{z\left[h_2h_3 + (z-1)h_1\right]}{D} & \dfrac{z\left[(z-1)^2 + h_3^2\right]}{D} \end{bmatrix} \tag{166}$$

where

$$D = (z-1)\left[(z-1)^2 + h_1^2 + h_2^2 + h_3^2\right] \quad .$$

The solution of Eq. 166 as a function of kT, $[C(kT)]$, requires the inverse Z-transform of Eq. 166. The inverse Z-transform of a function $F(z)$, denoted by $Z^{-1}[F(z)]$, is defined as

$$Z^{-1}[F(z)] = \frac{1}{2\pi j}\oint F(z)\, z^{k-1}\, dz = f(kT) \quad , \tag{167}$$

where the line integral is a suitably chosen contour in the complex plane (Ref. 31), generally the unit circle for stable functions.

Applying this definition to Eq. 165 the general form of the inverse of $[C(z)]$ is

$$[C(kT)] = [C(o)]\, Z^{-1}[A(z)] \quad , \tag{168}$$

where,

$$[A(z)] = z[zI] - [\Phi]]^{-1} \quad . \tag{169}$$

Then, from Eq. 166, for example,

$$A_{11}(z) = \frac{z\left[(z-1)^2 + h_1^2\right]}{(z-1)\left[(z-1)^2 + h_1^2 + h_2^2 + h_3^2\right]} \quad . \tag{170}$$

For this example, let the incremental angles on each axis be the same over each time period so that

$$h_1 = h_3 = h_2 = h \quad . \tag{171}$$

$$h_1^2 + h_2^2 + h_3^2 = Rh^2 \quad . \tag{172}$$

This allows Eq. 170 to be rewritten and factored as

$$A_{11}(z) = \frac{z\left|(z-1)^2 + h^2\right|}{(z-1)\left|\left(z^2 - 2z + 1\right) + 3h^2\right|}$$

$$= \frac{z\left|(z-1)^2 + h^2\right|}{(z-1)\ (z-1+j\sqrt{3}\,h)\ (z-1-j\sqrt{3}\,h)} \qquad (173)$$

The Residue Theorem (Ref. 31) states that if a function $g(z)$ is analytic inside and on a closed contour C except at a finite number of singular points $z_1, z_2, \ldots, z_n$ interior to C then

$$\oint g(z)\,dz = 2\pi j\left(K_1 + K_2 + \ldots + K_n\right) \qquad (174)$$

where K is a residue of $g(z)$ at $z = z_i$.

Applying this theorem to the example,

$$C(kT) = \frac{1}{2\pi j}\oint C(z)\,z^{k-1}\,dz = K_1 + K_2 + \ldots + K_n \quad , \qquad (175)$$

where K_i is a residue of $C(z)\,z^{k-1}$ at the pole $z = z_i$.

The Z-transform of the matrix $C(z)$ is performed by taking the Z-transform of each element of the matrix. By Eq. 175 then, the inverse of the terms $A_{11}(z)$ in Eq. 170 is found by

$$A_{11}(kT) = \left.\frac{z\left|(z-1)^2 + h^2\right|z^{k-1}}{(z-1)\left|z-1+j\sqrt{3}\,h\right|}\right|_{z=1+j\sqrt{3}\,h} + \left.\frac{z\left|(z-1)^2 + h^2\right|z^{k-1}}{(z-1)(z-1-j\sqrt{3}\,h)}\right|_{z=1-j\sqrt{3}\,h}$$

$$+ \left.\frac{z\left|(z-1)^2 + h^2\right|z^{k-1}}{(z-1-\quad h)(z-1+j\sqrt{3}\,h)}\right|_{z=1} \qquad (176)$$

Simplifying Eq. 176

$$A_{11}(kT) = \frac{1}{3}\left|(1 + j\sqrt{3}\,h)^k + (1 - j\sqrt{3}\,h)^k + 1\right|$$

$$= \frac{1}{3}\left|1 - \left(1 + 3h^2\right)^{k/2} e^{j\beta} + \left(1 + 3h^2\right)^{k/2} e^{-j\beta}\right| \qquad (177)$$

Thus,

$$A_{11}(kT) = \frac{1}{3}\left[1 + 2\left(1 + 3h^2\right)^{k/2} \cos\beta\right] \quad , \qquad (178)$$

where

$$\beta = k \tan^{-1} \sqrt{3}\,h \quad .$$

If the remaining terms in the $[A(kT)]$ matrix are derived in the same way, then the entire matrix $[A(kT)]$ appears as

$$[A(kT)] = \begin{bmatrix} \left[1 + 2(L)^{k/2} \cos\beta\right] & \left[1 - (L)^{k/2} (\cos\beta + \sqrt{3} \sin\beta)\right] & \left[1 - (L)^{k/2} (\cos\beta + \sqrt{3} \sin\beta)\right] \\ \left[1 - (L)^{k/2} (\cos\beta + \sqrt{3} \sin\beta)\right] & \left[1 + 2(L)^{k/2} \cos\beta\right] & \left[1 - (L)^{k/2} (\cos\beta - \sqrt{3} \sin\beta)\right] \\ \left[1 - (L)^{k/2} (\cos\beta - \sqrt{3} \sin\beta)\right] & \left[1 - (L)^{k/2} (\cos\beta + \sqrt{3} \sin\beta)\right] & \left[1 + 2(L)^{k/2} \cos\beta\right] \end{bmatrix} , \qquad (179)$$

where L is $\left(1 + 3h^2\right)$ and β is $k \tan^{-1} \sqrt{3}\,h$.

It should be noted that Eq. 179, which defines the inverse of Eq. 169, was obtained on the assumption of the truncated first-order exponential series. This assumption actually makes $[A(kT)]$ a first-order direction-cosine updating algorithm.

If in Eq. 179, or actually Eq. 171 and 172,

$$h_1 = h_2 = 0$$

$$h_3 = h \quad , \tag{180}$$

then $[A(kT)]$ is reduced to the first-order algorithm for single-axis motion. The terms A_{11} and A_{12} become the same as the terms on the right side of Eq. 145, assuming that $[C(o)] = [I]$, the identity matrix, and that the same approximations leading to Eq. 145 are also made here, namely,

$$1 - \sqrt[k]{1 + h^2} \cong \frac{-kh^2}{2} \quad ; \quad \tan^{-1} h \cong h \quad .$$

It is clear after reviewing both the classical numerical solution and the Z-transform method that the latter method is certainly less tedious to compute. This is especially true when solutions of higher ordered matrices are concerned.

MATRIX APPROACH (Ref. 1)

This method is useful in making a qualitative assessment of the accuracies of given algorithms.

Equation 132 can be expanded to include the third axis by writing $[C_{ij}]$ for $i = 1, 2, 3$ and $j = 1, 2, 3$. Thus,

$$C_{i1,\,k+1} = C_{i1,\,k} + h_3 C_{i2,\,k} - h_2 C_{i3,\,k}$$

$$C_{i2,\,k+1} = C_{i2,\,k} + h_1 C_{i3,\,k} - h_3 C_{i1,\,k}$$

$$C_{i3,\,k+1} = C_{i3,\,k} + h_2 C_{i1,\,k} - h_1 C_{i2,\,k} \quad . \tag{181}$$

This can be written in matrix form as

$$\left[C_{k+1}\right] = \left[C_k\right][H] \quad , \tag{182}$$

with

$$[H] = \begin{bmatrix} 1 & -h_3 & h_2 \\ h_3 & 1 & -h_1 \\ -h_2 & h_1 & 1 \end{bmatrix} ,$$

and $[C]$ the direction-cosine matrix with elements C_{ij} for $i, j = 1, 2, 3$. Equation 182 can be written with C_{ij} as a row vector as shown in Eq. 183.

$$\left[C_{i1,\,k+1}, C_{i2,\,k+1}, C_{i3,\,k+1}\right] = \left[C_{i1,\,k}, C_{i2,\,k}, C_{i3,\,k}\right][H] \quad , \tag{183}$$

or, as a column vector in Eq. 184.

$$\begin{bmatrix} C_{i1,\,k+1} \\ C_{i2,\,k+1} \\ C_{i3,\,k+1} \end{bmatrix} = \left[\gamma\right]^T \begin{bmatrix} C_{i1,\,k} \\ C_{i2,\,k} \\ C_{i3,\,k} \end{bmatrix} \; ; \; \underline{C_{k+1}} = [H]^T \cdot \underline{C_k} \quad . \tag{184}$$

$[H]^T$ indicates the transpose of $[H]$, and the underscore, __, indicates a vector quantity.

$\underline{C_{k+1}}$ can be interpreted as a transformation of the original vector $\underline{C_k}$ by use of the transformation matrix $[H]^T$. In general, $[H]^T$ represents a nonorthogonal approximation to the orthogonal matrix $[\phi]^T$. If, in Eq. 184, the true updating matrix $[\phi]^T$ is inserted in place of $[H]^T$, then

Eq. 185 represents the orthogonal transformation of the direction cosine vector,

$$\underline{C_{k+1}} = [\phi]^T \underline{C_k} \quad . \tag{185}$$

Now, assume that $\underline{\xi_k}$ is an eigenvector of $[\phi]^T$. This implies that

$$[\phi]^T \underline{\xi_k} = \lambda \underline{\xi_k} \quad , \tag{186}$$

where λ is a scalar multiplier. Equation 186 can be rearranged so that

$$\left(\lambda [I] - [\phi]^T\right) \underline{\xi_k} = 0 \quad . \tag{187}$$

This homogeneous problem possesses a nontrivial solution if, and only if, the determinant of $|\lambda [I] - [\phi]^T|$ vanishes. The values of λ that yield nontrivial solutions can be determined by setting

$$\left|\left[\lambda [I] - [\phi]^T\right]\right| = 0 \quad . \tag{188}$$

Equation 188 is the characteristic equation of the matrix $[\phi]^T$, and λ_1, λ_2, λ_3 are the eigenvalues of $[\phi]^T$.

A useful property of orthogonal transformations is that the eigenvalues are all of unit magnitude. Let $[\phi]^T$ now represent the orthogonal rotation about a single axis of rotation given by Eq. 90. For $h_1 = h_2 = h_3 = h$ in this equation, $[\phi]^T$ can be shown to have the eigenvalues

$$\lambda_{1,2,3} = 1, \; 1 \underline{/\pm \psi_\phi} \quad , \quad \psi_\phi = \sqrt{3}\, h \tag{189}$$

for $h = \omega T$. If, then, the eigenvalues of the $[H]^T$ matrix are determined, their deviation from the eigenvalues $1, \; 1 \underline{/\pm \psi_\phi}$ is a comparative indication of the errors associated with the algorithm $[H]$ approximation to $[\phi]$.

As an example, consider the rotation in two dimensions ($h_1 = h_2 = 0$, $h_3 = h$). The $[\phi]^T$ matrix for this case has eigenvalues $\lambda_{1,2,3} = 1$, $1 \angle \pm h$ (note that the eigenvalue phase angle is the angle of rotation). If this rotation is approximated by

$$[C_{k+1}] = [C_k]\begin{bmatrix} 1 & -h & 0 \\ h & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = [C_k][M] \quad ,$$

then M has characteristic roots

$$\lambda'_{1,2,3} = 1 \, , \quad 1 \pm jh \quad ,$$

or in polar form

$$\lambda'_{1,2,3} = 1 \, , \quad re^{j\phi} \quad ,$$

where $r = \sqrt{1+h^2}$ and $\phi = \pm \tan^{-1} h$ is the error in approximating the orthogonal matrix of the rotation by M for this one iteration.

For the modulus error, r is expanded in series,

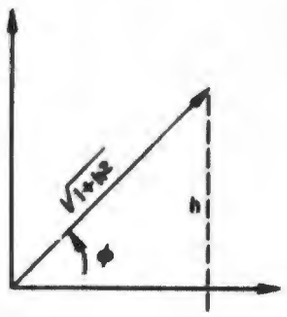
$$r = \left(1 + h^2\right)^{1/2} \cong 1 + \frac{h^2}{2} \quad ,$$

and it is compared with the eigenvalue magnitudes of $\lambda_{1,2,3}$ obtained above for the single axis rotation $[\phi]^T$ matrix. Thus, the modulus error is -

$$\epsilon_{mod} = r - 1 = 1 + \frac{h^2}{2} - 1 = \frac{h^2}{2} \quad ,$$

for $\left|\lambda'_{2,3}\right| - \left|\lambda_{2,3}\right|$.

The phase error results from having rotated the unit vector through the angle ϕ (with $\tan^{-1} h$) instead of through the angle h. Since $\phi = \tan^{-1} h \quad h - h^3/3$, the phase error is $|\phi - h|$ or $h^3/3$.



This type of analysis can be extended to three dimensions whenever the rotation is about a single axis. The analysis then takes place in the plane perpendicular to the axis of rotation in the case where this axis coincides with the axis of the approximating transformation.

In general, for any given algorithm, or multiple sequence of application of any given algorithm, this eigenvalue method is a useful tool. Also, when the axis of rotation is fixed in space for n iterations, an eigenvalue $\lambda_{i,n}$ of the matrix representing n iterations is equal to a corresponding eigenvalue $\lambda_{i,1}$ of the matrix for a single rotation raised to the nth power,

$$\lambda_{i,n} = (\lambda_i)^n \quad .$$

The following example is for a 1:1 limit cycle applied to an algorithm that updates according to a parallel updating scheme. A considerable number of other examples are to be found in Ref. 1 that make use of this same analysis technique. The difference equations representing the parallel scheme are

$$C_{i1,\,k+1} = C_{i1,\,k} + h\,C_{i2,\,k} - h\,C_{i3,\,k}$$

$$C_{i2,\,k+1} = C_{i2,\,k} + h\,C_{i3,\,k} - h\,C_{i1,\,k}$$

$$C_{i3,\,k+1} = C_{i3,\,k} + h\,C_{i1,\,k} - h\,C_{i2,\,k} \qquad (190)$$

Reversing the sign of the h for the limit cycle,

$$C_{i1,\,k+2} = C_{i1,\,k+1} - h\,C_{i2,\,k+1} + h\,C_{i3,\,k+1}$$

$$C_{i2,\,k+2} = C_{i2,\,k+1} - h\,C_{i3,\,k+1} + h\,C_{i1,\,k+1}$$

$$C_{i3,\,k+2} = C_{i3,\,k+1} - h\,C_{i1,\,k+1} + h\,C_{i2,\,k+1} \qquad (191)$$

Thus, the updating matrix over the period k to $k + 2$ is

$$\left[H_{k+2,\,k}\right] = \left[H_{k+2,\,k}\right]^T = \begin{bmatrix} 1 + 2h^2 & -h^2 & -h^2 \\ -h^2 & 1 + 2h^2 & -h^2 \\ -h^2 & -h^2 & 1 + 2h^2 \end{bmatrix} \qquad (192)$$

Forming

$$\left|\lambda\,[I] - \left|H_{k+2,\,k}\right|^{T}\right| = 0 \quad .$$

it is determined that

$$(1-\lambda)^3 + 6h^2\,(1-\lambda)^2 + 9h^4\,(1-\lambda) = 0 \quad . \tag{193}$$

The solutions to the characteristic equation give the eigenvalues

$$\lambda_1 = 1$$

$$\lambda_{2,\,3} = \left(1 + 3h^2\right) \quad . \tag{194}$$

The eigenvalues for $[\phi]^T\,[\phi]$, where ϕ is an orthogonal matrix, are

$$\lambda_{1,2,3} = 1,1,1 \quad .$$

Thus, by comparison, the parallel rule has a modulus error of $3h^2$ after two inputs. For this algorithm, after two updatings, there is no phase error.

For a monotonic input in the parallel case, the updating matrix is simply

$$[H]^T = \begin{bmatrix} 1 & h & -h \\ -h & 1 & h \\ h & -h & 1 \end{bmatrix} \tag{195}$$

for the interval k to $k + 1$ for a single increment about each axis of magnitude h. The characteristic equation of $[H]^T$ in Eq. 194 is

$$(1 - \lambda)^3 + 3h^2 (1 - \lambda) = 0 \quad , \tag{196}$$

with eigenvalues $\lambda_{1,2,3} = 1 \ , \ 1 \pm j \sqrt{3}\, h$ or in polar form

$$\lambda_{1,2,3} = 1 \ , \ \rho e^{j\psi} \quad , \tag{197}$$

where $\rho = \sqrt{1 + 3h^2}$ and $\psi = \tan^{-1} \sqrt{3}\, h$.

Comparison of Eq. 197 and 189 shows that the algorithm again has no phase error, but it does have a modulus error.

$$\left(1 + 3h^2\right)^{1/2} \approx 1 + 3h^2/2 \quad . \tag{198}$$

$$\epsilon_{mod} \approx 3h^2/2 \quad . \tag{199}$$

For a qualitative measure, this method does have an advantage over the direct numerical method of undetermined coefficients, although hand solution of a cubic equation can be tedious. Also the method lends itself without modification to the cases where axes of approximating transformation and true rotation coincide.

