2
mix

SRRC-CR-70-8

STRAPDOWN NAVIGATION COMPUTER STUDIES

Final Report

Contract No. NAS 9-9705

MARCH 1970

Prepared for
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
MANNED SPACECRAFT CENTER
HOUSTON, TEXAS

SPERRY RAND RESEARCH CENTER
SUDBURY, MASSACHUSETTS 01776

149

NASA CR 108429

SPERRY RAND
CORPORATION

SRRC-CR-70-8

STRAPDOWN NAVIGATION COMPUTER STUDIES

Final Report

Contract No. NAS 9-9705

MARCH 1970

Prepared for
National Aeronautics and Space Administration
Manned Spacecraft Center
Houston, Texas

SPERRY RAND RESEARCH CENTER
SUDBURY MASSACHUSETTS 01776

TABLE OF CONTENTS

TABLE OF CONTENTS (cont)

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS (cont.)

LIST OF TABLES

CHAPTER I

INTRODUCTION

The text which follows constitutes the final report on work performed at the Sperry Rand Research Center (SRRC) under contract NAS 9-9705 with the NASA Manned Spacecraft Center, Houston, Texas. The objectives of this program were twofold. First, it was to consider optimum organization and component technology for a strapdown transformation computer based upon the assumption of gyroscope and accelerometer inputs denoting components of angular and velocity change measured about an orthogonal axis triad. Second, it was to treat optimum methods of implementing a reliable attitude reference or control system when the gyroscope and accelerometer inputs were redundant and configured about the six normals to faces of a dodecahedron.

The work which is reported here divides into four major activites, each corresponding to a chapter. These will be described briefly. The first part of the activity, described in Chapter 2, is a study of means whereby a strapdown computer such as the one constructed for laboratory use under a previous contract may best be organized and implemented in reliable, flight-ready form. The second activity, covered in Chapter 3, is an investigation of optimum means for performing attitude reference computations when redundant dodecahedron sensor arrays are utilized. The third part of the study, described in Chapter 4, is a study of an attitude control system employing redundant control moment gyros as actuators. This part of the work was performed at the Sperry Flight Systems Division, under subcontract to SRRC. The final part of the study, Chapter 5, contains the results of further computer simulations of the truncation-error-free direction-cosine computation used in SIMU, methods and results are described.

CHAPTER 2

CONFIGURATION AND COMPONENT TECHNOLOGY FOR A STRAPDOWN COMPUTER

In a previous contract a strapdown coordinate-conversion computer was designed and constructed which used a truncation-error-free algorithm to compute direction cosines relating a rotating-axis system to one which was inertially stabilized. The organization of this computer was based upon a building block concept in which most blocks were digital accumulators with overflow detection and ternary input gating. In the present study, where the goal is to determine optimum system organization and component technology for such a computer, several structures, including the accumulator block, were studied.

Before describing the system, the underlying reasons for the particular choice of implementation will be given. First of all, since discussion centers about a spaceborne computer, weight, volume and power must be minimized and reliability must be maximized. From the point of view of weight, volume and power, it is clear that minimality is provided by incorporating as high a level of circuit integration as is feasible within the constraints of present day technology. Studies have shown that maximizing the level of integration also provides the greatest reliability,[1] since circuit bonds and off-chip interconnections, which are the primary sources of failure, are minimized. At the present time, reasonable device yields are obtained for integrated circuit chips with dimensions of 120 mils on each side. Interline spacings of 0.1 mil are readily obtainable. For metal-oxide-silicon (MOS) circuitry, active device areas are approximately 1.0 mil square. Allowing for metallization patterns and bonding pads, this means that an active device count of 2000 to 5000 is reasonable for MOS integrated circuits. Four-phase or ratioless MOS circuitry allows transfer rates from 2 to 5 MHz to be realized, and provides the highest circuit density presently available. For this reason, the preferred mechanization of the strapdown computer would use four-phase MOS LSI.

As in most applications, cost is also of importance. Development and manufacturing costs for the strapdown computer can be reduced by limiting the number of different chip types required. For this reason and the reasons above, the approach which was followed in formulating the systems structures was one in which maximum use was made of the fewest possible different circuit types, while maintaining the highest level of integration consistent with present technology and while trying to minimize the number of interconnections between chips.

The block diagram assumed for the triad system is shown in Fig. 2-1. It is essentially that of the SIMU computer, except that sections intended primarily for laboratory use have been eliminated. A set of three signals from orthogonal triad gyroscopic sensors provides the input to the direction cosine computation. These signals are assumed to be corrected for drift scale factor errors by equipment preceding the navigation computer. The gyro inputs are added in the buffer to triad rotational rate commands. These commands replace the earth-rate correction system in the SIMU computer and are assumed to be generated by the main guidance and navigation computer. They enter the strapdown computer referred to stable coordinates and are transformed to rotational coordinates by the "commanded rate transformation" unit. Triad strapdown accelerometer signals are transformed to stable coordinates by the "ΔV

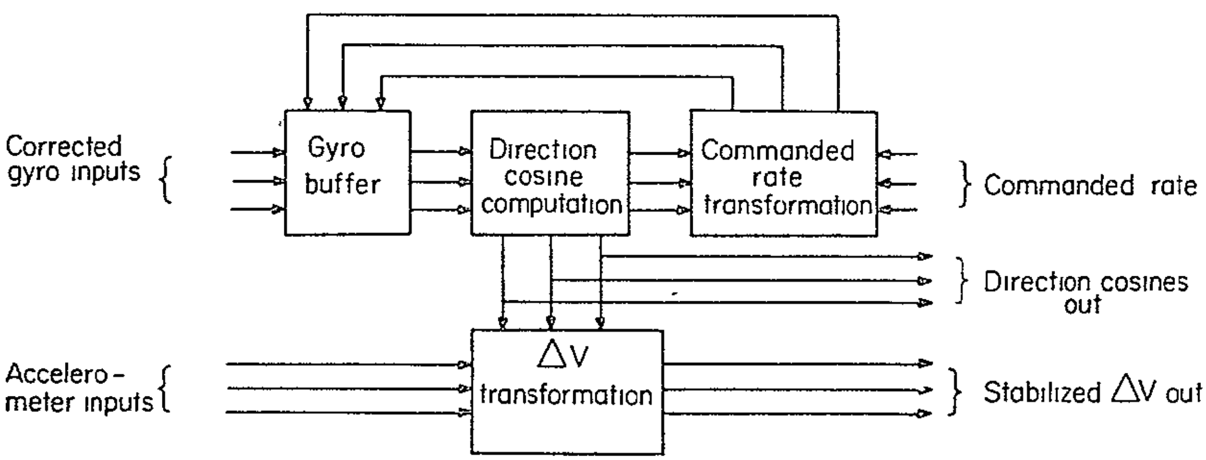


FIG. 2-1 Block diagram of SIMU computer.

transformation" unit, as in SIMU. System outputs are the stabilized ΔV's and the nine direction cosines.

Besides the configuration used in the SIMU computer, a new form has been developed for system implementation. This will be described first, and then methods for increasing system reliability will be discussed for both cases.

The new method of organization is very similar in form to SIMU except that 16-bit digital resolvers are used for the direction cosine computation instead of the accumulators. A diagram of one of the three identical direction cosine units is shown in Fig. 2-2. Each direction cosine resides in a triple of resolvers, the Y resolver storing the upper 16 bits, the R resolver storing the middle 16 bits, and the E resolver storing the lower 16 bits.

The updating algorithm used is the original truncation-error-free method involving three passes. The operations for a θ_x pulse are specified by the matrix equation

$$M_x = \begin{vmatrix} 1 & 0 & 0 \\ 0 & 1-h^2 & 0 \\ 0 & 0 & 1-h^2 \end{vmatrix} \begin{vmatrix} 1 & 0 & 0 \\ 0 & 1 & h \\ 0 & -h & 1 \end{vmatrix} \begin{vmatrix} 1 & 0 & 0 \\ 0 & 1 & h \\ 0 & -h & 1 \end{vmatrix} \tag{2.1}$$

Similar operations hold for θ_y and θ_z.

Each of the resolvers is equipped with its own serial adder operating in a closed loop with the storage register Thus, simultaneous accumulation by all resolvers is allowable. By judicious sequencing of operations, a three-to-one speedup in processing rate is obtained over the originally proposed SIMU computer. This sequencing is best clarified by means of the schedule shown in Fig. 2-3. Each term in the schedule signifies the updating of the contents stored in the specified resolver by adding to it the contents of another resolver, as determined by the updating matrices. The superscripts indicate whether it is the x-, y-, or z-matrix equation, and the priming of the superscripts specifies the right, middle, or left matrix, respectively. Thus, $E_1^{y'}$ indicates the updating of resolver E_1 for the second pass of θ_y updating. This updating involves subtracting the contents of resolver R_3 from the contents of resolver E_1 and storing the difference in E_1. It is clear that the updating for x, y and z pulses of any rank of resolvers (say, the E resolvers) requires just nine word-times. Since overlapping of cycles is allowable, a new set of gyro pulses may be entered every nine word-times. If the clock used is the same as in the SIMU computer, a processing rate speedup of two-to-one over SIMU is obtained The speedup over the originally proposed system is three-to-one. Besides the speedup in processing rate (or, conversely, the ability to use a lower clock rate), another advantage of this form of implementation is that it provides greater uniformity of components throughout the system, since the resolvers used in the cosine computation are identical to those used to provide the coordinate transformations elsewhere in the system. An appropriate circuit embodiment would be the realization of one or more resolvers as a single integrated circuit. The approximate device count for a simplex realization of such a circuit is 350 transistors, so that such an embodiment is quite feasible using MOS/LSI technology.

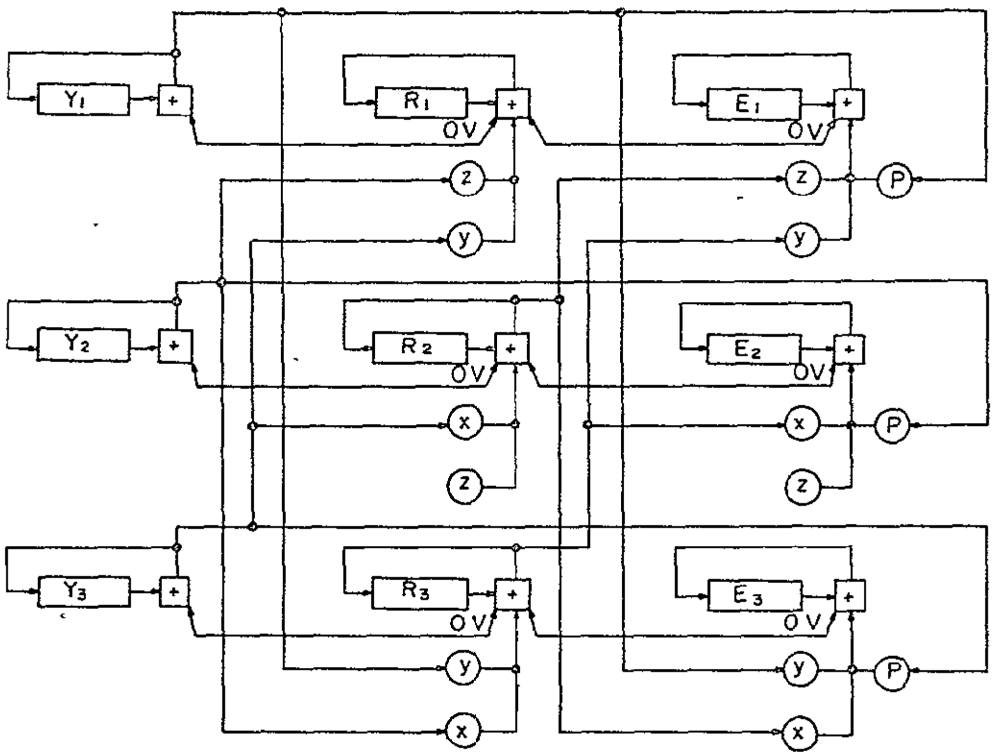


FIG. 2-2 Resolver direction cosine unit.

—	—	—	E_1^{Y}	$E_1^{Y'}$	$E_1^{Y''}$	E_1^{Z}	$E_1^{Z'}$	$E_1^{Z''}$	—	—
E_2^{X}	$E_2^{X'}$	$E_2^{X''}$	—	—	—	E_2^{Z}	$E_2^{Z'}$	$E_2^{Z''}$	—	—
E_3^{X}	$E_3^{X'}$	$E_3^{X''}$	E_3^{Y}	$E_3^{Y'}$	$E_3^{Y''}$	—	—	—	—	—
—	—	—	—	R_1^{Y}	$R_1^{Y'}$	$R_1^{Y''}$	R_1^{Z}	$R_1^{Z'}$	$R_1^{Z''}$	—
—	R_2^{X}	$R_2^{X'}$	$R_2^{X''}$	—	—	—	R_2^{Z}	$R_2^{Z'}$	$R_2^{Z''}$	—
—	R_3^{X}	$R_3^{X'}$	$R_3^{X''}$	R_3^{Y}	$R_3^{Y'}$	$R_3^{Y''}$	—	—	—	—
—	—	—	—	—	Y_1^{Y}	$Y_1^{Y'}$	$Y_1^{Y''}$	Y_1^{Z}	$Y_1^{Z'}$	$Y_1^{Z''}$
—	—	Y_2^{X}	$Y_2^{X'}$	$Y_2^{X''}$	—	—	—	Y_2^{Z}	$Y_2^{Z'}$	$Y_2^{Z''}$
—	—	Y_3^{X}	$Y_3^{X'}$	$Y_3^{X''}$	Y_3^{Y}	$Y_3^{Y'}$	$Y_3^{Y''}$	—	—	—

FIG. 2-3 Sequencing of all-resolver cosine computer.

Several techniques have been investigated for increasing reliability over the "simplex" SIMU computer. These will be described, although no revolutionary developments have emerged. The first technique makes use of the orthogonality properties of the direction cosine matrix for self checking. If C is the cosine matrix and C^T is its transpose, then orthogonality implies that

$$C \cdot C^T = C^T \cdot C = I \tag{2.2}$$

where I is the identity matrix. Carrying out the term-by-term multiplications, the twelve distinct equations in Table 2.1 result.

TABLE 2.1

Direction Cosine Orthogonality Conditions

Eq. No.	Equation
1	$C_{11}^2 + C_{12}^2 + C_{13}^2 - 1 = 0$
2	$C_{11}C_{21} - C_{12}C_{22} + C_{13}C_{23} = 0$
3	$C_{11}C_{31} + C_{12}C_{32} + C_{13}C_{33} = 0$
4	$C_{21}^2 + C_{22}^2 + C_{23}^2 - 1 = 0$
5	$C_{21}C_{31} + C_{22}C_{32} + C_{23}C_{33} = 0$
6	$C_{31}^2 + C_{32}^2 + C_{33}^2 - 1 = 0$
7	$C_{11}^2 + C_{21}^2 + C_{31}^2 - 1 = 0$
8	$C_{11}C_{12} + C_{21}C_{22} + C_{31}C_{32} = 0$
9	$C_{11}C_{13} + C_{21}C_{23} + C_{31}C_{33} = 0$
10	$C_{12}^2 + C_{22}^2 + C_{32}^2 - 1 = 0$
11	$C_{12}C_{13} + C_{22}C_{23} + C_{32}C_{33} = 0$
12	$C_{13}^2 + C_{23}^2 + C_{33}^2 - 1 = 0$

The failure of some subset of the equations in Table 2.1 to be satisfied may be used as a form of parity check to determine faulty computation of the direction cosines. For instance, an error in C_{11} alone is indicated by lack of satisfaction of Eqs. (2.1) and (2.7). Similar criteria may be derived from the covering relationships expressed by Table 2.2. At "X" at the intersection of a particular column and row indicates that the cosine corresponding to that row is covered by the equation corresponding to the column.

TABLE 2.2

Direction Cosine Covering by Orthogonality Equations (see Fig. 2-1)

Cosine	Equation											
	1	2	3	4	5	6	7	8	9	10	11	12
C_{11}	X	X	X				X	X	X			
C_{12}	X	X	X					X		X	X	
C_{13}	X	X	X						X		X	X
C_{21}		X		X	X		X	X	X			
C_{22}		X		X	X			X		X	X	
C_{23}		X		X	X				X		X	X
C_{31}			X		X	X	X	X	X			
C_{32}			X		X	X		X		X	X	
C_{33}			X		X	X			X		X	X

If a single bit error is introduced into one of the direction cosines, say C_{11}, the error will propagate to cosines C_{12} and C_{13} as a function of angular input increments received. No error, however, will propagate to the other cosines, since their computation "loops" do not intersect those of C_{11}. The dependence of C_{12}, for instance, upon C_{11} may be expressed by the partial differential equation pair below

$$\frac{\partial C_{11}}{\partial \theta_z} = -C_{12} \tag{2.3}$$

$$\frac{\partial C_{12}}{\partial \theta_z} = C_{11} \quad . \tag{2.4}$$

Consider the inversion of a single bit in the representation of C_{11} and its effect upon C_{12}. The bit inversion will be equivalent to adding some quantity S to C_{11}. Let the error in C_{11} be E_1 and the error in C_{12} be E_2. Then

$$\frac{\partial C_{11}}{\partial \theta_z} + \frac{\partial E_1}{\partial \theta_z} = -C_{12} - E_2 \tag{2.5}$$

$$\frac{\partial C_{12}}{\partial \theta_z} + \frac{\partial E_2}{\partial \theta_z} = C_{11} + E_1 \tag{2.6}$$

with initial values

$$E_1 = S$$

$$E_2 = 0 \quad .$$

Subtracting left and right equivalents from this set, the resulting error equations are

$$\frac{\partial E_1}{\partial \theta_z} = -E_2 \tag{2.7}$$

$$\frac{\partial E_2}{\partial \theta_z} = E_1 \tag{2.8}$$

with the obvious solution

$$E_1 = S \cos \theta_z \tag{2.9}$$

$$E_2 = S \sin \theta_z \tag{2.10}$$

If the checking equations of Table 2.1 are solved at fixed intervals and involve the upper n bits of the cosines, then the maximum period between solutions such that a single bit error in one of the direction cosines can be detected before it affects the other cosines in its loop is a function of n and the maximum angular rate which the system can accommodate. If the worst case bit inversion (i.e., $S=1$) is assumed, the maximum period between solutions is set by the inequality below, where ω_{max} is the maximum angular velocity and T_{max} is the maximum period

$$T_{max} \leq \frac{1}{\omega_{max}} \sin^{-1} 2^{-n} \quad . \tag{2.11}$$

After detecting the cosine which is in error, the problem of what to do about it still remains. If the angles were unchanging, it would be a simple matter to recompute the correct values of the cosines using the relationships of Table 2.1. This, however, is not generally the case. One possibility is to retain in storage at some point the last complete set of correct cosines computed before the error was detected. These values can be reinserted and the computations restarted. Although this method introduces an overall error if vehicle attitude has changed during the time since restart values were observed, the amount of error can be limited if the checking equations are run frequently enough. Another method which leads to no error accumulation is to stop the direction cosine computations as soon as an error is detected and store the input angle increments in a buffer, maintaining the proper sequence among the three axes. The equations in Table 2.1 then can be used to restore the proper value to the faulty cosine, after which the stored input increments are utilized to update the cosine matrix to its proper value. This method will succeed only if the allowable updating rate of the cosine matrix is greater than the maximum allowable pulse input rate from the gyros.

In the event that the error is caused by permanent failure of one of the components within the direction cosine computer, more than one-time correction will be required. A possible solution to this situation is obtained by noting that the nine cosines are computed as three independent triads. If a spare triad computer is included and is equipped with the proper connections to allow its output to be switched to replace any of the three primary computer outputs, then upon detection of a failure in one of the three primary units the restart operation can involve the spare unit, after which it is switched into the system in place of the faulty element.

In the methods described above that make use of the orthogonality properties of the direction cosine matrix for self correction, the nature of the computations and their rate is such that corrections can be handled by thegeneral purpose guidance computer associated with the system. It is possible, and also more straightforward, to include logical redundance within the direction cosine computer itself rather than using external means. In both the SIMU and all-resolver systems, numerous computational feedback loops are used. If it is desired that the addition of redundancy will effect correction of transient errors as well as permanent component failures, it is advisable to add the redundancy in such a way that error correction occurs within all feedback loops. One scheme for providing this sort of correction is shown in Fig. 2-4, which depicts a triply modular redundant resolver. Voting circuits are introduced in such a way that they intersect the tightest feedback paths and still provide correct outputs from all three of the resolver segments for the maximum number of component failures within the segment.

It is difficult to supply all-inclusive arguments for the level at which redundancy should be introduced in the strapdown computer, but since the resolver is the element which seems best suited for use as the basic building block, larger or smaller elements not providing the same lead-count and circuit-type minimality, and since the packaging of three interconnected resolvers with voting elements on a single substrate is presently within the state of the art, this configuration is recommended as the form to be used for implementation of the strapdown computer.

REFERENCES

1. "Failure Rate/Temperature Data for Univac Military Computers," Univac Federal Systems Division, Publication PX-4388-2, July 1969.

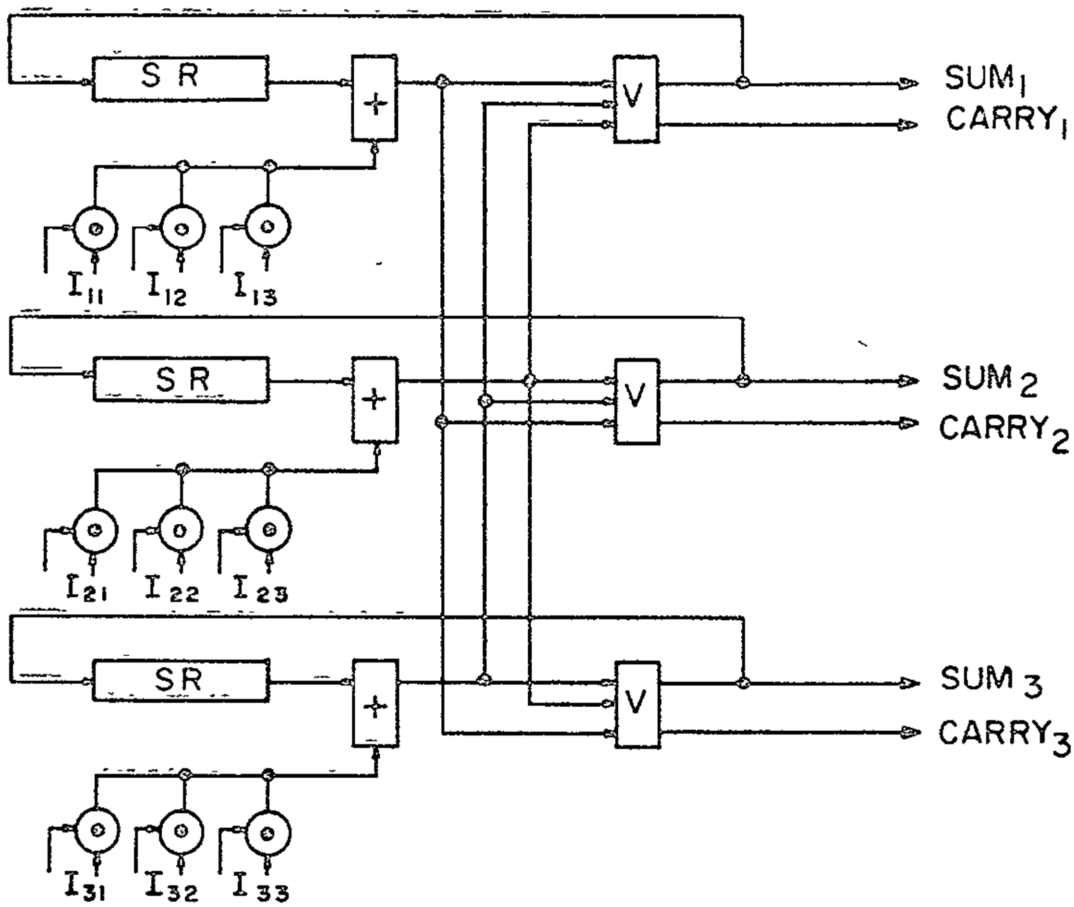


FIG. 2-4 TMR resolver.

CHAPTER 3

CONVERSION METHODS FOR TRANSFORMATION FROM DODECAHEDRON-TO-TRIAD AXES

3.1 DIRECT REALIZATION BY USE OF PSEUDO-INVERSE AND STATUS MATRICES

3.1.1 Theoretical Analysis of Approach

In this part of the report we describe a method of transforming the six-vector

$$m = \begin{pmatrix} m_a \\ m_b \\ m_c \\ m_d \\ m_e \\ m_f \end{pmatrix} \tag{3.1}$$

of the dodecahedron reference system into the three-vector

$$b = \begin{pmatrix} b_x \\ b_y \\ b_z \end{pmatrix} \tag{3.2}$$

of the orthogonal triad reference system.

The geometric relationship between the two systems is given by

$$m = Hb \tag{3.3}$$

where

$$H = \begin{bmatrix} S & 0 & C \\ -S & 0 & C \\ C & S & 0 \\ C & -S & 0 \\ 0 & C & S \\ 0 & C & -S \end{bmatrix} \tag{3.4}$$

$$S = \sin\alpha = \sqrt{\frac{5-\sqrt{5}}{10}} \tag{3.5}$$

$$C = \cos\alpha = \sqrt{\frac{5+\sqrt{5}}{10}} \tag{3.6}$$

Since every 3 × 3 submatrix of H is nonsingular, any three out of the six equations of (3.3) can be used to determine the b components in terms of the corresponding three components of m . The nature of the problem, however, is such that the instruments providing the components of m are subject to errors and, as was pointed out by Gilmore,[1] the solution of Eq. (3.3) for b which will best fit the geometric configuration of the well functioning instruments is given by

$$b = \left(H^T \lambda H\right)^{-1} H^T \lambda m \tag{3.7}$$

where $\lambda = \mathrm{diag}(\lambda_a, \lambda_b, \lambda_c, \lambda_d, \lambda_e, \lambda_f)$ is the "status" matrix, with $\lambda_i = 1$ or 0 according to whether instrument i, i=a,b,c,d,e,f , is error free or not.

Of course, for this solution to be meaningful, at least three out of the six instruments has to be error free. There are 42 different combinations with at least three of the λ_i being equal to 1 . After having evaluated explicitly the matrix

$$H^I = \left(H^T \lambda H\right)^{-1} H^T \lambda \tag{3.8}$$

for all the proper 42 combinations of the λ_i , it becomes evident that the total number of different nonzero entries of H^* for the various combinations is only 25. In terms of S and C , these 25 entries are as shown in Table 3.1.

TABLE 3.1

Twenty Five Required Data-Word Matrix

$N_1 = \frac{1}{10}(2S + C)$	$N_{11} = \frac{1}{4}(S + 4C)$	$N_{21} = 2S + C$
$N_2 = \frac{1}{10}(7S + C)$	$N_{12} = \frac{1}{4}(3S + C)$	$N_{22} = \frac{1}{2}(3S - C)$
$N_3 = \frac{1}{5}(S + 3C)$	$N_{13} = \frac{3}{4} S$	$N_{23} = \frac{1}{2}(S + 3C)$
$N_4 = \frac{1}{10}(-S + 7C)$	$N_{14} = \frac{3}{4} C$	$N_{24} = \frac{1}{4}(S - C)$
$N_5 = \frac{1}{10}(S - 2C)$	$N_{15} = \frac{1}{4}(-S + 3C)$	$N_{25} = \frac{1}{4}(S + C)$
$N_6 = \frac{1}{5}(3S - C)$	$N_{16} = \frac{1}{4}(2S - C)$	
$N_7 = \frac{1}{10}(2S - C)$	$N_{17} = \frac{1}{4}(S + 2C)$	
$N_8 = \frac{1}{2} S$	$N_{18} = \frac{1}{2}(-S + 2C)$	
$N_9 = \frac{1}{2} C$	$N_{19} = \frac{1}{2}(2S + C)$	
$N_{10} = \frac{1}{4}(4S - C)$	$N_{20} = -S + 2C$	

The 42 explicit evaluations of H^I are given in Table 3.2 in terms of the indices of the above listed N_i's . For instance, the H^I matrix for the combination $\lambda_a=\lambda_b=\lambda_e=1$ and $\lambda_c=\lambda_d=\lambda_f=0$, which is given by

$$H^I = \begin{bmatrix} N_{19} & -N_{19} & 0 & 0 & 0 & 0 \\ -N_{22} & -N_{22} & 0 & 0 & N_{20} & 0 \\ N_{18} & N_{18} & 0 & 0 & 0 & 0 \end{bmatrix} . \tag{3.9}$$

In Table 3.2 ,0 stands for the number zero and the numbers shown represent the N_i terms by listing only the subscript number (±i), e.g., $N_{12}=12$. For the expression in Eq. (3.9), this yields

$$\lambda_a,\lambda_b,\lambda_e = 1 \qquad \begin{bmatrix} 19 & -19 & 0 & 0 & 0 & 0 \\ -22 & -22 & 0 & 0 & 20 & 0 \\ 18 & 18 & 0 & 0 & 0 & 0 \end{bmatrix} . \tag{3.10}$$

3.1.2 Implementation of Pseudo-Inverse and Status Matrices Method Using LSI Packaged Basic Elements

The implementation of the transformation described in 3.1.1 essentially involves the use of a read-only memory and an arithmetic unit to provide the appropriate overflows to the 3-axis system's input buffer logic.

The following discussion will treat each section of the system in enough detail to make clear the functional interaction of the sections in achieving the required final results. There are four sections of the system: they are (1) the six input gyro-sync logic, (2) the read-only memory, (3) the arithmetic unit, and (4) the direction cosine computer's (SIMU) input buffer logic.

The gyro-sync logic performs two operations. First, the six (±x, ±y, ±z) asynchronous gyro inputs are synchronized to computer word time. Second, these six signals are encoded so as to be sent out as three octal variables (see Fig. 3-1).

The read-only memory's address selection is a function of two sets of input variables. These are encoded gyro outputs described above and the set of variables ($\lambda_1,\lambda_2,\lambda_3,\lambda_4,\lambda_5$ and λ_6) that indicate which gyros are operating satisfactorily at any given time. These nine selection lines are used to select one of 252 memory locations. In each of these memory locations is an address of a second read-only memory. The second read-only memory contains 25 sixteen-bit words.

To understand the need for this memory configuration refer to Table 3.2, which shows the manner in which the basic constants shown in Table 3.1 are used. The entries in Table 3.1 are a summary of the various products of a constant times $\frac{1}{2}\sin\alpha$ and $\frac{1}{2}\cos\alpha$ as discussed in Sec. 3.1.1.

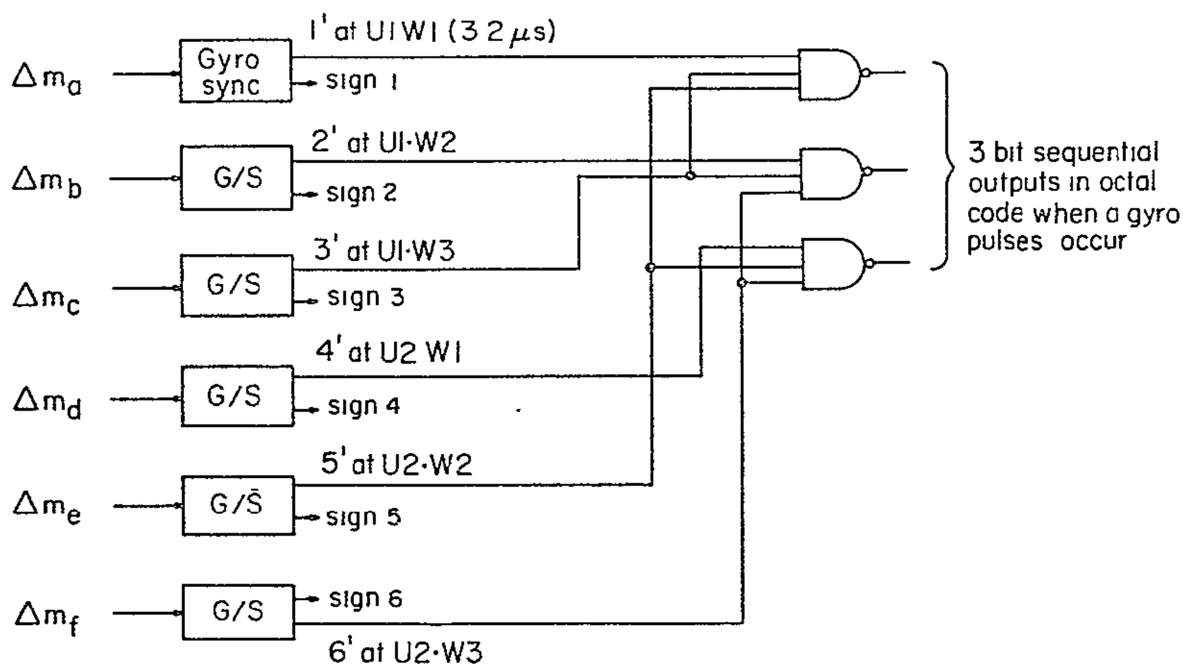


FIG. 3-1 Incremental input encoding for the six-gyro system.