DIGITAL COMPUTER ANALYSIS APPROACH (Ref. 10, 11, 16 and 17)

By using analytic techniques such as those outlined earlier in this report, it is possible to determine at least the orders of magnitude of the errors that occur when an algorithmic approximation to a direction cosine matrix is made.

As discussed previously, it requires considerable effort to progress much beyond simple analytical discussions with these described techniques.

In the several analytical methods discussed in this report for studying the equation $[\dot{C}] = [C][\Omega]$, the point was reached in most cases where the simplifying of approximations had to be made. Otherwise, the arithmetic required would have made these types of solution horrendous to complete.

The solution to a difficulty such as this naturally lies in proper utilization of a digital computer to handle these otherwise tedious and sometimes difficult analysis tasks.

Analysis Method

The digital computer programming scheme suggested for these studies is given by the simplified diagram below (Fig. 10).

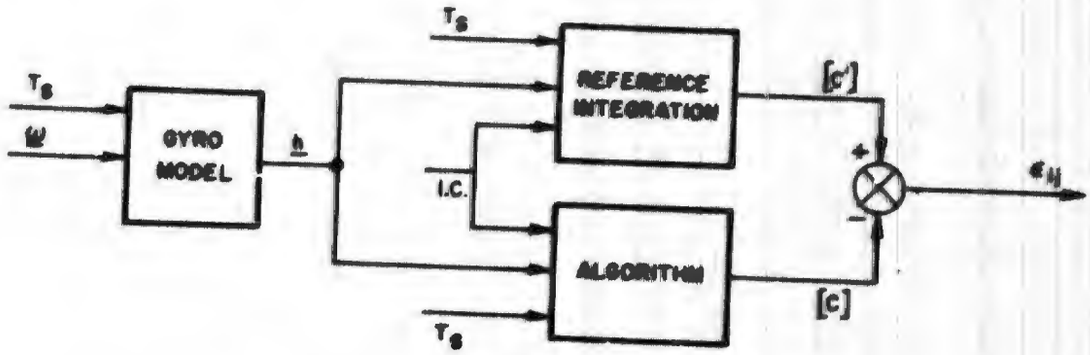


FIG. 10. Algorithm Truncation Error Analysis Program Diagram.

The essential program inputs are the angular velocities

$$\underline{\omega} = \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} .$$

These angular rates are then converted, on the clock pulse, to discrete pulses of weight $h_i = \pm 2^{-n}$, 0. The T_s entries to each block in the diagram indicate that the entire digital program is synchronized to the

gyro model clock period, T_g. If T_g is the gyro clock period and k is the updating cycle number, then the time in operation is given by

$$t = kT_g \quad .$$

The h-pulses represent the discrete three-axis rate information. Each pulse is sent to both the given algorithm form and to a reference updating form. The resultant updated reference direction cosine matrix, $[C']$, and the algorithm updated matrix, $[C]$, are then compared term by term. The program output is the error matrix defined by differencing the individual terms of $[C]$ and $[C']$ in

$$[C'] - [C] = [\epsilon] \quad . \tag{200}$$

Identical initial conditions are supplied to both the reference updating and the algorithm. The term by term comparison is made, in the analysis program, at any desired multiple of updating numbers.

With the digital computer program performing in the manner described here, the error matrix, $[\epsilon]$, is composed of errors primarily resulting from the series truncation of the given algorithm.

Basic Program Equations

The standard in this analysis is the reference updating against which the various algorithm updatings might be compared as a function of the number of updatings, type of inputs, etc. In the investigation of the equation of the form

$$\left|C_{k+1}\right| = \left|C_k\right| [\Phi] \tag{201}$$

where,

$$[\Phi] = e^{[\Omega] T_g} \quad ,$$

$$[\Omega] = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix} \quad .$$

we saw that [Φ] could be expanded in a series, and by proper rearrangement and collection of terms,

$$[\Phi] = [I] + \frac{[\Omega]\, T_s \sin\psi}{\psi} + \frac{[\Omega]^2 T_s^2 (1 - \cos\psi)}{\psi^2} \quad .$$

The preceding is a closed form of the exponential series. For this equation, ψ is as defined in Eq. 86.

By letting

$$h_i = \int_0^{T_s} \omega_i \, dt = \omega_i T_s = t2^{-n} \quad , \quad 0 \text{ for } i = 1,2,3 \quad ,$$

Eq. 89 became Eq. 90, which is repeated below. If we denote by primes the reference updating equation terms, then $\left|C'_{k+1}\right| = \left|C'_k\right|[\Phi']$ is the equation to be programmed on the computer.

$$[\Phi'] = \begin{bmatrix} \frac{h_1^2}{\psi^2} + \frac{h_2^2 + h_3^2}{\psi^2}\cos\psi & \frac{-h_3}{\psi}\sin\psi + \frac{h_1 h_2}{\psi^2}(1 - \cos\psi) & \frac{h_2}{\psi}\sin\psi + \frac{h_1 h_3}{\psi^2}(1 - \cos\psi) \\ \frac{h_3}{\psi}\sin\psi + \frac{h_1 h_2}{\psi^2}(1 - \cos\psi) & \frac{h_2^2}{\psi^2} + \frac{h_3^2 + h_1^2}{\psi^2}\cos\psi & \frac{h_1}{\psi}\sin\psi + \frac{h_2 h_3}{\psi^2}(1 - \cos\psi) \\ \frac{-h_2}{\psi}\sin\psi + \frac{h_1 h_3}{\psi 2}(1 - \cos\psi) & \frac{h_1}{\psi}\sin\psi + \frac{h_2 h_3}{\psi^2}(1 - \cos\psi) & \frac{h_3^2}{\psi^2} + \frac{h_1^2 + h_2^2}{\psi^2}\cos\psi \end{bmatrix}$$

The algorithm equation is of the same basic form as the reference equation. The difference lies in the updating approximation [*H*] used for [ϕ].

$$|C_{k+1}| = |C_k|\,[H] \quad .$$

TRUNCATION ERROR INVESTIGATIONS BY IBM 7094 COMPUTER (Ref. 11, 16, and 17)

The analytical effort in this section is concerned primarily with the algorithmic form itself and not with its mechanization. The justification for employing digital computer analysis techniques to aid in this study have been given in the preceding section, where it was noted that to progress beyond simple analytical discussions required considerable effort. Having reviewed these alternative analytical methods, it was obvious that the digital computer should be added to the repertoire of algorithm analysis tools.

The results of the digital computer analyses include single- and triple-axis representations of first- and second- order algorithms. Similar investigations on particular serial updating rules and on a modified first-order form have been pursued. These results will be shown and discussed.

ASSUMPTIONS FOR INVESTIGATIONS

For the studies discussed in this section, the following assumptions apply:

$$h_i = t2^{-12} \text{ radians, } 0, \text{ for } i = 1, 2, 3$$

$$T_s = (1/2000) \text{ sec (computer cycle time)}$$

$$\omega_{max} = 0.48828125 \text{ rad/sec} \quad .$$

Variations from these are made, but attention is called whenever it is done. For the results shown following, only the simple gyro model has been employed. For the simple inputs tried, there is no noticeable difference in the gyro outputs for the two separate models.

FIRST- AND SECOND-ORDER ALGORITHMS

Based on the developments of the section describing algorithms for updating the direction cosine matrix (Eq. 61), the form of $[\Phi]$ programmed here is given by

$$\left|C_{k+1}\right| = \left|C_k\right|[\Phi] = \left|C_k\right|\left[[I] + [H] + B \cdot \frac{[H]^2}{2}\right] \tag{202}$$

where $B = 0$ or 1 for a first-order approximation or a second-order approximation, respectively.

Computer Flow Diagram of Analysis Program (Fig. 11)

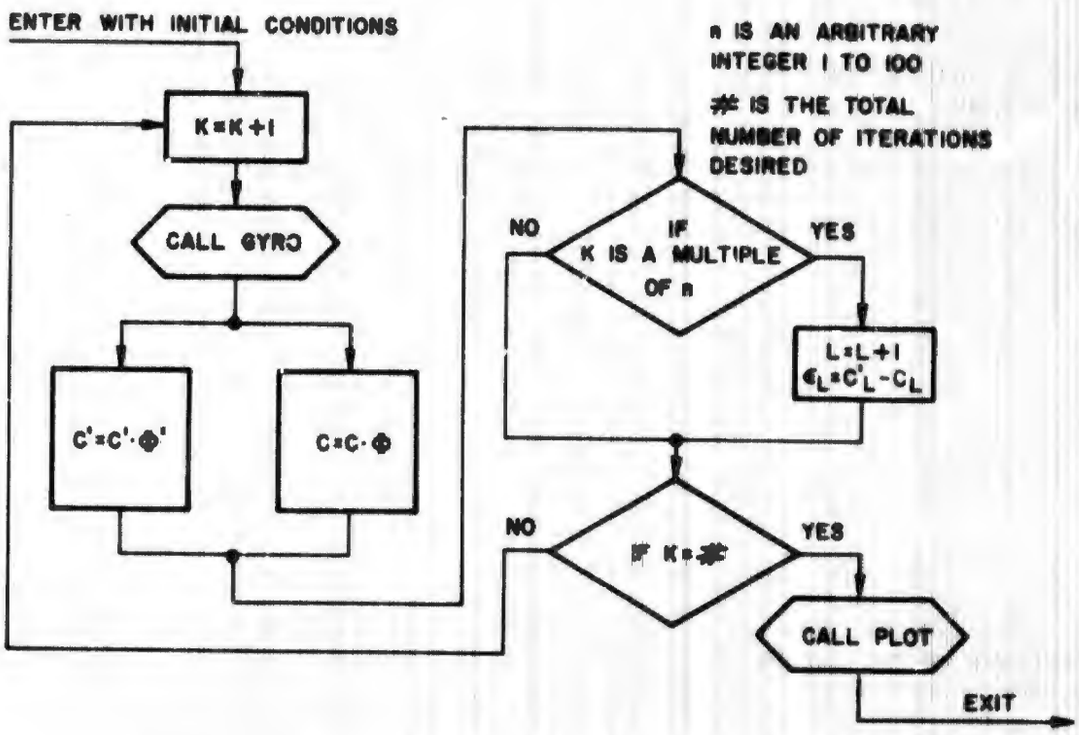


FIG. 11. First- and Second-Order Flow Diagram.

Results—First Order

FIRST ORDER/SINGLE AXIS/MONOTONIC. Figures 12 and 13 are the error traces resulting from two positive monotonic inputs (equal to ω_{max}) applied about the 3-axis. With $\omega_3 = \omega_{max}$, on each clock pulse $h_i = +2^{-12}$. ω_{max} is that ω_{max} as determined by Eq. 43. Corresponding to body roll-pitch and yaw, the axes are numbered as

1-axis = pitch

2-axis = yaw

3-axis = roll .

ω_{max} for $h_3 = 2^{-12}$ is 0.48828125 rad/sec and for $h_3 = 2^{-16}$, ω_{max} = 0.03051758 rad/sec. The initial condition matrix for these figures is unity, for T_s = (1/2000) sec.

$$\left|C_o\right| = [I] \quad ,$$

and the quantitization is $h = 2^{-12}$ radians in Fig. 12 and $h = 2^{-16}$ radians in Fig. 13.

The error growth in all cases is an increasing oscillatory function. These curves agree well with the analytical single-axis solution obtained in Eq. 149.

$$\varepsilon_{11,k} = -\frac{1}{2} h_3 \psi \cos \psi$$

$$\varepsilon_{12,k} = +\frac{1}{2} h_3 \psi \sin \psi \quad ,$$

where $\psi = kh_3$, with k the updating number.

As expected, the truncation error, over the same 80,000 iterations, decreases as the angular increment, h, size decreases. However, 80,000 increments of $h = 2^{-12}$ represents a much larger total angle than does $h = 2^{-16}$ over the same number of updatings. To compare the two sets of error curves, Fig. 12 and 13, a particular angle is chosen

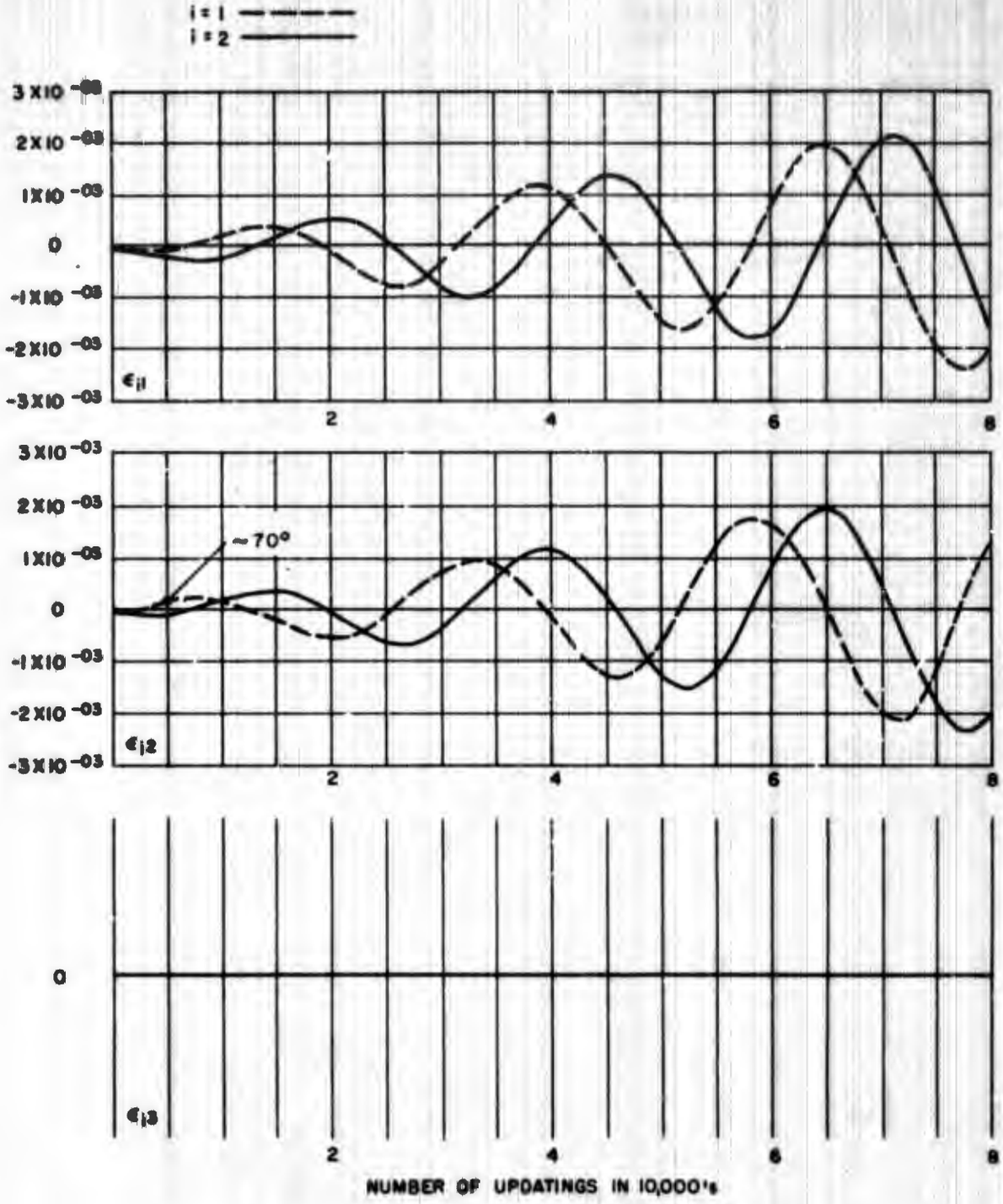


FIG. 12. Error Curves for First-Order Algorithm, Single-Axis Monotonic Input.

$h = 2^{-12}$, $[C(o)] = [I]$.

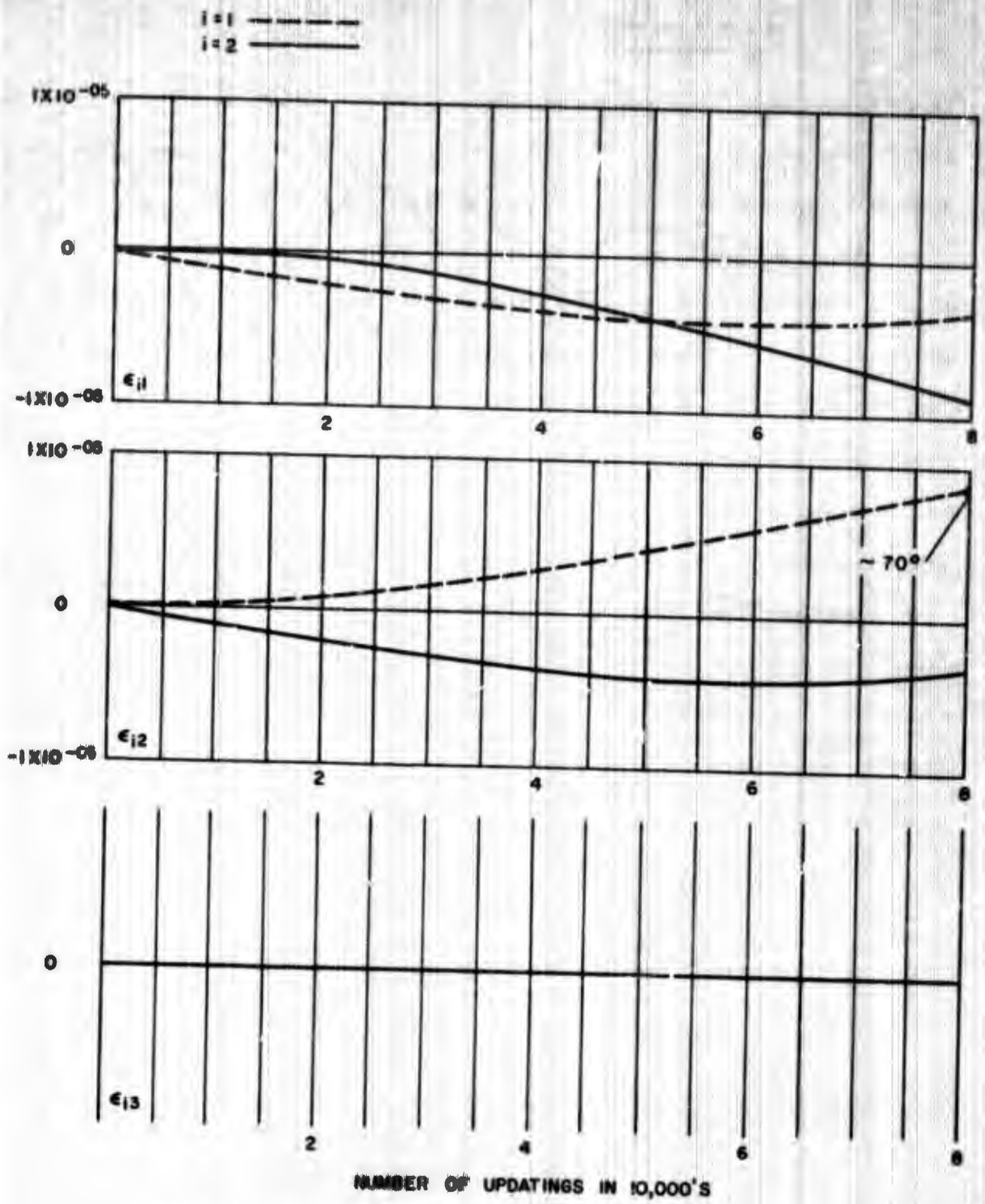


FIG. 13. Error Curves for First-Order Algorithm, Single-Axis Monotonic Input.

$$h = 2^{-16}, \quad [C(o)] = [I] \quad .$$

and then the updating number, at which this angle occurs, is determined. For example, let $kh_3 = 70°$. For $h = 2^{-12}$, this occurs at approximately $k = 5{,}000$ updatings. For $h = 2^{-16}$, 70° requires about 80,000 updatings. For this total angle, the first-order algorithm with a single-axis monotonic input has a truncation error, at 80,000 updatings, of

$$\epsilon_{12}\left(h = 2^{-12}\right) \cong 1.4 \times 10^{-4}$$

$$\epsilon_{12}\left(h = 2^{-16}\right) \cong 0.875 \times 10^{-5} \qquad (203)$$

The truncation error improvement suggests an error improvement factor of 16. By using Eq. 149, it is possible to check the accuracy of the digital program. From Eq. 149

$$\epsilon_{12,k} = \frac{1}{2} h_3 \phi \sin \phi \quad .$$

Setting $k = 80{,}000$ and $h_3 = 2^{-12}$ we get

$$\epsilon_{12} \cong 1.4 \times 10^{-4}$$

For $h_3 = 2^{-16}$

$$\epsilon_{12} \cong 0.875 \times 10^{-5} \quad .$$

FIRST-ORDER, SINGLE-AXIS, LIMIT CYCLE. The first order single-axis error response to a 1:1 limit cycle is shown in Fig. 14 and 15 for $h = 2^{-12}$. A 1:1 limit cycle is defined as a series of gyro pulses with signs alternating every cycle:

$$\ldots\ldots, +h, -h, +h, -h, +h, -h, +h, \ldots\ldots \qquad (204)$$

In the single-axis case, if the initial condition matrix $[C(o)]$ is unity,

$$[C(o)] = [I] \quad ,$$

no off-diagonal error can ever result for limit-cycle inputs.

$$|C_1| = [C(o)] \cdot \begin{bmatrix} 1 & -h_3 & 0 \\ h_3 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & -h_3 & 0 \\ h_3 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

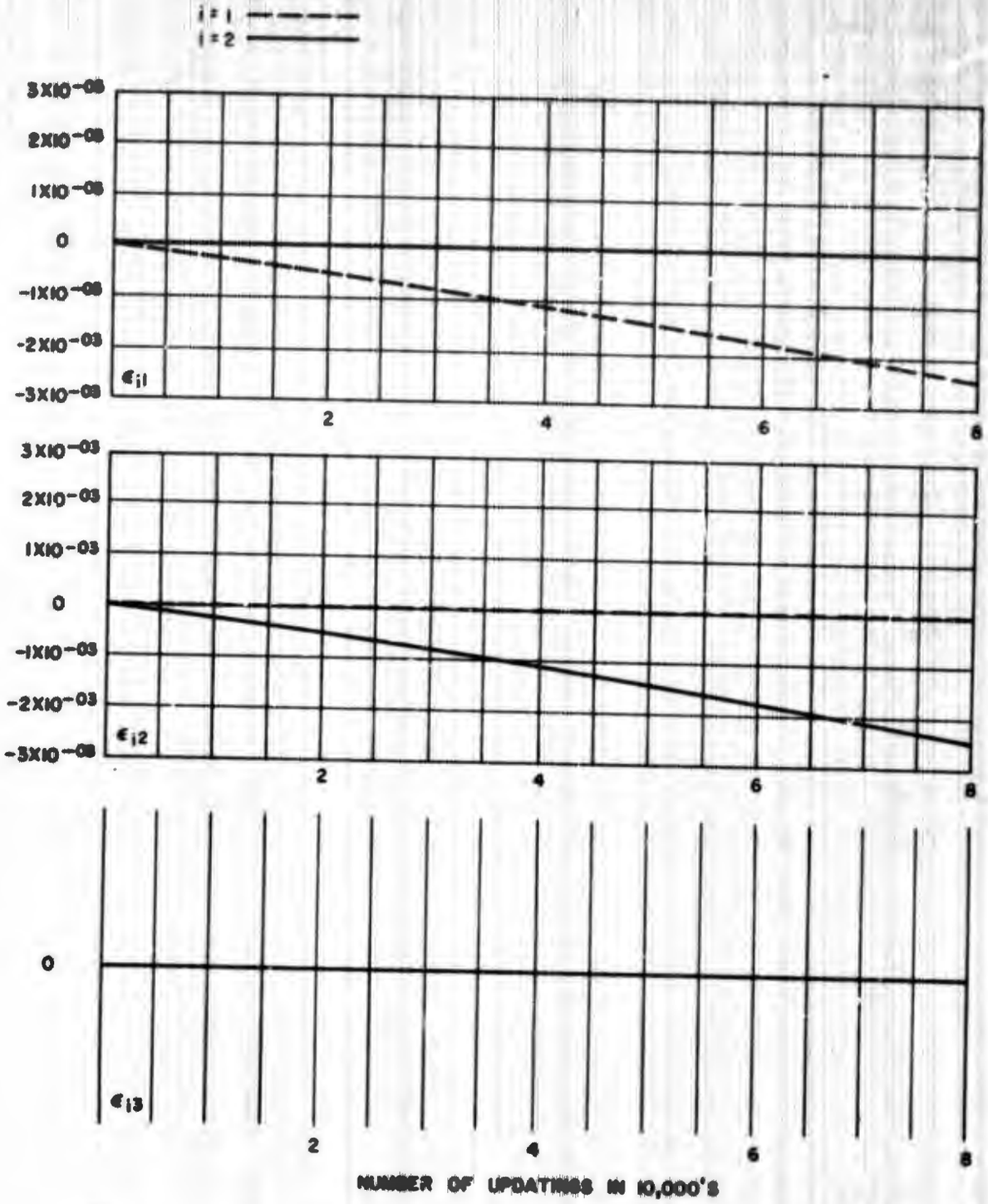


FIG. 14. Error Curves for First-Order Algorithm, Single-Axis Input Limit Cycle.

$$h = 2^{-12}, \quad [C(o)] = [I] \quad .$$

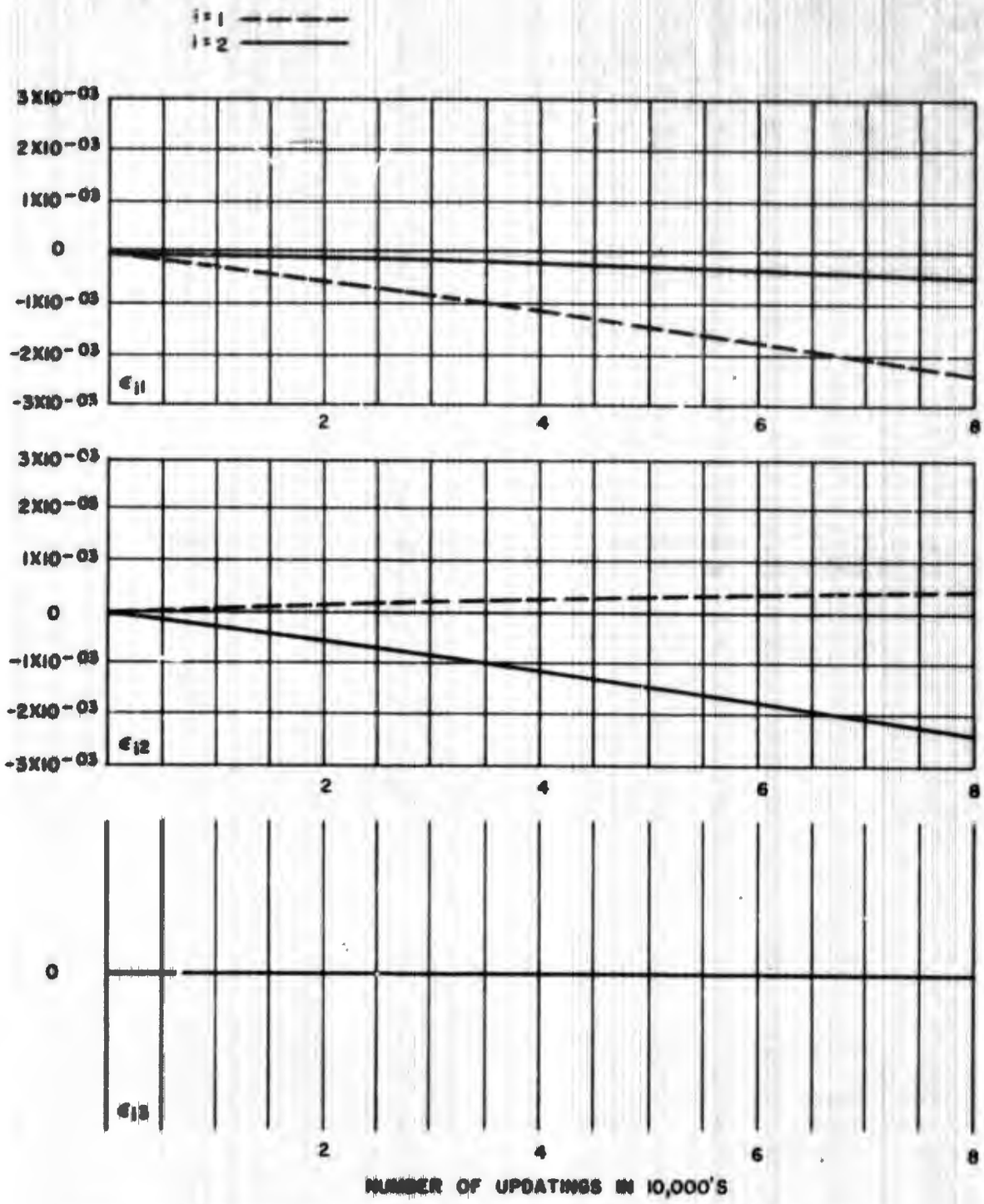


FIG. 15. First-Order Algorithm. Single-Axis Input Limit Cycle.

$$h = 2^{-12}, \quad |C(0)| = \begin{bmatrix} \cos 11^\circ & -\sin 11^\circ \\ \sin 11^\circ & \cos 11^\circ \end{bmatrix}$$

$$|C_2| = |C_1| \cdot \begin{bmatrix} 1 & h_3 & 0 \\ -h_3 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \left(1+h_3^2\right) & 0 & 0 \\ 0 & \left(1+h_3^2\right) & 0 \\ 0 & 0 & 1 \end{bmatrix} , \tag{205}$$

and so on. Figure 15 used the initial condition

$$|C_o| = \begin{bmatrix} \cos 11^\circ & -\sin 11^\circ & 0 \\ \sin 11^\circ & \cos 11^\circ & 0 \\ 0 & 0 & 1 \end{bmatrix} . \tag{206}$$

With $h = 2^{-12}$, the monotonic error bound because of a maximum input at 80,000 updatings was about (Fig. 12)

$$\epsilon_{11}\ (bound) \cong \pm 2.4 \times 10^{-3}$$

$$\epsilon_{12}\ (bound) \cong \pm 2.4 \times 10^{-3} . \tag{207}$$

In Fig. 15, the limit-cycle error at 80,000 iterations is

$$\epsilon_{11} \cong -2.4 \times 10^{-3}$$

$$\epsilon_{12} \cong -2.4 \times 10^{-3} . \tag{208}$$

FIRST-ORDER, TRIPLE-AXIS, MONOTONIC MAXIMUM RATE AND 1:1 LIMIT CYCLE. In the next two sets of curves, the results of subjecting the first-order updating to equal maximum monotonic and 1:1 limit cycle inputs about all three axes simultaneously are shown, $h_1 = h_2 = h_3 = +2^{-12}$.

For the monotonic inputs, Fig. 16 reveals an error bound at 80,000 updatings of

$$\epsilon_{ij} \cong \pm 4.8 \times 10^{-3} \quad , \quad \text{for} \quad i, j = 1, 2, 3. \tag{209}$$

With a 1:1 limit-cycle input (Fig. 17), the diagonal errors were

$$\epsilon_{11} = \epsilon_{22} = \epsilon_{33} \cong -4.8 \times 10^{-3} \quad . \tag{210}$$

The off-diagonal terms appeared as

$$\epsilon_{ij} \ (i \neq j) \cong 2.5 \times 10^{-3} \quad . \tag{211}$$

UNEQUAL RATES, TRIPLE-AXIS, FIRST-ORDER RESPONSE. Figure 18 demonstrates the first-order error response to the body rates $\omega = 0.025$, $\omega_2 = 0.0025$, and $\omega_3 = 0.05$ (rad/sec).

Results—Second Order.

The second-order algorithm was studied only for the triple-axis maximum-rate monotonic and 1:1 limit-cycle cases. It was felt that sufficient insight had been gained from the first-order rule that it would have been unnecessarily repetitious to make the same sequence of runs.

It is obvious that the second-order algorithm, being a second refinement of the same series used for the first-order algorithm, will have an error response similar in form but smaller in magnitude than the first-order results. This is demonstrated by comparing Fig. 19 and 20 with their first-order counterparts.

The error bound for the maximum-rate monotonic inputs at 80,000 updatings is

$$\epsilon_{ij} \cong 0.7 \times 10^{-6} \quad , \quad i, j = 1, 2, 3, \tag{212}$$

and for the 1:1 limit cycle

$$\epsilon_{ij} \cong -2.2 \times 10^{-10}$$

$$\epsilon_{ij} \ (i \neq j) \cong 1.1 \times 10^{-10} \quad , \quad i, j = 1, 2, 3. \tag{213}$$

The results obtained by applying input angular rates of $\omega_1 = 0.025$, $\omega_2 = 0.0025$, $\omega_3 = 0.05$ (rad/sec) are shown in Fig. 21.