TABLE 3.2

Status Matrices

$$\lambda_a,\lambda_b,\lambda_c = 1 \quad \begin{bmatrix} 19 & -19 & 0 & 0 & 0 & 0 \\ -23 & 23 & 21 & 0 & 0 & 0 \\ 18 & 18 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\lambda_a,\lambda_b,\lambda_d = 1 \quad \begin{bmatrix} 19 & -19 & 0 & 0 & 0 & 0 \\ 23 & -23 & 0 & -21 & 0 & 0 \\ 18 & 18 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\lambda_a,\lambda_b,\lambda_e = 1 \quad \begin{bmatrix} 19 & -19 & 0 & 0 & 0 & 0 \\ -22 & -22 & 0 & 0 & 20 & 0 \\ 18 & 18 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\lambda_a,\lambda_b,\lambda_f = 1 \quad \begin{bmatrix} 19 & -19 & 0 & 0 & 0 & 0 \\ 22 & 22 & 0 & 0 & 0 & 20 \\ 18 & 18 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\lambda_a,\lambda_c,\lambda_d = 1 \quad \begin{bmatrix} 0 & 0 & 18 & 18 & 0 & 0 \\ 0 & 0 & 19 & -19 & 0 & 0 \\ 20 & 0 & -22 & -22 & 0 & 0 \end{bmatrix}$$

$$\lambda_a,\lambda_c,\lambda_e = 1 \quad \begin{bmatrix} 22 & 0 & 19 & 0 & -18 & 0 \\ -18 & 0 & 22 & 0 & 19 & 0 \\ 19 & 0 & -18 & 0 & 22 & 0 \end{bmatrix}$$

$$\lambda_a,\lambda_c,\lambda_f = 1 \quad \begin{bmatrix} -18 & 0 & 23 & 0 & 0 & -19 \\ 19 & 0 & -18 & 0 & 0 & 23 \\ 23 & 0 & -19 & 0 & 0 & 18 \end{bmatrix}$$

$$\lambda_a,\lambda_d,\lambda_e = 1 \quad \begin{bmatrix} -18 & 0 & 0 & 23 & 19 & 0 \\ -19 & 0 & 0 & 18 & 23 & 0 \\ 23 & 0 & 0 & -19 & -18 & 0 \end{bmatrix}$$

$$\lambda_a,\lambda_d,\lambda_f = 1 \quad \begin{bmatrix} 22 & 0 & 0 & 19 & 0 & 18 \\ 18 & 0 & 0 & -22 & 0 & 19 \\ 19 & 0 & 0 & -18 & 0 & -22 \end{bmatrix}$$

$$\lambda_a,\lambda_e,\lambda_f = 1 \quad \begin{bmatrix} 21 & 0 & 0 & 0 & -23 & 23 \\ 0 & 0 & 0 & 0 & 18 & 18 \\ 0 & 0 & 0 & 0 & 19 & -19 \end{bmatrix}$$

$$\lambda_b,\lambda_c,\lambda_d = 1 \quad \begin{bmatrix} 0 & 0 & 18 & 18 & 0 & 0 \\ 0 & 0 & 19 & -19 & 0 & 0 \\ 0 & 20 & 22 & 22 & 0 & 0 \end{bmatrix}$$

$$\lambda_b,\lambda_c,\lambda_e = 1 \quad \begin{bmatrix} 0 & 18 & 23 & 0 & -19 & 0 \\ 0 & -19 & -18 & 0 & 23 & 0 \\ 0 & 23 & 19 & 0 & -18 & 0 \end{bmatrix}$$

$$\lambda_b,\lambda_c,\lambda_f = 1 \quad \begin{bmatrix} 0 & -22 & 19 & 0 & 0 & -18 \\ 0 & 18 & 22 & 0 & 0 & 19 \\ 0 & 19 & 18 & 0 & 0 & -22 \end{bmatrix}$$

$$\lambda_b,\lambda_d,\lambda_e = 1 \quad \begin{bmatrix} 0 & -22 & 0 & 19 & 18 & 0 \\ 0 & -18 & 0 & -22 & 19 & 0 \\ 0 & 19 & 0 & 18 & 22 & 0 \end{bmatrix}$$

$$\lambda_b,\lambda_d,\lambda_f = 1 \quad \begin{bmatrix} 0 & 18 & 0 & 23 & 0 & 19 \\ 0 & 19 & 0 & 18 & 0 & 23 \\ 0 & 23 & 0 & 19 & 0 & 18 \end{bmatrix}$$

$$\lambda_b,\lambda_e,\lambda_f = 1 \quad \begin{bmatrix} 0 & -21 & 0 & 0 & 23 & -23 \\ 0 & 0 & 0 & 0 & 18 & 18 \\ 0 & 0 & 0 & 0 & 19 & -19 \end{bmatrix}$$

$$\lambda_c,\lambda_d,\lambda_e = 1 \quad \begin{bmatrix} 0 & 0 & 18 & 18 & 0 & 0 \\ 0 & 0 & 19 & -19 & 0 & 0 \\ 0 & 0 & -23 & 23 & 21 & 0 \end{bmatrix}$$

$$\lambda_c,\lambda_d,\lambda_f = 1 \quad \begin{bmatrix} 0 & 0 & 18 & 18 & 0 & 0 \\ 0 & 0 & 19 & -19 & 0 & 0 \\ 0 & 0 & 23 & -23 & 0 & -21 \end{bmatrix}$$

TABLE 3.2

Status Matrices (cont.)

$$\lambda_c, \lambda_e, \lambda_f = 1 \quad \begin{bmatrix} 0 & 0 & 20 & 0 & -22 & -22 \\ 0 & 0 & 0 & 0 & 18 & 18 \\ 0 & 0 & 0 & 0 & 19 & -19 \end{bmatrix} \qquad \lambda_d, \lambda_e = 0 \quad \begin{bmatrix} 13 & -12 & 17 & 0 & 0 & -25 \\ 16 & 17 & 10 & 0 & 0 & 11 \\ 15 & 14 & -24 & 0 & 0 & -16 \end{bmatrix}$$

$$\lambda_d, \lambda_e, \lambda_f = 1 \quad \begin{bmatrix} 0 & 0 & 0 & 20 & 22 & 22 \\ 0 & 0 & 0 & 0 & 18 & 18 \\ 0 & 0 & 0 & 0 & 19 & -19 \end{bmatrix} \qquad \lambda_c, \lambda_e = 0 \quad \begin{bmatrix} 12 & -13 & 0 & 17 & 0 & 25 \\ 17 & 16 & 0 & -10 & 0 & 11 \\ 14 & 15 & 0 & 24 & 0 & -16 \end{bmatrix}$$

$$\lambda_e, \lambda_f = 0 \quad \begin{bmatrix} 8 & -8 & 9 & 0 & 0 & 0 \\ 0 & 0 & 19 & -19 & 0 & 0 \\ 18 & 18 & 0 & 0 & 0 & 0 \end{bmatrix} \qquad \lambda_b, \lambda_e = 0 \quad \begin{bmatrix} 16 & 0 & 15 & 14 & 0 & -24 \\ 25 & 0 & 13 & -12 & 0 & 17 \\ 11 & 0 & -16 & -17 & 0 & -10 \end{bmatrix}$$

$$\lambda_c, \lambda_d = 0 \quad \begin{bmatrix} 19 & -19 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 18 & 18 \\ 9 & 9 & 0 & 0 & 8 & -8 \end{bmatrix} \qquad \lambda_a, \lambda_e = 0 \quad \begin{bmatrix} 0 & -16 & 14 & 15 & 0 & 24 \\ 0 & 25 & 12 & -13 & 0 & 17 \\ 0 & 11 & 17 & 16 & 0 & -10 \end{bmatrix}$$

$$\lambda_a, \lambda_b = 0 \quad \begin{bmatrix} 0 & 0 & 18 & 18 & 0 & 0 \\ 0 & 0 & 8 & -8 & 9 & 9 \\ 0 & 0 & 0 & 0 & 19 & -19 \end{bmatrix} \qquad \lambda_b, \lambda_d = 0 \quad \begin{bmatrix} 10 & 0 & 11 & 0 & -17 & -16 \\ 24 & 0 & 16 & 0 & 14 & 15 \\ 17 & 0 & -25 & 0 & 12 & -13 \end{bmatrix}$$

$$\lambda_d, \lambda_f = 0 \quad \begin{bmatrix} 12 & -13 & 17 & 0 & -25 & 0 \\ -17 & -16 & 10 & 0 & 11 & 0 \\ 14 & 15 & 24 & 0 & 16 & 0 \end{bmatrix} \qquad \lambda_a, \lambda_d = 0 \quad \begin{bmatrix} 0 & -10 & 11 & 0 & -16 & -17 \\ 0 & -24 & 16 & 0 & 15 & 14 \\ 0 & 17 & 25 & 0 & 13 & -12 \end{bmatrix}$$

$$\lambda_c, \lambda_f = 0 \quad \begin{bmatrix} 13 & -12 & 0 & 17 & 25 & 0 \\ -16 & -17 & 0 & -10 & 11 & 0 \\ 15 & 14 & 0 & -24 & 16 & 0 \end{bmatrix} \qquad \lambda_b, \lambda_c = 0 \quad \begin{bmatrix} 19 & 0 & 0 & 11 & 16 & 17 \\ -24 & 0 & 0 & -16 & 15 & 14 \\ 17 & 0 & 0 & -25 & 13 & -12 \end{bmatrix}$$

$$\lambda_b, \lambda_f = 0 \quad \begin{bmatrix} 16 & 0 & 14 & 15 & 24 & 0 \\ -25 & 0 & 12 & -13 & 17 & 0 \\ 11 & 0 & -17 & -16 & 10 & 0 \end{bmatrix} \qquad \lambda_a, \lambda_c = 0 \quad \begin{bmatrix} 0 & -10 & 0 & 11 & 17 & 16 \\ 0 & 24 & 0 & -16 & 14 & 15 \\ 0 & 17 & 0 & 25 & 12 & -13 \end{bmatrix}$$

$$\lambda_a, \lambda_f = 0 \quad \begin{bmatrix} 0 & -16 & 15 & 14 & -24 & 0 \\ 0 & -25 & 13 & -12 & 17 & 0 \\ 0 & 11 & 16 & 17 & 10 & 0 \end{bmatrix} \qquad \lambda_f = 0 \quad \begin{bmatrix} 8 & -8 & 9 & 9 & 0 & 0 \\ -1 & -1 & 2 & -2 & 3 & 0 \\ 4 & 4 & 5 & -5 & 6 & 0 \end{bmatrix}$$

TABLE 3.2

Status Matrices (cont.)

$$\lambda_e = 0 \quad \begin{bmatrix} 8 & -8 & 9 & 9 & 0 & 0 \\ 1 & 1 & 2 & -2 & 0 & 3 \\ 4 & 4 & -5 & 5 & 0 & -6 \end{bmatrix} \qquad \lambda_b = 0 \quad \begin{bmatrix} 6 & 0 & 4 & 4 & 5 & -5 \\ 0 & 0 & 8 & -8 & 9 & 9 \\ 3 & 0 & -1 & -1 & 2 & -2 \end{bmatrix}$$

$$\lambda_d = 0 \quad \begin{bmatrix} 2 & -2 & 3 & 0 & -7 & -7 \\ 5 & -5 & 6 & 0 & 4 & 4 \\ 9 & 9 & 0 & 0 & 8 & -8 \end{bmatrix} \qquad \lambda_a = 0 \quad \begin{bmatrix} 0 & -6 & 4 & 4 & -5 & 5 \\ 0 & 0 & 8 & -8 & 9 & 9 \\ 0 & 3 & 1 & 1 & 2 & -2 \end{bmatrix}$$

$$\lambda_c = 0 \quad \begin{bmatrix} 2 & -2 & 0 & 3 & 7 & 7 \\ -5 & 5 & 0 & -6 & 4 & 4 \\ 9 & 9 & 0 & 0 & 8 & -8 \end{bmatrix} \qquad \lambda_a,\lambda_b,\lambda_c,\lambda_d,\lambda_e,\lambda_f = 1 \quad \begin{bmatrix} 8 & -8 & 9 & 9 & 0 & 0 \\ 0 & 0 & 8 & -8 & 9 & 9 \\ 9 & 9 & 0 & 0 & 8 & -8 \end{bmatrix}$$

In operation, the first read-only memory determines which matrix in Table 3 2 will be required to process the sensors which are operable (λ_i=1). The second memory selects the words specified by this matrix. These words are stored as in Table 3.1. They are processed with the appropriate timing to allow each of the triad axes to be updated for any gyro outputs that have occurred. These words are used in 42 different sequences in performing the transformations in the arithmetic unit. There are three sets of memories as described above for processing from six to three axes, one for each of the triad axes.

The arithmetic unit consists of an adder and five data registers used in parallel processing to transform to $\Delta\theta x$, $\Delta\theta y$ and $\Delta\theta z$ (see Fig. 3-2). The sequence of events is as follows. Starting the updating to obtain a $\Delta\theta x$ input to the computer, the first specified word from the second read-only memory (ROM) (of the set of six words specified by the first ROM for the x-axis update)is loaded into the constant buffer. From the buffer the scaled increment is added to the previous remainder of $\Delta\theta x$ and the result is held in the general $\Delta\theta$ buffer register. Once this is done the selection of the next constant for the x-axis update from the ROM can begin. The $\Delta\theta x$ register is loaded again to prepare for the next component of the $\Delta\theta x$ update. After the six cycles are completed the remainder will be in the $\Delta\theta x$ remainder register and an output will be sent to the gyro-buffer logic if the entire update produced either a positive or negative overflow.

The gyro-buffer will recognize this overflow as the signal to produce a positive or negative output pulse that will update the appropriate direction cosines. This sequence will be followed by the second set of ROM's and its arithmetic register to provide a $\Delta\theta y$ input to the direction cosine matrix, if the inputs are of sufficient magnitude to produce an overflow. The same sequence will take place for the final set of ROM's to produce a $\Delta\theta z$ input pulse when required.

The above constitutes a complete update cycle and takes place within the system update time of 57.6 μs. The computer described above can function continuously at any input rate desired that does not exceed this pulse repetition rate.

3.2 CONVERSION FROM DODECAHEDRON TO TRIAD BY SENSOR CORRECTION

3.2.1 Description

In this section a method for converting sensor signals from redundant dodecahedron to equivalent triad axes is described in which signals missing because of sensor failure are restored by synthesis from operational sensor outputs. The resulting six inertial signals are then used as the input to a fixed transformation whose output is the desired equivalent triad set. A block diagram of the system is shown in Fig. 3-3. It consists of a sensor failure detection unit, a correction unit, and a dodecahedron-to-triad conversion unit. The detection unit is the parity equation integrator, which will be described in Sec. 3.2.2. The detection unit produces as its output a six-element binary vector, λ , each of whose elements, when equal to unity, affirms the operation of one of the sensors. The λ vector and the sensor outputs serve as inputs to the sensor correction unit. If a particular sensor is operative, the output of the correction unit corresponding to that sensor is the sensor signal itself. If a sensor has failed, then the correction unit output corresponding to that sensor is an estimate of what the sensor signal should be, derived from the operative sensor signals It is clear that at least three sensors must be operative in order that such an estimate can be made.

The six sensor signals (synthetic or actual) from the correction unit are used as inputs to the dodecahedron-to-triad conversion unit. Since six sensor signals are always assumed to be present at the input of this unit, even when as many as three sensors have actually failed, a simple, fixed conversion algorithm may be used. The techniques employed and the system itself will now be described in greater detail.

The method used for synthesizing failed-sensor signals is based upon solution of one of the parity equations for the missing signal in terms of the three valid signals. The parity equations, as defined by Gilmore,[1] are reproduced here as Table 3.3. As an example, if sensor A has failed but sensors B, C , and D are still operating, then Eq. (1) in Table 3.3 may be used for estimating A , viz,

$$A = B + \gamma(C + D) \qquad (3.11)$$

where

$$\gamma = \frac{\sin(\alpha)}{\cos(\alpha)} = \frac{\sqrt{5} - 1}{2} = 0.618033 \quad . \qquad (3.12)$$

A is the estimate of the output that would be obtained from sensor A if it were operating and α is the angle between two intersecting normals to adjacent faces of a dodecahedron. A general equation may be written for sensor correction :

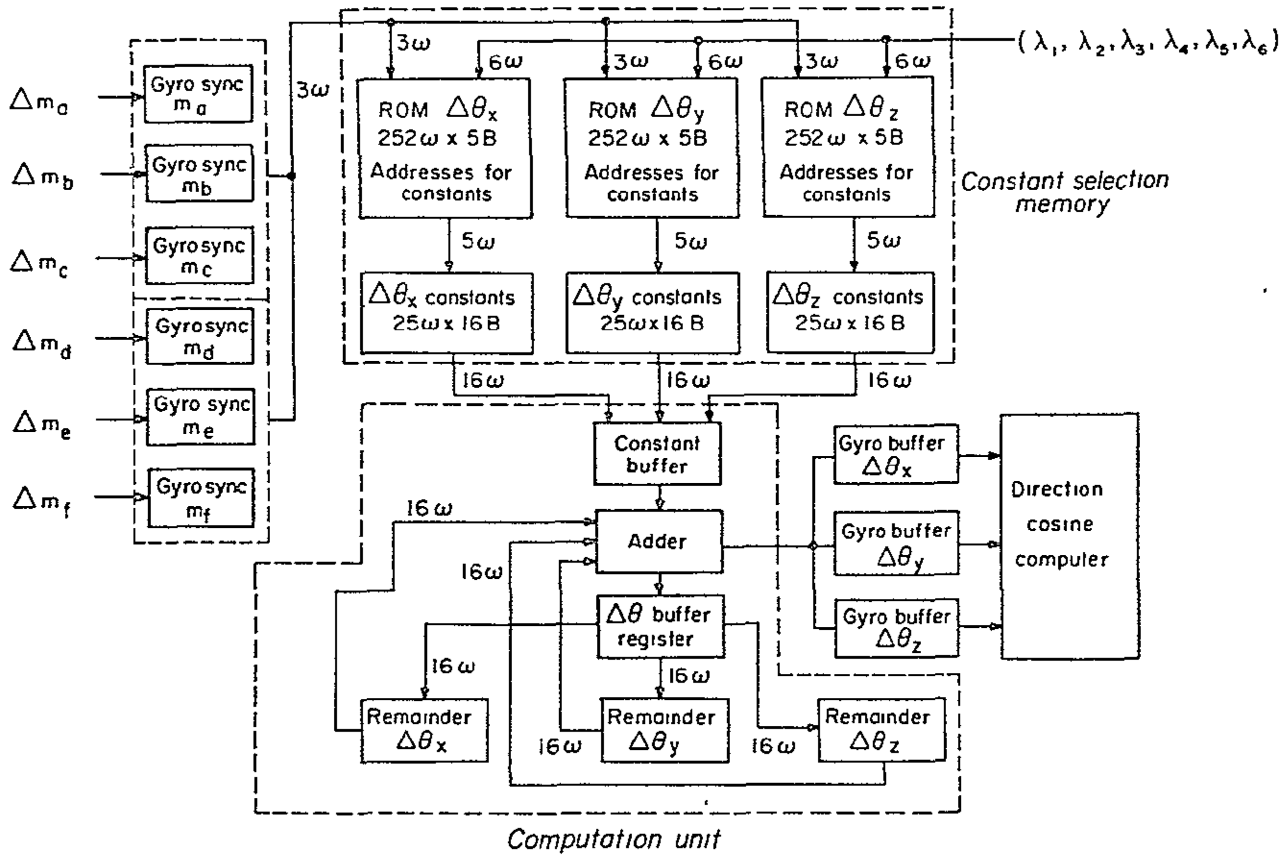


FIG. 3-2 Block diagram of six-gyro system using pseudo-inverse and status matrices method.

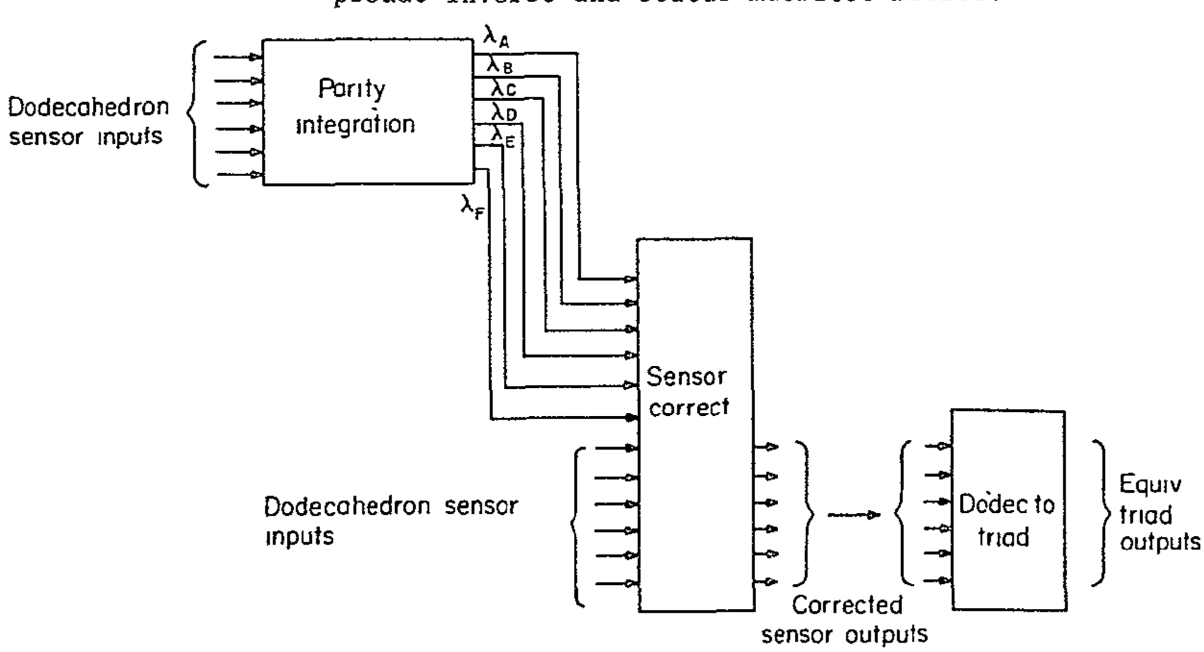


FIG. 3-3 Block diagram of sensor correction method.

$$
\begin{aligned}
\hat{1} = \lambda_1 1 + \bar{\lambda}_1 \Big\{ & \left(\lambda_2\lambda_3\lambda_4\bar{\lambda}_6 + \lambda_2\lambda_3\lambda_4\lambda_5\right)E_A \\
& + \lambda_2\lambda_3\bar{\lambda}_4\lambda_5\bar{\lambda}_6E_B + \lambda_2\lambda_3\bar{\lambda}_5\lambda_6E_C \\
& + \lambda_2\bar{\lambda}_3\lambda_4\lambda_5E_D + \lambda_2\bar{\lambda}_3\lambda_4\bar{\lambda}_5\lambda_6E_E \\
& + \lambda_2\bar{\lambda}_3\bar{\lambda}_4\lambda_5\lambda_6E_F + \bar{\lambda}_2\lambda_3\lambda_4\lambda_5\bar{\lambda}_6E_G \\
& + \bar{\lambda}_2\lambda_3\lambda_4\bar{\lambda}_5\lambda_6E_4 + \lambda_3\bar{\lambda}_4\lambda_5\lambda_6E_I \\
& + \bar{\lambda}_2\lambda_4\lambda_5\lambda_6E_J \Big\} \quad ;
\end{aligned}
\tag{3.13}
$$

TABLE 3.3

Failure Detection Parity Equations

No.	Instrument	Equation
1	ABCD	(A - B)c - (C + D)s = 0
2	ABCE	(B + C)c - (A + E)s = 0
3	ABCF	(C - A)c + (B - F)s = 0
4	ABDE	(D - A)c + (B + E)s = 0
5	ABDF	(B + D)c - (A - F)s = 0
6	ABEF	(E - F)c - (A + B)s = 0
7	ACDE	(D + E)c - (A - C)s = 0
8	ACDF	(F - C)c - (A + D)s = 0
9	ACEF	(A + F)c - (C + E)s = 0
10	ADEF	(E - A)c + (D - F)s = 0
11	BCDE	(E - C)c + (D - B)s = 0
12	BCDF	(F + D)c + (B - C)s = 0
13	BCEF	(B - E)c + (C + F)s = 0
14	BDEF	(B + F)c + (D - E)s = 0
15	CDEF	(C - D)c - (E + F)s = 0

$$c = \cos(\alpha) = \left(\frac{\sqrt{5} + 5}{10}\right)^{1/2} \approx 0.85065$$

$$s = \sin(\alpha) = \left(\frac{5 - \sqrt{5}}{10}\right)^{1/2} \approx 0.52574$$

To interpret Eq. (3.13) let 1 represent the output of sensor 1 and let $\hat{1}$ represent the estimate of this quantity. The various λ's are the operational indicators of their respective sensors or, when barred, the complements of these quantities. Finally, the quantities E_A, E_B, etc., represent the parity equations to be used in computing $\hat{1}$. Thus, Eq. (3.13) serves to select which (if any) equation will be used to synthesize the sensor output, based upon the operational status of all sensors in the system.

Equation (3.13) can be made specific to any particular sensor by means of Table 3.4, which indicates the substitution of subscripts for correction of a given sensor. The synthesis of a failed-sensor output from one of the parity equations involves multiplying certain of the operational outputs by either unity, γ or $\gamma^{-1} = 1+\gamma$ and then summing these quantities. The selection of quantities to be summed to produce the desired synthetic sensor output is governed by the λ vector. A resolver implementation of sensor output synthesis is shown in Fig 3-4 for the general case. Six such resolvers are required, one for each dodecahedron axis. The resolver signals S_i, either multiplied by γ or by unity, are gated into a resolver by sequencing signals T_{ij} and control signals C_{ij}. The overflow output of this resolver is the dodecahedron sensor signal, either synthetic or real, to be used in the dodecahedron-to-triad conversion. The control signals C_{ij} are derived from Eq. (3.13) as functions of the λ vector and are given in general form by Table 3.5. Table 3.5 is made specific to a particular sensor by means of Table 3.4. Implementation of the control functions is by means of a read-only memory whose inputs are the five λ_i and whose outputs are the C_{ij}. This memory has a capacity of 16 words, each 15 bits long. Six identical memories are used, one for each of the dodecahedron sensors. Each memory is made specific to an axis by the combination and permutation of λ's used as its input. These are specified by Table 3.4.

TABLE 3.4

Table for Substitution in General 1-Gyro Restoring Equation

Sensor	General Variable															
	λ_1	λ_2	λ_3	λ_4	λ_5	λ_6	E_A	E_B	E_C	E_D	E_E	E_F	E_G	E_H	E_I	E_J
A	λ_A	λ_B	λ_C	λ_D	λ_E	λ_F	E_1	E_2	E_3	E_4	E_5	E_6	E_7	E_8	E_9	E_{10}
B	λ_B	λ_A	λ_C	λ_D	λ_F	λ_E	E_1	E_3	E_2	E_5	E_4	E_6	E_{12}	E_{11}	E_{13}	E_{14}
C	λ_C	λ_D	λ_E	λ_F	λ_A	λ_B	E_{15}	E_7	E_{11}	E_8	E_{12}	E_1	E_9	E_{13}	E_2	E_3
D	λ_D	λ_C	λ_E	λ_F	λ_B	λ_A	E_{15}	E_{11}	E_7	E_{12}	E_8	E_1	E_{14}	E_{10}	E_4	E_5
E	λ_E	λ_F	λ_A	λ_B	λ_C	λ_D	E_6	E_9	E_{10}	E_{13}	E_{14}	E_{15}	E_2	E_4	E_7	E_{11}
F	λ_F	λ_E	λ_A	λ_B	λ_D	λ_C	E_6	E_{10}	E_9	E_{14}	E_{13}	E_{15}	E_5	E_3	E_8	E_{12}

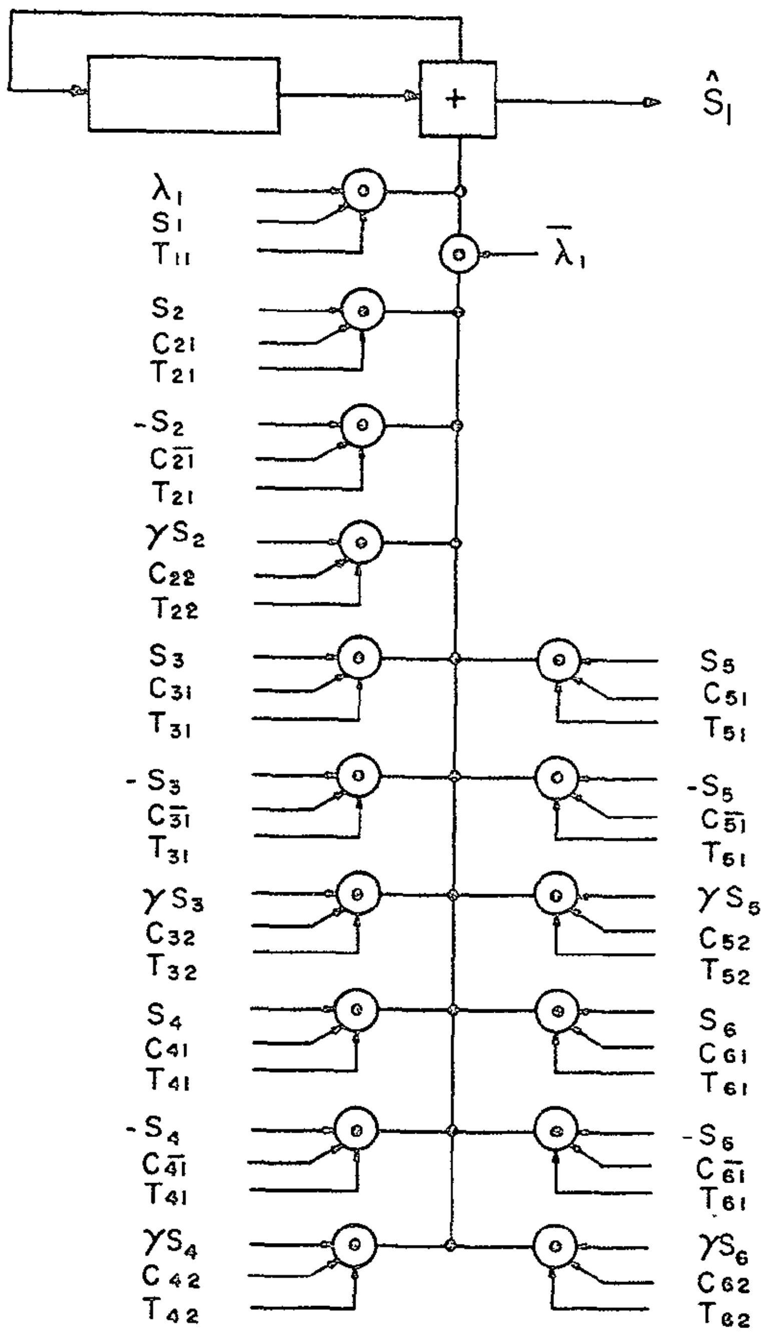


FIG. 3-4 Dodecahedron gyro correction resolver (6 required).

TABLE 3.5

Table of Binary Constants for Dodecahedron Gyro Correction (see Fig. 3-4)

Eq.	λ_2 λ_3 λ_4 λ_5 λ_6	c_{21}	$\overline{c_{21}}$	c_{22}	c_{31}	$\overline{c_{31}}$	c_{32}	c_{41}	$\overline{c_{41}}$	c_{42}	c_{51}	$\overline{c_{51}}$	c_{52}	c_{61}	$\overline{c_{61}}$	c_{62}
A	1 1 1 0 0 1 1 1 1 0 1 1 1 1 1	1	0	0	0	0	1	0	1	0	0	0	0	0	0	0
B	1 1 0 1 0	1	0	1	1	0	1	0	0	0	0	1	0	0	0	0
C	1 1 0 0 1 1 1 1 0 1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	1
D	1 0 1 1 0 1 0 1 1 1	0	0	1	0	0	0	1	0	0	0	0	1	0	0	0
E	1 0 1 0 1	1	0	1	0	0	0	1	0	1	0	0	0	1	0	0
F	1 0 0 1 1	0	1	0	0	0	0	0	0	0	1	0	1	0	1	1
G	0 1 1 1 0	0	0	0	1	0	0	1	0	1	1	0	1	0	0	0
H	0 1 1 0 1	0	0	0	1	0	1	0	1	0	0	0	0	0	1	1
I	0 1 0 1 1 1 1 0 1 1	0	0	0	0	0	1	0	0	0	0	0	1	0	1	0
J	0 0 1 1 1 0 1 1 1 1	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1

When all six dodecahedron sensor signals are assumed present, the dodecahedron-to-triad conversion takes the form

$$\begin{bmatrix} S_x \\ S_y \\ S_z \end{bmatrix} = \frac{1}{2} \begin{bmatrix} \sin\alpha & -\sin\alpha & \cos\alpha & \cos\alpha & 0 & 0 \\ 0 & 0 & \sin\alpha & -\sin\alpha & \cos\alpha & \cos\alpha \\ \cos\alpha & \cos\alpha & 0 & 0 & \sin\alpha & -\sin\alpha \end{bmatrix} \cdot \begin{bmatrix} \hat{S}_A \\ \hat{S}_B \\ \hat{S}_C \\ \hat{S}_D \\ \hat{S}_E \\ \hat{S}_F \end{bmatrix} \qquad (3.14)$$

The implementation of the conversion is by means of three resolver circuits, each identical to the one shown in Fig. 3-5. Whole-number representations of the constant half-sines and half-cosines are gated into the resolver by synthetic sensor outputs $\hat{S}_i$, in accordance with Eq. (3.14) and by sequencing signals T_i. The overflow output of each resolver is the respective triad signal.