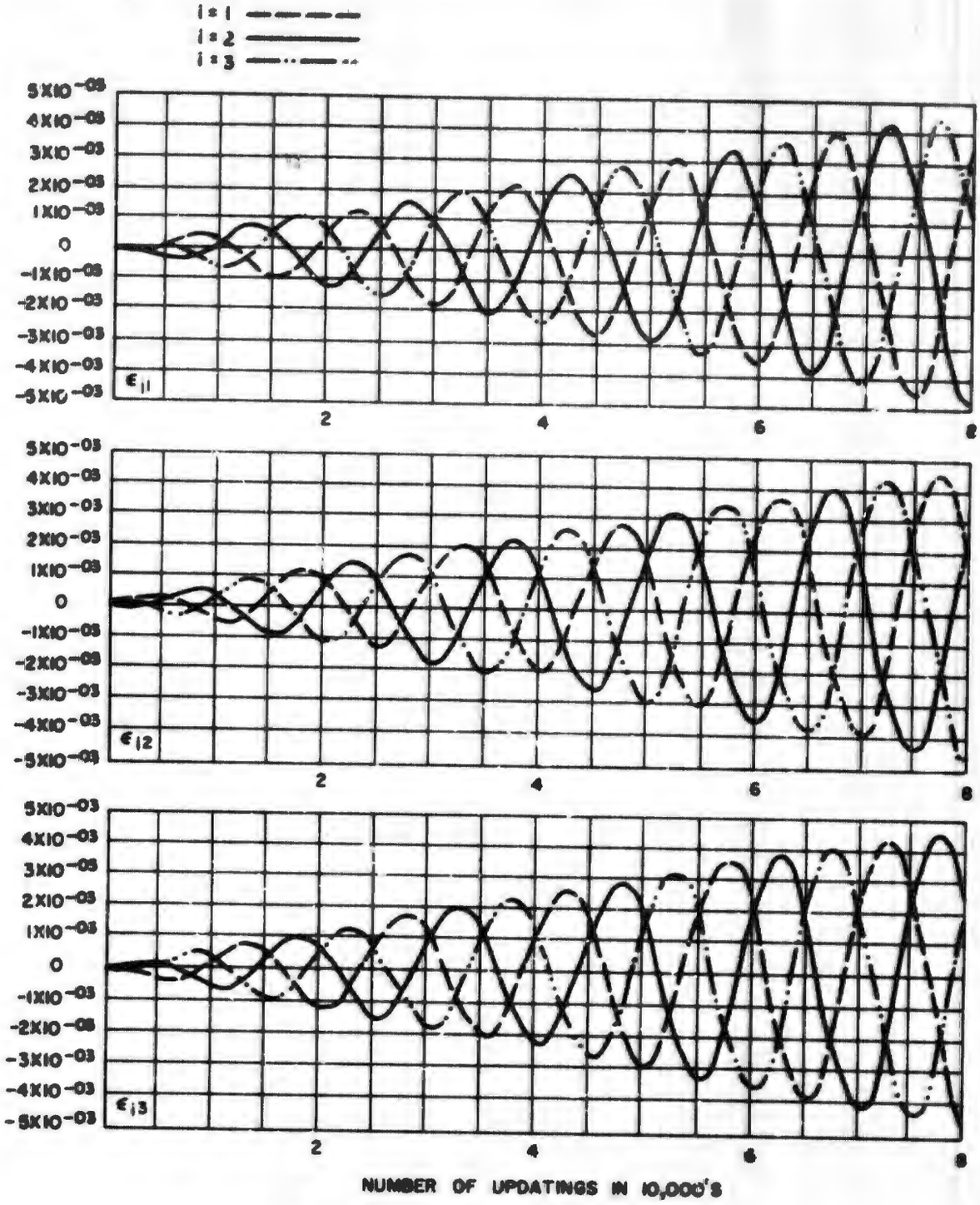


FIG. 16. Error Curves for First Order Algorithm, Triple-Axis Input, Monotonic.

$h = 2^{-12}$, $[C(o)] = [I]$.

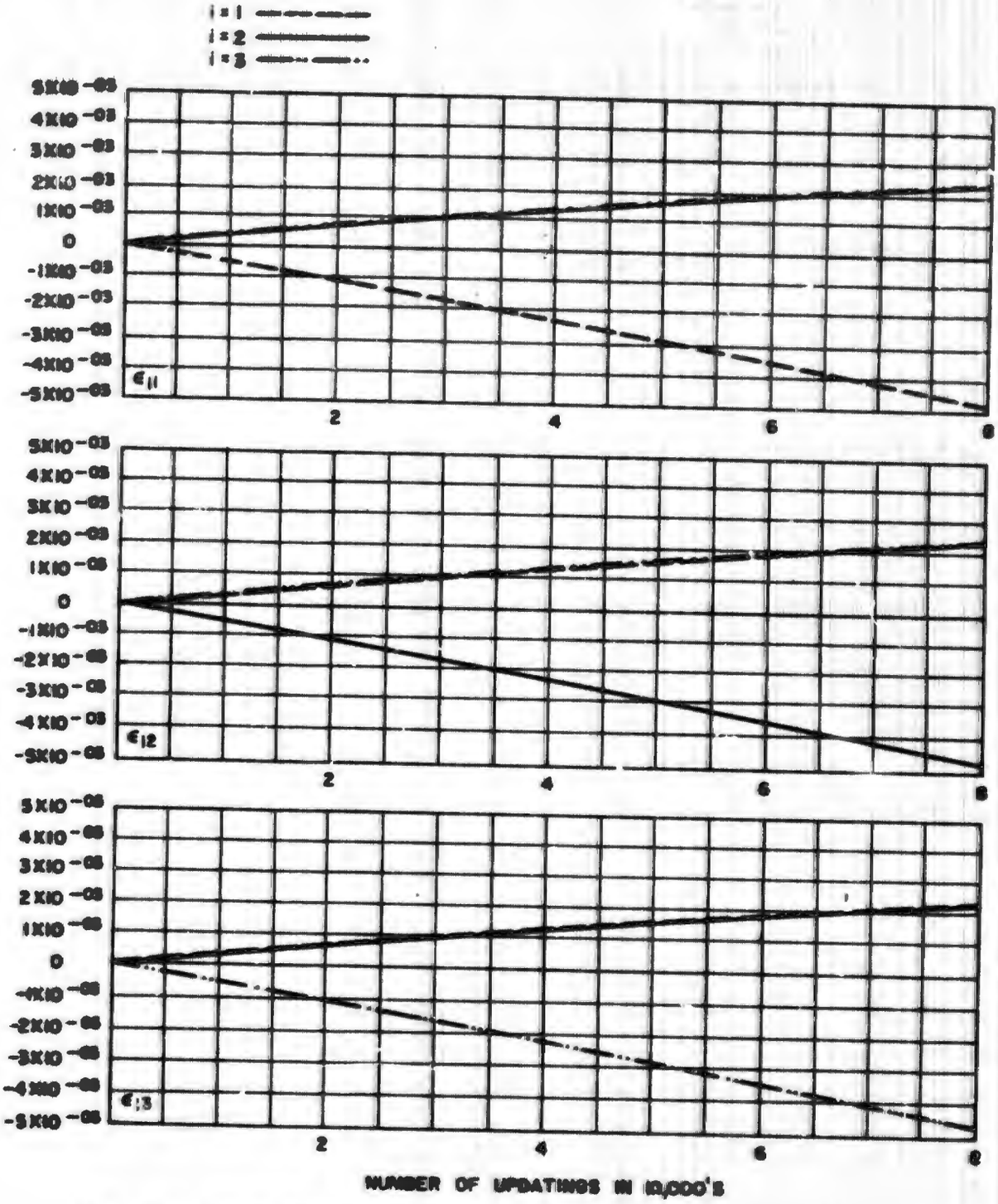


FIG. 17. Error Curves for First Order Algorithm. Triple-Axis Input, Limit Cycle.

$$h = 2^{-12}, \quad [C(o)] = [I] \quad .$$

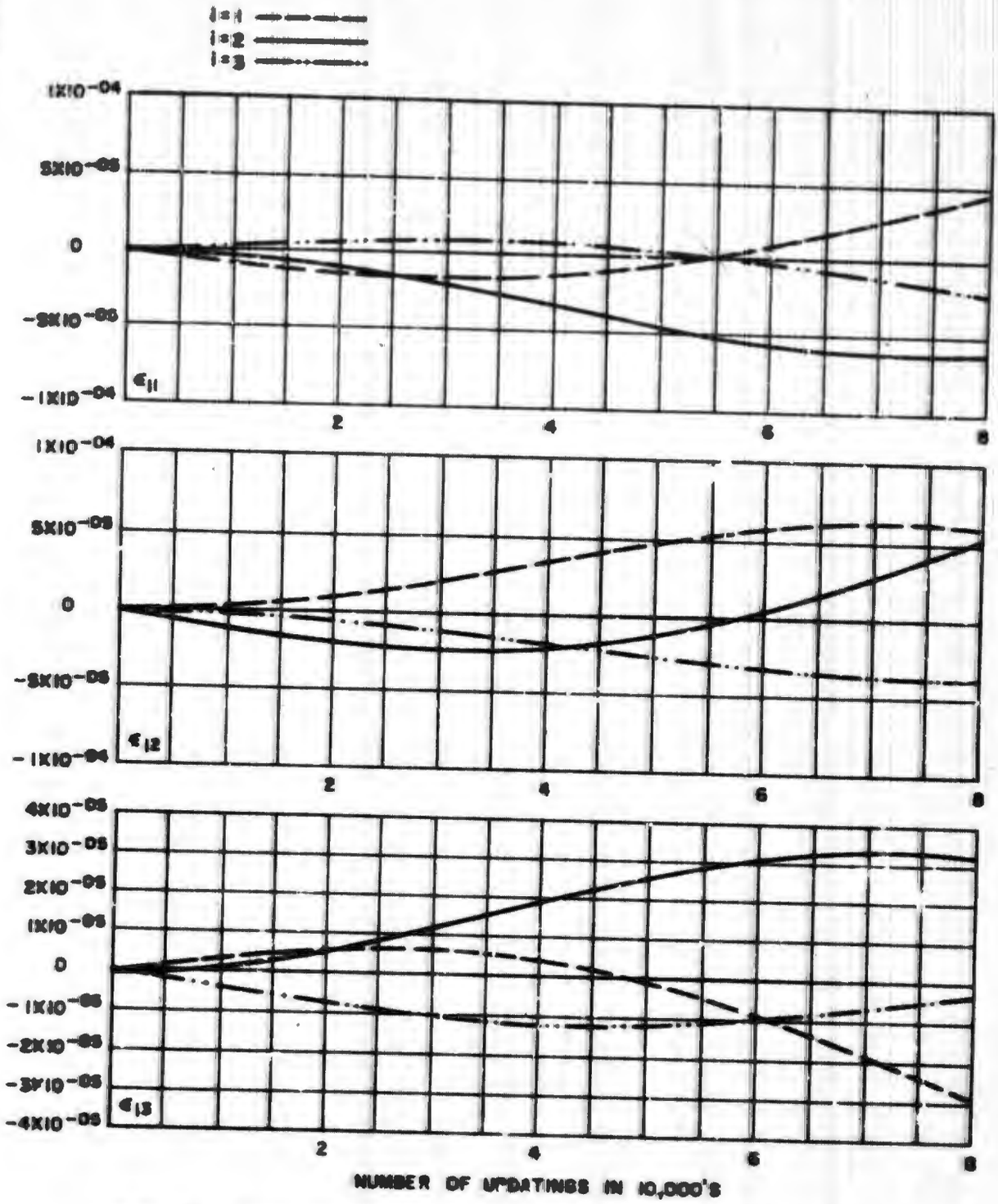


FIG. 18. Error Curves for First-Order Algorithm, Gyro Model.

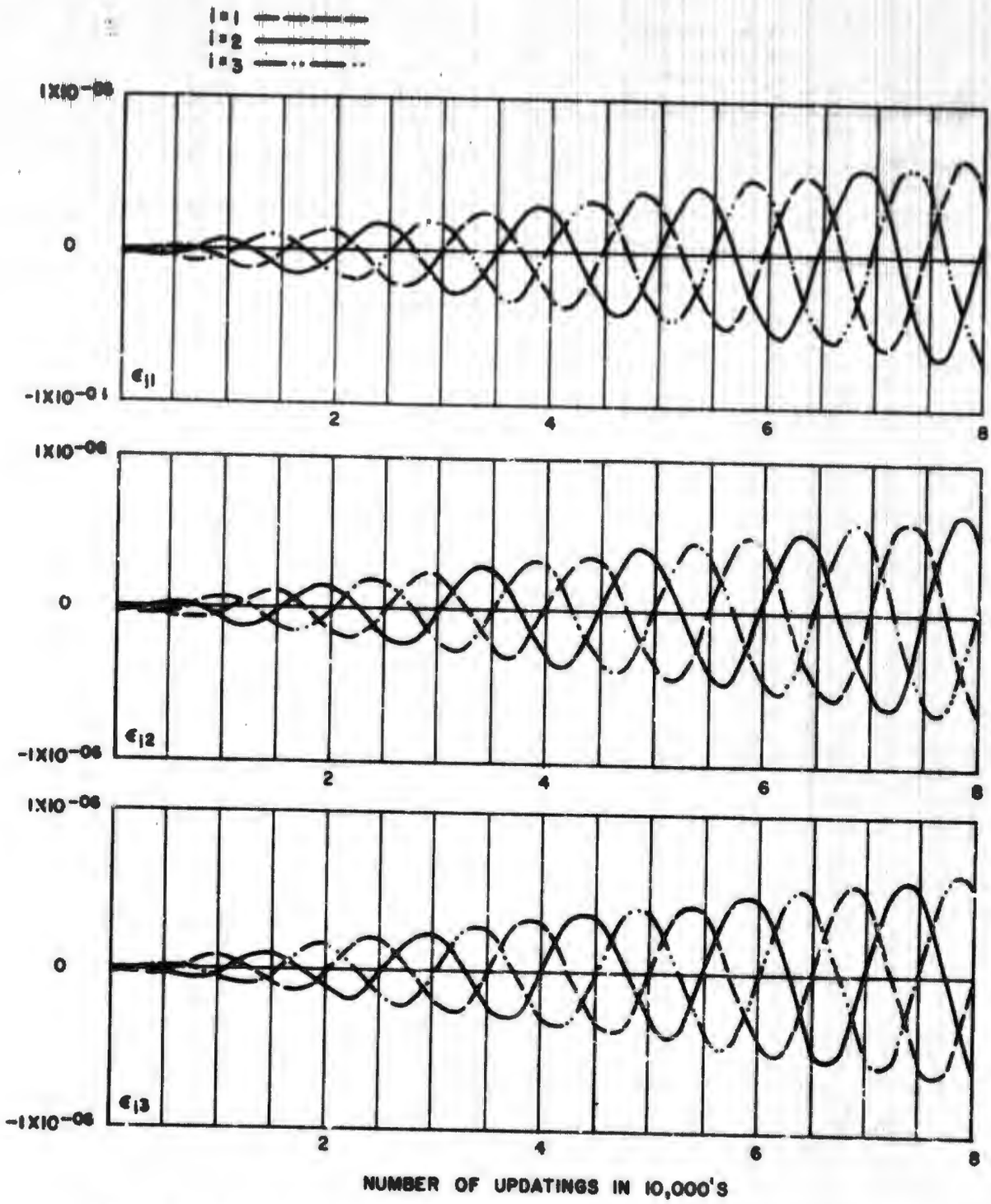


FIG. 19. Error Curves for Second-Order Algorithm. Triple-Axis Monotonic Input.

$$h = 2^{-12}, \quad [C(o)] = [I] \quad .$$

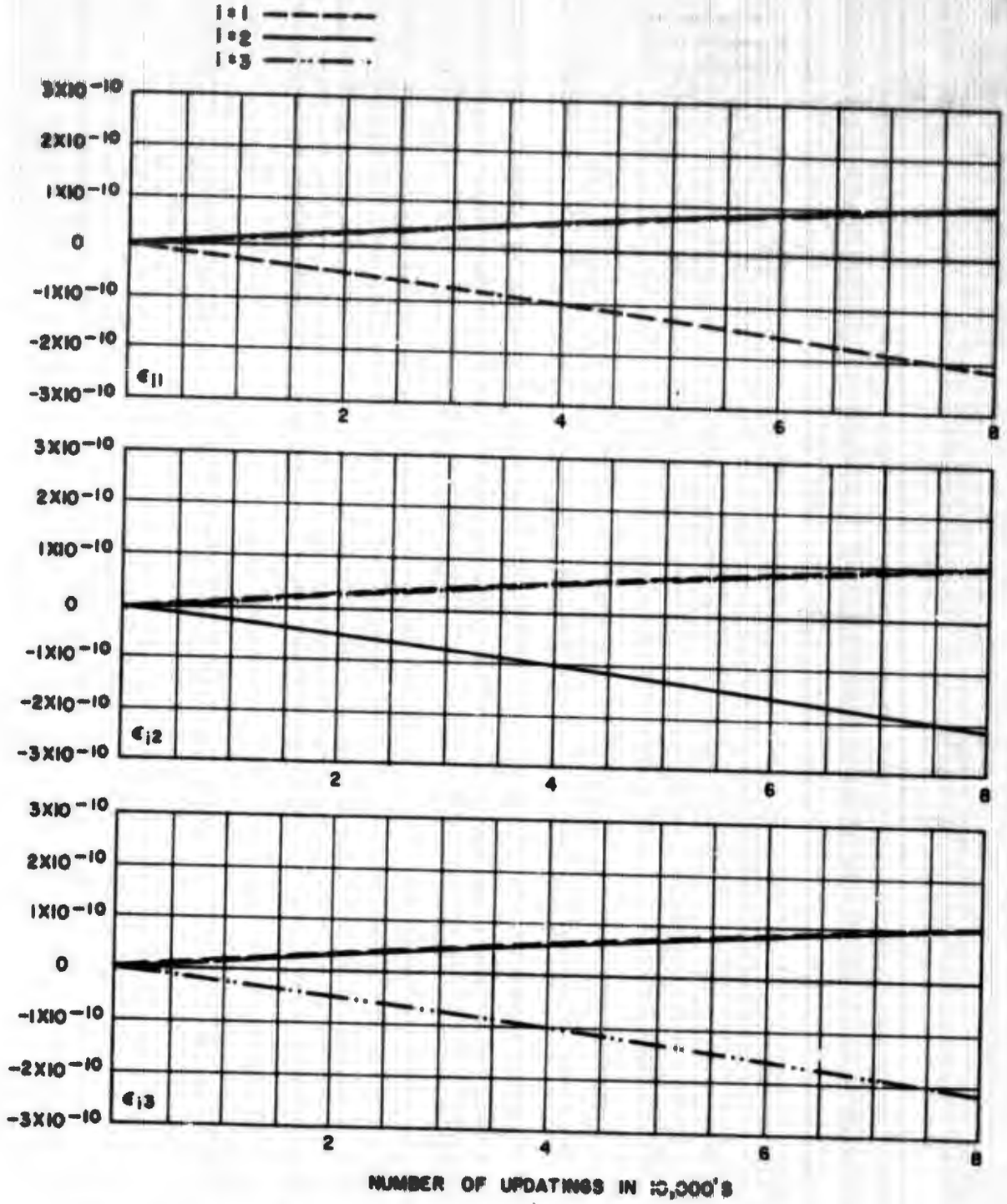


FIG. 20. Error Curves for Second-Order Algorithm, Triple-Axis Input Limit Cycle.

$$h = 2^{-18}, \quad [C(o)] = [I] \quad .$$

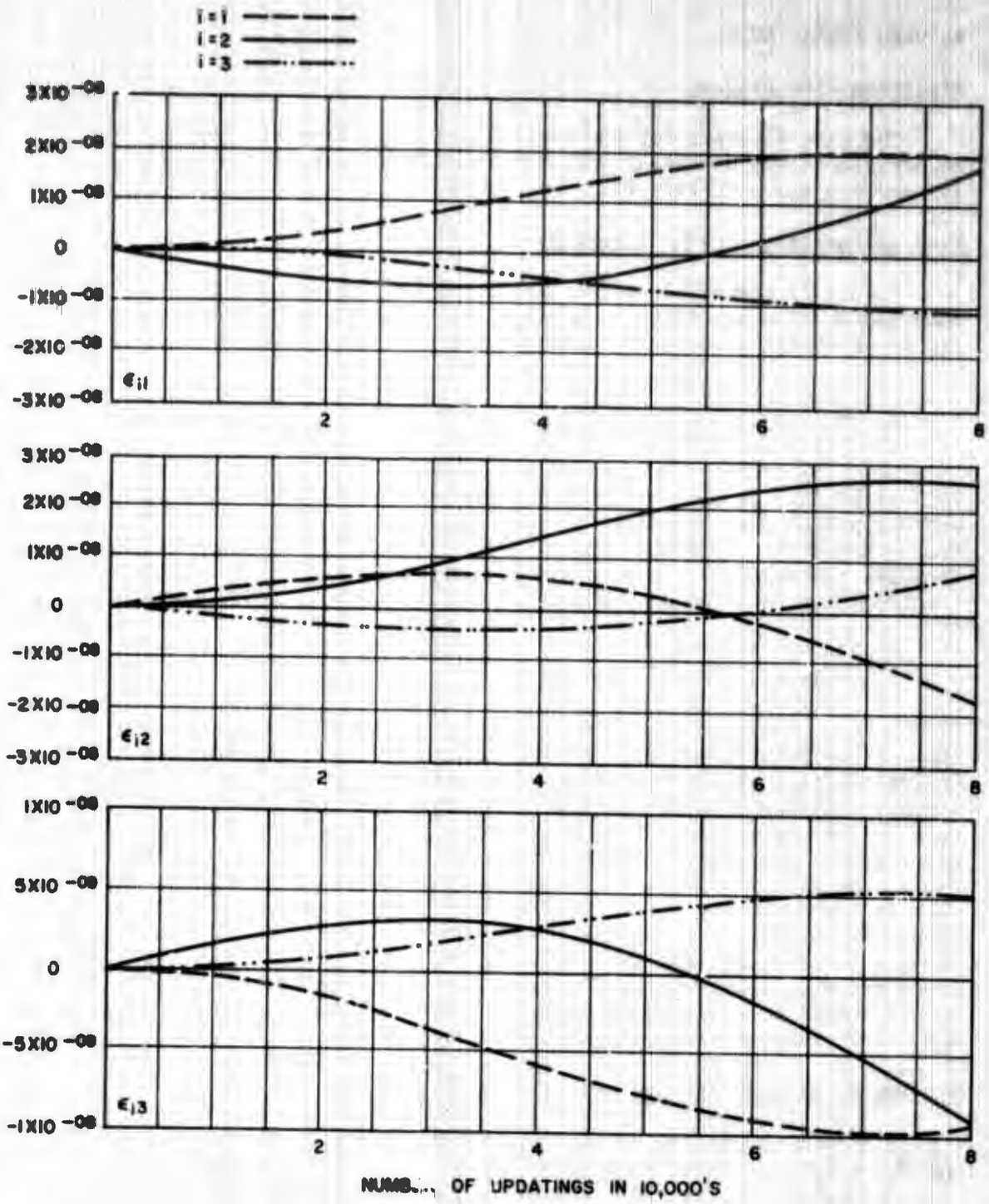


FIG. 21. Error Curves for Second-Order Algorithm, Gyro Model.

RESULTS-SERIAL RULES

Algorithm Flow Diagram

The flow diagrams for the Bumstead-Hession, Crowder-Hession, and Marmon algorithms are identical, except for the form of the matrix [Φ] and are shown in Fig. 22.

Results-Bumstead-Hession Algorithm

Figure 23 shows the error response of the forward portion, M_f, of the Bumstead-Hession algorithm when subjected to equal maximum-rate monotonic inputs about all three axes. The error bounds at 80,000 updatings were

$$\epsilon_{ij} \cong \pm 1 \times 10^{-3} \quad .$$

This gives the forward part of the Bumstead-Hession algorithm, as described previously in the section on algorithm form variations, a slight truncation error advantage over a simple first-order scheme. Figure 24 includes the reversal portion [M_r] alternatively with [M_f] subjected to monotonic inputs. The 80,000 point error bounds were approximately

$$\epsilon_{ij} \cong \pm 4 \times 10^{-8} \quad .$$

These error values made the resultant algorithm form approximately equivalent to a third-order parallel algorithm. Over the 80,000 iterations tried, the truncation error, which is predicted analytically to be of the order of h^4, was not observed to diverge. To observe any error growth would have required about 2^{36} iterations. For a 1:1 limit cycle, the eigenvalues of the Bumstead-Hession algorithm after each $(+h, -h)$ sequence are unity. Figure 25 shows zero error in off-diagonal terms and, at 80,000 updatings, about 10^{-11} error in the diagonal terms. It is concluded that the error shown along the diagonal is due to 7094 round-off accumulation rather than to the algorithm itself. Figure 26 is the result of applying an input rate $\omega_1 = \omega_{max} = 10$ $\omega_2 = 100$ $\omega_3 = 0.48828125$ rad/sec to the Bumstead-Hession algorithm.

Crowder-Hession Algorithm Results

Figure 27 shows the Crowder-Hession algorithm error response to a maximum-rate monotonic input about three axes. The 1:1 limit-cycle result is shown in Fig. 28. The same comment as was made previously about the Bumstead-Hession limit cycle main diagonal errors hold for this case also. Figure 29 is the result of the input $\omega_1 = \omega_{max} = 10$ $\omega_2 = 100$ $\omega_3 = 0.48828125$ rad/sec.

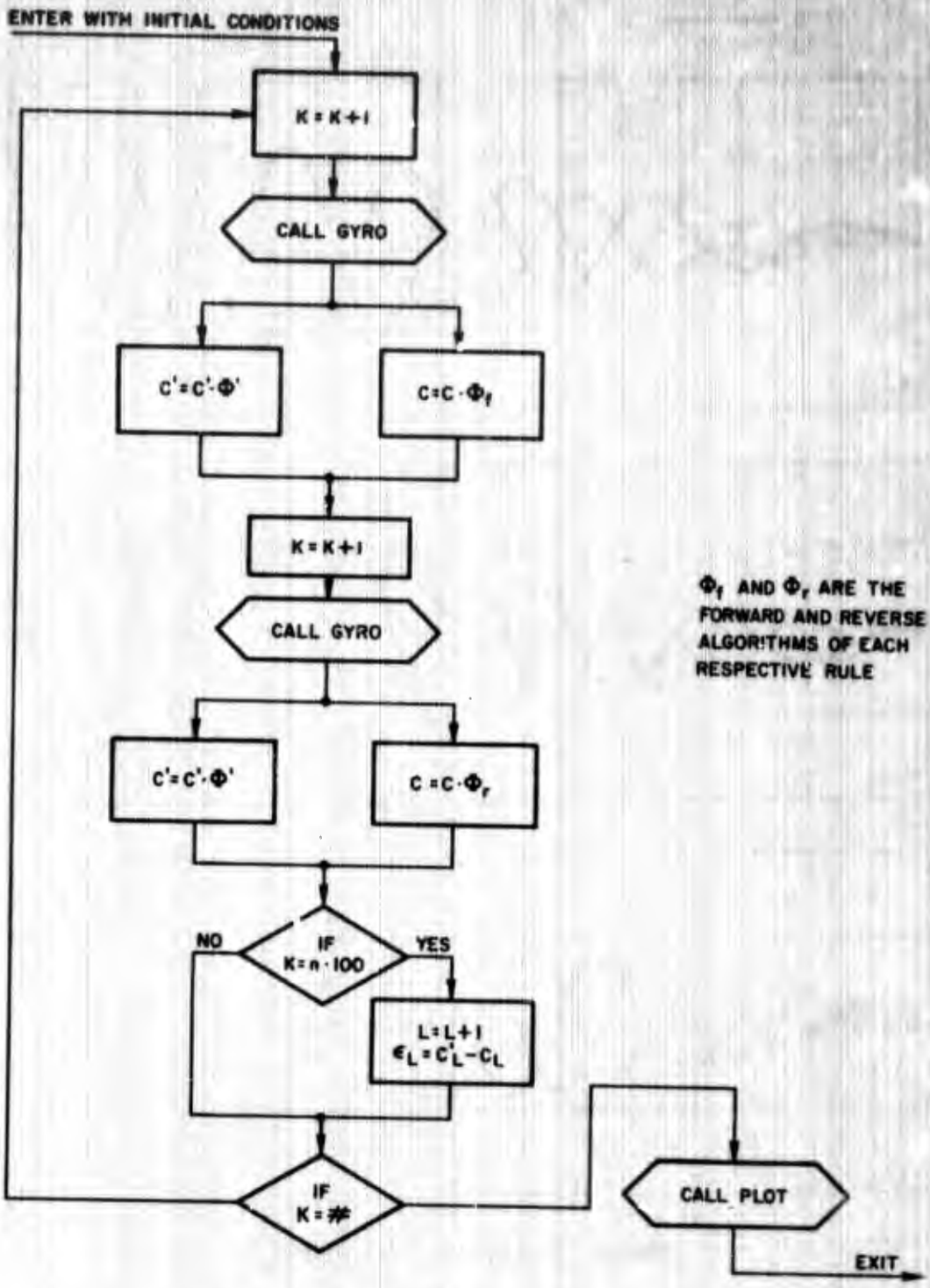


FIG. 22. Serial Algorithm Flow Diagram.

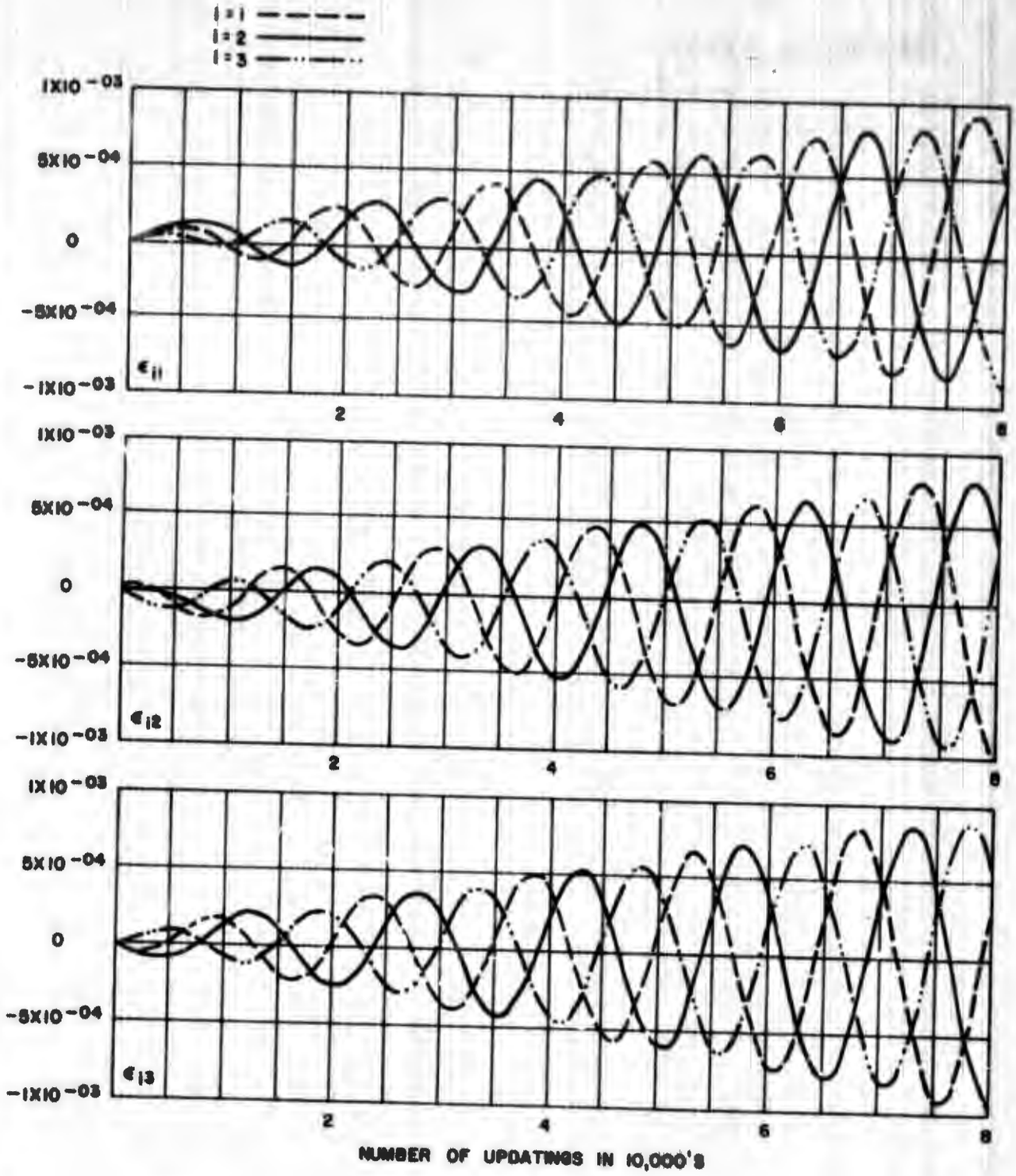


FIG. 23. Error Curves for Bumstead-Hession Algorithm. Triple-Axis Input Monotonic. Forward Algorithm Only.