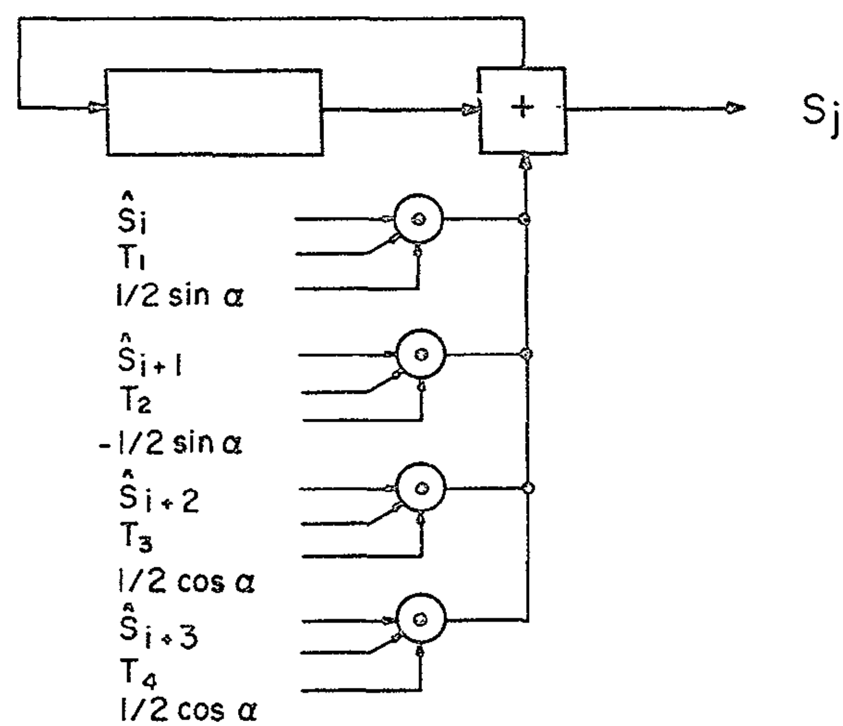


FIG. 3-5 Triad axis generator (3 required).

3.2.2 Parity Integration

To implement the parity equation determination of gyro failure, a weighted integration technique will be adopted. The reasoning behind the use of integration is that quantization effects in both the gyros and the computer must be separated from significant gyro deviations. Weighting is included to mask out long-term drift due to roundoff and other effects within the parity computation itself. Consistent with other sections of the computer, an operational digital procedure will be used in the parity section of the system.

The form of the parity conditions used closely resembles that shown by Gilmore.[1] That is, for the first parity equation

$$Q_1 = \left(S_A - S_B\right)\cos\alpha - \left(S_C + S_D\right)\sin\alpha \tag{3.15}$$

where S_i is the output of gyro i and is a measure of ω_i. For perfect gyros, $Q_1=0$. Thus Q_i represents an angular-rate error. Since pulsed gyros are assumed, where each output pulse represents an increment of angle, the first parity equation may be written in angle form as

$$dX_1 \mathrel{\dot=} Q_1 dt = \left(d\theta_A - d\hat{\theta}_B\right)\cos\alpha - \left(d\theta_C + d\theta_D\right)\sin\alpha \tag{3.16}$$

and similarly for the other equation. Temporal weighting and integration are introduced by defining the quantity P_1,

$$P_1 = \int_0^t e^{-K(t-\tau)}\, dX_1 \tag{3.17}$$

or

$$P_1 = \int_0^t e^{-K(t-\tau)} \left(\frac{dX_1(\tau)}{d\tau}\right) d\tau \quad . \tag{3.18}$$

If $dX_1/dt = 0$ for $t<0$ then P_1 is seen to be the convolution of dX_1/dt with e^{-Kt} The differential form of the above integral equation may readily be shown to be

$$\frac{dP_1}{dt} = -KP_1 + \frac{dX_i}{dt} \tag{3.19}$$

or

$$dP_1 = -KP_1 dt + dX_1 \tag{3.20}$$

The final form implies a simple DDA solution, as shown in Fig. 3-6.

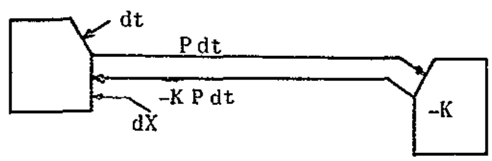


FIG. 3-6 DDA solution.

The implementation of this configuration using resolvers is shown in Fig. 3-7. A complete solution to allow correction of two failures and detection of three simultaneously requires 15 units of the type in Fig. 3-7, although the elements so designated may be shared among the 15 systems. It is also possible to mechanize a single-failure correction scheme using just six of the units in Fig. 3-7 in which the dX_i inputs are switched among the parity networks. The gyro failures (up to two) are indicated by the binary quantities λ_i, i=A,B,C,D,E,F . The value $\lambda_i=1$ corresponds to the condition that gyro i is operative. The λ_i are binary functions of the parity threshold functions E_j . Specifically,

$$\bar{\lambda}_i = \prod_{\substack{\text{subset} \\ i}} \left(E_j\right) \tag{3.21}$$

The E_j subset for each λ_i is indicated by Table 3.6. The E_j numbering corresponds to Table 2 in Gilmore. Detection of three or more failures is given by

$$\bar{\lambda}_T = \prod_{j=1}^{15} E_j \quad . \tag{3.22}$$

The condition $\lambda_T = 0(\bar{\lambda}_T = 1)$ indicates that at least three gyros have failed, but the failures cannot be isolated. This is equivalent to complete system failure.

The rationale behind the use of the weighted integral form of parity equation follows that discussed by Keene.[2] The prevalent type of sensor failure is performance degradation, evidenced by either scale factor change or by excessive bias. Let g be the sensor output. Then

$$g = (a + \epsilon)\omega + b \tag{3.23}$$

where a is the scale factor, ϵ is the scale factor error, b is the bias, and ω is the measured quantity. Since integrating sensors are normally used, the output of interest is

$$\int g dt = (a + \epsilon) \int \omega dt + bt \quad . \tag{3.24}$$

Then for constant $\omega = \omega_0$,

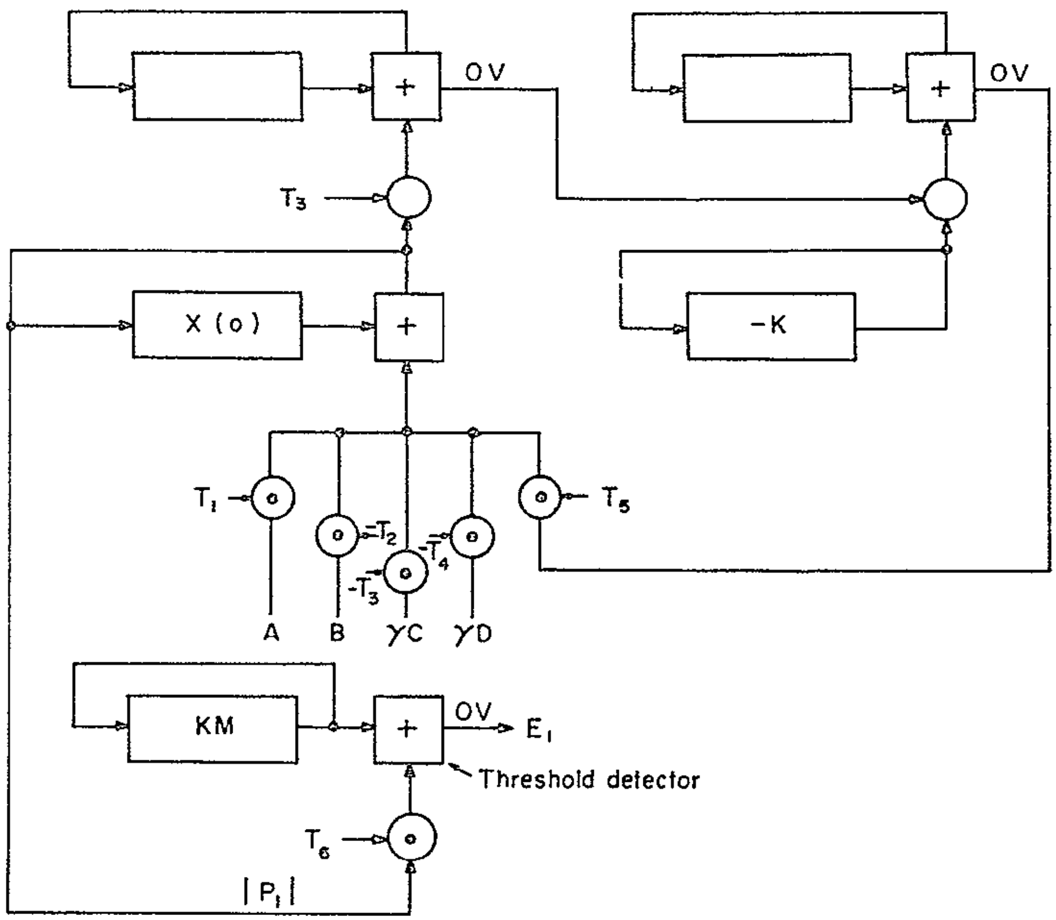


FIG. 3-7 Parity integrator.

TABLE 3.6

E_J Subset for λ_i

	λ_A	λ_B	λ_C	λ_D	λ_E	λ_F
E_1	1	1	1	1		
E_2	1	1	1		1	
E_3	1	1	1			1
E_4	1	1		1	1	
E_5	1	1		1		1
E_6	1	1			1	1
E_7	1		1	1	1	
E_8	1		1	1		1
E_9	1		1		1	1
E_{10}	1			1	1	1
E_{11}		1	1	1	1	
E_{12}		1	1	1		
E_{13}		1	1		1	1
E_{14}		1		1	1	1
E_{15}			1	1	1	1

$$\hat{\theta} = \int g dt = a\omega_0 t + (\varepsilon\omega_0 + b)t \qquad (3.25)$$

implying a ramp error behavior. Since X_i is a linear combination of the $\hat{\theta}$'s, it too exhibits ramp behavior under these conditions. Let $X_i = \theta_0 t$. Then Eq. (3.19) becomes

$$\frac{dP_i}{dt} = -KP_i + Q_0 \qquad (3.26)$$

or

$$P_i = \frac{Q_0}{K}\left(1 - e^{-Kt}\right) \quad . \qquad (3.27)$$

If M is defined as the maximum allowable value of P_i, then threshold is exceeded for

$$P_i > M$$

or

$$\frac{Q_o}{K}\left(1 - e^{-Kt}\right) > M \quad . \tag{3.28}$$

Solving for the time t_M at which the inequality is satisfied,

$$t_M > \frac{1}{K}\,\ell n\left(\frac{Q_o}{Q_o - KM}\right) \tag{3.29}$$

Thus, KM is established as the minimum error rate for parity failure, and the time until such failure is detected is adjusted by K and decreases with increasing Q_o .

3.2.3 Constant Rate Multipliers

Although it has been suggested that standard resolvers be used to effect the multiplication of a sensor or other pulse-rate signal by a constant quantity, another interesting approach has been developed. The basis for this approach is given in Appendix A (Smooth Sequences) and stems from the realization that the pulse rate resulting from the multiplication of the source rate by a fixed quantity (less than unity) should preserve as well as possible the relative pulse density of the source. Although the use of resolver-type rate multipliers does in fact preserve relative densities optimally, several techniques have been developed which perform the same function with appreciable circuit savings. One of these well suited for applications in the dodecahedron sensor transformation computer will be described. Consider Fig. 3-8, which shows an interconnection of pulse counters and inhibit elements. The upper counter produces one output pulse for every a_1 input pulses, and so forth. Each time a pulse counter produces an output pulse it inhibits one input pulse to the counter immediately above it. Analysis of this circuit shows that the pulse output frequency is equal to the pulse output frequency multiplied by P/Q , where P and Q are relatively prime integers and

$$\frac{P}{Q} = \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3 + \cfrac{1}{a_4 + \ddots \cfrac{1}{a_n}}}}} \tag{3.30}$$

Equivalently, P output pulses are produced for every Q input pulses. The P output pulses so produced are synchronous with input pulses, but are spaced as uniformly over the input pulses as is possible.

The representation of P/Q in Eq. (3.30) is as a simple continued fraction. The properties of such fractions are well known. One property of particular interest here is the

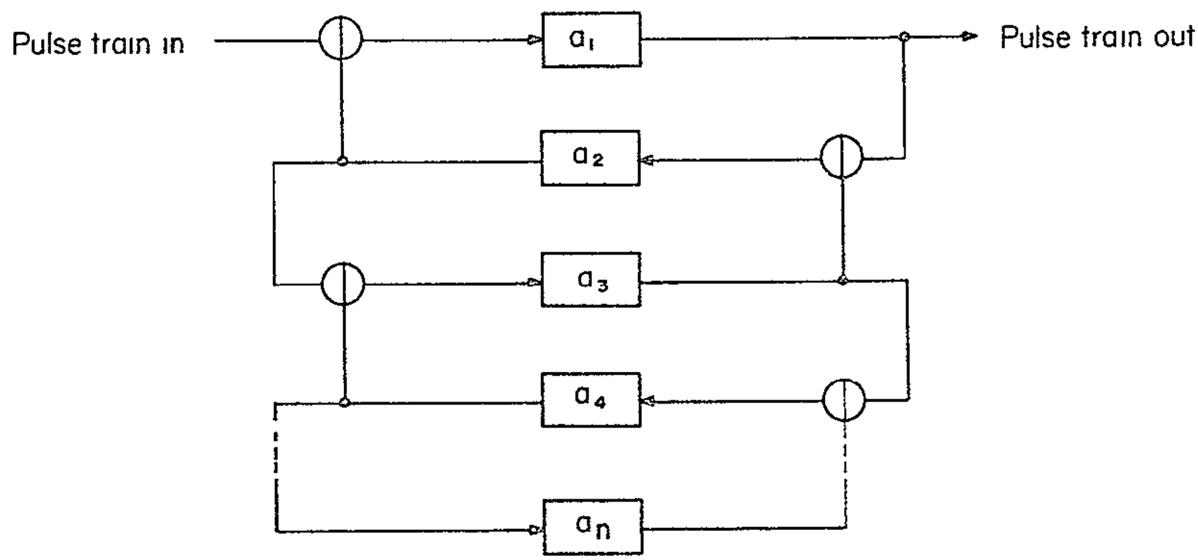


FIG. 3-8 Pulse rate multiplier.

representation of irrational numbers by their approximation in continued fraction form. Theory shows that all algebraic irrational numbers may be represented by an infinite continued fraction whose partial quotients (a_i in Eq. (3.30)) are bounded. The theory shows further that the approximation of any rational or irrational number by a truncated simple continued fraction is optimal in the sense that the absolute error incurred is smallest over all rational approximations with denominator no larger than that of the continued-fraction-derived approximation.

A case of special interest in the dodecahedron-to-triad conversion system is the multiplication of sensor output pulse rates by the quantities

$$\gamma = \frac{\sin \alpha}{\cos \alpha} = \frac{\sqrt{5} - 1}{2} \tag{3.31}$$

and

$$\gamma^{-1} = \frac{\cos \alpha}{\sin \alpha} = \frac{\sqrt{5} + 1}{2} = 1 + \gamma \tag{3.32}$$

where 2α is the angle measured between any two dodecahedron normals. The expansion of γ as an infinite simple continued fraction is given in Eq. (3.33)

$$\gamma = \cfrac{1}{1 + \cfrac{1}{1 + \cfrac{1}{1 + \cfrac{1}{1 + \ddots}}}} \tag{3.33}$$

This leads to a very simple pulse-rate multiplier of the type shown in Fig. 3-8, since all the a_i are unity and are therefore just direct connections. The resolution of this multiplier is determined by the number of stages used. For a resolver-type multiplier the resolution is doubled with each stage that is added to the storage register, since this has the effect of adding one more bit to the number added. It is interesting to compare the resolution per stage of the resolver to that of the continued fraction multiplier. Clearly, the resolution of an n-bit resolver-type multiplier is $1/2^n$ and is independent of the quantity by which the rate is multiplied. The approximate resolution of the continued fraction multiplier can be calculated using known properties of continued fraction expansions. The n^{th} convergent of a continued fraction is defined as the rational fraction obtained by truncating the continued fraction after the n^{th} partial quotient. If C_n is the n^{th} convergent of some continued fraction, then

$$C_n = P_n/Q_n \tag{3.34}$$

and

$$C_n - C_{n-1} = \frac{(-1)^{n-1}}{Q_n Q_{n-1}} \quad . \tag{3.35}$$

In Eq. (3.33)

$$\lim_{n\to\infty} C_n = \gamma \quad .$$

The odd-order convergents form a monotonic sequence that converges to γ from above and the even-order convergents form a monotonic sequence that converges to γ from below. The absolute error $E_n = |C_n - \gamma|$ is therefore bounded by the difference between the n^{th} and $n-1^{st}$ convergents. Letting this difference be D_n,

$$|D_n| = |C_n - C_{n-1}| = \frac{1}{Q_n Q_{n-1}} \tag{3.36}$$

and

$$|D_{n-1}| = |C_{n-1} - C_{n-2}| = \frac{1}{Q_{n-1} Q_{n-2}} \tag{3.37}$$

but

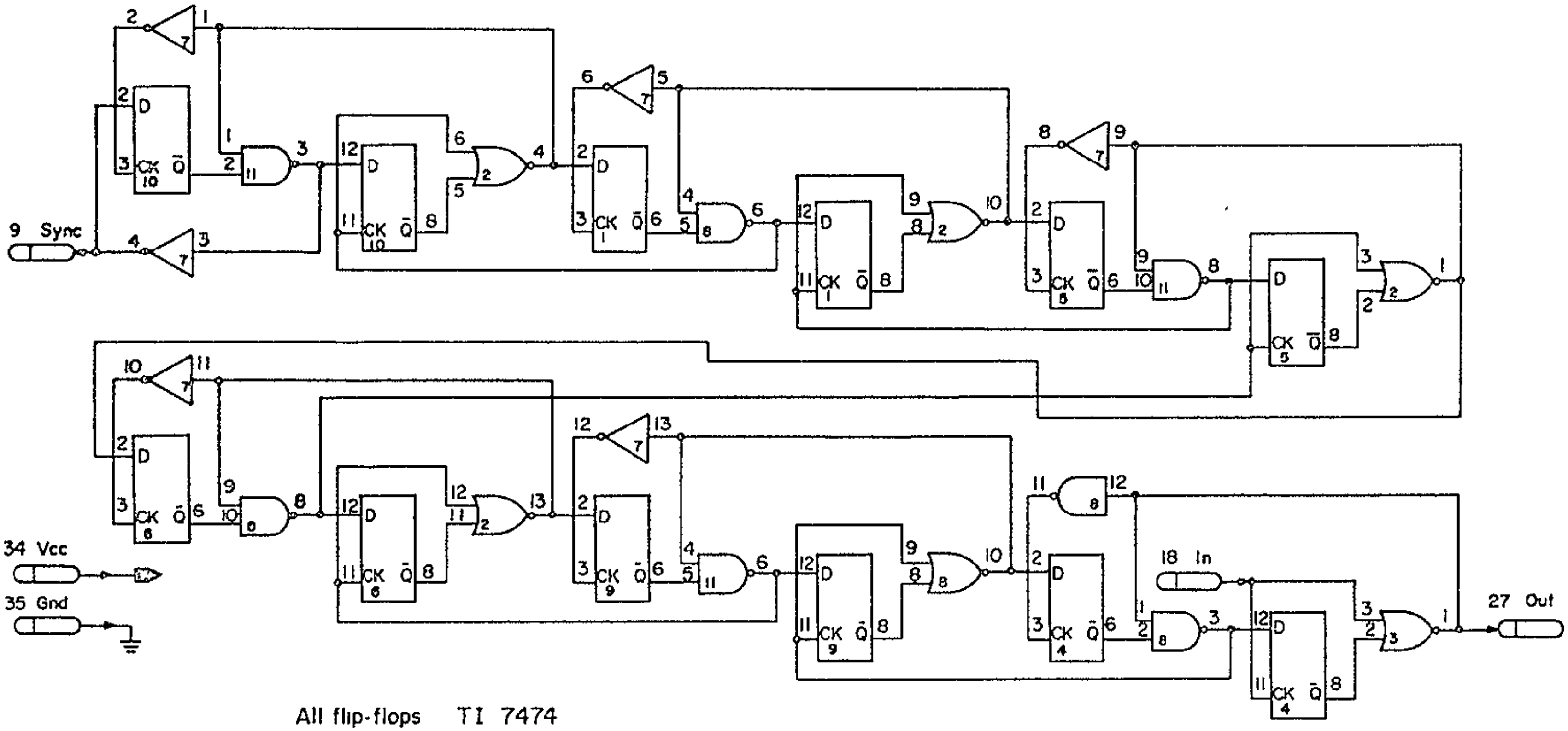
$$\lim_{n\to\infty} Q_{n-1} = \gamma Q_n \quad . \tag{3.37}$$

Therefore, in the limit,

$$D_n \rightarrow \gamma^2 D_{n-1} \quad . \tag{3.38}$$

Thus, each additional partial quotient included in the continued fraction approximation to γ increases the resolution by a factor of approximately $\gamma^{-2} = (3+\sqrt{5})/2 = 2.618$, and as n increases the resolution approaches $1/\gamma^{-2n} = 1/(1+\gamma)^{2n}$. The resolution of resolver-type and continued-fraction rate multipliers is compared in Table 3.7. In the strapdown system a resolution equivalent to a word length of 16 bits is generally required. It may be seen from Table 3.7 that this resolution is obtained using only twelve stages of the continued fraction multiplier. Another advantage of the continued fraction multiplier is that it is essentially a parallel system with no carry propagation delays. Therefore, slower electronic components may be used to attain the samespeeds as the equivalent resolver.

A working model of a 12-stage continued fraction multiplier was constructed. The circuit diagram of this unit appears as Fig. 3-9.



All flip-flops TI 7474
All NAND gates TI 7400
All NOR gates TI 7402
All inverters TI 7404

FIG. 3-9 Continued-fraction pulse rate multiplier.

TABLE 3.7

Comparison of Resolution of Resolver
and Continued Fraction Ratio Multipliers

n	Resolution	Cont. Fraction Resolution
1	2	2
2	4	6
3	8	15
4	16	40
5	32	104
6	64	273
7	128	714
8	256	1870
9	512	4895
10	1024	12816
11	2048	33552
12	4096	87841
13	8192	229970
14	16384	602070
15	32768	1576239
16	65536	4126648
17	131072	10803704

REFERENCES

1. "A Non-Orthogonal Multi-Sensor Strapdown Inertial Reference Unit," J. P. Gilmore, MIT/IL No. E2308, August 1968.

2. "The Meaning of Parity," D. Keene, MIT/IL, SIRU Memo No. 276, 28 Nov. 1969.

CHAPTER 4

INCREMENTAL COMPUTER CONFIGURATION FOR A REDUNDANT CMG CONTROL SYSTEM

4.1 INTRODUCTION

The objective of this study is to present an incremental computer structure for a space vehicle attitude control system that uses a set of six skewed, single-gimbal, control moment gyros (CMG's) as actuators and a set of six rate gyros in a dodecahedron configuration as rate sensors. The computer is to employ the redundancy in both the sensors and actuators to provide fail-operational performance. The CMG configuration of six skewed, single-gimbal gyros (exemplified by the Sperry 6-GAMS configuration) can be sized to provide three-axis control with failures in as many as three of its gyros, and the dodecahedron-configured rate gyro package can provide vehicle rate information from any three of its six rate gyros.

4.2 GENERAL DESCRIPTION OF THE CMG CONTROL SYSTEM

Since the computations to be performed by the CMG control computer are directly related to the specific configuration and steering law selected for the CMG's and also to the rate sensor configuration, these items are described briefly in paragraphs 4.2.1, 4.2.2, and 4.2.3, respectively. Paragraph 4.2.4 identifies the functions to be performed by the computer, including a discussion on three-variable versus six-variable input data to the steering law computer. A brief discussion on computer failure detection is presented in paragraph 4.2.5.

4.2.1 The CMG Configuration

Previous studies conducted jointly by Sperry and Lockheed for the Air Force[1] have identified a family of skewed, single-gimbal CMG configurations, denoted as the GAMS family of CMG configurations, to be the most efficient in terms of weight, power, and volume in providing redundant three-axis attitude control. Specifically, the 6-GAMS configuration, illustrated in Figure 4.2-1, is particularly suitable where control is to be maintained with any two of the six CMG's not operating. Although the control computer in this study is directed principally toward this configuration, the results are expected to be adaptable to other configurations.

The black arrows in the CMG configuration model shown in Figure 4-1 depict the angular momentum vectors $\{h_i\}$ of the six gyros, all of approximately equal magnitude. These vectors are shown in their respective reference directions in the figure. The notation used in describing the configuration angles is shown in Figure 4-2. All gimbal axes are tilted at an angle β from the x-axis, and the projections of the gimbal axes on the yz-plane are directed at angles $\gamma = \{0, 60, 120, 180, 240, 300\}$ degrees from the y-axis. The gimbal angles denoted by $\alpha = \{\alpha_1, \ldots, \alpha_6\}$ are measured from their respective reference directions in the counterclockwise direction about the gimbal axes as shown by

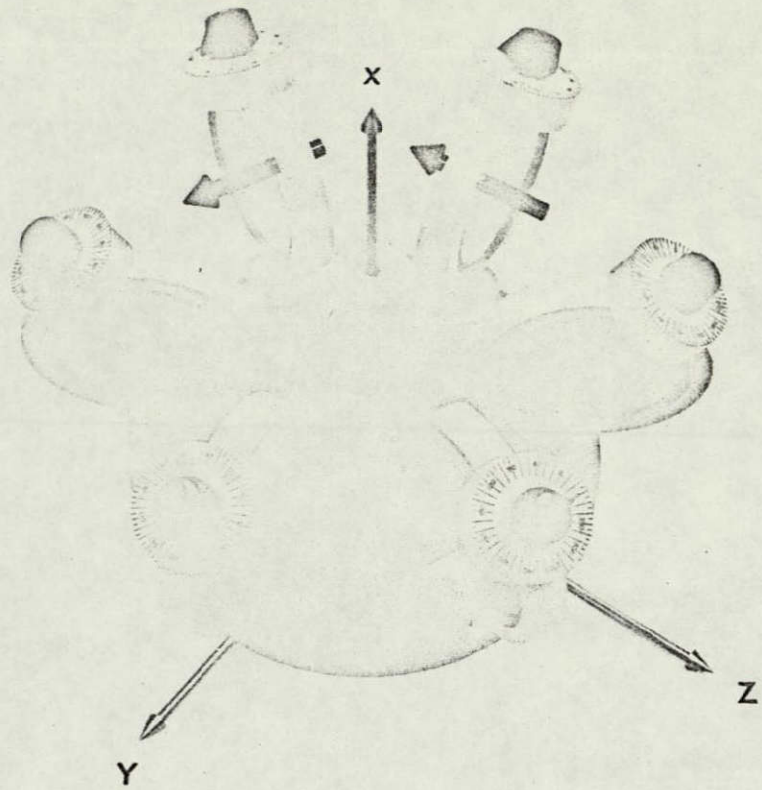


FIG. 4-1 Model of the Sperry 6-GAMS configuration.

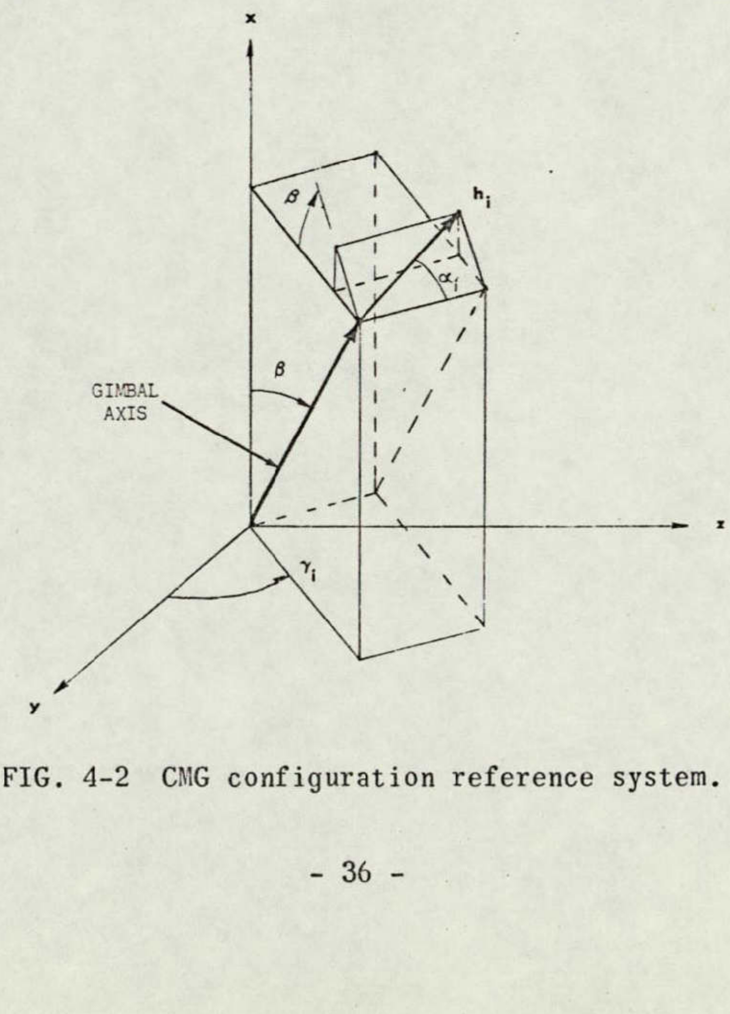


FIG. 4-2 CMG configuration reference system.

the angular scales of each gyro in Figure 4-1. For $\alpha = \{0, \ldots, 0\}$, the net angular momentum vector, H, of the CMG configuration is zero, assuming that all the individual momenta, $\{h_i\}$, are of equal magnitude. In general, the projections of $\vec{H}$ onto the vehicle principal axes are given by equations (4.1), where $S_{(\ )}$ denotes sin ( ) and $C_{(\ )}$ denotes cos ( ).

$$
\begin{aligned}
H_x &= \sum_{i=1}^{6} h_i \, S_\beta S_{\alpha_i} \\
H_y &= \sum_{i=1}^{6} -h_i \left(S_{\gamma_i} C_{\alpha_i} + C_{\beta_i} C_{\gamma_i} S_{\alpha_i} \right) \\
H_z &= \sum_{i=1}^{6} h_i \left(C_{\gamma_i} C_{\alpha_i} - C_{\beta_i} S_{\gamma_i} S_{\alpha_i} \right) \quad .
\end{aligned}
\tag{4.1}
$$

Equations (4.1) do not include the angular momentum due to the rotations of the gimbals and rotors about the gimbal axes which, for the purpose of steering law computations, may be assumed to be negligibly small in comparison with the angular momentum due to the rotation of the rotors about their spin axes.

CMG systems provide attitude control by appropriate transfer of angular momentum between the CMG'S and the vehicle. The mathematical law by which desired vehicle angular accelerations (or torques) are translated to gimbal rates is referred to as a steering law.

4.2.2 CMG Steering Laws

The equation of rigid-body angular motion for the vehicle (not including the CMG rotors and gimbals) is given by

$$T_v = I_v \dot{\omega} + \Omega \, I_v \, \omega \tag{4.2}$$

where

T_v = 3 x 1 matrix of projections (onto the vehicle principal axes) of the net torque vector $\vec{T}_v$ acting on the vehicle

I_v = diagonal matrix of vehicle inertias about the principal axes

ω = 3 x 1 matrix of projections of the vehicle rate vector $\vec{\omega}$

$\dot{\omega}$ = 3 x 1 matrix of projections of the vehicle acceleration $\dot{\vec{\omega}}$

$$\Omega = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \tag{4.3}$$

Similarly, the equation of motion for each CMG rotor can be expressed in terms of angular momentum as

$$T_{R_i} = \dot{H}_i + \Omega H_i \tag{4.4}$$

where T_{R_i} and H_i are the 3 x 1 matrices of projections of the torque and angular momentum vectors, respectively, for CMG number i. The net torque, T, produced by all six CMG's on the vehicle, neglecting gimbal inertias, is then given by

$$\begin{aligned} T &= -\sum_{i=1}^{6} T_{R_i} \\ &= -\dot{H} - \Omega H \end{aligned} \tag{4.5}$$

where

$$H = \sum_{i=1}^{6} H_i \quad . \tag{4.6}$$

By the law of conservation of angular momentum,

$$I_v \omega + H = H_n \tag{4.7}$$

where

$$H_n = I_v \omega_o + \int_o^t T_d dt' \tag{4.8}$$

H_n = net angular momentum of the vehicle plus the CMG's

ω_o = ω at time t = o

T_d = torque acting on the vehicle not caused by the CMG,

i.e., $T_v = T + T_d$.

The CMG system is normally initialized so that H is close to zero when $\omega = 0$; i.e., ω_o is normally close to zero.

From the previous equations, the vehicle angular acceleration is given by

$$\begin{aligned} \dot{\omega} &= -I_v^{-1} \left[\dot{H} + \Omega\,(H_n) + T_d\right] \\ &\approx -I_v^{-1}\ \dot{H} . \end{aligned} \tag{4.9}$$

This approximation becomes an equality if $\omega_o = 0$ and $T_d = 0$. The vehicle acceleration, $\dot{\omega}$, can thus be controlled by controlling $\dot{H}$. If $\dot{H}$ can be controlled so that

$$\dot{H} = -I_V\dot{\omega}_c \tag{4.10}$$

where $\dot{\omega}_c$ is a commanded $\dot{\omega}$, the torque produced on the vehicle by the CMG's is given by

$$T = I_V\dot{\omega}_c + \Omega\ I_V\left(\omega - \omega_n\right) \tag{4.11}$$

where $\omega_n = I_V^{-1}H_n$, the vehicle rate not caused by the CMG system. For $\omega_n \approx 0$, the CMG system thus inherently compensates for the centrifugal couple, $\Omega I_V\omega$, produced when ω has components in two or more of the principal axes for vehicles with unequal inertias.

By taking the time-derivatives of the components of H, given in equations (4.1) for the 6-GAMS configuration, $\dot{H}$ can be expressed by

$$\dot{H} = A\ \dot{\alpha} \tag{4.12}$$

where A is a 3 x 6 matrix of elements, a_{ij}, given by

$$\begin{aligned}
a_{1j} &= h_j\ S_\beta\ C_{\alpha_i} \\
a_{2j} &= h_j\left(S_{\gamma_j}\ S_{\alpha_j} - C_\beta\ C_{\gamma_i}\ C_{\alpha_i}\right) \\
a_{3j} &= -h_j\left(C_{\gamma_j}\ S_{\alpha_j} + C_\beta S_{\gamma_j} C_{\alpha_j}\right)
\end{aligned} \tag{4.13}$$

for the stated configuration. Thus, the matrix A is a function of the gimbal angles $\alpha = \{\alpha_1, \ldots, \alpha_6\}$.

$\dot{H}$ can therefore be controlled by controlling the gimbal rates, $\dot{\alpha}$. To control $\dot{H}$ according to equation (4.10), a solution of equation (4.12) is required. Since A is singular, any solution that exists is not unique. The specific solution that is adopted for a CMG system, even if it is only an approximate solution, is commonly referred to as the CMG steering law.

The selection of a steering law significantly affects the sizing (magnitude of h_i) required for the CMG's in order for the system to be able to meet a specified angular momentum envelope requirement. The most prominent steering law is based on the pseudo-inverse [2,3] of A and is referred to as the pseudo-inverse steering law. The significant feature of this inverse, denoted by $A^{\dagger}$, is that it provides the unique minimum norm solution if a solution exists. In this case, $A^{\dagger}$ can be computed by

$$A^{\dagger} = A^{T}\left(AA^{T}\right)^{-1} \quad . \tag{4.14}$$

In minimizing the norm of the gimbal rates, this steering law tends to emphasize the gimbal rates of the CMG's that most efficiently provide momentum transfer in the commanded direction, thereby tending to maximize the net angular momentum envelope obtainable for the CMG configuration. Although other steering laws have been investigated (such as the superposition of solutions for three CMG's, and the transpose steering law which provides a very approximate solution), they have been found to be either less efficient with no simplification in the computations, or very inefficient with considerable computational simplification. It may be possible to improve on the pseudo-inverse steering law to increase the efficiency of angular momentum utilization even further, but a study to identify such a law is beyond the scope of this project. Without further justification, the pseudo-inverse steering law is selected for the CMG control computer described in this report.

The steering law computations thus provide gimbal rate commands, $\dot{\alpha}_c$, to the individual gimbal control systems in response to vehicle angular acceleration commands, $\dot{\omega}_c$, in accordance with the following equation:

$$\dot{\alpha}_c = A^{\dagger}\dot{H} = -A^{\dagger}I_V\dot{\omega}_c \quad . \tag{4.15}$$

4.2.3 The Rate Gyro Configuration

The vehicle rate information required for attitude control system stability and/or vehicle rate control is to be obtained from a set of six rate gyros configured to sense vehicle rates along the six dodecahedron axes shown in Figure 4-3. These axes correspond to the normals of the faces of a regular dodecahedron and have the unique property that the acute angles between any two axes are equal, given by $2\delta \approx 63.4$ degrees. This configuration is described by Gilmore[4] The significant feature of this sensor configuration is that the vehicle rates can be obtained from any three of the six dodecahedron rates, denoted by $r = \{r_1, \ldots, r_6\}$.

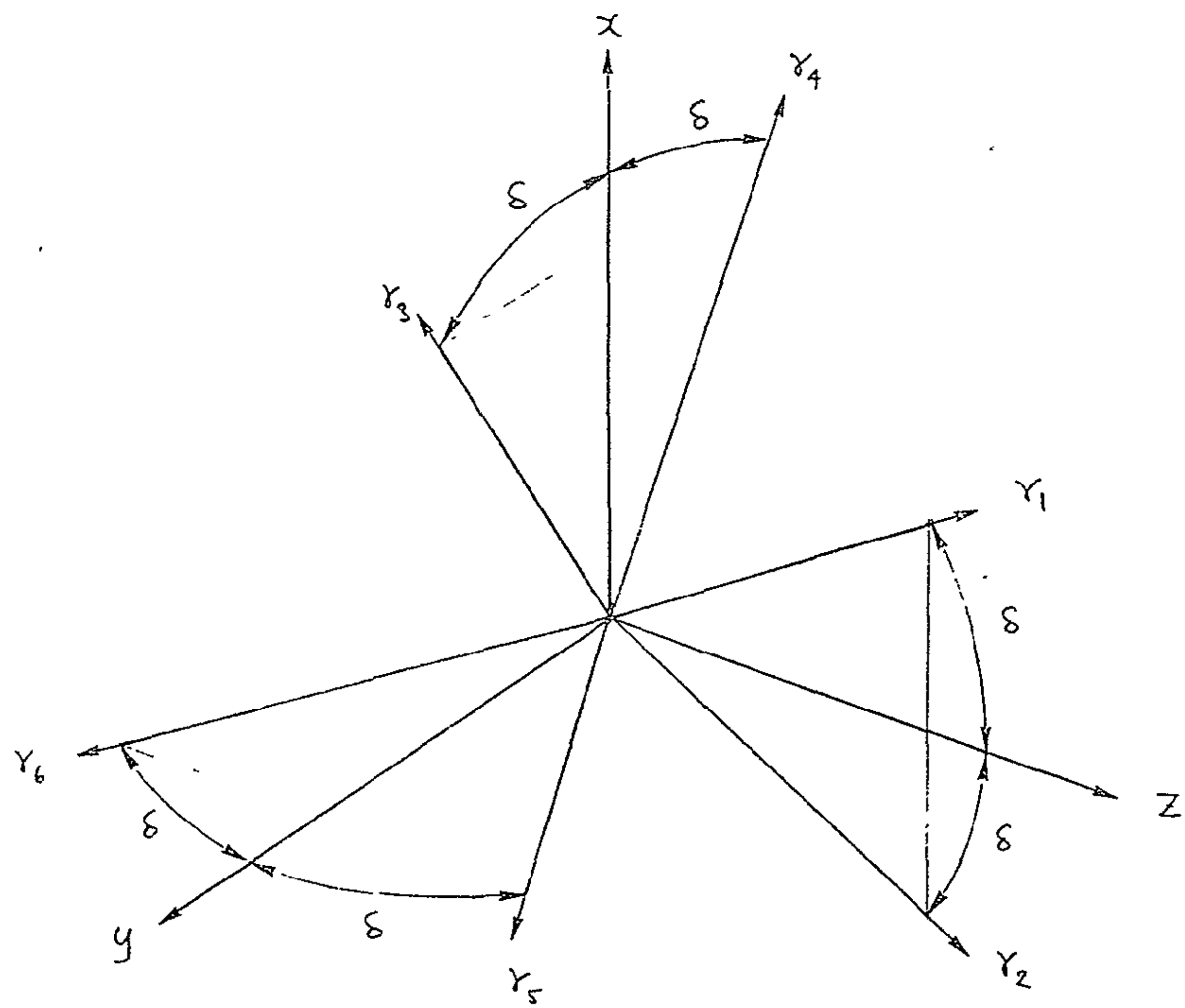


FIG. 4-3 Dodecahedron axis system.

The vehicle rates, ω, can be projected onto the dodecahedron axes to obtain the dodecahedron rates r by

$$r = \bar{E}\,\omega \tag{4.16}$$

where

$$\bar{E} = \begin{bmatrix} s & 0 & c \\ -s & 0 & c \\ c & s & 0 \\ c & -s & 0 \\ 0 & c & s \\ 0 & c & -s \end{bmatrix} \tag{4.17}$$

and

$$\begin{aligned} s &= \sin\delta = \left(\frac{5-\sqrt{5}}{10}\right)^{1/2} \approx 0.526 \\ c &= \cos\delta = \left(\frac{5+\sqrt{5}}{10}\right)^{1/2} \approx 0.850 \quad . \end{aligned} \tag{4.18}$$

If all six rate gyros are operating properly, the signals they generate, denoted by r_s, are approximately equal to r. If a failure is detected in rate gyro number i, however, this gyro is disengaged; i.e., $r_{si} = 0$. Let

$$\lambda = \mathrm{diag}\left\{\lambda_1,\ \ldots,\ \lambda_6\right\} \tag{4.19}$$

where $\lambda_i = 1$ if rate gyro number i is engaged, and $\lambda_i = 0$ if it is disengaged.

Then

$$r_s = \lambda r = E\omega \tag{4.20}$$

where $E = \lambda\bar{E}$. To determine ω from r_s requires the solution of equation (4.20) Although E is a constant matrix for every state of λ, there are 42 states of λ for which at least three of its elements are 1's; and there are 20 states for which exactly three of its elements are 1's. Therefore, at least 20 different solutions are required in order to find ω under all failure conditions (up to three failures).

Since any three rate gyros contain sufficient information to find ω, only the 20 solutions are required to meet all failure conditions. By using all the rate gyros available, however, the effects of inaccuracies in the signals can be minimized. The pseudo-inverse solution given by equations (4.21) and (4.22) provides the solution for ω that minimizes the norm of $\lambda r - r_s$ (the "least-squares-fit" solution)

$$\omega_s = E^\dagger r_s \tag{4.21}$$

where

$$\begin{aligned} E^\dagger &= \left(E^T E\right)^{-1} E^T \\ &= \left(\bar{E}^T \lambda \bar{E}\right)^{-1} \left(\bar{E}^T \lambda\right) \end{aligned} \tag{4.22}$$

and where ω_s is the set of sensed and transformed signals for ω. This solution is identical to a three-signals-only solution when all but the three respective elements of λ are set to zero.

The detection and isolation of failures in the rate gyro package is also a major task for the CMG control computer in order to generate λ. Catastrophic failures can usually be detected by various types of monitors, but errors in the signals are not as easy to detect. Gilmore[5] describes a set of parity equations that isolate two gyro failures and detect a third failure. These equations are presented in Table 4.1 under the notational convention adopted for this report.

TABLE 4.1

Parity Equations for Rate Gyro Failure Detection and Isolation

Equation No.	Equation	Signals Compared
1	$(r_1 - r_2)\ c - (r_3 + r_4)\ s = 0$	1234
2	$(r_2 + r_3)\ c - (r_1 + r_5)\ s = 0$	1235
3	$(r_3 - r_1)\ c + (r_2 - r_6)\ s = 0$	1236
4	$(r_4 - r_1)\ c + (r_2 - r_5)\ s = 0$	1245
5	$(r_2 + r_4)\ c - (r_1 - r_6)\ s = 0$	1246
6	$(r_5 - r_6)\ c - (r_1 + r_2)\ s = 0$	1256
7	$(r_4 + r_5)\ c - (r_1 - r_3)\ s = 0$	1345
8	$(r_6 - r_3)\ c + (r_1 + r_4)\ s = 0$	1346
9	$(r_1 + r_6)\ c - (r_3 + r_5)\ s = 0$	1356
10	$(r_5 - r_1)\ c + (r_4 - r_6)\ s = 0$	1456
11	$(r_5 - r_3)\ c + (r_4 - r_2)\ s = 0$	2345
12	$(r_6 + r_4)\ c + (r_2 - r_3)\ s = 0$	2346
13	$(r_2 - r_5)\ c + (r_3 + r_6)\ s = 0$	2356
14	$(r_2 + r_6)\ c + (r_4 - r_5)\ s = 0$	2456
15	$(r_4 - r_3)\ c + (r_5 - r_6)\ s = 0$	3456

4.2.4 Control Computer Description

The principal function of the CMG control computer is to generate gimbal rate command signals to the six CMG's of the configuration described in paragraph 4.2.1, such that the vehicle responds in accordance with input command signals for vehicle rate and acceleration, using the rate signals provided by the six dodecahedron configured rate gyros described in paragraph 4.2.3. An objective of the computer structure is to take advantage of the redundancy in both the CMG and rate gyro configurations to provide fail-operational performance, including failures in the computer.

The control law for the vehicle rate control system, illustrated in Figure 4-4 can be written in the form

$$\dot{\omega}_c = \dot{\omega}_{cc} + G^3 \left(\omega_c - \omega_s\right) \tag{4.23}$$

where

$\dot{\omega}_{cc}$ = vehicle acceleration commanded directly as an input to the rate control system

$\omega_c - \omega_s$ = vehicle rate error

G^3 = 3 x 3 diagonal matrix of compensation functions

The input signals $\dot{\omega}_{cc}$ and ω_c are provided either by a control law for the vehicle attitude control system or by manual commands. The vehicle rate signal ω_s is obtained by converting the six dodecahedron axes rate signals r_s to vehicle axes signals.

Since both the sensor signals r_s and the gimbal rate command signals $\dot{\alpha}_c$ are 6-tuples, the question has been raised as to whether there are any advantages in performing the computations in six variables without first reducing the rate signals to the three-variable format. It has been suggested that by performing computations of data in a redundant format, perhaps some reliability advantage can be obtained with incremental computers, for which simultaneous computations are performed by separate components. The thought is that if one or more of such components should fail without affecting the remainder of the computations, the redundancy of the data and the computations would permit fail-operational performance. Investigations into this approach indicate that fail-operational performance is obtainable for certain component failures, but the complexity of such a system is considerably greater than for a system with redundancy at the overall computer level. In addition, there are many types of computer failures that do not permit fail-operational performance. Two complete computers with failure monitoring or three such computers with majority voting provide greater reliability and appear to be less complex for reasons discussed subsequently.

The vehicle accelerations in the dodecahedron axes for which rate gyro signals are available, $\lambda\dot{r}$, can be related to gimbal rates by combining equations (4.9), (4.12) and (4.20) to obtain

$$\lambda\dot{r} = -R\dot{\alpha} \tag{4.24}$$

where

$$R = EI_V^{-1}A \tag{4.25}$$

and where the disturbance terms of equation (4.9) are neglected for simplicity. The steering law in this case is required to solve equation (4.24) for α_c in terms of vehicle acceleration command signals $\dot{r}_c$ in the six-variate format. This command signal is given by

$$\dot{r}_c = E\dot{\omega}_c \tag{4.26}$$

and for the vehicle rate control law of equation (4.22)

$$\dot{r}_c = E\dot{\omega}_{cc} + E\ G^3 \left(\omega_c - \omega_s\right) \quad . \tag{4.27}$$

By requiring that the three diagonal elements of G^3 be identical, thereby requiring that the control system characteristic be identical in the three vehicle axes, the product $E\ G^3$ can be written as $G^6\ E$, where G^6 is a diagonal 6 x 6 matrix having diagonal elements identical to G^3. Equation (4.27) can then be written as

$$\begin{aligned} \dot{r}_c &= E\dot{\omega}_{cc} + G^6 \left(E\omega_c - r_s\right) \\ &= \dot{r}_{cc} + G^6 \left(r_c - r_s\right) \quad . \end{aligned} \tag{4.28}$$

Figure 4-5 shows the structure for this control law.

The steering law in this case is required to invert the 6 x 6 matrix R and the pseudo-inverse is selected for the reasons given in paragraph 4.2.2. The simplest method for computing the pseudo-inverse of a 6 x 6 matrix of rank 3 is to first factor it into the product of a 6 x 3 matrix M times a 3 x 6 matrix N, which is always possible[6]. The pseudo-inverse S of R = MN can then be computed by

$$S = R^{\dagger} = N^T \left(NN^T\right)^{-1} \left(M^TM\right)^{-1} M^T \quad . \tag{4.29}$$

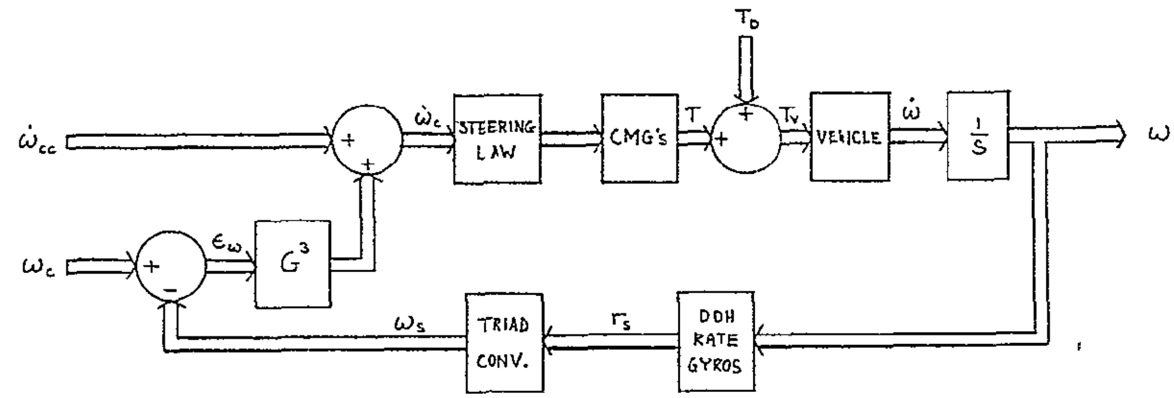


FIG. 4-4 Three-variate control law.

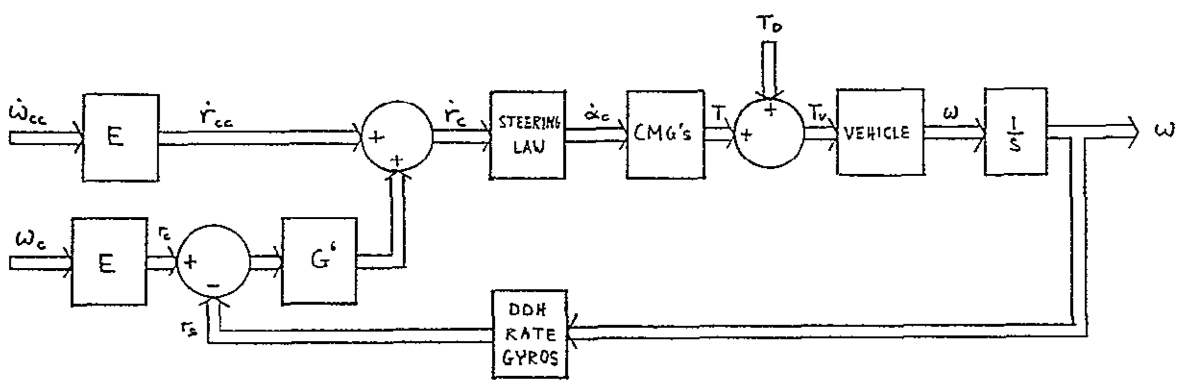


FIG. 4-5 Six-variate control law.

Some authors define the pseudo-inverse by equation (4.29).[7] Therefore

$$S = N^{\dagger} M^{\dagger} \tag{4.30}$$

S can be factored as follows to obtain

$$\begin{aligned} S &= \left(I_V^{-1} A\right)^{\dagger} E^{\dagger} \\ &= A^{\dagger} I_V E^{\dagger} \quad . \end{aligned} \tag{4.31}$$

Although there are infinitely many ways of factoring R to obtain the unique $S = R^{\dagger}$, equation (4.30) indicates that the pseudo-inverse of this 6 x 6 matrix is equivalent to a transformation of the six-variate acceleration commands $\dot{r}_c$ to some three-axis format, followed by a transformation to the six-variable gimbal rate commands $\dot{\alpha}_c$. The question is whether implementation of the 6 x 6 matrix S can provide any advantages in terms of simplicity or reliability over cascading the 6 x 3 and 3 x 6 matrix factors of S.

Direct computation of the elements of S can provide limited fail-operational performance if such failures can be detected and isolated. By expanding the steering law computations

$$\dot{\alpha}_c = -S\, \dot{r}_c \tag{4.32}$$

to obtain

$$\begin{aligned} \dot{\alpha}_{c1} &= -S_{11}\, \dot{r}_{c1} - S_{12}\, \dot{r}_{c2} - \cdots - S_{16}\, \dot{r}_{c6} \\ \dot{\alpha}_{c2} &= -S_{21}\, \dot{r}_{c1} - S_{22}\, \dot{r}_{c2} - \cdots - S_{26}\, \dot{r}_{c6} \\ &\cdots\cdot \\ \dot{\alpha}_{c6} &= -S_{61}\, \dot{r}_{c1} - S_{62}\, \dot{r}_{c2} - \cdots - S_{66}\, \ddot{r}_{c6} \end{aligned} \tag{4.33}$$

it can be observed that the effects of a failure in the computation of the element, S_{ij}, can be nullified by letting $\lambda_j = 0$, thus causing $\dot{r}_{cj}$ to be zero. The steering law is a function of λ, and is therefore automatically modified to compensate for this step. Since the elements of column j in S are nullified by this procedure, isolated computation of the elements of this column is required. However, all 18 elements of $A^{\dagger}$ must be computed for each of the six columns of S, necessitating six isolated computations of these elements, each element being a relatively complex function of the six gimbal angles. An increase in these computations, which constitute the bulk of the steering law computations, by a factor of six represents a very significant increase in the complexity of the entire computer. Yet it provides fail-operational performance for only a limited set of failure types, namely those failures which can be isolated to the computation of the elements of a column. Further use of the rate gyro corresponding to the failed column is unfortunately also nullified by this procedure, thereby reducing the redundancy of the rate gyro package.

To perform steering law computations on the six-variable signals also requires conversion of the three-variate input command signals, $\dot{\omega}_{cc}$ and ω_c, to $\dot{r}_{cc}$ and r_c. In addition, six rate-loop compensation filters are required instead of the three required with the three-variable system.

Based on these observations, it appears that parallel redundancy of the entire computer is preferred both from standpoints of complexity and reliability. The remainder of this report, therefore, considers only the three-variate structure for the control law.

Figure 4-6 shows the basic structure for the CMG control system in terms of its subcomputers, the CMG configuration, the vehicle, and the rate gyro package. Redundancy in the computation is not shown; this aspect is discussed in the following section.

4.2.5 Fail-Operational Computation

Fail-operational performance for a computation function can be obtained with two or more complete channels of computation plus monitors that are able to detect and identify a channel failure. In some cases, the only reliable technique for identifying a failure is to "vote" between at least three channels for a single failure, at least four channels for two failures, and so on. If it is possible to reliably detect a failure in a single channel by a simple method, one less channel of computation is required in comparison with the voting method. It is therefore worthwhile to investigate techniques for failure monitoring.

The steering law computation and the dodecahedron inversion both consist of the inversion of matrices that are very much simpler to compute in the forward direction than to invert. By performing the forward transformation M on the output of an inversion transformation $M^{\dagger}$ and comparing the resulting output with the original input, an indication of a failure in either of these transformations is obtained. This principle is illustrated in Figure 4-7.

The scope of this study does not permit comparisons of various methods for detecting and isolating failures, and the approach suggested here requires further study.

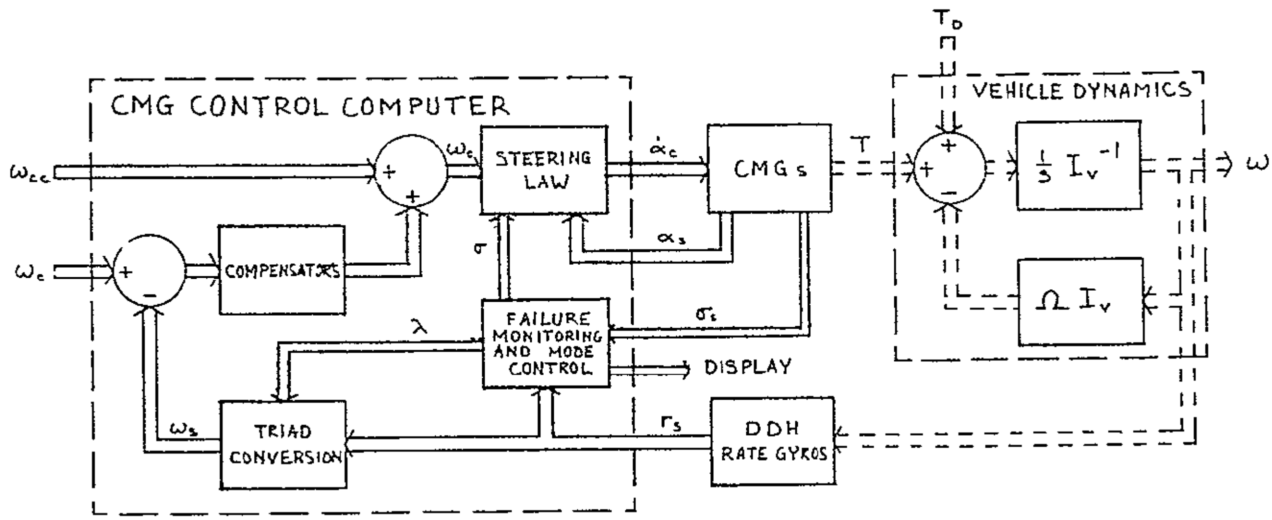


FIG. 4-6 CMG control system block diagram.

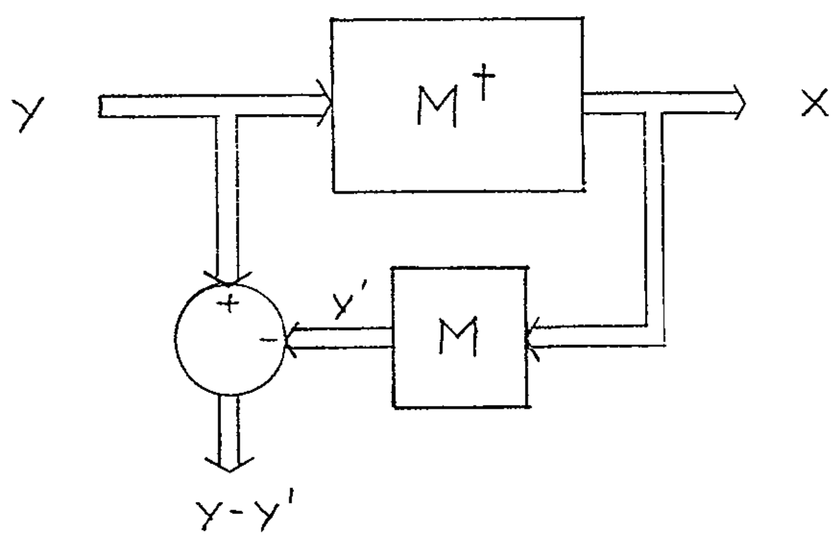


FIG. 4-7 Method of failure detection.

4.3 STEERING LAW COMPUTATION

Since the steering law computation is by far the most complex function to be performed by the CMG control computer, most of the effort under this program has been devoted to this function. Paragraph 4.3.1 presents the equations to be computed by the steering law computer, paragraph 4.3.2 describes the principles of incremental computation, and the remaining paragraphs describe how the equations can be implemented with incremental computation.

4.3.1 Equations and Computer Structure

The steering law computer is required to generate gimbal rate command signals $\dot{\alpha}_c$ to the individual gimbal control systems of the CMG's in response to vehicle angular acceleration signals $\dot{\omega}_c$ in accordance with equations given in paragraph 4.2.2 and repeated below for convenience

$$\dot{\alpha}_c = -A^{\dagger} I_v \dot{\omega}_c$$

$$\dot{\alpha}_c = \begin{bmatrix} \dot{\alpha}_{c1} \\ \dot{\alpha}_{c2} \\ \dot{\alpha}_{c3} \\ \dot{\alpha}_{c4} \\ \dot{\alpha}_{c5} \\ \dot{\alpha}_{c6} \end{bmatrix} \qquad \dot{\omega}_c = \begin{bmatrix} \dot{\omega}_{cx} \\ \dot{\omega}_{cy} \\ \dot{\omega}_{cz} \end{bmatrix}$$

$$I_v = \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix}$$

I_x, I_y, and I_z are the inertias of the vehicle about its principal axes, and $\left[A^{\dagger} = A^T (AA^T)^{-1}\right]$ is the pseudo-inverse of A.

To simplify notations in the subsequent development, the following definitions are made.

$$B = AA^T \tag{4.34}$$

$$C = \text{adj } B \tag{4.35}$$

$$d_o = \det B \tag{4.36}$$

$$D = A^T C \tag{4.37}$$

$$u = \left(\frac{1}{d_o}\right)\left(I_v \dot{\omega}_c\right) \quad . \tag{4.38}$$

Then,

$$A^\dagger = D \left(\frac{1}{d_o}\right) \tag{4.39}$$

and

$$\dot{\alpha}_c = -Du \quad . \tag{4.40}$$

By leaving $A^\dagger$ in the factored form, as in equation (4.39) only three divisions by the scalar d_o are required compared to 18 such divisions if the elements of $A^\dagger$ were to be computed. Since B is only a 3x3 matrix, the adjoint method for inverting B is used. The steering law computer can now be structured of sub-computers related to the above defined variables so that they can be discussed separately. Figure 4-8 illustrates this structure of the steering law computer.

The A computer computes the elements of A given by

$$\begin{aligned} a_{ij} &= h_j \, S_\beta \, C_{\alpha_j} \\ a_{2j} &= h_j \left(S_{\gamma_j} S_{\alpha_j} - C_\beta \, C_{\gamma_j} C_{\alpha_j}\right) \\ a_{3j} &= -h_j \left(C_{\gamma_j} \, S_{\alpha_j} + C_\beta \, S_{\gamma_j} C_{\alpha_j}\right) . \end{aligned} \tag{4.41}$$

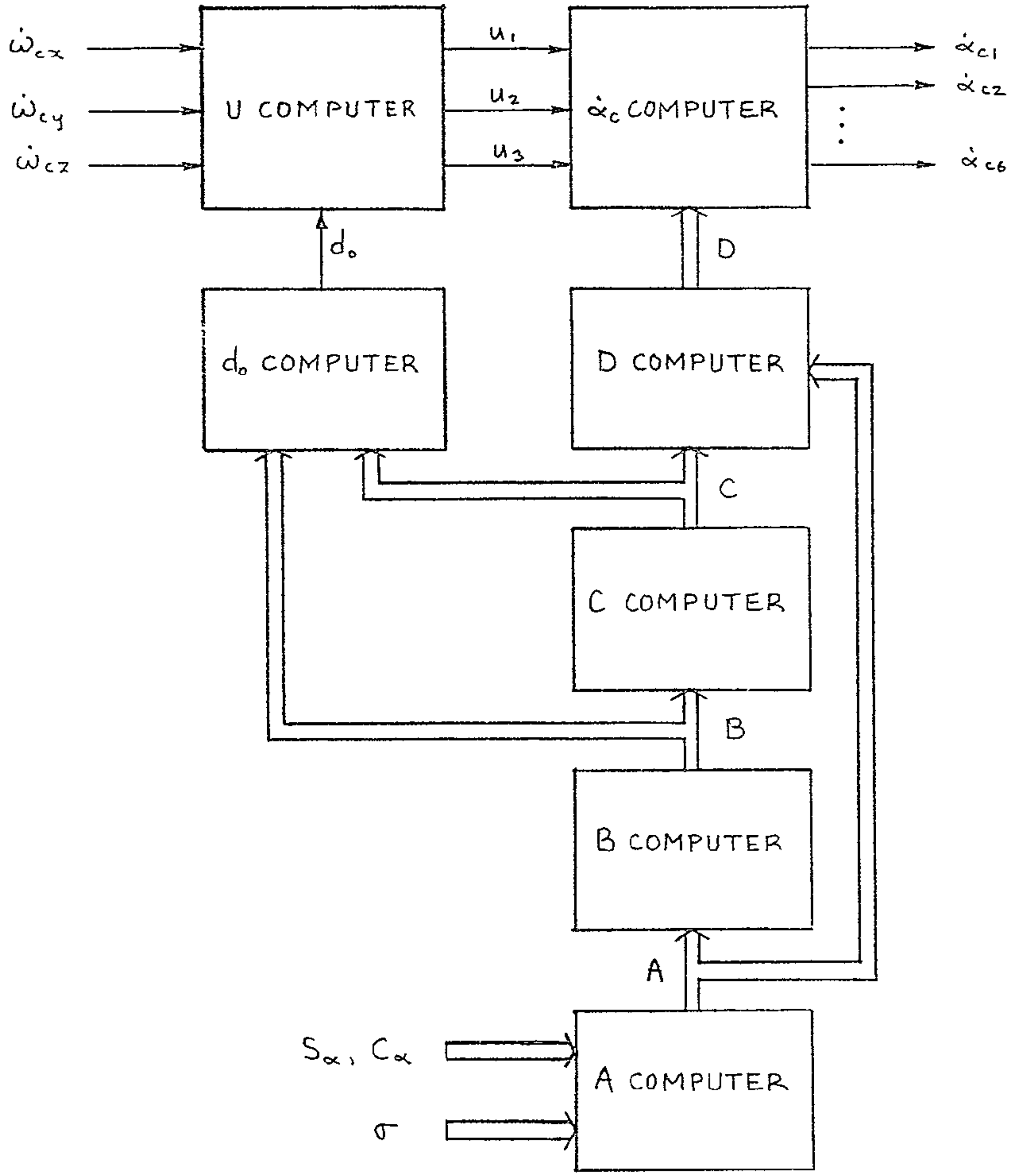


FIG. 4-8 Organization of the pseudo-inverse steering law computer.