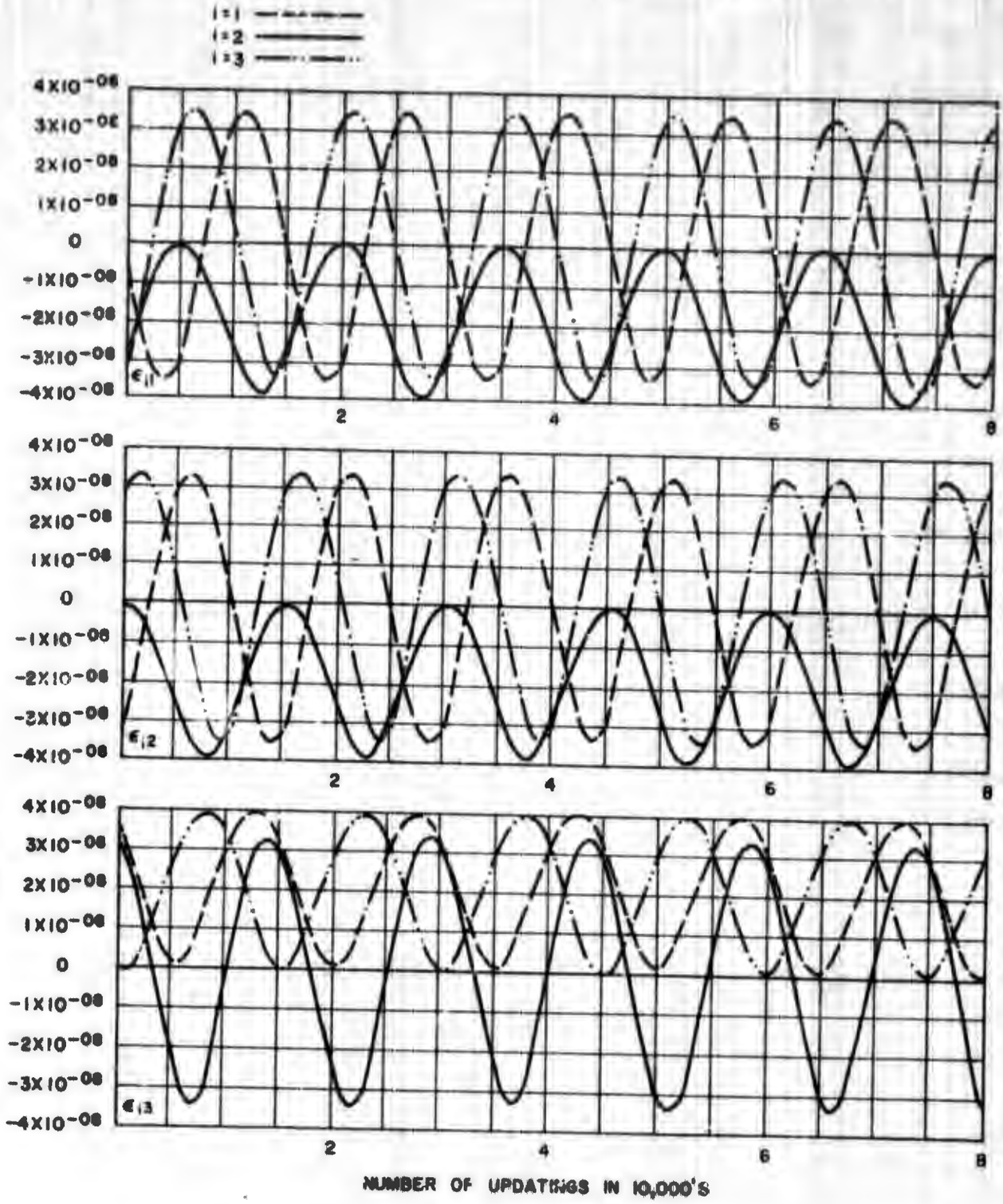


FIG. 24. Error Curves for Bumstead-Hession Algorithm. Triple-Axis Input Monotonic. Forward and Reverse.

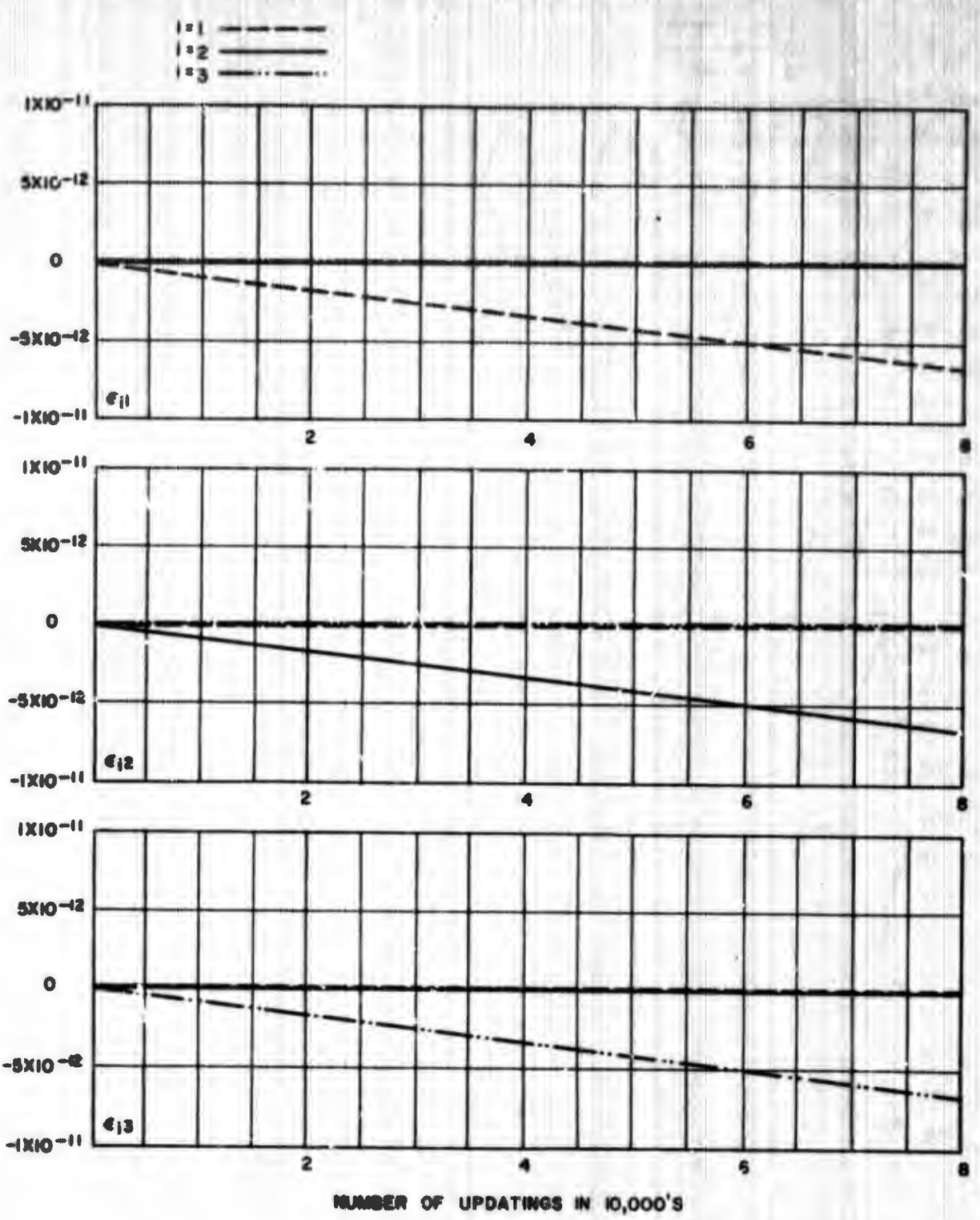


FIG. 25. Error Curves for Bumstead-Hession Algorithm. Triple-Axis Input Limit Cycle. Forward and Reversed.

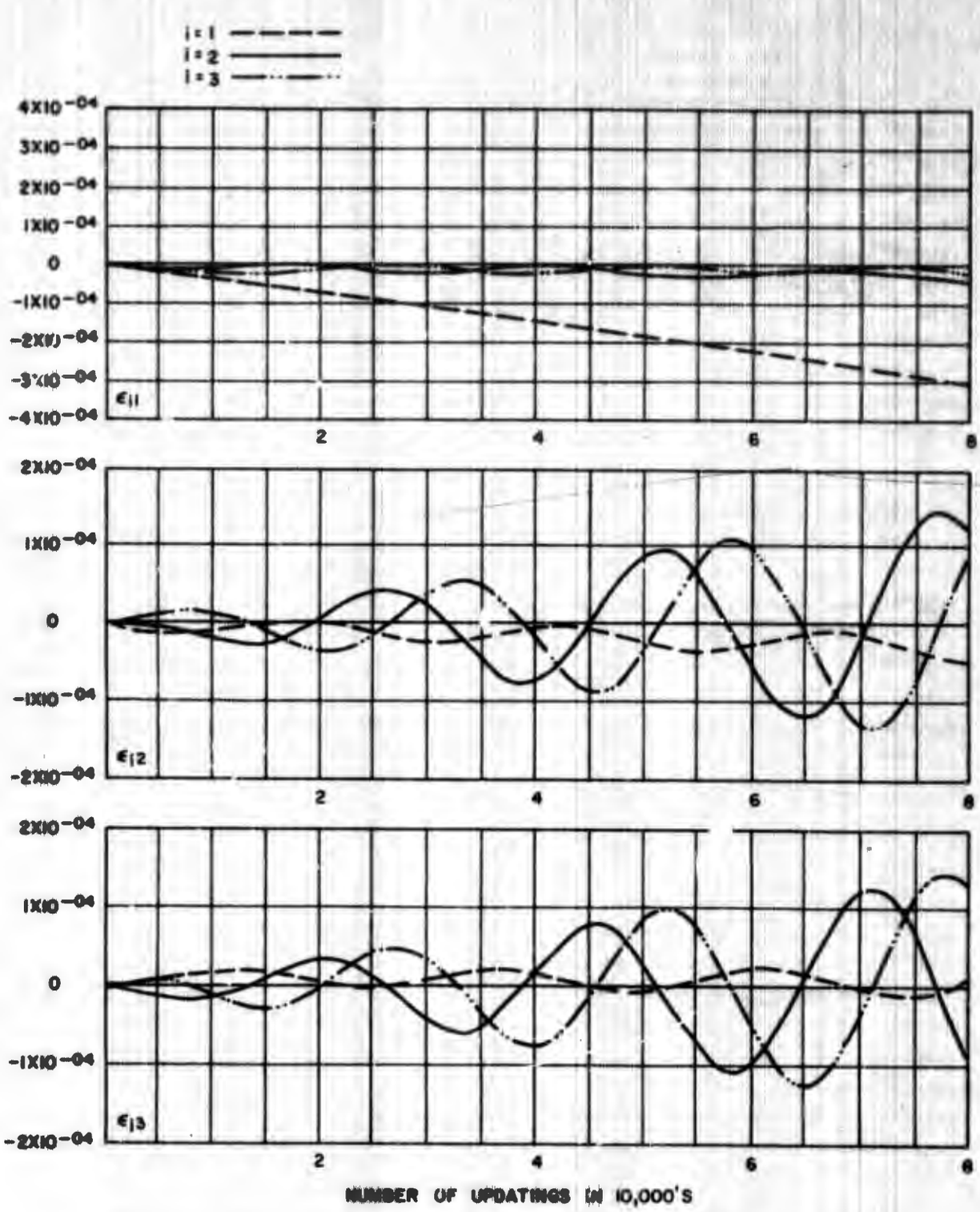


FIG. 26. Error Curves for Bumstead-Hession Algorithm Gyro Model.

$\omega_1 = \omega_{max} = 10\ \omega_2 = 100\ \omega_3$; $\omega_{max} = 0.48828125$.

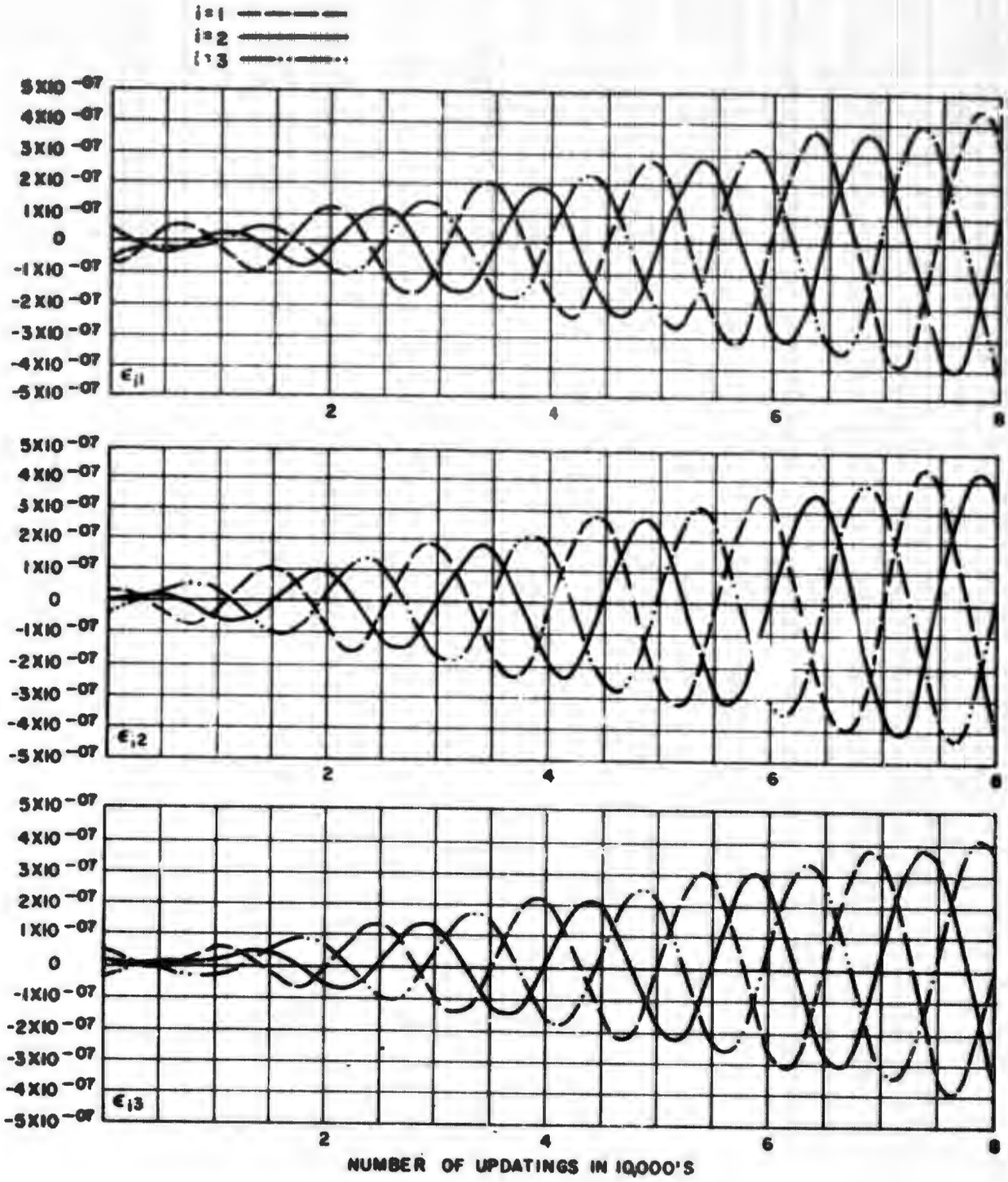


FIG. 27. Error Curves for Crowder-Hession Algorithm. Triple-Axis Input Monotonic. Forward and Reversal.