2

Element a_{ij} may be factored into

$$a_{ij} = \rho_{ij}\sigma_j \tag{4.42}$$

where

$$\rho_{ij} = \left(m_{ij}\, S_{\alpha_j} + n_{ij}\, C_{\alpha_j}\right) \tag{4.43}$$

and where σ_j is the speed of rotor j in revolutions per second. The constants m_{ij} and n_{ij} are given by

$$\begin{aligned}
m_{1j} &= 0 & n_{1j} &= 2\pi I_R S_\beta \\
m_{2j} &= 2\pi I_R S_{\gamma_j} & n_{2j} &= -2\pi I_R C_\beta C_{\gamma_j} \\
m_{3j} &= -2\pi I_R C_{\gamma_j} & n_{3j} &= -2\pi I_R C_\beta S_{\gamma_j}
\end{aligned} \tag{4.44}$$

where I_R is the inertia of the CMG rotor about its spin axis. For the 6-GAMS configuration described in paragraph 4.2.1, the γ angles are 0, 60, 120, 180, 240, and 300 degrees so $\{S_{\gamma_j}\}$ and $\{C_{\gamma_j}\}$ are specified, but the gimbal axis tilt angle β depends on the requirements for the net angular momentum envelope.

Thus, the A computer for each of the 18 elements $\{a_{ij}\}$ performs the scaling and addition of the input signals, S_{α_j} and C_{α_j}, as indicated by equation (4.3-10) and multiplies the result by the input signal for rotor speed, σ_j.

To help visualize the complexity associated with each of the subcomputers of the steering law computer, the related matrix expressions are expanded in the following discussion. Since $B = AA^\dagger$ is a symmetric 3x3 matrix, only six elements such as those making up the upper triangle of B, are required. These elements are

$$\begin{aligned}
b_{11} &= a_{11}^2 + a_{12}^2 + a_{13}^2 + a_{14}^2 + a_{15}^2 + a_{16}^2 \\
b_{12} &= a_{11}a_{21} + a_{12}a_{22} + a_{13}a_{23} + a_{14}a_{24} + a_{15}a_{25} + a_{16}a_{26} \\
b_{13} &= a_{11}a_{31} + a_{12}a_{32} + a_{13}a_{33} + a_{14}a_{34} + a_{15}a_{35} + a_{16}a_{36} \\
b_{22} &= a_{21}^2 + a_{22}^2 + a_{23}^2 + a_{24}^2 + a_{25}^2 + a_{26}^2 \\
b_{23} &= a_{21}a_{31} + a_{22}a_{32} + a_{23}a_{33} + a_{24}a_{34} + a_{25}a_{35} + a_{26}a_{36} \\
b_{33} &= a_{31}^2 + a_{32}^2 + a_{33}^2 + a_{34}^2 + a_{35}^2 + a_{36}^2 .
\end{aligned} \tag{4.45}$$

The C computer evaluates the upper triangle elements of $C = \operatorname{adj} B$ since this matrix is also symetric:

$$
\begin{aligned}
c_{11} &= b_{22}b_{33} - b_{23}^{2} \\
c_{12} &= b_{13}b_{23} - b_{12}b_{33} \\
c_{13} &= b_{12}b_{23} - b_{13}b_{22} \\
c_{22} &= b_{11}b_{33} - b_{13}^{2} \\
c_{23} &= b_{12}b_{13} - b_{11}b_{23} \\
c_{33} &= b_{11}b_{22} - b_{12}^{2} \;.
\end{aligned}
\tag{4.46}
$$

Elements of B and C are used to compute d_o :

$$
d_o = b_{11}c_{11} + b_{12}c_{12} + b_{13}c_{13} \tag{4.47}
$$

The 18 elements of $D = A^T C$ are computed as follows:

$$
\begin{aligned}
d_{11} &= a_{11}c_{11} + a_{21}c_{12} + a_{31}c_{13} \\
d_{12} &= a_{11}c_{12} + a_{21}c_{22} + a_{31}c_{23} \\
d_{13} &= a_{11}c_{13} + a_{21}c_{23} + a_{31}c_{33} \\
d_{21} &= a_{12}c_{11} + a_{22}c_{12} + a_{32}c_{13} \\
d_{22} &= a_{12}c_{12} + a_{22}c_{22} + a_{32}c_{23} \\
d_{23} &= a_{12}c_{13} + a_{22}c_{23} + a_{32}c_{33} \\
d_{31} &= a_{13}c_{11} + a_{23}c_{12} + a_{33}c_{13} \\
d_{32} &= a_{13}c_{12} + a_{23}c_{22} + a_{33}c_{23} \\
d_{33} &= a_{13}c_{13} + a_{23}c_{23} + a_{33}c_{33} \\
d_{41} &= a_{14}c_{11} + a_{24}c_{12} + a_{34}c_{13} \\
d_{42} &= a_{14}c_{12} + a_{24}c_{22} + a_{34}c_{23} \\
d_{43} &= a_{14}c_{13} + a_{24}c_{23} + a_{34}c_{33}
\end{aligned}
\tag{4.48}
$$

$$d_{51} = a_{15}c_{11} + a_{25}c_{12} + a_{35}c_{13}$$

$$d_{52} = a_{15}c_{12} + a_{25}c_{22} + a_{35}c_{23}$$

$$d_{53} = a_{15}c_{13} + a_{25}c_{23} + a_{35}c_{33} \tag{4.48}$$

$$d_{61} = a_{16}c_{11} + a_{26}c_{12} + a_{36}c_{13}$$

$$d_{62} = a_{16}c_{12} + a_{26}c_{22} + a_{36}c_{23}$$

$$d_{63} = a_{16}c_{13} + a_{26}c_{23} + a_{36}c_{33} \quad .$$

The only divisions required in the steering law computer are in the u computer. The elements of this 3 x 1 matrix are

$$u_1 = \frac{I_x \dot{\omega}_{cx}}{d_o}$$

$$u_2 = \frac{I_y \dot{\omega}_{cy}}{d_o} \tag{4.49}$$

$$u_3 = \frac{I_z \dot{\omega}_{cz}}{d_o} \quad .$$

The $\dot{\alpha}_c$ computer then provides the outputs of the steering law computer by performing the following computations:

$$\dot{\alpha}_{c1} = d_{11}u_1 + d_{12}u_2 + d_{13}u_3$$

$$\dot{\alpha}_{c2} = d_{21}u_1 + d_{22}u_2 + d_{23}u_3$$

$$\dot{\alpha}_{c3} = d_{31}u_1 + d_{32}u_2 + d_{33}u_3$$

$$\dot{\alpha}_{c4} = d_{41}u_1 + d_{42}u_2 + d_{43}u_3 \tag{4.50}$$

$$\dot{\alpha}_{c5} = d_{51}u_1 + d_{52}u_2 + d_{53}u_3$$

$$\dot{\alpha}_{c6} = d_{61}u_1 + d_{62}u_2 + d_{63}u_3 \quad .$$

A summary of the types and numbers of mathematical operations required for the subcomputations of the steering law computer is presented in Table 4.2.

TABLE 4.2

Mathematic Operations for the Steering Law Computer

Subcomputer	Additions	Multiplications	Divisions
A	18	18	0
B	30	36	0
C	6	12	0
d_o	2	3	0
D	36	54	0
u	0	0	3
$\dot{\alpha}_c$	12	18	0
Total	104	141	3

4.3.2 Incremental Computation

An incremental computer is a special purpose digital computer that has a number of features which make it attractive for solving equations of the type required for the pseudo-inverse steering law. In contrast with the general purpose type digital computer, which performs entire computations during each computation cycle, an incremental computer updates previous computations. Such computations are generally much simpler and are performed simultaneously on the numerous variables of the problem. The time required to complete each computation cycle is therefore much less than for a general purpose computer, permitting higher sampling rates.

In an incremental computer, the variables of the computations are stored in binary registers, referred to as Y registers. The content (numerical value) stored in a Y register is increased by one when so commanded by a pulse, ΔY, representing an incremental increase in the variable Y. The ΔY pulse can also be negative, in which case the Y register is decreased by one. In addition, the content of a Y register is added to the content of an R register when so commanded by a positive ΔX pulse, or it is subtracted from the R register for a negative ΔX pulse.

Figure 4.3-2 shows the schematic symbols adopted for the incremental computer elements. Since the R register has the same capacity as the Y register, repeated ΔX pulses of the same sign will cause the R register to overflow. Each time the R register overflows in the positive direction, a positive ΔZ pulse is generated; if the overflow is in the negative direction, a negative ΔZ pulse is generated. This ΔZ pulse can then serve as a ΔX pulse or a ΔY pulse for other Y and R registers. Let

$$Z(n) = \sum_{i=1}^{n} \Delta Z(i) \tag{4.51}$$

where $\Delta Z(i)$ is the ΔZ pulse produced by the i'th ΔX pulse, $\Delta X(i)$ [$\Delta Z(i)$ may be 0, +1, or -1]. Let Y(i) and R(i) be the contents of the Y and R registers, respectively, at the occurrence of $\Delta X(i)$, and let c be the capacity of the registers. Then

$$cZ(n) + R(n) = R(1) + \sum_{i=1}^{n} Y(i)\, \Delta X(i) \quad . \tag{4.52}$$

Therefore, for $\left| Y(i) \right| >> 1$,

$$Z(n) \approx \frac{1}{c} \sum_{i=1}^{n} Y(i)\, \Delta X(i) \, . \tag{4.53}$$

For the case where Y(i) is a function of X(i),

$$Z(n) \approx \frac{1}{c} \int_0^{X(n)} Y[X(i)]\ dX(i) \quad . \tag{4.54}$$

This computation structure may thus be used to integrate functions of independent variables, and combinations of such elements may be used to generate functions that are solutions of differential equations. Computers structured by interconnecting integrators of this type are commonly referred to as DDA's (digital differential analyzers).

The computations for the steering law require only additions, multiplications, and divisions. The sum of several variables can be obtained by feeding their respective Δ pulses to a common Y register and properly controlling their pulse times so that they do not occur simultaneously. Such timing control can be implemented by various methods, and will be discussed later.

Multiplication is readily obtained by noting that

$$d\left(Y_1 Y_2\right) = Y_1 dY_2 + Y_2 dY_1 \tag{4.55}$$

Therefore, let $\Delta X_1 = \Delta Y_2$ and $\Delta X_2 = \Delta Y_1$, and let the two Y registers add into a common R register as shown in Figure 4-10. Then

$$\Delta Z \approx \frac{1}{c}\left(Y_1 \Delta Y_2 + Y_2 \Delta Y_1\right) \approx \frac{1}{c}\,\Delta\left(Y_1 Y_2\right) \quad . \tag{4.56}$$

The accuracy of this computation improves with the size c of the registers. It can also be improved by proper sequencing of the operations when ΔY_1 and ΔY_2 occur simultaneously. Investigations of such methods are beyond the scope of this study.

4.3.3 ΔA Computation

The ΔA computer generates increments $\{\Delta a_{ij}\}$ to increase or decrease the values in the set of 18 registers that store the elements of the A matrix, given by equations (4.42), (4.43) and (4.44). Column j of A corresponds to CMG number j. The incremental computer structure for the three elements of this column is shown in Figure 4-11.

The Y register corresponding to some variable, y, is identified by this variable, and it stores the numerical value $K_y y$ where K_y is a scale factor. The R register that generates Δy is identified by $\tilde{y}$. Constant numbers are represented by the oval-shaped areas in Figure 4-11. For example, when a positive pulse for $\Delta\sigma_j$ occurs, the Y register containing $K_{\sigma_j}\sigma_j$ is increased by the constant number K_{σ_j}.

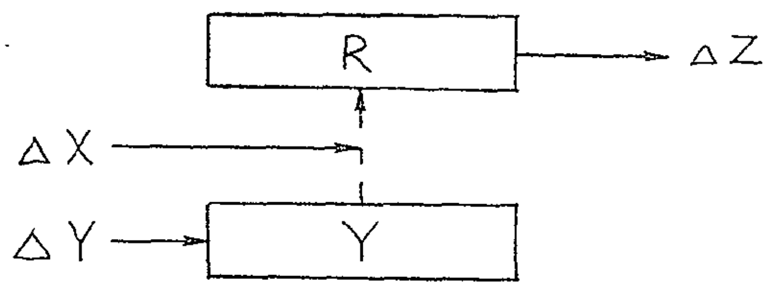


FIG. 4-9 Basic incremental computer elements.

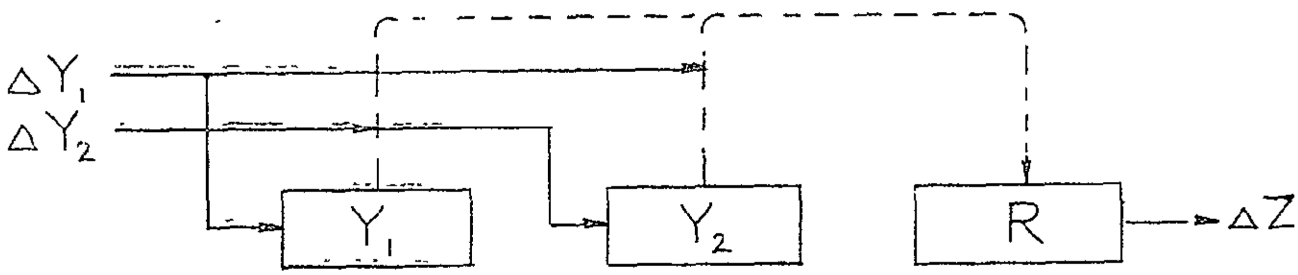


FIG. 4-10 Incremental multiplier.

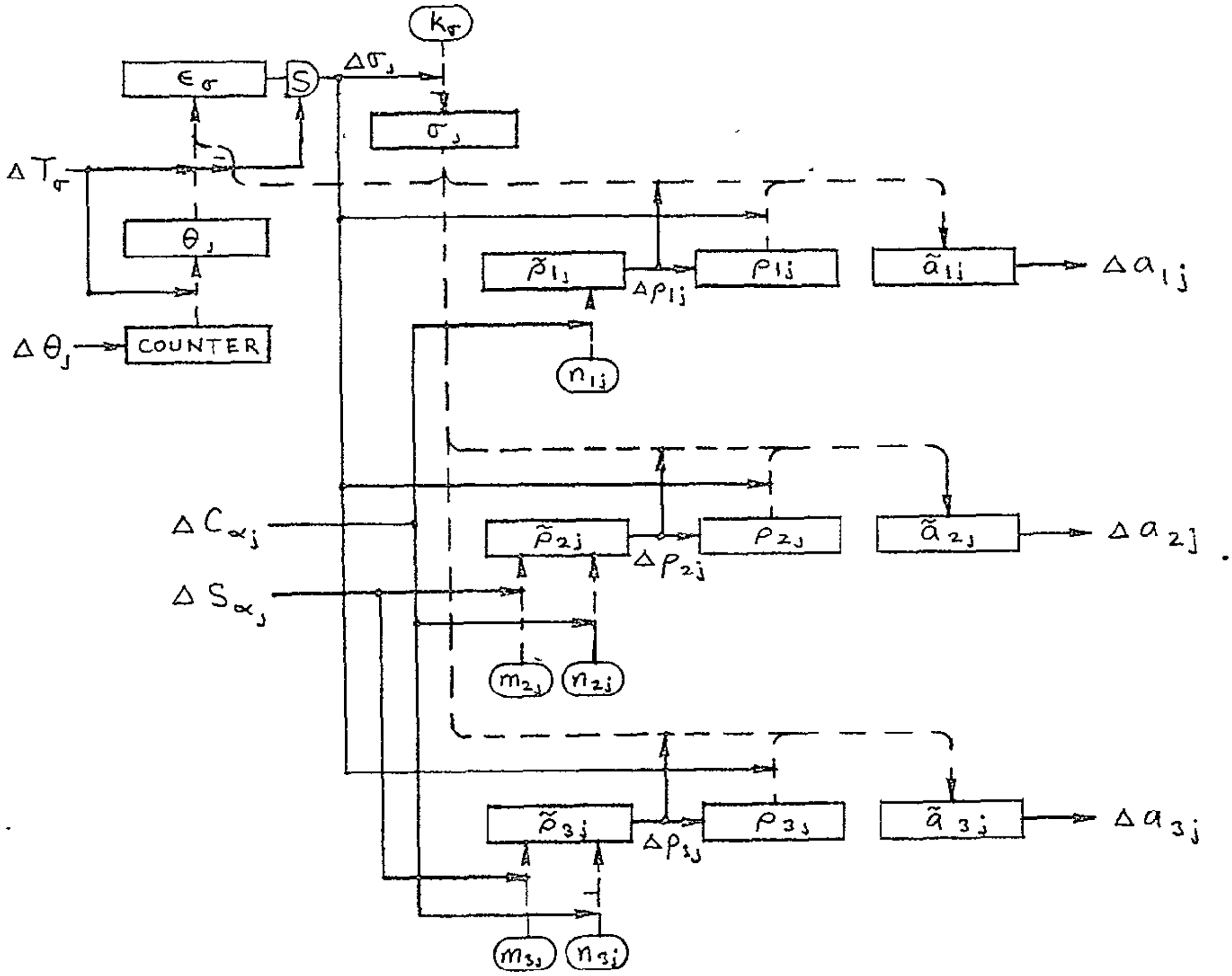


FIG. 4-11 ΔA computer.

The rotor speed σ_j (revolutions per second) can be held constant by the spin motor electronics to some degree of accuracy, but high accuracy may be difficult to achieve. If the rotor speed is sensed and included in the steering law computations, rotor speed control is not required at all, permitting simple spin motor electronics. The rotor speed is therefore included in the steering law computations.

The rotor speed is computed by counting revolutions θ_j over a period of time T_σ. A pulse $\Delta\theta_j$ produced for each rotor revolution increases the counter by one. A pulse ΔT_σ. is produced each T_σ seconds by some clock. T_σ must be quite long, such as several seconds, to permit sufficient resolution in σ_j. When the ΔT_σ pulse occurs, the θ_j register is first set to $K_{\theta_j}\theta_j$, where θ_j is the present number in the counter and where $K_{\theta_j} = K_{\sigma_j}/T_\sigma$ (chosen to be a power of 2 for simplicity). The counter is then reset to zero. Next, $K_{\theta_j}\theta_j$ and $K_{\sigma_j}\sigma_j$ are compared by adding the first to, and subtracting the second from, the ε_{σ_j} register. If this difference is positive, a positive $\Delta\sigma_j$ pulse is generated and causes the σ_j register to increase by K_{σ_j}, or vice versa for a negative pulse.

To obtain maximum accuracy in incremental computations, the Y registers should be scaled so low that they are as full as possible without overflowing. Therefore, K_{σ_j} is selected so that K_{σ_j} times maximum σ_j is close to but does not exceed the register capacity c.

T_σ $\left(\text{or } K_{\theta_j} = K_{\sigma_j}/T_\sigma\right)$ should be selected to minimize the error in σ_j. If T_σ is very large, the maximum error in σ_j due to resolution, given by $\sigma_j/\theta_j = 1/T_\sigma$, is small. However, the maximum error due to a change in σ_j during the T_σ interval, $\dot{\sigma}_m T_\sigma$ (where $\dot{\sigma}_m = \max \dot{\sigma}$, assumed constant in this interval), is large. The T_σ for which these errors are equal is given by $T_\sigma = 1/\sqrt{\dot{\sigma}_m}$.

Incremental signals for sin α_j and cos α_j can be obtained by numerous methods, but at the present state of the art a set of six resolvers with a single time-shared A/D converter for all six gimbal angles appears to be the best candidate. Selection of a preferred method is not performed under this study. The scale factor for the registers in the A/D converter that store S_{α_j} and C_{α_j} is K_{sc}; i.e., a ΔS_{α_j} pulse corresponds to a change of $1/K_{sc}$ in S_{α_j}, etc.

When a positive ΔS_{α_j} pulse or ΔC_{α_j} pulse occurs, the $\tilde{\rho}_{ij}$ registers are increased by the respective constants, m_{ij} and n_{ij}. If this produces a positive overflow of a ρ_{ij} register, the resulting $\Delta\rho_{ij}$ pulse causes ρ_{ij} to increase by one, and σ_j to be added to $\tilde{a}_{ij}$. Overflows in the $\tilde{a}_{ij}$ registers then produce the output pulses of the ΔA computer, which are held in flip-flops until called for in the next computation cycle. Additions to the $\tilde{a}_{ij}$ registers are also produced by a $\Delta\sigma_j$ pulse.

Although there are many ways to perform the functional operations symbolized in Figure 4-11, considerable savings in hardware may be realized by serializing many of the variables on a single, long, circulating register. Table 4.3 describes one method for sequencing the additions into the $\tilde{a}_{ij}$ registers. All 18 variables $\{\rho_{ij}\}$ are serialized on a single, long register in the sequence shown in the table, and the size variables $\{\sigma_j\}$ are serialized on a single register such that exactly three cycles of the σ register coincide with one cycle of the ρ register, and such that the least significant bits of σ_1, ρ_{11}, and $\tilde{a}_{ij}$ coincide, etc, as shown. The 18 $\tilde{a}$ registers are the same length as each word of ρ and σ. During word interval 1, the ρ register adds into the $\tilde{a}_{11}$ register if $\Delta\sigma_1 = 1$ (or subtracts if $\Delta\sigma_1 = -1$), the σ register adds into $\tilde{a}_{16}$ if $\Delta\rho_{16} = 1$ (or subtracts if $\Delta\rho_{16} = -1$), etc, as shown in the table. It is also possible to serialize the a_{ij} variable with increased logic complexity and longer cycle time.

TABLE 4.3

$\tilde{A}$ Computation Sequence

Interval:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
ρ:	ρ_{11}	ρ_{21}	ρ_{31}	ρ_{12}	ρ_{22}	ρ_{32}	ρ_{13}	ρ_{23}	ρ_{33}	ρ_{14}	ρ_{24}	ρ_{34}	ρ_{15}	ρ_{25}	ρ_{25}	ρ_{16}	ρ_{26}	ρ_{36}
σ:	σ_1	σ_2	σ_3	σ_4	σ_5	σ_6	σ_1	σ_2	σ_3	σ_4	σ_5	σ_6	σ_1	σ_2	σ_3	σ_4	σ_5	σ_6
$\tilde{a}_{11}$	$\rho\Delta\sigma_1$						$\sigma\Delta\rho_{11}$											
$\tilde{a}_{21}$		$\rho\Delta\sigma_1$					$\sigma\Delta\rho_{21}$											
$\tilde{a}_{31}$			$\rho\Delta\sigma_1$				$\sigma\Delta\rho_{31}$											
$\tilde{a}_{12}$				$\rho\Delta\sigma_2$				$\sigma\Delta\rho_{12}$										
$\tilde{a}_{22}$					$\rho\Delta\sigma_2$			$\sigma\Delta\rho_{22}$										
$\tilde{a}_{32}$						$\rho\Delta\sigma_2$		$\sigma\Delta\rho_{32}$										
$\tilde{a}_{13}$							$\rho\Delta\sigma_3$		$\sigma\Delta\rho_{13}$									
$\tilde{a}_{23}$								$\rho\Delta\sigma_3$	$\sigma\Delta\rho_{23}$						$\sigma\Delta\rho_{23}$			
$\tilde{a}_{33}$									$\rho\Delta\sigma_3$						$\sigma\Delta\rho_{33}$			
$\tilde{a}_{14}$										$\rho\Delta\sigma_4$						$\sigma\Delta\rho_{14}$		
$\tilde{a}_{24}$											$\rho\Delta\sigma_4$					$\sigma\Delta\rho_{24}$		
$\tilde{a}_{34}$												$\rho\Delta\sigma_4$				$\sigma\Delta\rho_{34}$		
$\tilde{a}_{15}$													$\rho\Delta\sigma_5$				$\sigma\Delta\rho_{15}$	
$\tilde{a}_{25}$														$\rho\Delta\sigma_5$			$\sigma\Delta\rho_{25}$	
$\tilde{a}_{35}$															$\rho\Delta\sigma_5$		$\sigma\Delta\rho_{35}$	
$\tilde{a}_{16}$	$\sigma\Delta\rho_{16}$															$\rho\Delta\sigma_6$		
$\tilde{a}_{26}$	$\sigma\Delta\rho_{26}$																$\rho\Delta\sigma_6$	
$\tilde{a}_{36}$	$\sigma\Delta\rho_{36}$																	$\rho\Delta\sigma_6$

The Δa_{ij} pulses accumulate in the a_{ij} register (which is part of the ΔB computer). The number stored in this register is given by [per equation (4.56)]

$$K_{a_{ij}} a_{ij} = \frac{1}{c} \left(K_\rho \rho_{ij}\right) \left(K_\sigma \sigma_j\right) \quad . \tag{4.57}$$

Therefore, since $a_{ij} = \rho_{ij}\sigma_j$ [equation (4.42)], the scale factors are required to satisfy

$$K_{a_{ij}} = \frac{K_{\rho_{ij}} K_{\sigma_j}}{c} \tag{4.58}$$

In addition, the Y registers cannot be allowed to overflow. Therefore,

$$K_{\rho_{ij}} \max \left|\rho_{ij}\right| \le c \tag{4.59}$$

$$K_{\sigma_{ij}} \max \left|\sigma_{ij}\right| \le c \tag{4.60}$$

$$K_{a_{ij}} \max \left|a_{ij}\right| \le c \quad . \tag{4.61}$$

If more than one Δa_{ij} pulse is permitted to occur for each computation cycle, the complexity of the Δa_{ij} adder must be increased significantly. To avoid this situation, let

$$K_{\rho_{ij}} \max \left|\rho_{ij}\right| + K_{\sigma_j} \max \left|\sigma_j\right| \le c \quad . \tag{4.62}$$

Maximum accuracy is obtained with the scale factors as large as possible, subject to the above constraints.

Let the constants m_{ij} and n_{ij} have the same scale factor, $K_{m_{ij}}$. Then

$$K_{\rho_{ij}} \rho_{ij} = \frac{1}{c} K_{m_{ij}} K_{sc} \left(m_{ij} s_{\alpha_j} + n_{ij} c_{\alpha_j}\right) \quad . \tag{4.63}$$

Since $\rho_{ij} = m_{ij} s_{\alpha_j} + n_{ij} c_{\alpha_j}$ [equation (4.43)], the scale factors must satisfy

$$K_{\rho_{ij}} = \frac{K_{m_{ij}} K_{sc}}{c} \quad . \tag{4.64}$$

Also, to prevent more than one $\Delta\rho_{ij}$ pulse per computation cycle,

$$K_{m_{ij}} \left(\left|m_{ij}\right| + \left|n_{ij}\right|\right) \le c \tag{4.65}$$

4.3.4 ΔB Computation

The ΔB computer calculates the increments for the six upper triangle elements of the B matrix given in equations (4.45). Figure 4-12 shows the structure of this computer, which has 18 Y registers to store the A matrix elements and six R registers to generate the Δb_{ij} pulses.

Each of the three diagonal elements of B is the sum of the squares of the six elements in a row of A. By letting $\Delta X = \Delta Y$ (see Figure 4-9), we obtain

$$\Delta Z = \frac{1}{c} Y \Delta Y \approx \frac{1}{2c} \Delta(Y^2) . \tag{4.66}$$

Therefore, by adding $K_{a_{ij}} a_{ij}$ to $K^*_{b_{ii}} b_{ii}$ when $\Delta a_{ij} = +1$ for each j, the content of the Y register that accumulates the Δb_{ii} pulses will be

$$K^*_{b_{ii}} b_{ii} \approx \frac{1}{2c} \sum_{j=1}^{6} \left(K_{a_{ij}} a_{ij}\right)^2 . \tag{4.67}$$

In order for

$$b_{ii} = \sum_{j=1}^{6} a_{ij}^2 \tag{4.68}$$

the scale factors for the diagonal elements of B must satisfy

$$K^*_{b_{ii}} = \frac{K_{a_i}^2}{2c} \tag{4.69}$$

where $K_{a_i} = K_{a_{ij}}$ for all j.

For the off-diagonal elements, multiplication is performed as described previously. In this case,

$$K_{b_{ij}} b_{ij} = \frac{1}{c} \sum_{K=1}^{6} K_{a_i} a_{ik} K_{a_j} a_{jk} . \tag{4.70}$$

Therefore,

$$K_{b_{ij}} = \frac{K_{a_i} K_{a_j}}{c} , \quad i \neq j . \tag{4.71}$$

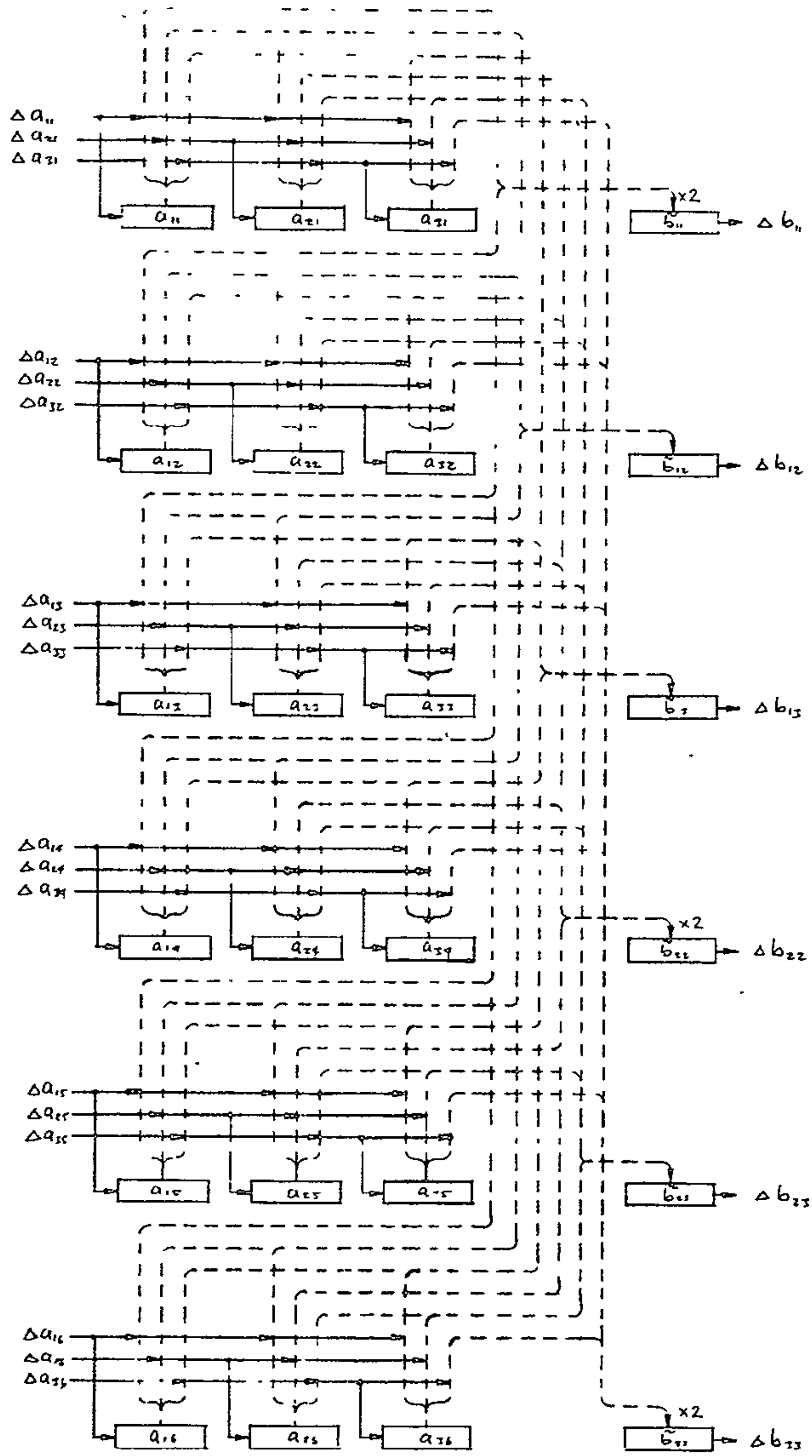


FIG. 4-12 ΔB computer.

The ΔC computer, described in paragraph 4.3.5, requires that all elements of B have the same scale factor. Therefore, let $K_{b_{ij}} = K_b$ for all i and j. The result is that $K_{\tilde{b}_{ii}}$ must equal $K_b/2$, which may be accomplished either by adding twice to the $\tilde{b}_{ii}$ registers for each Δa_{ij} or by delaying the additions by one bit, thus doubling the number added to the register.

In order to ensure that, at most, one Δb_{ij} pulse is produced during each computation cycle, the following constraint is placed on K_a:

$$K_a \sum_{K=1}^{6} \max \left|a_{ik}\right| + \max \left|a_{jk}\right| \leq c \quad \text{for all i, j} \quad . \tag{4.72}$$

This constraint limits K_a to a lower value than the constraint given by equation (4.71). Since $K_{a_{ij}} = K_a$ for all i and j and since it is reasonable to let $K_{\sigma_j} = K_\sigma$ for all j (all wheels have the same speed), equation (4.58) indicates that $K_{\rho_{ij}} = K_\rho$ for all i and j. Equation (4.58) is therefore replaced with

$$K_a = \frac{K_\rho K_\sigma}{c} \quad . \tag{4.73}$$

Table 4.4 shows a computation sequence for the additions to the $\tilde{b}_{ij}$ registers where the variables a_{ij} are stored serially on a single circulating register in the order shown. Each $\tilde{b}_{ij}$ register recycles in the period of a one-word interval of the A register. During interval 1, twice the number in the A register during that interval is added to the $\tilde{b}_{11}$ register if $\Delta a_{11} = 1$. The A register also adds to the $\tilde{b}_{12}$ register if $\Delta a_{21} = 1$ during this interval, and to the $\tilde{b}_{13}$ register if $\Delta a_{31} = 1$. All the computations of the ΔB computer are completed in one cycle of the A register.

TABLE 4.4

$\tilde{B}$ Computation Sequence

Interval	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
A	a_{11}	a_{21}	a_{31}	a_{12}	a_{22}	a_{32}	a_{13}	a_{23}	a_{33}	a_{14}	a_{24}	a_{34}	a_{15}	a_{25}	a_{35}	a_{16}	a_{26}	a_{30}
$\tilde{b}_{11}$	$2\Lambda\Delta a_{11}$			$2\Lambda\Delta a_{12}$			$2\Lambda\Delta a_{13}$			$2\Lambda\Delta a_{14}$			$2\Lambda\Delta a_{15}$			$2\Lambda\Delta a_{16}$		
$\tilde{b}_{12}$	$\Lambda\Delta a_{21}$	$\Lambda\Delta a_{11}$		$\Lambda\Delta a_{22}$	$\Lambda\Delta a_{12}$		$\Lambda\Delta a_{23}$	$\Lambda\Delta a_{13}$		$\Lambda\Delta a_{24}$	$\Lambda\Delta a_{14}$		$\Lambda\Delta a_{25}$	$\Lambda\Delta a_{15}$		$\Lambda\Delta a_{26}$	$\Lambda\Delta a_{16}$	
$\tilde{b}_{13}$	$\Lambda\Delta a_{31}$		$\Lambda\Delta a_{11}$	$\Lambda\Delta a_{32}$		$\Lambda\Delta a_{12}$	$\Lambda\Delta a_{33}$		$\Lambda\Delta a_{13}$	$\Lambda\Delta a_{24}$		$\Lambda\Delta a_{14}$	$\Lambda\Delta a_{35}$		$\Lambda\Delta a_{15}$	$\Lambda\Delta a_{36}$		$\Lambda\Delta a_{16}$
$\tilde{b}_{22}$		$2\Lambda\Delta a_{21}$			$2\Lambda\Delta a_{22}$			$2\Lambda\Delta a_{23}$			$2\Lambda\Delta a_{24}$			$2\Lambda\Delta a_{25}$			$2\Lambda\Delta a_{26}$	
$\tilde{b}_{23}$		$\Lambda\Delta a_{31}$	$\Lambda\Delta a_{21}$		$\Lambda\Delta a_{32}$	$\Lambda\Delta a_{22}$		$\Lambda\Delta a_{33}$	$\Lambda\Delta a_{23}$		$\Lambda\Delta a_{34}$	$\Lambda\Delta a_{24}$		$\Lambda\Delta a_{35}$	$\Lambda\Delta a_{25}$		$\Lambda\Delta a_{36}$	$\Lambda\Delta a_{26}$
$\tilde{b}_{33}$			$2\Lambda\Delta a_{31}$			$2\Lambda\Delta a_{32}$			$2\Lambda\Delta a_{33}$			$2\Lambda\Delta a_{34}$			$2\Lambda\Delta a_{35}$			$2\Lambda\Delta a_{36}$

4.3.5 ΔC Computation

The ΔC computer, illustrated in Figure 4.13, computes increments for the adjoint of the B matrix given by equations (4.46). As in the ΔB computer, when squaring computation is performed, the addition to an R register is doubled. This can be accomplished by delaying the number to be added by one bit time. The doubled addition is signified in the figure by a "2" placed next to the ΔX arrowhead which indicates addition of Y to R. As for the previous computers, the scale factors must satisfy

$$K_c = \frac{K_b^2}{c} \quad . \tag{4.74}$$

By requiring that

$$K_b \; \Sigma \max |b_{ij}| \leq c \tag{4.75}$$

where the summation is taken over the four terms b_{ij} of each of equations (4.46), only one Δc pulse will be produced during each computation cycle.

Table 4.5 presents a computation sequence for the additions to the $\tilde{c}$ registers for this computation, where the elements of B are stored serially on a single circulating register in the sequence shown.

TABLE 4.5
$\tilde{C}$ and $\tilde{d}_o$ Computation Sequences

Interval:	1	2	3	4	5	6
B:	b_{11}	b_{12}	b_{13}	b_{22}	b_{23}	b_{33}
$\tilde{c}_{11}$				$B\Delta b_{33}$	$-2B\Delta b_{23}$	$B\Delta b_{22}$
$\tilde{c}_{12}$		$-B\Delta b_{33}$	$B\Delta b_{23}$		$B\Delta b_{13}$	$-B\Delta b_{12}$
$\tilde{c}_{13}$		$B\Delta b_{23}$	$-B\Delta b_{22}$	$-B\Delta b_{13}$	$B\Delta b_{12}$	
$\tilde{c}_{22}$	$B\Delta b_{33}$		$-2B\Delta b_{13}$			$B\Delta b_{11}$
$\tilde{c}_{23}$	$-B\Delta b_{23}$	$B\Delta b_{13}$	$B\Delta b_{12}$		$-B\Delta b_{11}$	
$\tilde{c}_{33}$	$B\Delta b_{22}$	$-2B\Delta b_{12}$	$B\Delta b_{11}$			
C:	c_{22}	c_{23}	c_{33}	c_{11}	c_{12}	c_{13}
$\tilde{d}_o$:	$B\Delta c_{11}$	$B\Delta c_{12}$	$B\Delta c_{13}$	$C\Delta b_{11}$	$C\Delta b_{12}$	$C\Delta b_{13}$

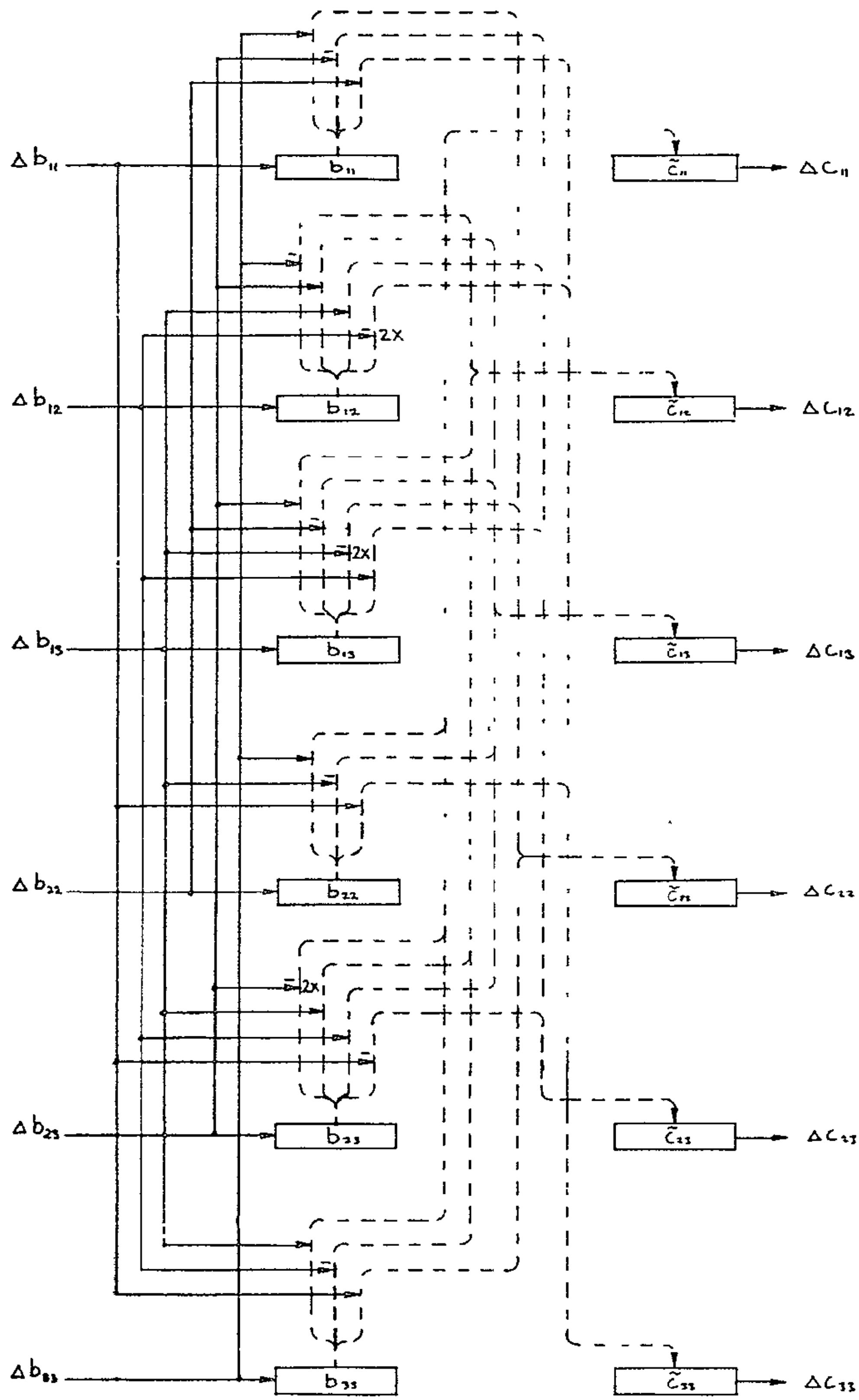


FIG. 4-13 ΔC computer.

4.3.6 Δd_o Computation

Computation of the increments for d_o, the determinant at the B matrix [equation (4.47)] is performed by adding elements of B and C to a single R register, as shown in Figure 4-14. The sequence of these additions is shown in Table 4.5, where the elements of C are stored on a single register in the sequence shown. The scale factor for d_o is given by

$$K_{d_o} = \frac{K_b K_c}{c} \tag{4.76}$$

and by constraining K_b and K_c so that

$$K_b K_c \sum_{j=1}^{3} \max|b_{1j}| + \max|c_{1j}| \leq c \tag{4.77}$$

with only one Δd_o pulse produced during each computation cycle.

4.3.7 ΔD Computation

Figure 4-15 illustrates computation of the increments for the 18 D elements given by equations (4 48) . Similarly to the previous computers, the scale factors for the elements of D must satisfy

$$K_{d_{ij}} = \frac{K_a K_c}{c} = K_d . \tag{4.78}$$

Therefore, they must all be equal. By requiring that

$$K_a K_c = \sum_{k=1}^{3} \max|a_{ki}| + \max|\ \ _{kj}| \leq c \quad \text{for all } i,\ j \tag{4.79}$$

only one Δd pulse is produced during each computation cycle.

Table 4.6 presents a computation sequence for additions to the $\tilde{d}$ registers where the elements of A and C are stored serially on respective single registers in the relative sequence shown.

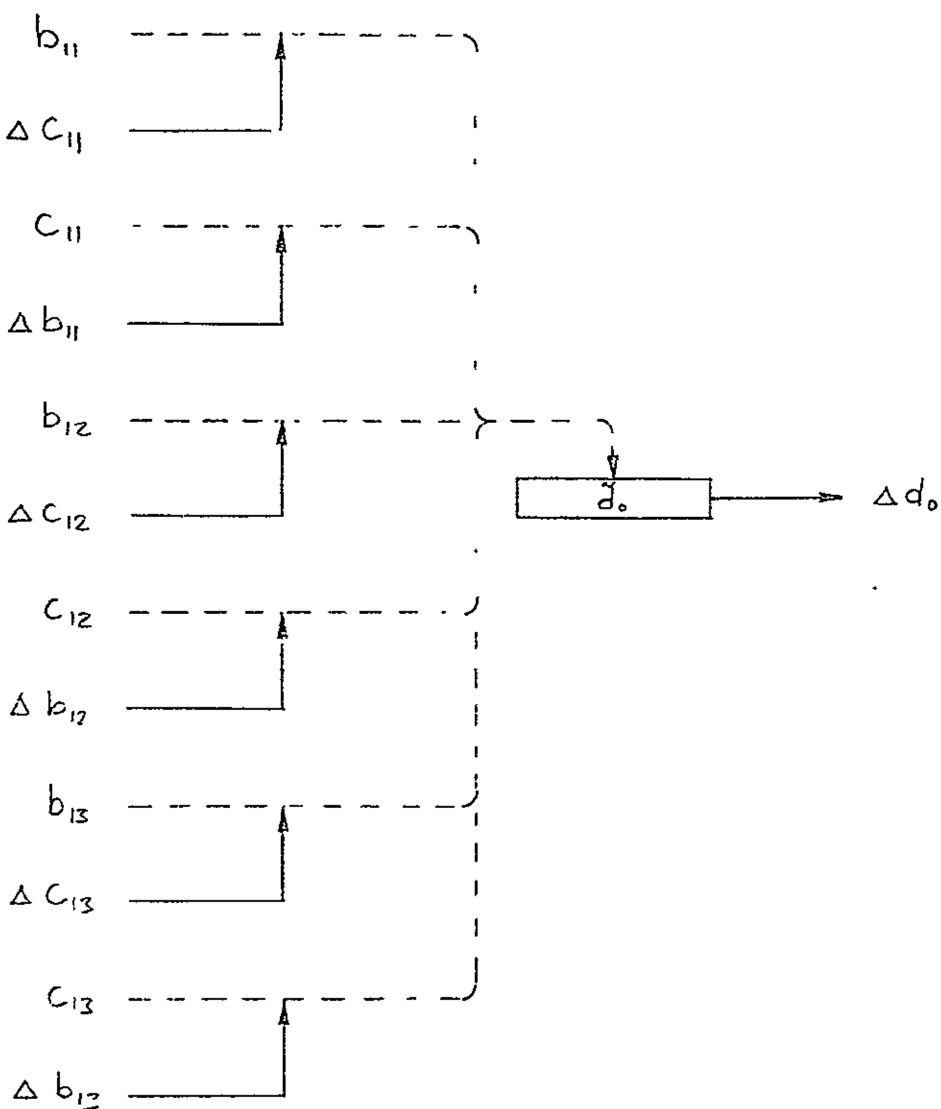


FIG. 4-14 Δd_0 computer.

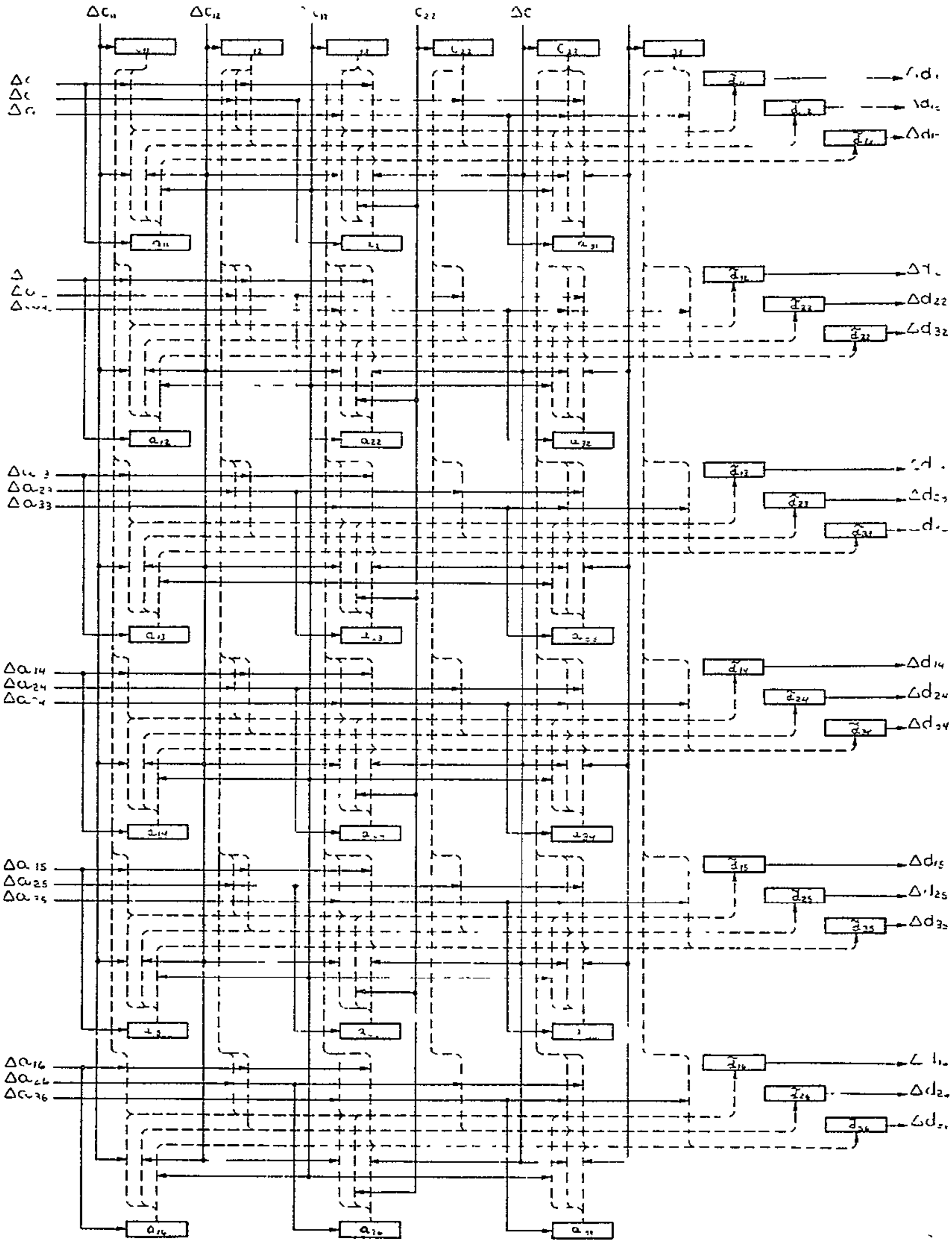


FIG. 4-15 ΔD computer.

TABLE 4.6

$\tilde{D}$ Computation Sequence

Interval:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
A:	a_{11}	a_{21}	a_{31}	a_{12}	a_{22}	a_{32}	a_{13}	a_{23}	a_{33}	a_{14}	a_{24}	a_{34}	a_{15}	a_{25}	a_{35}	a_{16}	a_{26}	a_{36}
C:	c_{11}	c_{12}	c_{13}	c_{22}	c_{23}	c_{33}	c_{11}	c_{12}	c_{13}	c_{22}	c_{23}	c_{33}	c_{11}	c_{12}	c_{13}	c_{22}	c_{23}	c_{33}
$\tilde{d}_{11}$	$A\Delta c_{11}$	$A\Delta c_{12}$	$A\Delta c_{13}$				$C\Delta a_{11}$	$C\Delta a_{21}$	$C\Delta a_{31}$									
$\tilde{d}_{12}$	$A\Delta c_{12}$	$A\Delta c_{22}$	$A\Delta c_{23}$					$C\Delta a_{11}$		$C\Delta a_{21}$	$C\Delta 2_{31}$							
$\tilde{d}_{13}$	$A\Delta c_{13}$	$A\Delta c_{23}$	$A\Delta c_{33}$						$C\Delta a_{11}$		$C\Delta a_{21}$	$C\Delta a_{31}$						
$\tilde{d}_{21}$				$A\Delta c_{11}$	$A\Delta c_{12}$	$A\Delta c_{13}$	$C\Delta a_{12}$	$C\Delta a_{22}$	$C\Delta a_{32}$									
$\tilde{d}_{22}$				$A\Delta c_{12}$	$A\Delta c_{22}$	$A\Delta c_{23}$		$C\Delta a_{12}$		$C\Delta a_{22}$	$C\Delta a_{32}$							
$\tilde{d}_{23}$				$A\Delta c_{13}$	$A\Delta c_{23}$	$A\Delta c_{33}$			$C\Delta a_{12}$		$C\Delta a_{22}$	$C\Delta a_{32}$						
$\tilde{d}_{31}$							$A\Delta c_{11}$	$A\Delta c_{12}$	$A\Delta c_{13}$				$C\Delta a_{13}$	$C\Delta a_{23}$	$C\Delta a_{33}$			
$\tilde{d}_{32}$							$A\Delta c_{12}$	$A\Delta c_{22}$	$A\Delta c_{23}$					$C\Delta a_{13}$		$C\Delta a_{23}$	$C\Delta a_{33}$	
$\tilde{d}_{33}$							$A\Delta c_{13}$	$A\Delta c_{23}$	$A\Delta c_{33}$						$C\Delta a_{13}$		$C\Delta a_{23}$	$C\Delta a_{33}$
$\tilde{d}_{41}$										$A\Delta c_{11}$	$A\Delta c_{12}$	$A\Delta c_{13}$	$C\Delta a_{14}$	$C\Delta a_{24}$	$C\Delta a_{34}$			
$\tilde{d}_{42}$										$A\Delta c_{12}$	$A\Delta c_{22}$	$A\Delta c_{23}$		$C\Delta a_{14}$		$C\Delta a_{24}$	$C\Delta a_{34}$	
$\tilde{d}_{43}$										$A\Delta c_{13}$	$A\Delta c_{23}$	$A\Delta c_{33}$			$C\Delta a_{14}$		$C\Delta a_{24}$	$C\Delta a_{34}$
$\tilde{d}_{51}$	$C\Delta a_{15}$	$C\Delta a_{25}$	$C\Delta a_{35}$										$A\Delta c_{11}$	$A\Delta c_{12}$	$A\Delta c_{13}$			
$\tilde{d}_{52}$		$C\Delta a_{15}$		$C\Delta a_{25}$	$C\Delta a_{35}$								$A\Delta c_{12}$	$A\Delta c_{22}$	$A\Delta c_{23}$			
$\tilde{d}_{53}$			$C\Delta a_{15}$		$C\Delta a_{25}$	$C\Delta a_{35}$							$A\Delta c_{13}$	$A\Delta c_{23}$	$A\Delta c_{33}$			
$\tilde{d}_{61}$	$C\Delta a_{16}$	$C\Delta a_{26}$	$C\Delta a_{36}$													$A\Delta c_{11}$	$A\Delta c_{12}$	$A\Delta c_{13}$
$\tilde{d}_{62}$		$C\Delta a_{16}$		$C\Delta a_{26}$	$C\Delta a_{36}$											$A\Delta c_{12}$	$A\Delta c_{22}$	$A\Delta c_{23}$
$\tilde{d}_{63}$			$C\Delta a_{16}$		$C\Delta a_{26}$	$C\Delta a_{36}$										$A\Delta c_{13}$	$A\Delta c_{23}$	$A\Delta c_{33}$

4.3.8 Δu Computation

The Δu computer performs the divisions of equations (4.49) by multiplications in the feedback paths, as illustrated in Figure 4-16.

When the number stored in the $\tilde{\varepsilon}_1$ register is positive, the element identified by the letter S in the figure (and inappropriately but conventionally termed a "servo"), produces positive Δu pulses at the computation cycle rate. When $\tilde{\varepsilon}_1$ is negative, the servo produces negative Δu pulses. No pulses are produced when $\tilde{\varepsilon}_1$ is zero.

When a positive $\Delta\omega_{cx}$ pulse occurs, the number $K_{I_x} I_x$, which is scaled to be nearly as large as c, is added to the $\tilde{\varepsilon}_1$ register. Similarly, a positive Δd_o pulse causes $K_u u_1$ to be added to $\tilde{\varepsilon}_1$. The resulting Δu_1 pulses produced cause $K_{d_o} d_o$ to be repeatedly subtracted from $\tilde{\varepsilon}_1$ until $\tilde{\varepsilon}_1$ is zero. (The register may not go to zero exactly, in which case the servo will generate alternate positive and negative pulses.)

The change in $\tilde{\varepsilon}_1$ due to $\Delta\dot{\omega}_{cx}$ pulses or Δd_o pulses can be expressed as

$$\Delta\varepsilon_1 = K_{I_x} I_x \, \Delta\dot{\omega}_{cx} - K_u u_1 \Delta d_o - K_{d_o} d_o \Delta u_1 \tag{4.80}$$

$$= \Delta \left[\left(K_{I_x} I_x \right) \left(K_{\dot{\omega}_x} \dot{\omega}_{cx} \right) \right] - \Delta \left[\left(K_u u_1 \right) \left(K_{d_o} d_o \right) \right] . \tag{4.81}$$

Since $\tilde{\varepsilon}_1 \approx 0$, and with proper initialization of the d_o and u_1 registers,

$$u_1 \approx \frac{K_{I_x} K_{\omega_x}}{K_u K_{d_o}} \frac{I_x \dot{\omega}_{cx}}{d_o} . \tag{4.82}$$

To satisfy equations (4.49), the scale factors must therefore satisfy

$$\frac{K_{I_x} K_{\dot{\omega}_x}}{K_u K_{d_o}} = 1 . \tag{4.83}$$

Similar requirements apply to the y and z axes.

To ensure that the ε_1 register does not overflow, the scale factors should also satisfy

$$K_{I_x} I_x + K_u \max \left| u_1 \right| + K_{d_o} \max \left| d_o \right| \leq c . \tag{4.84}$$

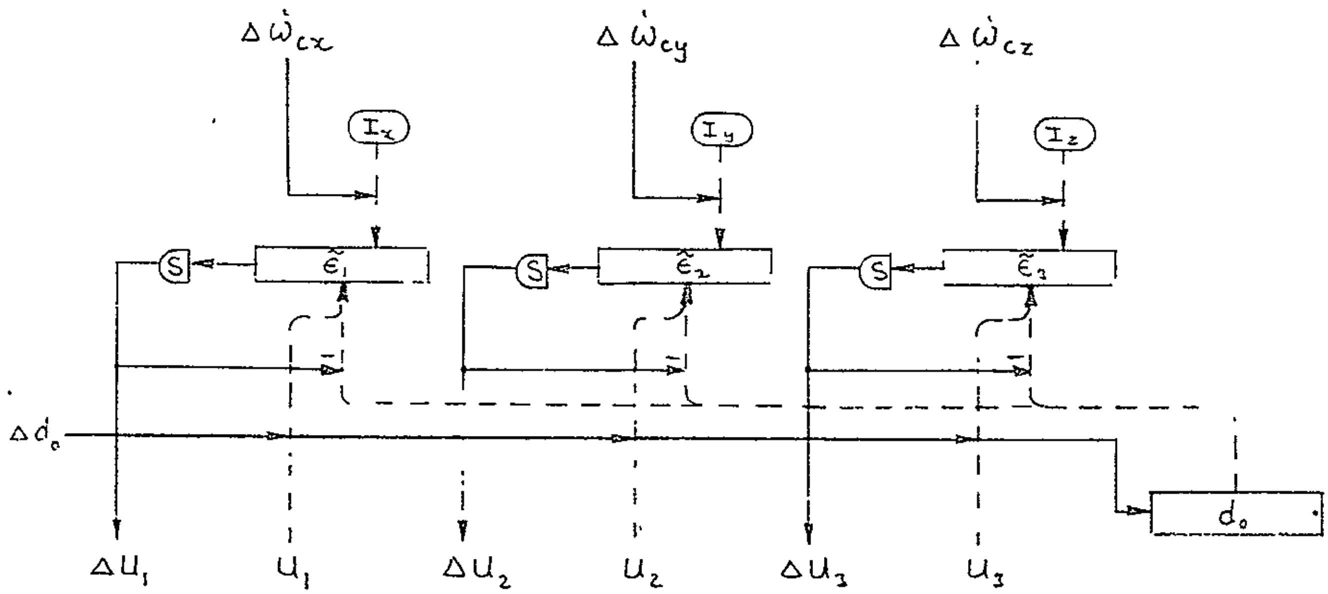


FIG. 4-16 Δu computer.

Table 4.7 presents a computation sequence for $\tilde{\varepsilon}_1$, $\tilde{\varepsilon}_2$, and $\tilde{\varepsilon}_3$, where u_1, u_2, and u_3 are stored serially on a single register as shown, and where d_o is stored on a single register that recycles at the rate of each element of u. The computation is completed in two cycles of the u register.

TABLE 4.7

$\tilde{\varepsilon}$ Computation Sequence

Interval:	1	2	3	4	5	6
u:	u_1	u_2	u_3	u_1	u_2	u_3
d_o:	d_o	d_o	d_o	d_o	d_o	d_o
$\tilde{\varepsilon}_1$	$u\Delta d_o$			$d_o\Delta u_1$		
$\tilde{\varepsilon}_2$		$u\Delta d_o$			$d_o\Delta u_2$	
$\tilde{\varepsilon}_3$.			$u\Delta d_o$			$d_o\Delta u_3$

4.3.9 $\Delta\dot{\alpha}_c$ Computation

The $\Delta\dot{\alpha}_c$ computer, illustrated in Figure 4-17, computes increments for the desired outputs of the steering law computer [equations (4.50)]. As in the previous cases, the scale factor for the registers that accumulate the $\Delta\dot{\alpha}_c$ pulses (not shown) must satisfy

$$K_\alpha = \frac{K_d K_u}{c} \quad . \tag{4.85}$$

By requiring that

$$K_\alpha \sum_{j=1}^{3} \max \left| d_{ij} \right| + \max \left| u_j \right| \leq c \quad \text{for each } i \tag{4.86}$$

only one $\tilde{\dot{\alpha}}_c$ register overflow can occur during each computation cycle.

Table 4.8 shows a computation sequence for the $\tilde{\dot{\alpha}}_c$ registers, where the elements of D and u are each stored serially on single long registers in the sequence shown.

Since $\dot{\alpha}_c$ is most likely desired in analog form for the CMG gimbal control system, the $\Delta\dot{\alpha}_c$ pulses may be fed into counters that are parts of a D/A conversion system.