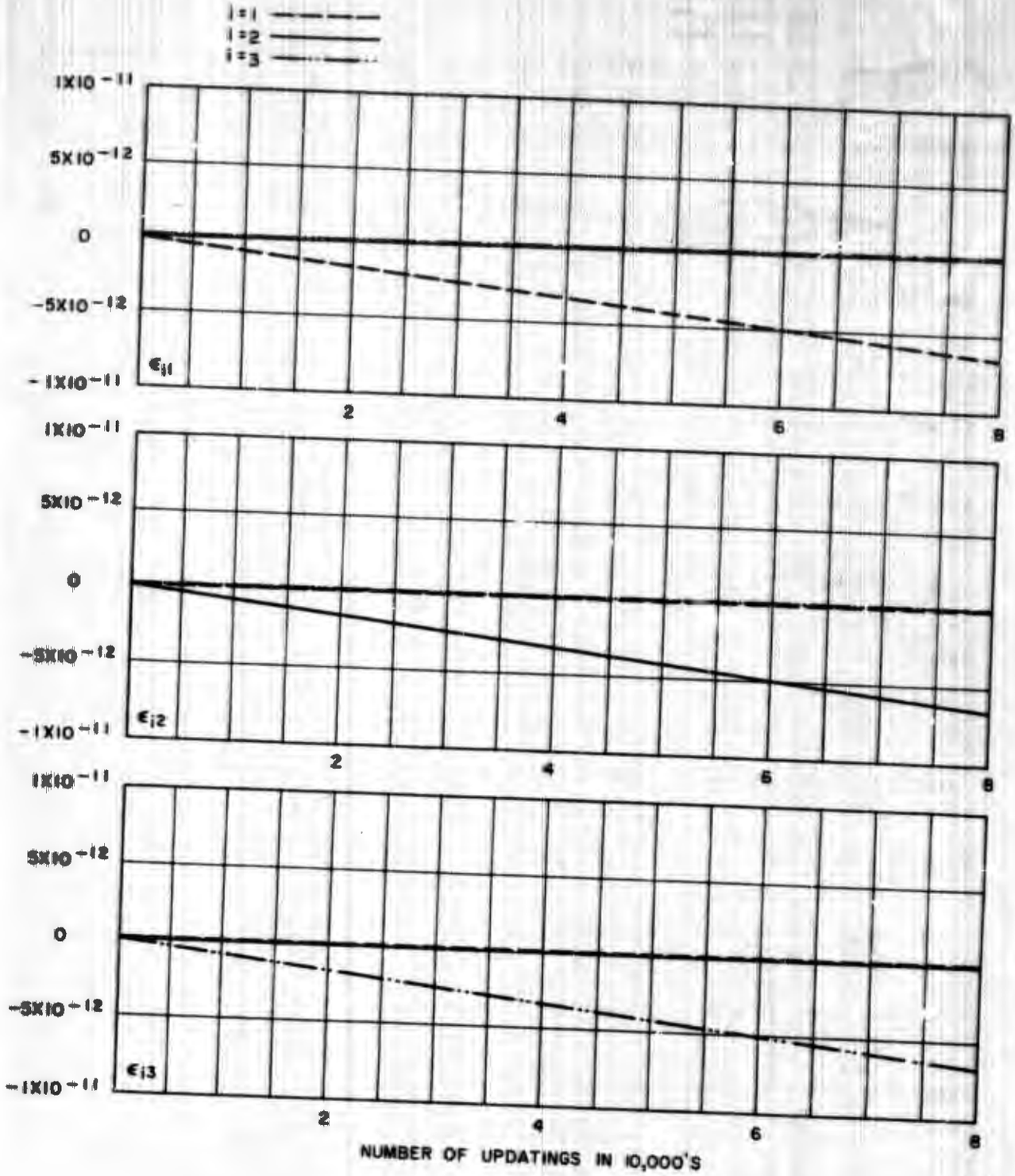


FIG. 28. Error Curves for Crowder-Hession Algorithm. Triple-Axis Input Limit Cycle. Forward and Reversal.

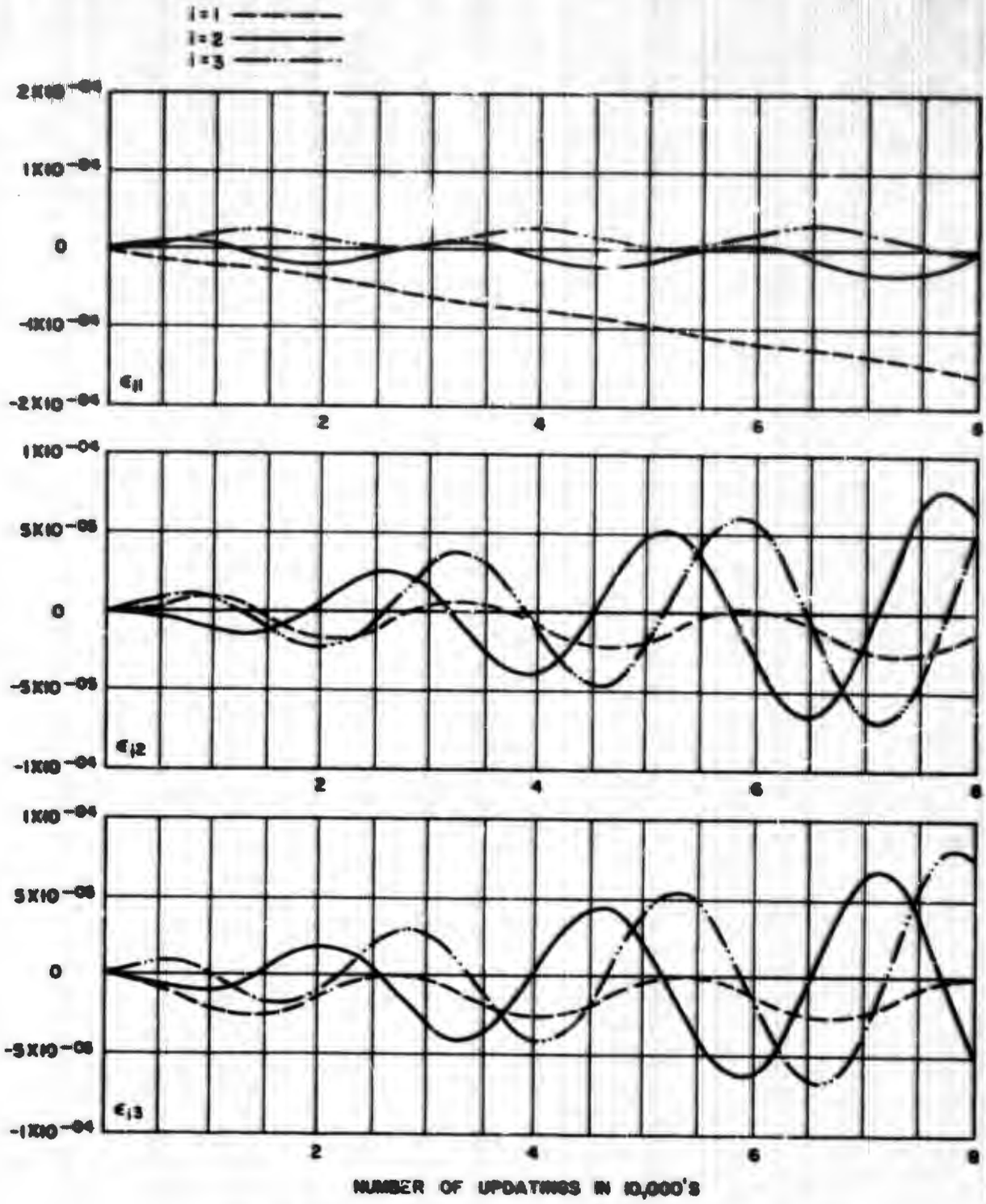


FIG. 29. Error Curves for Crowder-Session Gyro Model

$\omega_1 = \omega_{max} = 10\ \omega_2 = 100\ \omega_3$; $\omega_{max} = 0.$ [illegible] d/sec.

Marmon Algorithm Results

The results obtained by applying a monotonic rate equally about three axes is displayed in Fig. 30. The error is oscillating in all nine cosines between zero and a linearly growing upper bound. Attention is called to this error rectification for this special case. If a 1:1 limit cycle is applied, the error at 80,000 updatings would be

$$\varepsilon_{11} = \varepsilon_{22} = \varepsilon_{33} = 0$$

$$\varepsilon_{ij}\ (i \neq j) \cong 2.4 \times 10^{-3} \quad .$$

Finally, Fig. 31 shows the Marmon algorithm error response to $\omega_1 = 0.025$, $\omega_2 = 0.0025$, $\omega_3 = 0.05$ (rad/sec).

DISCUSSION AND SUMMARY

We review briefly the material covered in this report. Following an introduction, which gave a broad overview of the problems of strapped-down inertial guidance, attention was focused on the problems of updating the transformation matrix. Briefly discussed were schemes for mechanizing simple models of the strapped-down gyros. These models served as the sources of angular information for the updating algorithm analysis. A simple digital computer mechanization, which could effect the updating of direction cosines using only SHIFT and ADD operations, was discussed. The scope of the report was narrowed by introducing and discussing the transformation method of direction cosines. With the aid of the preceding information, it was then possible to make a definite statement of the direction-cosine updating problem: "... perform a numerical integration of [the direction cosine differential equation] $[\dot{C}] = [C]\ [\Omega]$ given quantitized angular rate information."

Sufficient information and concepts were available to begin to study algorithms useful in integrating $[\dot{C}] = [C]\ [\Omega]$. Some of the more conventional integration schemes, such as rectangular integration, trapezoidal integration, and Simpson's 1/3 rule, were studied. The Taylor series proved to be of particular interest and was seen to be an extremely useful tool for forming integration rules that did not require closed-loop solution. Several variations on the Taylor series algorithm form were studied. These studies included serial processing algorithms and the technique of processing reversal under limit-cycle conditions. Sequential pulse-processing and pulse-grouping schemes were also reviewed.

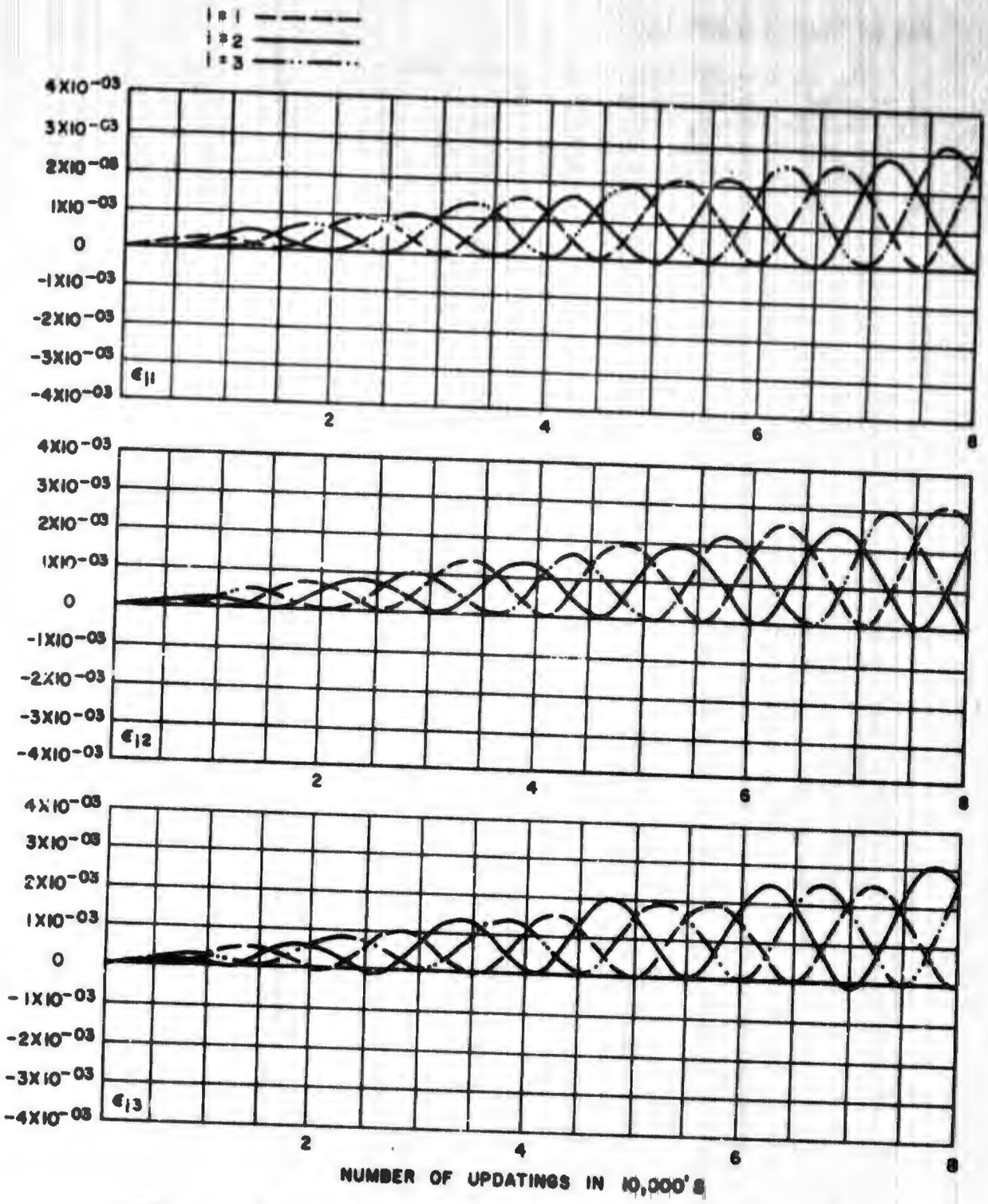


FIG. 30. Error Curves for Marmon Algorithm, Triple-Axis Monotonic Input.

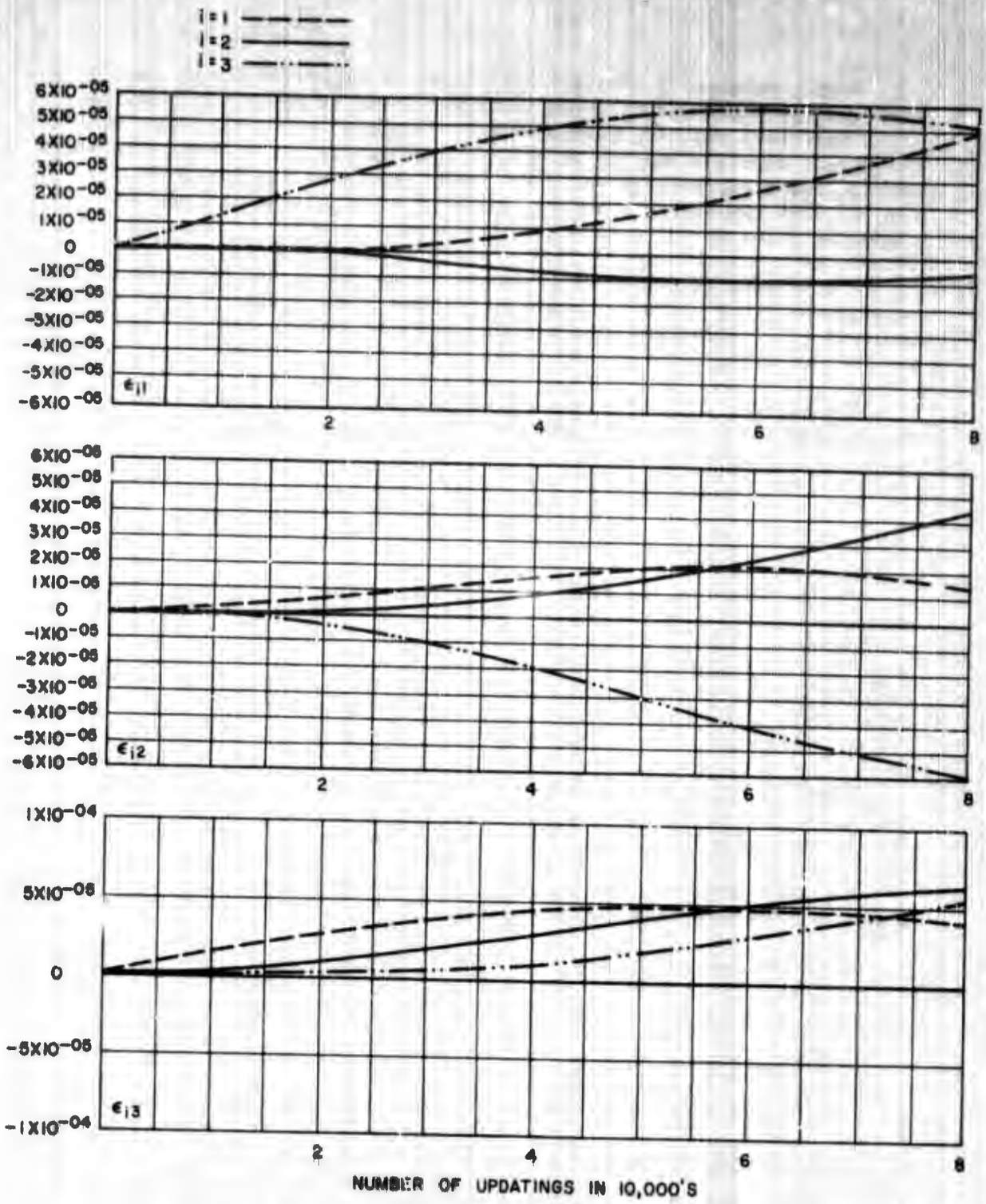


FIG. 31. Error Curves for Marmon Algorithm Gyro Model $\omega_1 = 0.025$, $\omega_2 = 0.0085$, $\omega_3 = 0.05$ (rad/sec).