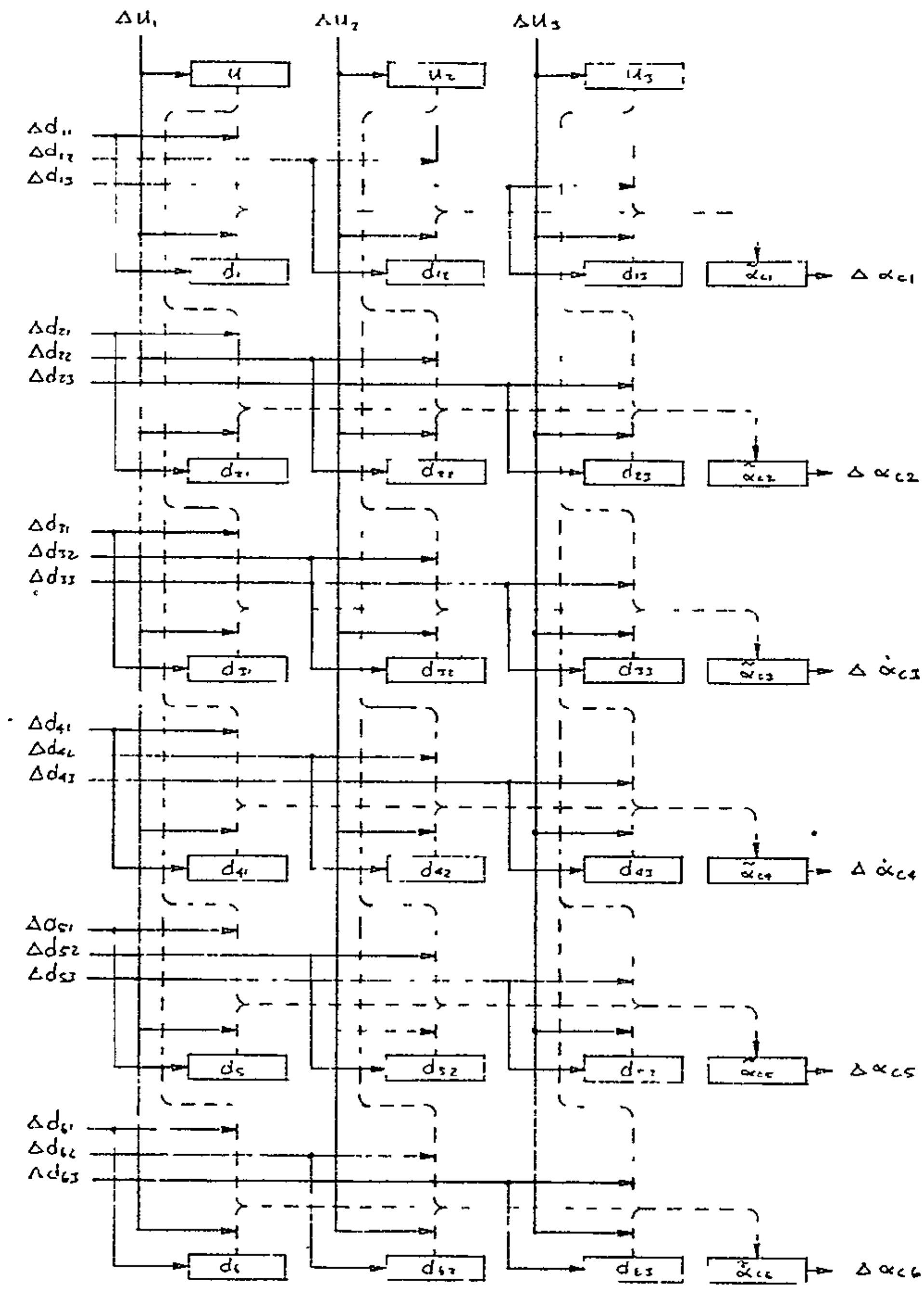


FIG. 4-17 $\Delta\dot{\alpha}_c$ computer.

TABLE 4.8

$\tilde{\ddot{\alpha}}_c$ Computation Sequence

Interval:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
D:	d_{11}	d_{12}	d_{13}	d_{21}	d_{22}	d_{23}	d_{31}	d_{32}	d_{33}	d_{41}	d_{42}	d_{43}	d_{51}	d_{52}	d_{53}	d_{61}	d_{62}	d_{63}
u:	u_1	u_2	u_3	u_1	u_2	u_3	u_1	u_2	u_3	u_1	u_2	u_3	u_1	u_2	u_3	u_1	u_2	u_3
$\tilde{\ddot{\alpha}}_{c1}$	$D\Delta u_2$	$D\Delta u_2$	$D\Delta u_3$	$u\Delta d_{11}$	$u\Delta d_{12}$	$u\Delta d_{13}$												
$\tilde{\ddot{\alpha}}_{c2}$		$u\Delta d_{22}$	$u\Delta d_{23}$	$D\Delta u_1$	$D\Delta u_2$	$D\Delta u_3$	$u\Delta d_{21}$											
$\tilde{\ddot{\alpha}}_{c3}$			$u\Delta d_{33}$	$u\Delta d_{31}$	$u\Delta d_{32}$		$D\Delta u_1$	$D\Delta u_2$	$D\Delta u_3$									
$\tilde{\ddot{\alpha}}_{c4}$				$u\Delta d_{41}$	$u\Delta D_{42}$	$u\Delta d_{43}$				$D\Delta u_1$	$D\Delta u_2$	$D\Delta u_3$						
$\tilde{\ddot{\alpha}}_{c5}$					$u\Delta d_{52}$	$u\Delta d_{53}$	$u\Delta d_{51}$						$D\Delta u_1$	$D\Delta u_2$	$D\Delta u_3$			
$\tilde{\ddot{\alpha}}_{c6}$						$u\Delta d_{63}$	$u\Delta d_{61}$	$u\Delta d_{62}$								$D\Delta u_1$	$D\Delta u_2$	$D\Delta u_3$

4.4 OTHER COMPUTATIONS

4.4.1 Triad Conversion

The triad converter changes the six rate signals $r_s = \{r_{s1}, \cdots, r_{s6}\}$ from the dodecahedron configured rate gyros, described in paragraph 4.2.3, to vehicle rates $\omega_s = \{\omega_{sx}, \omega_{sy}, \omega_{sz}\}$. The equations to be computed are given by equations (4.21) and (4.22) and are repeated here for convenience :

$$\omega_s = E^{\dagger} r_s$$

where

$$E^{\dagger} = (E^T E)^{-1} E^T$$

$$E = \lambda \bar{E}$$

$$\bar{E} = \begin{bmatrix} s & o & c \\ -s & o & c \\ c & s & o \\ s & -s & o \\ o & c & s \\ o & c & -s \end{bmatrix}$$

$$\lambda = \text{diag} \{\lambda_1, \dots, \lambda_6\}$$

$$s = \sin \delta \approx 0.526 \quad .$$

$$c = \cos \delta \approx 0.850$$

If rate gyro number i is used, $\lambda_i = 1$; if it is switched off because it has failed, $\lambda_i = 0$. Let

$$F = E^T E \tag{4.87}$$

$$P = \text{adj } F \tag{4.88}$$

$$Q = PE^T \tag{4.89}$$

$$q_o = \det F \quad . \tag{4.90}$$

Then

$$\omega_s = \frac{1}{q_o} Q r_s \quad . \tag{4.91}$$

Since E changes with λ, the Q matrix can be recomputed by performing these computations each time λ changes. This method requires minimum memory and is probably the preferred method with a general purpose computer. The elements of Q can also be expressed in terms of the elements of λ. This method is preferred for incremental and analog computation, and is described as follows. Let

$$\mu_1 = 2s^4c \tag{4.92}$$

$$\mu_2 = 2s^5 + \mu_1 \tag{4.93}$$

$$\mu_3 = \mu_1 + \mu_2 \tag{4.94}$$

$$\mu_4 = \mu_2 + \mu_3 \tag{4.95}$$

$$\lambda_{ijk} = \lambda_i\lambda_j\lambda_k . \tag{4.96}$$

Then the 18 elements of Q are given by the following expressions:

$$\begin{aligned}
q_{11} &= \mu_2\lambda_{123} + \mu_2\lambda_{124} + \mu_4\lambda_{125} \\
&\quad + \mu_4\lambda_{126} + \mu_2\lambda_{135} - \mu_1\lambda_{136} \\
&\quad - \mu_1\lambda_{145} + \mu_2\lambda_{146} + 2\mu_2\lambda_{156} \\
q_{12} &= -\mu_2\lambda_{123} - \mu_2\lambda_{124} - \mu_4\lambda_{125} \\
&\quad - \mu_4\lambda_{126} + \mu_1\lambda_{235} - \mu_2\lambda_{236} \\
&\quad - \mu_2\lambda_{245} + \mu_1\lambda_{246} + 2\mu_2\lambda_{256} \\
q_{13} &= \mu_3\lambda_{134} + \mu_4\lambda_{135} + \mu_3\lambda_{136} \\
&\quad + \mu_3\lambda_{234} + \mu_3\lambda_{235} + \mu_4\lambda_{236} \\
&\quad + \mu_1\lambda_{346} + \mu_1\lambda_{346} + 2\mu_3\lambda_{356} \\
q_{14} &= \mu_3\lambda_{134} + \mu_3\lambda_{145} + \mu_4\lambda_{146} \\
&\quad + \mu_3\lambda_{234} + \mu_4\lambda_{245} + \mu_3\lambda_{246} \\
&\quad + \mu_1\lambda_{345} + \mu_1\lambda_{346} + 2\mu_3\lambda_{456}
\end{aligned} \tag{4.97}$$

$$q_{15} = -\mu_3\lambda_{135} + \mu_2\lambda_{145} - \mu_3\lambda_{156}$$

$$-\mu_2\lambda_{235} + \mu_3\lambda_{245} + \mu_3\lambda_{256}$$

$$-\mu_2\lambda_{356} + \mu_2\lambda_{456}$$

$$q_{16} = -\mu_2\lambda_{136} + \mu_3\lambda_{146} + \mu_3\lambda_{156}$$

$$-\mu_3\lambda_{236} + \mu_2\lambda_{246} - \mu_3\lambda_{256}$$

$$-\mu_2\lambda_{356} + \mu_2\lambda_{456}$$

$$q_{21} = -\mu_3\lambda_{123} + \mu_3\lambda_{124} - \mu_2\lambda_{125}$$

$$+\mu_2\lambda_{126} - \mu_3\lambda_{135} + \mu_2\lambda_{135}$$

$$-\mu_2\lambda_{145} + \mu_3\lambda_{146}$$

$$q_{22} = \mu_3\lambda_{123} - \mu_3\lambda_{124} - \mu_2\lambda_{125} \tag{4.97}$$

$$+\mu_2\lambda_{126} - \mu_2\lambda_{235} + \mu_3\lambda_{236}$$

$$-\mu_3\lambda_{245} + \mu_2\lambda_{246}$$

$$q_{23} = 2\mu_2\lambda_{123} + \mu_4\lambda_{134} + \mu_2\lambda_{135}$$

$$-\mu_1\lambda_{136} + \mu_4\lambda_{234} - \mu_1\lambda_{235}$$

$$+\mu_2\lambda_{236} + \mu_2\lambda_{345} + \mu_2\lambda_{346}$$

$$q_{24} = -2\mu_2\lambda_{124} - \mu_4\lambda_{134} - \mu_1\lambda_{145}$$

$$-\mu_2\lambda_{146} - \mu_4\lambda_{234} - \mu_2\lambda_{245}$$

$$+\mu_1\lambda_{246} - \mu_2\lambda_{345} - \mu_2\lambda_{346}$$

$$q_{25} = 2\mu_3\lambda_{125} + \mu_4\lambda_{135} + \mu_3\lambda_{145}$$

$$+\mu_1\lambda_{156} + \mu_3\lambda_{235} + \mu_4\lambda_{245}$$

$$+\mu_1\lambda_{256} + \mu_3\lambda_{356} + \mu_3\lambda_{456}$$

$$
\begin{aligned}
q_{26} = {} & 2\mu_3\lambda_{126} + \mu_3\lambda_{136} + \mu_4\lambda_{146} \\
& +\mu_3\lambda_{156} + \mu_4\lambda_{236} + \mu_3\lambda_{246} \\
& +\mu_1\lambda_{256} + \mu_3\lambda_{356} + \mu_3\lambda_{456} \\
q_{31} = {} & \mu_1\lambda_{123} + \mu_1\lambda_{124} + \mu_3\lambda_{135} \\
& +\mu_3\lambda_{126} + 2\mu_3\lambda_{134} + \mu_4\lambda_{135} \\
& -\mu_3\lambda_{136} + \mu_3\lambda_{145} + \mu_4\lambda_{146} \\
q_{32} = {} & \mu_1\lambda_{123} + \mu_1\lambda_{124} + \mu_3\lambda_{125} \\
& +\mu_3\lambda_{126} + 2\mu_3\lambda_{234} + \mu_3\lambda_{235} \\
& +\mu_4\lambda_{236} + \mu_4\lambda_{245} + \mu_3\lambda_{246} \\
q_{33} = {} & -\mu_2\lambda_{134} - \mu_3\lambda_{135} - \mu_2\lambda_{136} \\
& +\mu_2\lambda_{234} + \mu_2\lambda_{235} + \mu_3\lambda_{236} \\
& -\mu_3\lambda_{345} + \mu_3\lambda_{346} \\
q_{34} = {} & -\mu_2\lambda_{134} - \mu_2\lambda_{145} - \mu_3\lambda_{146} \\
& +\mu_2\lambda_{234} + \mu_3\lambda_{245} + \mu_2\lambda_{246} \\
& +\mu_3\lambda_{345} - \mu_3\lambda_{346} \\
q_{35} = {} & \mu_2\lambda_{135} - \mu_1\lambda_{145} + \mu_2\lambda_{156} \\
& -\mu_1\lambda_{235} + \mu_2\lambda_{245} + \mu_2\lambda_{256} \\
& +2\mu_2\lambda_{345} + \mu_4\lambda_{356} + \mu_4\lambda_{456} \\
q_{36} = {} & \mu_1\lambda_{136} - \mu_2\lambda_{146} - u_2\lambda_{156} \\
& -\mu_2\lambda_{236} + \mu_1\lambda_{246} - \mu_2\lambda_{256} \\
& -2\mu_2\lambda 346 - \mu_4\lambda_{356} - \mu_4\lambda_{456} \; .
\end{aligned}
\tag{4.97}
$$

To express q_o in a similar manner, let

$$\mu_5 = 4s^4c^2 = \left(c^3 - s^3\right)^2 \tag{4.98}$$

$$\mu_6 = 4s^2c^4 = \left(c^3 + s^3\right)^2 \quad . \tag{4.99}$$

Then

$$\begin{aligned} q_o = {} & \mu_5\lambda_{123} + \mu_5\lambda_{124} + \mu_6\lambda_{125} \\ & + \mu_6\lambda_{126} + \mu_6\lambda_{134} + \mu_6\lambda_{135} \\ & + \mu_5\lambda_{136} + \mu_5\lambda_{145} + \mu_6\lambda_{146} \\ & + \mu_5\lambda_{156} + \mu_6\lambda_{234} + \mu_5\lambda_{235} \\ & + \mu_6\lambda_{236} + \mu_6\lambda_{245} + \mu_5\lambda_{246} \\ & + \mu_5\lambda_{256} + \mu_5\lambda_{345} + \mu_6\lambda_{346} \\ & + \mu_6\lambda_{356} + \mu_6\lambda_{456} \quad . \end{aligned} \tag{4.100}$$

All of the preceding elements are sums of the constants $\{\mu_1, \ldots, \mu_6\}$ weighted by λ's that are either one or zero. A possible method for computing ω_s [equation (4.91)] by the incremental method described in paragraph 4.3 is to initially store the 16 q's and the q_o that correspond to no failures in the Y registers; then when a failure occurs, change the q's in accordance with the preceding equations. Only the six constants $\{\mu_1, \ldots, \mu_6\}$ need to be stored to accomplish the required changes in the q's and q_o. The incremental computation for Δr_s will then be similar to the Δu and $\Delta\dot{\alpha}c$ computations described in paragraph 4.3.

4.4.2 Control Compensation, Failure Monitoring, and Mode Control

The three compensators required for the vehicle rate loop (see Figure 4-6) can readily be accomplished by incremental computation. The application of this computation technique to flight controls, for example, has been studied by several flight control system manufacturers, including Sperry. Due to the limited scope of this study, this section of the CMG control computer will not be investigated.

The failure monitoring and mode control function of the CMG control computer (Figure 4-6) generates the λ's for the triad conversion; the rotor speeds σ for the steering law computer; and supplies the display signals required to monitor the failure status of the CMG's and sensors. The major computations required for this function are the parity check equations for the sensors listed in Table 4.1. A computer structure for the computations required by the failure monitoring and mode control function is not presented in this study due to its limited scope.

4.5 CONCLUSIONS AND RECOMMENDATIONS FOR FURTHER STUDY

The steering law computations are by far the most complex computations required by a CMG control computer, and they outweigh the remaining computing functions in selecting the type of computer to employ for controlling CMG's. Since these computations are time-variant and nonlinear, digital computation is required. A general purpose computer can certainly perform these computations but it is considerably more limited in speed than an incremental computer. The time required to do the steering law computations on a UNIVAC® 1819 computer (designed for airborne use) is estimated to be 6.37 milliseconds. If we roughly estimate that the time required to perform the total CMG control computer computations is 10 milliseconds, then a sampling rate of 100 samples per second saturates the computer. For some of the anticipated requirements for future CMG control systems, sampling rates of this order are required. It can therefore be concluded that the use of a central, general purpose computer, which is also required to perform other than CMG control functions, is not feasible for the type of CMG system described in this report.

An incremental computer can be made much faster than a general purpose computer, but its speed depends on the level of serialization of its computations. A bit-time of 0.5 microsecond is easily accomplished, and if a 300-bit register is used to serially store a set of 18 Y variables (such as the 18 a's or d's), a computation cycle takes 0.15 millisecond. Of course, there are also lags in incremental computers that must be considered when comparing them with the general purpose type, but it is clear that incremental computers are much faster. Also, since computations are performed in parallel, adding functions to an incremental computer does not require additional computation time.

In terms of complexity, it is not obvious at this stage which type of computer is the simplest. Further study would be required to determine parts counts. They may be quite competitive in this aspect.

Since incremental CMG control computers have some definite advantages over general purpose computers for this application, it is recommended that a further study be initiated to design a prototype incremental computer for a specific CMG control system requirement, and to compare it with a general purpose type, programmed for the same function, in terms of complexity (parts count), accuracy, reliability, weight, and power consumption.

REFERENCES

1 Control Moment Gyro Study, Vol I, "Control Moment Gyro System Analysis", SAMSO-TR-67-74 (SECRET), Space and Missile Systems Organization, Air Force System Command, Los Angeles, California, November 1967.

2 L. A. Zadeh and A. Desoer, Linear System Theory, McGraw-Hill Book Co., Inc, New York, 1963, pp 577 - 582.

3 Ralph Deutsch, Estimation Theory, Prentice-Hall Inc, Englewood Cliffs, New Jersey, 1965, pp 82 - 87.

4 J. P. Gilmore, A Non-Orthogonal Multi-Sensor Strapdown Inertial Reference Unit, Instrumentation Laboratory, Massachusetts Institute of Technology, E-2308, August 1968, (NASA Contract NAS 9-8242).

5 Ibid.

6 Deutsch, p 83.

7 Ibid, pp 83 - 84.

CHAPTER 5
SIMU DIRECTION COSINE SIMULATION PROGRAM RUNS

5.1 DESCRIPTION OF METHOD AND RESULTS

The purpose of the simulation of SIMU operation was to determine the error accumulation for large angle changes in position and also the worst-case mode of error accumulation during extended runs.

5.1.1 Types of Runs

Three types of runs were made and are described below.

(1) Runs simulating a motion to a given angle and then back to the zero position. The angles used range from less than 1^{0} to 180^{0}.

(2) Oscillatory runs taken near the zero reference position where the curve of data from the test above indicates that the worst error accumulation occurs.

(3) Three-axis rotations where the inputs are: (a) up to one radian for x, followed by one radian for y and then one radian for z, followed in reverse order for the return to zero; (b) single pulse increments in x, y and z followed successively until each axis has had the pulse required to read up to one radian rotation (32,768), then the reverse order of the same increments of x, y and z until zero is reached.

5.1.2 Test Results

The results of these tests show the following:

Figure 5-1 shows that error accumulations in direction cosine values after rotation to a particular angle (of 180^{0} or less) and then returning to zero degrees were limited to a maximum value of approximately one part in 37. Refer also to Table 5.1.

The error moves up steeply until approximately 2000 unidirectional input pulses have been accumulated. This is approximately a range of one bit of error for every two pulses fed the system by the time the rotation back to zero is completed. (See steep initial slope on Fig 5-1.)

The three axis runs were made for continuous inputs of a fixed number of pulses for x, y and then z sequentially. They were also made for runs where the input pulses were increments of Δx, Δy and Δz repeated in that order until the required number of sets of pulses were processed. Thereafter the reverse sequences of pulses were given to return the system to the zero reference. Table 5.2 lists the above data plus that for each set of two axis updates for 8, 128 and 32,768 sets of input pulses, the value 22,768 approximating one radian of rotation. The results show the order of magnitude of error for the number of pulses specified to be the same as for single axis operation. However, as is expected, they are not identical values.

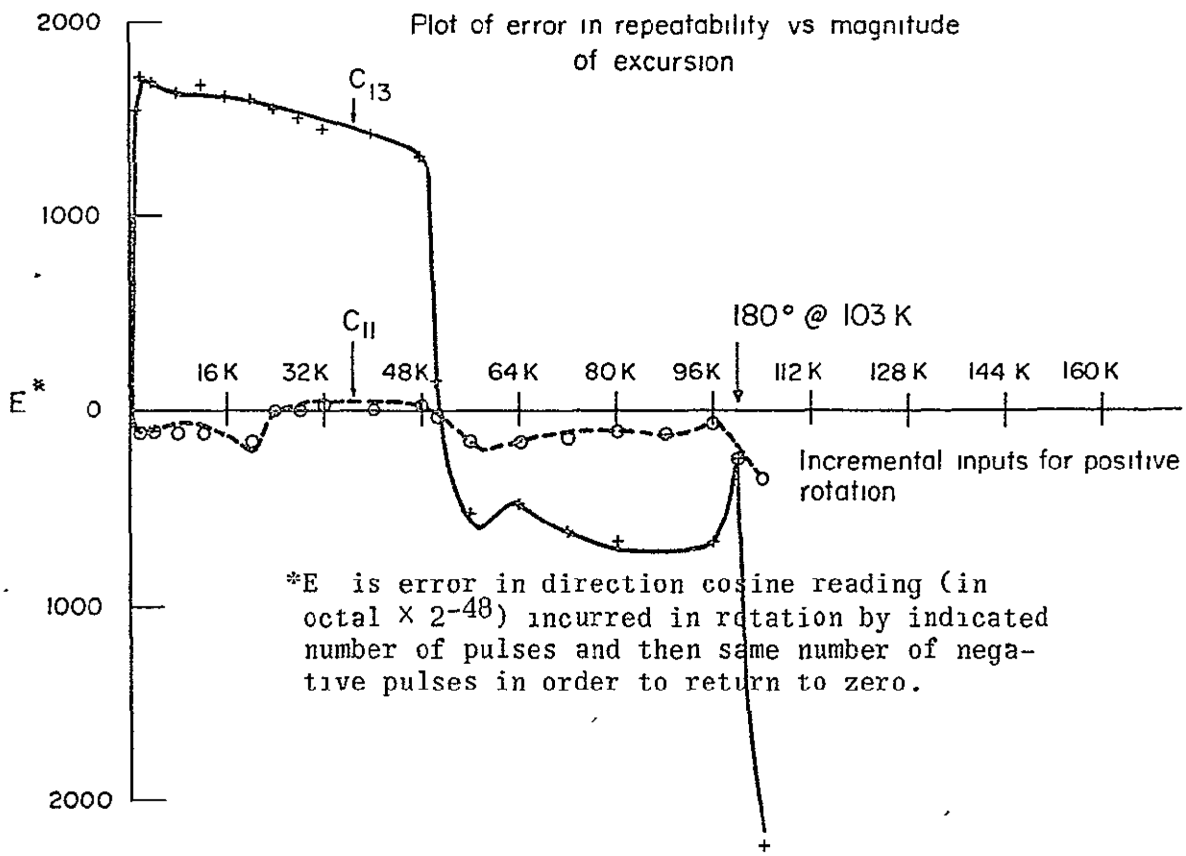


FIG. 5-1 Plot of error in repeatability.

The oscillatory runs consisted of ten cycles from 0 to 1024 pulses and back to determine the effect of extended testing on the maximum error rate that the system initially produced. (See Fig. 5-1.) The result was that the error increased proportionally for as many input pulses as were applied. See Table 5.3

These tests were run to provide a reference to determine the correct operation of the SIMU.

TABLE 5.1

SIMU Simulation Data

Single Axis Values After n(Δθ) Pulses

n	C11			C13		
1,024	037770	000052	165024	000777	165253	027743
2,048	037740	001252	151642	001777	052536	017055
4,096	037600	025245	035227	003772	125673	160345
8,192	037002	124477	036241	007725	073552	133023
32,768	021224	050037	135327	032732	124436	043614
106,496	140140	026754	104267	174423	052267	067173

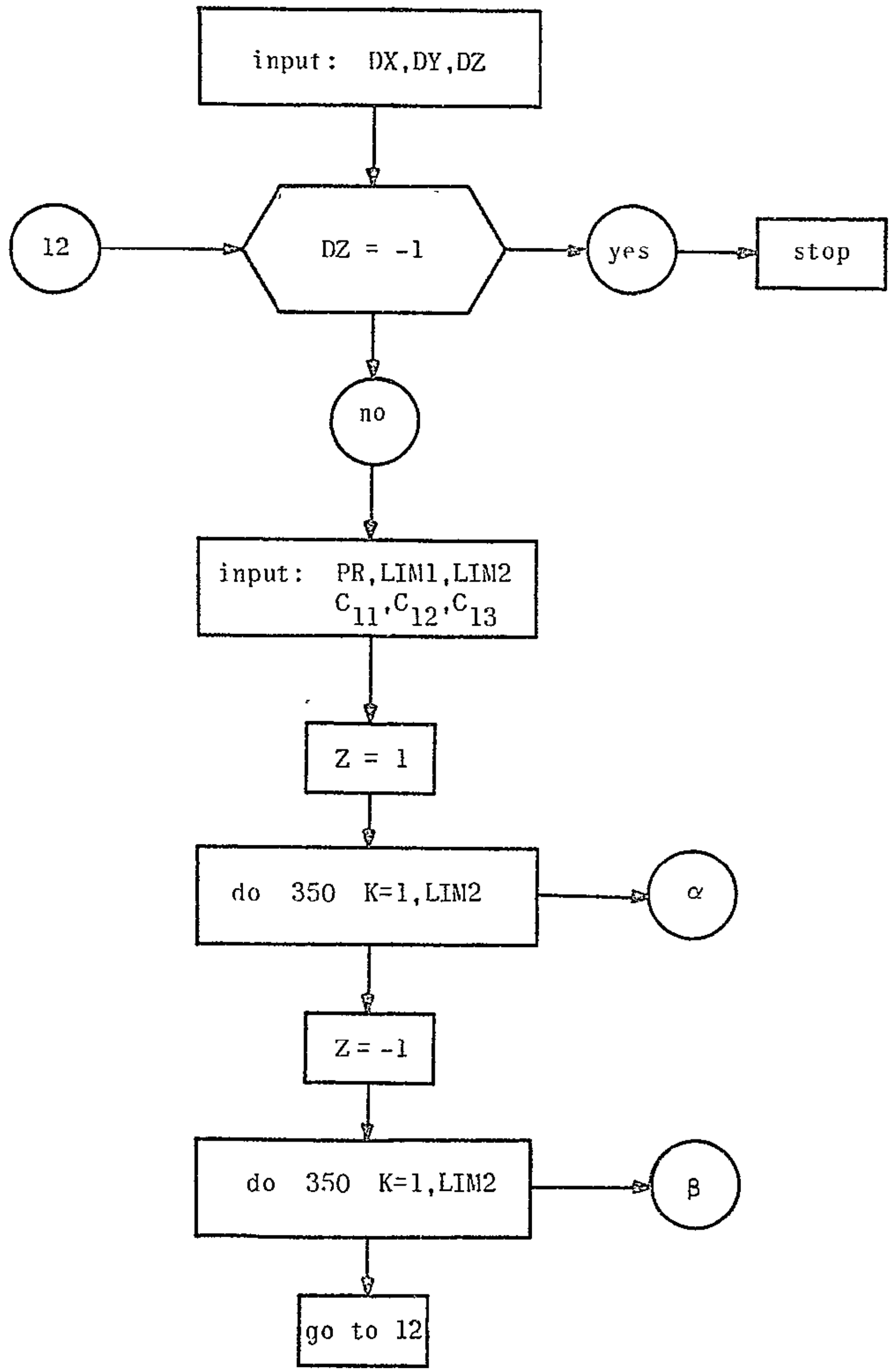
Single Axis Return to Zero Values After n(Δθ) Pulses and n(-Δθ)

n	C11			C13		
1,024	040000	000000	037670	000000	000000	041541
2,048	040000	000000	037666	000000	000000	041727
4,096	040000	000000	037675	000000	000000	041706
8,192	040000	000000	037675	000000	000000	041662
12,288	040000	000000	037663	000000	000000	041651
16,384	040000	000000	037711	000000	000000	041622
32,768	040000	000000	040031	000000	000000	041461
65,536	040000	000000	037631	000000	000000	037303
98,304	040000	000000	037717	000000	000000	037112
106,496	040000	000000	037444	000000	000000	035560

5.2 PROGRAM DESCRIPTION

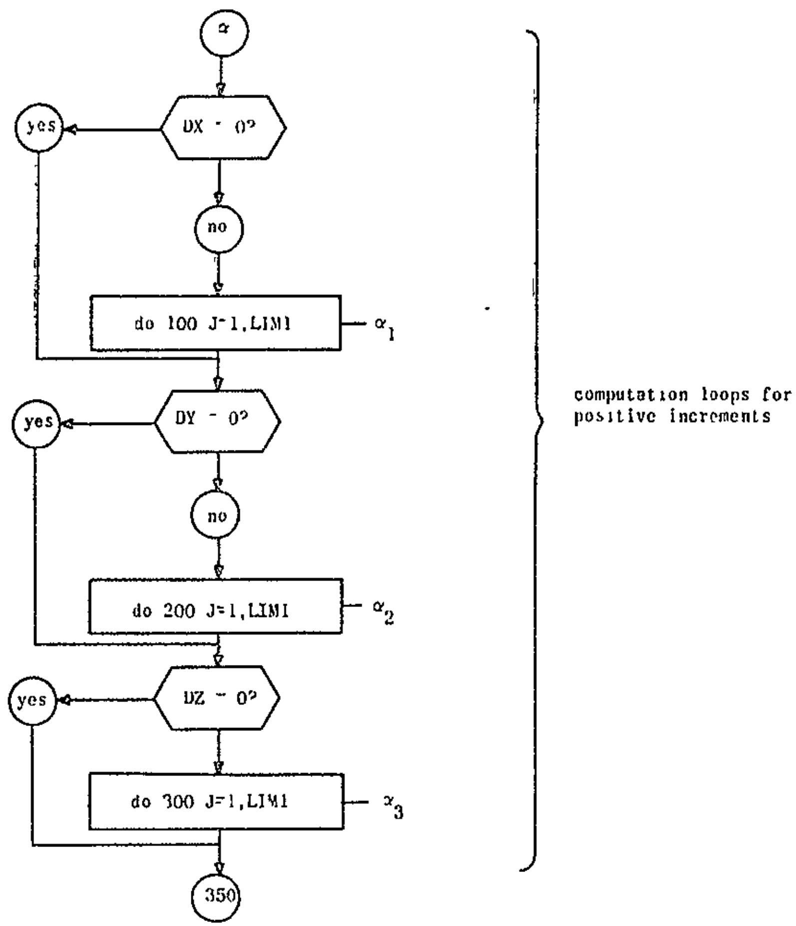
The program, written for the UNIVAC 1108 in Fortran IV, computes new direction cosines from old direction cosines using the truncation-error-free algorithm. See the flow chart (Fig. 5-2) to determine how this simulation was programmed.

The nature of the iterative process is controlled by punched card inputs in the format described below. Provision was made for controlling iteration by selecting angular increments in each of the three axial senses, $\vec{x}$, $\vec{y}$ and $\vec{z}$. Let the three selection variables be DX, DY, DZ . DX=0 means skip an angular increment in this sense. DX$\neq$0 means perform the appropriate iteration for an angular increment in this sense (this angular increment is assumed fixed). Refer to Fig. 5-3.

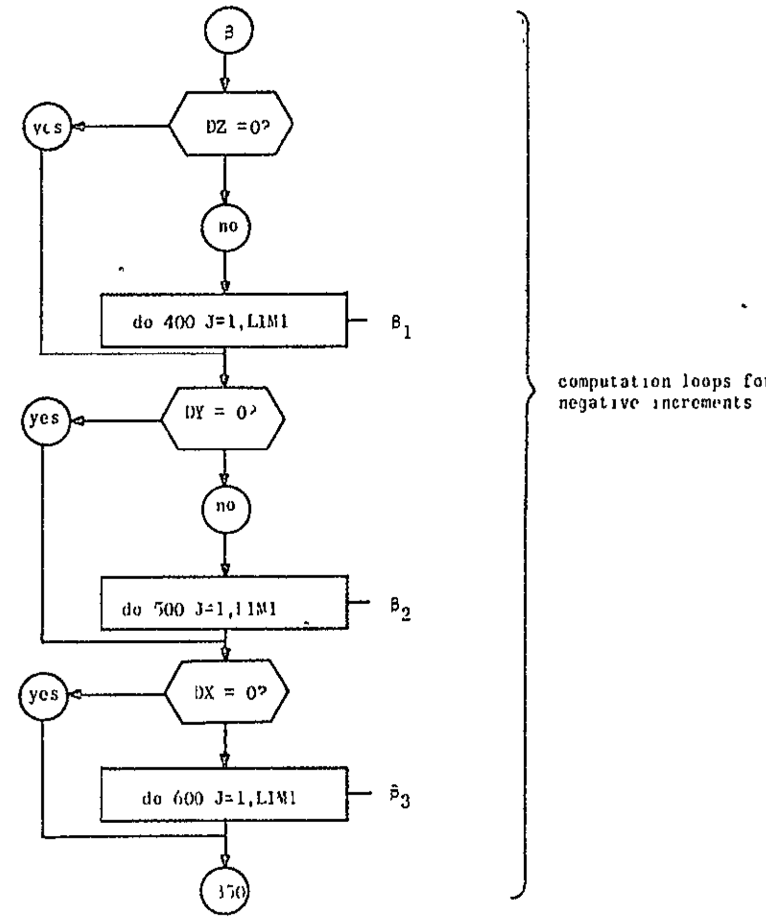


Input and Control Initialization

FIG. 5-2 Flow charts of simulation program.