Having surveyed a considerable number of algorithm forms, the analysis was continued by investigating methods suitable for studying these numerical updating forms. After reviewing sample solutions by the direct numerical method of undetermined coefficients, the Z-transform method, and a matrix-eigenvalue approach, the necessity for utilizing the IBM 7094 computer for successful study of the large classes of algorithm forms was discussed and justified.

The results of the digital computer analysis of truncation error were given. These results were detailed for three separate algorithm classes: (1) first-and second-order (parallel), (2) serial rules, and (3) modified first-order (nonserial).

The reader is referred to Ref. 17 for detailed discussions of the digital computer programming employed in this analysis.

CONCLUSIONS

On the basis of the algorithm truncation-error-response values observed, it is concluded that the algorithms studied are well behaved and their response is intuitively justifiable. Also, the numerical forms of these algorithms are acceptable methods for systems mechanization for direction cosine updating. Algorithms can be chosen, even under limit cycle conditions, with truncation error values sufficiently low to prevent this error from being a major design consideration.

Some attention must necessarily be given the truncation error, but, more importantly, the criteria for choosing and mechanizing the various algorithm form variations must be considered. For example, if high rate oscillations in the form of limit cycles are anticipated as part of the system functioning environment, then it might be required that a serial-reversal rule be employed.

With the probable factors playing a role in algorithm choice being

(1) computer size, memory, and speed capabilities

(2) inertial sensor errors

(3) errors inherent to the dynamics of the carrying vehicle

(4) errors inherent to the system mechanization mode,

ultimate algorithm choice will certainly be made on the basis of factors other than those of truncation alone.

REFERENCES

1. Massachusetts Institute of Technology, MIT Instrumentation Laboratory. Analysis of a Transformation Computer Used With a Gimballess IMU, by R. M. Hession. Cambridge, Mass., MIT, January 1965. (MIT Instr. Lab. Report No. R-481.)

2. IBM Federal Systems. Preliminary Specification for Strapdown Inertial System DDA, by G. A. Harasty. Owego, New York, IBM, 1 June 1964. (IBM No. 64-550-014.)

3. ------. Evaluation and Error Analysis of the DDA Transformation Matrix Computer for Strapdown Inertial Systems, by G. A. Harasty. Owego, New York, IBM, 10 June 1964. (IBM No. 64-550-018.)

4. U. S. Naval Ordnance Test Station. Vehicle-Fixed Component Inertial Guidance System Study, by C. Broxmeyer and H. Wishner. China Lake, Calif., NOTS, December 1964. (NAVWEPS Report 8668, NOTS TP 3715.)

5. Raytheon. Strapped Down Attitude Reference System, by H. Wishner and N. P. Coren. Sudbury, Mass., Raytheon, 5 January 1965. (No. FR-65-6.)

6. U. S. Naval Ordnance Test Station. Strapdown Analysis, Part 1, Algorithms for updating the direction cosine matrix, by W. F. Ball. China Lake, Calif., NOTS, 16 April 1965. (Technical Note 404-15.)

7. Lockheed Aircraft Corp., Missile Systems Division. Direction Cosine Coordinate Transformation for Strapped Down Inertial Reference, by W. C. Marmon. Sunnyvale, Calif., LMSD, 7 August 1964. (GTM 93.)

8. ------. Transformation Computer for Strapped-Down Guidance Systems, by L. F. Crowder. Sunnyvale, Calif., LMSD, 17 March 1964. (GTM 83.)

9. Massachusetts Institute of Technology, MIT Instrumentation Laboratory. Theoretical Analysis of Gimballess Inertial Reference Equipment Using Delta-Modulated Instruments, by T. F. Wiener. (A doctorate of science thesis), Cambridge, Mass., MIT, March 1962. (Thesis, T-300.)

10. U. S. Naval Ordnance Test Station. Strapdown Analysis, Part 2, Algorithm Error Analysis Methods, by W. F. Ball and Lt. H. A. Steele, Jr., USN. China Lake, Calif., NOTS, 30 June 1965. (Technical Note 404-19.)

11. ------. Strapdown Analysis, Part 3, Preliminary Results; Digital Computer Analysis of Direction Cosine Matrix Updating Algorithms, by William F. Ball. China Lake, Calif., NOTS, 15 December 1965. (Technical Note 404-26.)

12. Army Missile Command. Preliminary Analysis and Evaluation of Analytic Platform Systems...FY 63, by R. E. Alongi. Redstone, Ala., AMC, 12 August 1963. (RG-TR-63-23.)

13. ------. Supplementary Analysis and Evaluation of Analytic Platform Systems...FY 64, by R. E. Alongi. Redstone, Ala., AMC, 10 July 1964. (RG-TR-64-10.)

14. University of California at Los Angeles, Department of Engineering. An Investigation of the Alignment Calibration, and Inertial Component Error Compensation Techniques Required for a Platformless Inertial Guidance System, by W. B. Walker. (A Masters Thesis), Los Angeles, Calif., UCLA, June 1963.

15. Bumstead, R. M. and W. E. VanderVelde. "Navigation and Guidance Systems Employing a Gimballess IMU," AIAA G and C Conference, Massachusetts Institute of Technology, 12-14 August 1963. (Paper 63-307.)

16. U. S. Naval Ordnance Test Station. Algorithm Analysis, Part 4, Gyro Modeling and Gyro Storage Error, by William F. Ball. China Lake, Calif., NOTS, 15 June 1966. (Technical Note 404-38.)

17. ------. Algorithm Analysis, Part 5, Algorithm Analysis Programs for IBM 7094 Digital Computer, by William F. Ball. China Lake, Calif., NOTS, 30 June 1966. (Technical Note 404-39.)

18. Gantmacher, F. R. The theory of matrices. Chelsea, New York. 1959. Vol. 2.

19. Pipes, L. A. Matrix methods for engineering. Prentice-Hall, Englewood Cliffs, N. J. 1963.

20. Bronmeyer, C. Inertial navigation systems. New York, McGraw-Hill, 1964.

21. Ralston, A. and Wolf, H. S. Mathematical Methods for Digital Computers. Wiley, New York, 1964.

22. Salvadori, M. G. and Baron, M. L. Numerical Methods in Engineering. Prentice-Hall, Englewood Cliffs, N. J. 1952.

23. Todd, J. Survey of Numerical Analysis. McGraw-Hill, New York, 1962.

24. Kunz, K. S. Numerical Analysis. McGraw-Hill, New York, 1957.

25. Jury, E. I. Theory and Application of the Z-Transform Method. Wiley, New York, 1964.

26. Kuo, B. C. Analysis and Synthesis of Sampled-Data-Control Systems. Prentice-Hall, Englewood Cliffs, N. J., 1963.

27. Turley, A. R. "A Solution for Problems of the No-Gimbal Inertial Navigator Concept." AFAL-TR-64-307. Wright-Patterson AFB, Ohio. January 1965.

28. Hildebrand, F. B. Introduction to Numerical Analysis. McGraw-Hill. New York, 1956.

29. U. S. Air Force Aeronautical Laboratory. A Comparison of Three No-Gimbal Inertial Systems by First Test on an IBM 7094 Computer, by A. R. Turley. Wright-Patterson AFB, Ohio, AFAL, December 1965. (AFAL-TR-65-201.)

30. Begge, J. S. Advanced Mechanism. Macmillan, New York, 1966.

31. Ragazzini, J. R., and G. F. Franklin. Sampled Data Control Systems. McGraw-Hill, New York, 1958.

ABSTRACT CARD

U. S. Naval Ordnance Test Station
Strapped-Down Inertial Guidance. The Coordinate Transformation Updating Problem, by William F. Ball. China Lake, Calif., NOTS, April 1967. 116 pp. (NOTS TP 4267), UNCLASSIFIED.

ABSTRACT. This report provides a detailed introduction to the problem of updating the rotational transformation of a strapped-down inertial guidance system. Analyses are specifically devoted to problems of updating the direction cosine transformation. Various digital computer algorithms for updating the direction cosines are displayed and discussed.

(Over)
1 card, 8 copies

U. S. Naval Ordnance Test Station
Strapped-Down Inertial Guidance. The Coordinate Transformation Updating Problem, by William F. Ball. China Lake, Calif., NOTS, April 1967. 116 pp. (NOTS TP 4267), UNCLASSIFIED.

ABSTRACT. This report provides a detailed introduction to the problem of updating the rotational transformation of a strapped-down inertial guidance system. Analyses are specifically devoted to problems of updating the direction cosine transformation. Various digital computer algorithms for updating the direction cosines are displayed and discussed.

(Over)
1 card, 8 copies

U. S. Naval Ordnance Test Station
Strapped-Down Inertial Guidance. The Coordinate Transformation Updating Problem, by William F. Ball. China Lake, Calif., NOTS, April 1967. 116 pp. (NOTS TP 4267), UNCLASSIFIED.

ABSTRACT. This report provides a detailed introduction to the problem of updating the rotational transformation of a strapped-down inertial guidance system. Analyses are specifically devoted to problems of updating the direction cosine transformation. Various digital computer algorithms for updating the direction cosines are displayed and discussed.

(Over)
1 card, 8 copies

U. S. Naval Ordnance Test Station
Strapped-Down Inertial Guidance. The Coordinate Transformation Updating Problem, by William F. Ball. China Lake, Calif., NOTS, April 1967. 116 pp. (NOTS TP 4267), UNCLASSIFIED.

ABSTRACT. This report provides a detailed introduction to the problem of updating the rotational transformation of a strapped-down inertial guidance system. Analyses are specifically devoted to problems of updating the direction cosine transformation. Various digital computer algorithms for updating the direction cosines are displayed and discussed.

(Over)
1 card, 8 copies

NOTS TP 4267

Methods of performing error analyses on these algorithms are reviewed and results of IBM 7094 analyses are shown. The primary analytical results are representations of the truncation error resulting from the particular numerical forms studied.

NOTS TP 4267

Methods of performing error analyses on these algorithms are reviewed and results of IBM 7094 analyses are shown. The primary analytical results are representations of the truncation error resulting from the particular numerical forms studied.

NOTS TP 4267

Methods of performing error analyses on these algorithms are reviewed and results of IBM 7094 analyses are shown. The primary analytical results are representations of the truncation error resulting from the particular numerical forms studied.

NOTS TP 4267

Methods of performing error analyses on these algorithms are reviewed and results of IBM 7094 analyses are shown. The primary analytical results are representations of the truncation error resulting from the particular numerical forms studied.

UNCLASSIFIED
Security Classification

DOCUMENT CONTROL DATA - R&D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author)
U. S. Naval Ordnance Test Station
China Lake, California 93555

2a. REPORT SECURITY CLASSIFICATION
UNCLASSIFIED

2b. GROUP
---

3. REPORT TITLE
STRAPPED-DOWN INERTIAL GUIDANCE. The Coordinate Transformation Updating Problem

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)
Analysis

5. AUTHOR(S) (Last name, first name, initial)
BALL, William F.

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
April 1967	116	31

8a. CONTRACT OR GRANT NO.	9a. ORIGINATOR'S REPORT NUMBER(S)
b. PROJECT NO.	NOTS TP 4267
c. AIR TASK A36-533-205/216-1/F009-03-03	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)
d.	---

10. AVAILABILITY/LIMITATION NOTICES
THIS DOCUMENT IS SUBJECT TO SPECIAL EXPORT CONTROLS AND EACH TRANSMITTAL TO FOREIGN GOVERNMENTS OR FOREIGN NATIONALS MAY BE MADE ONLY WITH PRIOR APPROVAL OF THE U.S. NAVAL ORDNANCE TEST STATION.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
---	Naval Air Systems Command Naval Material Command Washington, D. C. 20360

13. ABSTRACT

ABSTRACT. This report provides a detailed introduction to the problem of updating the rotational transformation of a strapped-down inertial guidance system. Analyses are specifically devoted to problems of updating the direction cosine transformation. Various digital computer algorithms for updating the direction cosines are displayed and discussed. Methods of performing error analyses on these algorithms are reviewed and results of IBM 7094 analyses are shown. The primary analytical results are representations of the truncation error resulting from the particular numerical forms studied.

DD FORM 1 JAN 64 1473 0101-807-6800

UNCLASSIFIED
Security Classification

UNCLASSIFIED

Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Strapped-Down Inertial System Rotational Transformation Updating Direction Cosines						

INSTRUCTIONS

1. ORIGINATING ACTIVITY: Enter the name and address of the contractor, subcontractor, grantee, Department of Defense activity or other organization *(corporate author)* issuing the report.

2a. REPORT SECURITY CLASSIFICATION: Enter the overall security classification of the report. Indicate whether "Restricted Data" is included. Marking is to be in accordance with appropriate security regulations.

2b. GROUP: Automatic downgrading is specified in DoD Directive 5200.10 and Armed Forces Industrial Manual. Enter the group number. Also, when applicable, show that optional markings have been used for Group 3 and Group 4 as authorized.

3. REPORT TITLE: Enter the complete report title in all capital letters. Titles in all cases should be unclassified. If a meaningful title cannot be selected without classification, show title classification in all capitals in parenthesis immediately following the title.

4. DESCRIPTIVE NOTES: If appropriate, enter the type of report, e.g., interim, progress, summary, annual, or final. Give the inclusive dates when a specific reporting period is covered.

5. AUTHOR(S): Enter the name(s) of author(s) as shown on or in the report. Enter last name, first name, middle initial. If military, show rank and branch of service. The name of the principal author is an absolute minimum requirement.

6. REPORT DATE: Enter the date of the report as day, month, year; or month, year. If more than one date appears on the report, use date of publication.

7a. TOTAL NUMBER OF PAGES: The total page count should follow normal pagination procedures, i.e., enter the number of pages containing information.

7b. NUMBER OF REFERENCES: Enter the total number of references cited in the report.

8a. CONTRACT OR GRANT NUMBER: If appropriate, enter the applicable number of the contract or grant under which the report was written.

8b, 8c, & 8d. PROJECT NUMBER: Enter the appropriate military department identification, such as project number, subproject number, system numbers, task number, etc.

9a. ORIGINATOR'S REPORT NUMBER(S): Enter the official report number by which the document will be identified and controlled by the originating activity. This number must be unique to this report.

9b. OTHER REPORT NUMBER(S): If the report has been assigned any other report numbers *(either by the originator or by the sponsor)*, also enter this number(s).

10. AVAILABILITY/LIMITATION NOTICES: Enter any limitations on further dissemination of the report, other than those imposed by security classification, using standard statements such as:

(1) "Qualified requesters may obtain copies of this report from DDC."

(2) "Foreign announcement and dissemination of this report by DDC is not authorized."

(3) "U. S. Government agencies may obtain copies of this report directly from DDC. Other qualified DDC users shall request through

_____."

(4) "U. S. military agencies may obtain copies of this report directly from DDC. Other qualified users shall request through

_____."

(5) "All distribution of this report is controlled. Qualified DDC users shall request through

_____."

If the report has been furnished to the Office of Technical Services, Department of Commerce, for sale to the public, indicate this fact and enter the price, if known.

11. SUPPLEMENTARY NOTES: Use for additional explanatory notes.

12. SPONSORING MILITARY ACTIVITY: Enter the name of the departmental project office or laboratory sponsoring *(paying for)* the research and development. Include address.

13. ABSTRACT: Enter an abstract giving a brief and factual summary of the document indicative of the report, even though it may also appear elsewhere in the body of the technical report. If additional space is required, a continuation sheet shall be attached.

It is highly desirable that the abstract of classified reports be unclassified. Each paragraph of the abstract shall end with an indication of the military security classification of the information in the paragraph, represented as *(TS), (S), (C), or (U)*

There is no limitation on the length of the abstract. However, the suggested length is from 150 to 225 words

14. KEY WORDS: Key words are technically meaningful terms or short phrases that characterize a report and may be used as index entries for cataloging the report. Key words must be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location, may be used as key words but will be followed by an indication of technical context. The assignment of links, roles, and weights is optional.

UNCLASSIFIED

Security Classification

5 Massachusetts Institute of Technology, Cambridge (Instrumentation Laboratory)
- R. Huston (1)
- R. Hession (1)
- J. Sitomer (1)
- S. Fortier (1)
- R. Bumstead (1)

1 Raytheon Company, Sudbury, Mass. (J. Matthews)

3 Stanford Research Institute, Menlo Park, Calif.
- G. Branch (1)
- J. Merritt (1)
- J. Peschon (1)

1 The Analytic Sciences Corporation, Winchester, Mass. (A. Gelb)

1 University of California at Los Angeles (Engineering Department, J. S. Beggs)