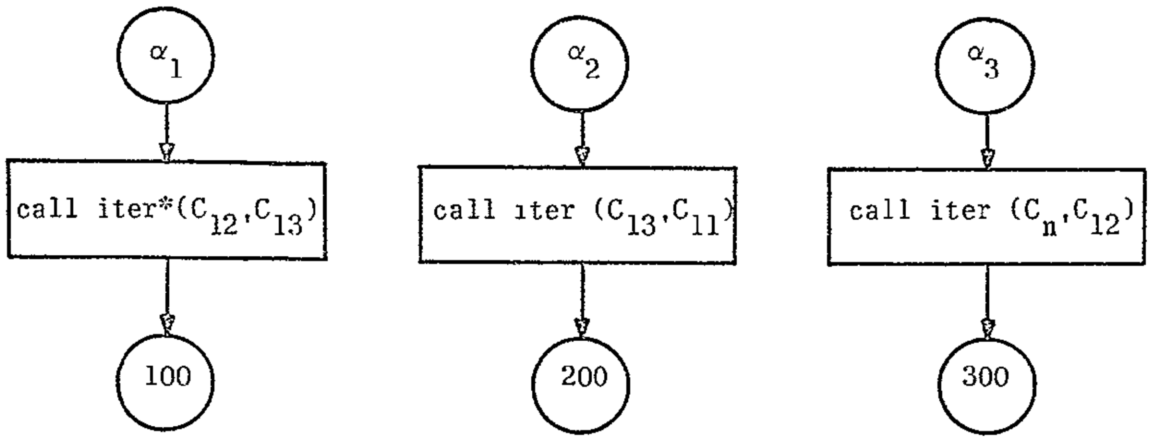


Choice of Angular Increment

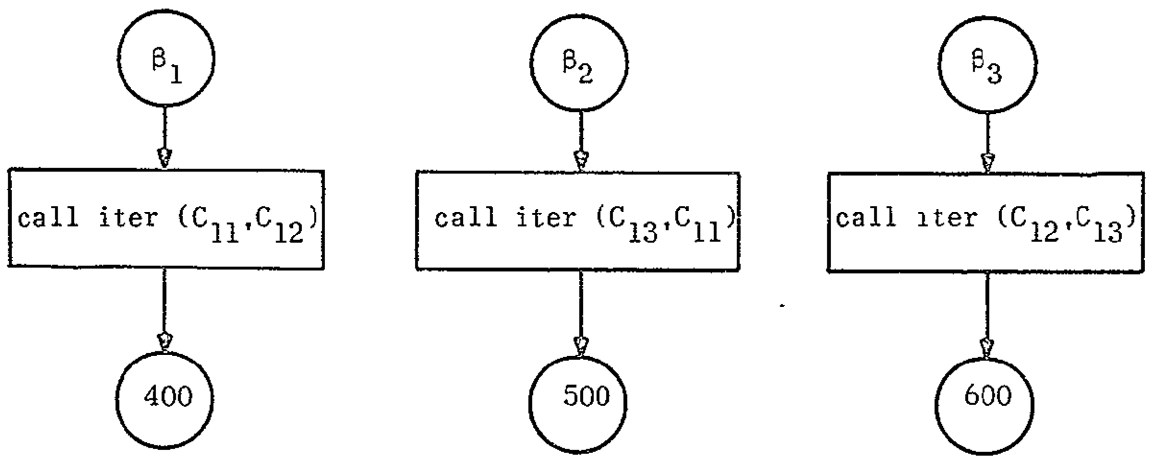


Choice of Angular Increment

FIG. 5-2 Flow charts of simulation program (cont.).



computation for positive increments



computation for negative increments

*ITER is the subroutine that actually performs the computation on the pair of direction cosines that are the arguments in the call statement.

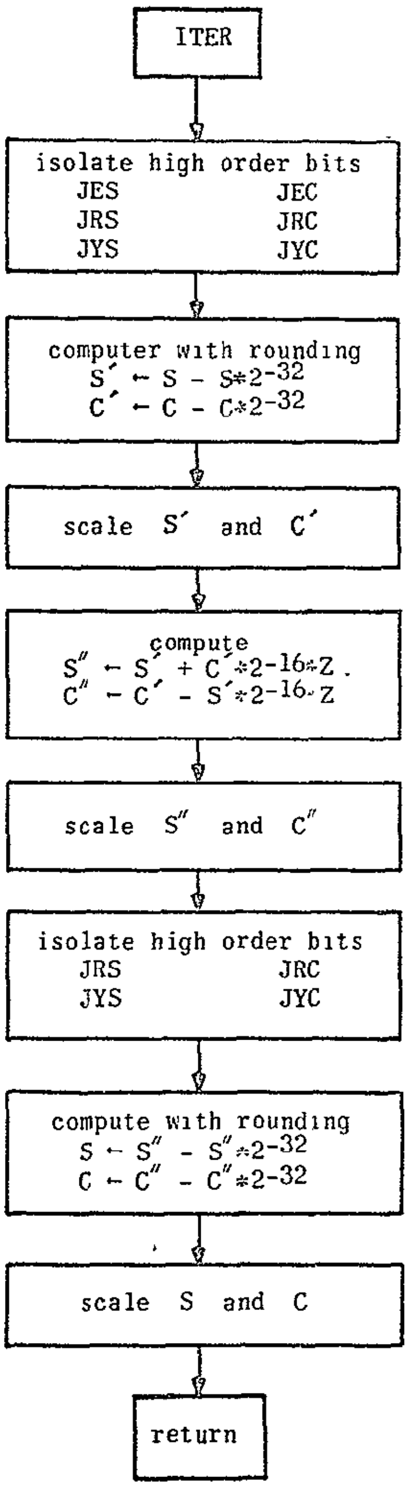
ITER itself calls another routine SCALE which scales the 3 integer variables of each C_{ij} to lie within 0 and $2^{16} - 1$;

The variable Z is set to +1 for positive increments and -1 for negative increments.

Printing is done after each call to iter depending on the control variable PR.

Computation for New Direction Cosines

FIG. 5-2 Flow charts of simulation program (cont.).



arguments: (S,C)

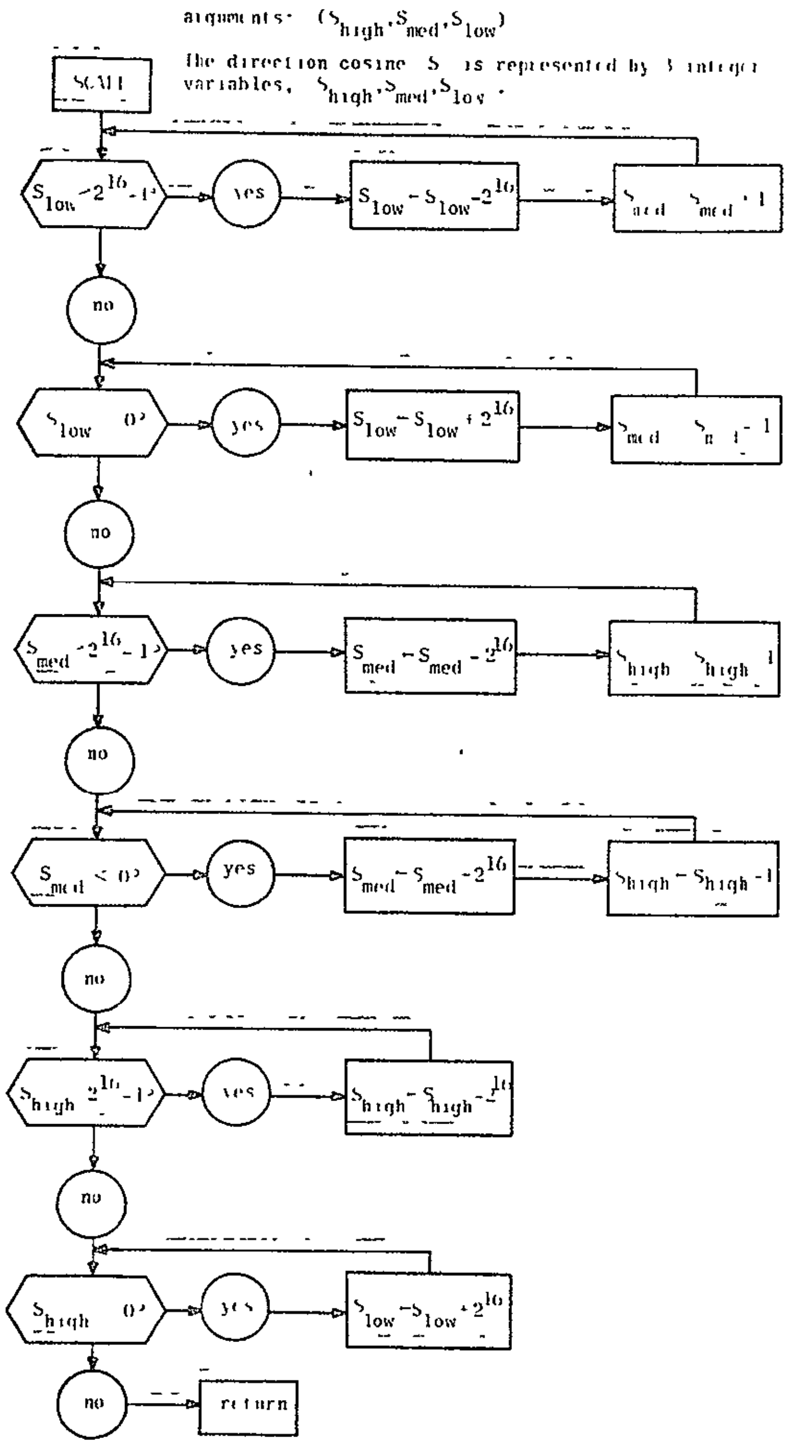
*S represents the three integer variables holding the three 16-bit parts of the 1st direction cosine, and C is used to represent the 2nd. JES, JEC, etc., represent corresponding high order bits for each part.

Computation is achieved by adding or subtracting the parts of S, C, $S*2^{-32}$, $C*2^{-32}$, etc., that are of corresponding order. The high order bits are used to perform rounding and "shifting". See program for details.

S', C', S'', C'' represent intermediary results which are in the same format as S and C.

Actual Computation of New Direction Cosines

FIG. 5-2 Flow charts of simulation program (cont.).



Scaling of 16-Bit Parts of Direction Cosine

FIG. 5-2 Flow charts of simulation program (cont.).

Two further variables, LIM1 and LIM2, control the length of the iterative procedure. Two specific settings of LIM1 and LIM2 are of interest:

(1) LIM1 = 1, LIM2 = K means iterate K times in the first selected axial sense, then K times in the second, etc.

(2) LIM1 = K, LIM2 = 1 means perform K times the sequence {one iteration in the first selected sense, followed by one iteration in the second selected sense} etc.

The program performs the "mirror-image" of the iteration sequence in order to return to the starting point.

One further variable, PR , is provided to control print frequency. If PR=K , then each N^{th} iteration in any selected axial sense forces output of all the direction cosines.

Input variables C11, C12, C13 are provided to enable the user to specify different initial direction cosines. C11, C12, C13 are always entered and printed out as 6 digit octal numbers.

The program performs only integer arithmetic, using one Fortran integer variable to hold the three 16-bit parts of each 48-bit C_{IJ} , as well as intermediary results (see Table 5.4).

Card Column
1

Card #1

DX	DY	DZ
I10	I10	I10

I10 = integer, field width of 10

Card #2

PR	LIM1	LIM2
+ I10	I10	I10

Card #3-5

high order 16 bits c_{ij}	medium order 16 bits c_{ij}	low order 16 bits c_{ij}
θF	θF	θF

C11 on Card 3
C12 on Card 4
C13 on Card 5

θF = octal integer, field width of F

Card 6

DX′	DY′	DZ′
I10	I10	I10

At the end of a sequence specified by one set of data, the program reads a new DX′, DY′, DZ′ . If DX′ = -1 the program stops, otherwise, computation resumes with a new set of data which is assumed to follow card 6, etc.

FIG. 5-3 Card input format.

C	1	0							
1	1	8							
[illegible]	[illegible]	040000	040000	[illegible]	000005	040000	[illegible]	[illegible]	[illegible]
1	[illegible]	177777	140000	[illegible]	000000	040000	000000	10000	[illegible]
2	037777	177776	040000	[illegible]	[illegible]	040000	000001	[illegible]	037774
3	037777	177773	140000	[illegible]	000003	040000	000001	100000	03776[illegible]
4	037777	17777[illegible]	040000	[illegible]	000004	040000	000002	000000	03775[illegible]
5	037777	177763	140000	[illegible]	000002	040000	000002	100000	03772[illegible]
6	077777	177756	040000	[illegible]	000000	040000	000007	[illegible]	0376[illegible]
7	077777	177747	140000	[illegible]	000000	040000	000007	100000	037611
8	037777	177745	040000	[illegible]	000000	040000	000004	000000	03752[illegible]
-1	037777	177747	140000	[illegible]	000000	040000	000007	100000	03761[illegible]
-2	077777	177756	040000	[illegible]	000000	040000	000007	000000	03766[illegible]
-3	077777	177763	140000	[illegible]	000000	040000	000002	100000	03772[illegible]
-4	037777	17777[illegible]	040000	[illegible]	000000	040000	000002	000000	037744
-5	077777	177773	140000	[illegible]	000000	040000	000001	100000	03775[illegible]
-6	037777	177776	040000	[illegible]	000000	040000	000001	000000	03776[illegible]
-7	037777	177777	140000	[illegible]	000000	040000	000000	100001	03777[illegible]
-8	040000	000000	040000	[illegible]	[illegible]	040000	000000	000000	[illegible]

Single axis - 0 to 8 pulses and return to 0 again.

```
CYC:1

PR:8

SEQ:128,:-128,:0

S:0,:0,:040000

C:040000,:0,:040000

=    8=     4=       0=   37520    =  37777= 177740=  40000
=   16=    10=       0=   35240    =  37777= 177600=  40000
=   24=    14=       0=   26760    =  37777= 177340=  40000
=   32=    20=       0=   12500    =  37777= 177000=  40000
=   40=    23= 177777= 166224    =  37777= 176340=  40003
=   48=    27= 177777= 127754    =  37777= 175600=  40013
=   56=    33= 177777=   55504    =  37777= 174740=  40026
=   64=    37= 177776= 165234    =  37777= 174000=  40047
=   72=    43= 177776=   54764    =  37777= 172740=  40100
=   80=    47= 177775= 122514    =  37777= 171600=  40143
=   88=    53= 177774= 144244    =  37777= 170340=  40222
=   96=    57= 177773= 137774    =  37777= 167000=  40320
=  104=    63= 177772= 103524    =  37777= 165340=  40440
=  112=    67= 177771=   15254    =  37777= 163600=  40605
=  120=    73= 177767=    73004    =  37777= 161740=  41002
=  128=    77= 177765= 112534    =  37777= 160000=  41233
=-   8=    73= 177767='  73014    =  37777= 161740=  40777
=-  16=    67= 177771=   15274    =  37777= 163600=  40576
=-  24=    63= 177772= 103554    =  37777= 165340=  40426
=-  32=    57= 177773= 140034    =  37777= 167000=  40304
=-  40=    53= 177774= 144314    =  37777= 170340=  40204
=-  48=    47= 177775= 122574    =  37777= 171600=  40123
=-  56=    43= 177776=   55054    =  37777= 172740=  40057
=-  64=    37= 177776= 165334    =  37777= 174000=  40025
=-  72=    33= 177777=   55614    =  37777= 174740=  40003
=-  80=    27= 177777= 130074    =  37777= 175600=  37767
=-  88=    23= 177777= 166354    =  37777= 176340=  37757
=-  96=    20=       0=   12630    =  37777= 177000=  37753
=-104=    14=       0=   27100    =  37777= 177340=  37753
=-112=    10=       0=   35350    =  37777= 177600=  37753
=-120=     4=       0=   37620    =  37777= 177740=  37753
=-128=     0=       0=   40070    =  40000*       0=  37753
*
```

Single axis - 0 to 128 pulses and return to 0 again.

TABLE 5.2

Results of SIMU Simulation Program for Checkout Runs

```
c98:1

PR:4096

SEO:32768,:-32768,:0

S:0,:0,:040000

C:040000,.0,:040000
```

= 4096=	3772=	125673=	160345	=	37600=	25245=	35227
= 8192=	7725=	73552=	133023	=	37002=	124477=	36241
= 12288=	13561=	1125=	130127	=	35615=	67727=	72357
= 16384=	17256=	164164=	101440	=	34052=	50014=	175554
= 20480=	22562=	35725=	43374	=	31746=	150733=	64506
= 24576=	25637=	174055=	132252	=	27323=	176614=	172114
= 28672=	30437=	67310=	46071	=	24406=	13657=	173402
= 32768=	32732=	124436=	43614	=	21224=	50037=	135327
=- 4096=	30437=	67310=	46071	=	24406=	13657=	173433
=- 8192=	25637=	174055=	132221	=	27323=	176614=	172210
=-12288=	22562=	35725=	43371	=	31746=	150733=	64761
=-16384=	17256=	164164=	101350	=	34052=	50014=	175747
=-20480=	13561=	1125=	130002	=	35615=	67727=	72606
=-24576=	7725=	73552=	132645	=	37002=	124477=	36440
=-28672=	3772=	125673=	160130	=	37600=	25245=	35404
=-32768=	0=	0=	41461	=	40000=	0=	40031

*

Single axis - 0 to 32,768 pulses and return to 0 again.

1	1	C							
1		1							
-	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
1	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
2	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	040000	[illegible]	[illegible]	[illegible]
3	040000	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
4	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
5	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
6	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
7	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
8	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
1	[illegible]	177777	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
2	[illegible]	177776	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
3	[illegible]	177777	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
4	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
5	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
6	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
7	[illegible]	177747	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
8	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
-1	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
-2	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
-3	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
-4	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
-5	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
-6	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
-7	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
-8	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
-1	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
-2	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
-3	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
-4	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
-5	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
-6	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
-7	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
-8	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

Two axis x,y sequentially - 0 to 8 pulses and return to 0 again.

TABLE 5.2 (cont.)

0	1	1							
1	P	1							
0	040000	000000	040000	000000	000000	040000	000000	000000	040000
1	037777	177777	140000	000000	000000	040000	000000	100000	037777
2	037777	177776	040000	000000	000000	040000	000001	000000	037774
3	037777	177773	140000	000000	000000	040000	000001	100000	037760
4	037777	177770	040000	000000	000000	040000	000002	000000	037750
5	037777	177763	140000	000000	000000	040000	000002	100000	037720
6	037777	177755	040000	000000	000000	040000	000003	000000	037664
7	037777	177747	140000	000000	000000	040000	000003	100000	037611
8	037777	177740	040000	000000	000000	040000	000004	000000	037520
1	037777	177737	140000	177777	100000	040100	000004	000000	037520
2	037777	177736	040000	177777	000000	040202	000004	000000	037520
3	037777	177737	140000	177776	100000	040313	000004	000000	037520
4	037777	177730	040000	177776	000000	040424	000004	000000	037520
5	037777	177723	140000	177775	100000	040556	000004	000000	037520
6	037777	177715	040000	177775	000000	040736	000004	000000	037520
7	037777	177707	140000	177774	100000	041066	000004	000000	037520
8	037777	177700	040000	177774	000000	[illegible]	000004	000000	037520
-1	037777	177707	140000	177774	100000	041057	000004	000000	037520
-2	037777	177715	040000	177775	000000	040704	000004	000000	037520
-3	037777	177727	140000	177775	100000	040545	000004	000000	037520
-4	037777	177730	040000	177776	000000	040420	000004	000000	037520
-5	037777	177733	140000	177776	100000	040303	000004	000000	037520
-6	037777	177735	040000	177777	000000	040174	000004	000000	037520
-7	037777	177737	140000	177777	100000	040071	000004	000000	03752
-8	037777	177740	040000	000000	000000	037776	000004	000000	037520
-1	037777	177747	140000	000000	000000	037770	000003	100000	03761
-2	037777	177756	040000	000000	000000	037770	000003	000000	037660
-3	037777	177763	140000	000000	000000	037770	000002	100000	037720
-4	037777	177770	040000	000000	000000	037770	000002	000000	037744
-5	037777	177773	140000	000000	000000	037773	000001	100000	037760
-6	037777	177776	040000	000000	000000	037770	000001	000000	037765
-7	037777	177777	140000	000000	000000	037770	000000	100000	03777
-8	040000	000000	040000	000000	000000	037770	000000	000000	037770

Two axis y,z sequentially - 0 to 8 pulses and return to 0 again.

1	n	1							
1	=	1							
0	040000	000000	040000	000000	000000	040000	000000	000000	04000
1	040000	000000	040000	000000	000000	040000	000000	000000	04000
2	040000	000000	040000	000000	000000	040000	000000	000000	04000
3	040000	000000	040000	000000	000000	040000	000000	000000	040000
4	040000	000000	040000	000000	000000	040000	000000	000000	040000
5	040000	000000	040000	000000	00000	040000	000000	000000	040000
6	040000	000000	040000	000000	000000	040000	000000	000000	040000
7	040000	000000	040000	000000	000000	040000	000000	000000	040000
8	040000	000000	040000	000000	000000	040000	000000	000000	040000
1	037777	177777	140000	177777	100000	040003	000000	000000	040000
2	037777	177776	040000	177777	000000	040002	000000	000000	040000
3	037777	177773	140000	177776	100000	040010	000000	000000	040000
4	037777	177770	040000	177776	000000	040024	000000	000000	04000
5	037777	177763	140000	177775	100000	040050	000000	000000	040000
-4	037777	177770	040000	177776	000000	040020	000000	000000	040000
-5	037777	177773	140000	177776	100000	040003	000000	000000	040000
-6	037777	177775	040000	177777	000000	037774	000000	000000	040000
-7	037777	177777	140000	177777	100000	037771	000000	000000	040000
-8	040000	000000	040000	000000	000000	037770	000000	000000	040000
-1	040000	000000	040000	000000	000000	037770	000000	000000	040000
-2	040000	000000	040000	000000	000000	037770	000000	000000	040000
-3	040000	000000	040000	000000	000000	037770	000000	000000	040000
-4	040000	000000	040000	000000	000000	037773	000000	000000	040000
-5	040000	000000	040000	000000	000000	037770	000000	000000	04000
-6	040000	000000	040000	000000	000000	037770	000000	000000	040000
-7	040000	000000	040000	000000	000000	037770	000000	000000	04000
-8	040000	000000	040000	000000	000000	037770	000000	000000	04000

Two axis x,z sequentially - 0 to 8 pulses and return to 0 again.

TABLE 5.2 (cont.)

1	1	1							
1	8	1							
0	040000	000000	040000	000000	000000	040000	000000	000000	040000
1	040000	000000	040000	000000	000000	040000	000000	000000	040000
2	040000	000003	040000	000000	000000	040000	000000	000000	040000
3	040000	000000	040000	000000	000000	040000	000000	000000	040000
4	040000	000000	040000	000000	000000	040000	000000	000003	040000
5	040000	000000	040000	000000	000000	040000	000000	000000	040000
6	040000	000000	040000	000000	000003	040003	000000	000000	040000
7	040000	000000	040000	000000	000000	040000	000000	000000	040000
8	040000	000000	040000	000000	000000	040000	000000	000000	040000
1	037777	177777	140000	000000	000000	040000	000000	100000	037777
2	037777	177776	040000	000000	000000	040000	000001	000000	037774
3	037777	177773	140000	000000	000000	040000	000001	100000	037765
4	037777	177770	040000	000000	000000	040000	000002	000000	037750
5	037777	177763	140000	000000	000000	040000	000002	100003	037727
6	037777	177756	040000	000000	000000	040000	000007	000000	037664
7	037777	177747	140000	000000	000000	040003	000003	100000	037611
-7	037777	177737	140000	177777	100000	040571	000004	000000	037520
-8	037777	177740	040000	000000	000000	037770	000004	000000	037520
-1	037777	177747	140000	000000	000005	037770	000003	100000	037610
-2	037777	177756	040000	000000	000000	037773	000003	000000	037660
-3	037777	177763	140000	000000	000000	037770	000002	100000	037720
-4	037777	177770	040000	000000	000000	037770	000002	000000	037744
-5	037777	177777	140000	000000	000000	037770	000001	100000	037760
-6	037777	177776	040000	000000	000000	037770	000001	000000	037765
-7	037777	177777	140000	000000	000005	037770	000000	100000	037770
-8	040000	000000	040000	000000	000000	037770	000000	000000	037770
-1	040000	000000	040000	000000	000000	037770	000000	000000	037770
-2	040000	000000	040000	000000	000000	037770	000000	000000	037770
-3	040000	000000	040000	000000	000000	037770	000000	000000	037770
-4	040000	000000	040000	000000	000000	037770	000000	000000	037770
-5	040000	000000	040000	000000	000000	037770	000000	000000	037770
-6	040000	000000	040000	000000	000000	037770	000000	000000	037770
-7	040000	000000	040000	000000	000000	037770	000000	000000	03777
-8	040000	000000	040000	000000	000000	037770	000000	000000	037770

Three axis x,y,z sequentially - 0 to 8 pulses and return to 0 again.

1	1	[illegible]							
1	1	[illegible]							
0	040000	000000	040000	000000	000000	040003	000000	000000	040000
1	040000	000003	040000	000000	000000	040000	000000	000000	040000
1	037777	177777	140000	000000	000000	040000	000000	100000	037777
2	037777	177777	140000	000000	000001	040000	000000	100000	037775
2	037777	177775	040000	000000	000001	040000	000001	000000	037777
3	037777	177776	140000	000000	000003	040000	000001	000000	037765
3	037777	177773	140000	000000	000003	040000	000001	100000	037757
4	037777	177773	140000	000000	000006	040000	000001	100000	037745
4	037777	177770	040000	000000	000006	040000	000002	000000	037730
5	037777	177770	140000	000000	000012	040000	000007	000000	037710
5	037777	177763	140000	000000	000012	040003	000002	100000	037663
-2	037777	177756	040000	000000	000017	040000	000003	000000	037566
-3	037777	177753	140000	000000	000017	040000	000002	100000	037624
-3	037777	177763	140000	000000	000012	040000	000002	100000	037656
-4	037777	177770	040000	000000	000012	040000	000002	000003	037700
-4	037777	177770	040000	000000	000006	040000	000007	000000	037720
-5	037777	177773	140000	000000	000006	040000	000001	100000	037734
-5	037777	177773	140000	000000	000003	040000	000001	100000	037744
-6	037777	177776	040000	000000	000003	040000	000001	000000	037752
-6	037777	177775	040000	000000	000001	040000	000001	000000	037755
-7	037777	177777	140000	000000	000001	040000	000000	100000	037760
-7	037777	177777	140000	000000	000000	040000	000000	100003	037760
-8	040000	000000	040000	000000	000000	040000	000000	000000	037760
-8	040000	000000	040000	000000	000000	040000	000000	000000	037760

Two axis $\Delta x, \Delta y$ incrementally - 0 to 8 pulses and return to 0 again.

TABLE 5.2 (cont.)

0	1	1							
1	1	8							
0	040000	000007	040000	000000	000000	040002	000000	000000	040000
1	037777	177777	140000	000000	000000	040000	000000	100001	037777
1	037777	177777	040000	177777	100000	040001	000000	100000	037777
2	037777	177775	140000	177777	100000	040001	000001	000000	037777
2	037777	177774	040000	177777	000000	040007	000001	000000	037777
3	037777	177771	140000	177777	000000	040007	000001	100007	037761
3	037777	177767	040000	177776	100000	040026	000001	100000	037760
4	037777	177763	140000	177776	100000	040026	000017	000000	037737
4	037777	177760	040000	177776	000000	040062	000002	000000	037737
5	037777	177753	140000	177776	000000	040062	000002	100000	03766
5	037777	177747	040000	177775	100000	040137	000002	100000	037665
6	037777	177741	140000	177775	100000	040137	000003	000000	037575
7	037777	177734	040000	177775	000000	040241	000007	000005	037573
7	037777	177725	140000	177775	000000	040241	000003	100003	037455
7	037777	177717	040000	177774	100000	040374	000003	100000	037455
8	037777	177707	140000	177774	100000	040374	000004	000000	037304
8	037777	177700	040000	177774	000000	040564	000004	000000	037304
-1	037777	177707	140000	177774	100000	040373	000074	000000	037374
-1	037777	177717	040000	177774	100000	040373	000000	100000	037455
-2	037777	177725	140000	177775	000000	040237	000000	100000	037457
-2	037777	177734	040000	177775	000000	040237	000003	000000	037573
-3	037777	177741	140000	177775	100000	040134	000003	000000	037577
-3	037777	177747	040000	177775	100000	040134	000002	100000	037667
-4	037777	177753	140000	177776	000000	040056	000002	100000	037667
-4	037777	177757	040000	177776	000000	040056	000002	000000	037725
-5	037777	177763	140000	177776	100000	040021	000002	000000	037727
-5	037777	177767	040000	177776	100000	040021	000001	100000	037757
-6	037777	177771	140000	177777	000000	040001	000001	100000	037757
-6	037777	177774	040000	177777	000000	040001	000001	000000	037765
-7	037777	177775	140000	177777	100000	037772	000001	000000	037765
-7	037777	177777	040000	177777	100000	037772	000000	100000	037777
-8	037777	177777	140000	000000	000000	037770	000000	100000	037777
-8	040000	000000	040000	000000	000000	037770	000000	000000	037777

Two axis $\Delta y, \Delta z$ incrementally - 0 to 8 pulses and return to 0 again.

1		1							
1	1	[illegible]							
0	040000	000000	040000	000000	000000	040000	000000	000000	040000
1	040000	000000	040000	000000	000000	040000	000000	000000	040000
1	037777	177777	140000	177777	100000	040000	000000	000000	040000
2	037777	177777	140000	177777	100000	040000	000000	000001	040000
2	037777	177775	140000	177777	000000	040002	000000	000001	040000
3	037777	177776	140000	177777	000000	040006	000000	000003	040000
3	037777	177777	140000	177775	100000	040014	000000	000003	040000
4	037777	177773	140000	177776	100000	040024	000000	000006	040000
4	037777	177770	040000	177776	000000	040040	000000	000006	040000
5	037777	177770	040000	177776	000000	040060	000000	000012	040000
5	037777	177753	140000	177775	100000	040104	000000	000012	040000
6	037777	177763	140000	177775	100000	040134	000000	000017	040000
6	037777	177756	040000	177775	000000	040172	000000	000017	040000
-2	037777	177756	040000	177775	000000	040166	000000	000017	040000
-3	037777	177763	140000	177775	100000	040127	000000	000017	040000
-3	037777	177763	140000	177775	100000	040075	000000	000012	040000
-4	037777	177770	040000	177776	000000	040050	000000	000012	040000
-4	037777	177770	040000	177776	000000	040030	000000	000006	040000
-5	037777	177773	140000	177776	100000	040013	000000	000006	040000
-5	037777	177773	140000	177776	100000	040001	000000	000003	040000
-6	037777	177776	040000	177777	000000	037772	000000	000003	040000
-6	037777	177776	040000	177777	000000	037766	000000	000001	040000
-7	037777	177777	140000	177777	100000	037763	000000	000001	040000
-7	037777	177777	140000	177777	100000	037761	000000	000000	040000
-8	040000	000000	040000	000000	000000	037760	000000	000006	040000
-8	040000	000000	040000	000000	000000	037760	000000	000006	040000

Two axis $\Delta x, \Delta z$ incrementally - 0 to 8 pulses and return to 0 again.

TABLE 5.2 (cont.)

1	1	1							
1	1	2							
0	040000	000000	040000	000000	000000	040000	000000	000000	040000
1	040000	000000	040000	000000	000000	040000	000000	000000	040000
1	037777	177777	140000	000000	000000	040000	000000	100000	037777
1	037777	177777	040000	177777	100000	040001	000000	100000	037777
2	037777	177777	040000	177777	100001	040001	000000	100001	037776
2	037777	177775	137776	177777	100001	040001	000001	000001	037771
2	037777	177774	040000	177777	000001	040037	000001	000001	037771
3	037777	177774	040000	177777	000003	040013	000001	000003	037760
3	037777	177771	137772	177777	000003	040013	000001	100003	037750
3	037777	177767	040000	177776	100003	040032	000001	100003	037750
4	037777	177767	040000	177776	100006	040042	000001	100006	03774
4	037777	177763	177764	177776	100006	040042	000002	000006	037710
4	037777	177760	040000	177775	000005	040075	000002	000006	037710
5	037777	177760	040000	177776	000012	040115	000002	000012	037670
-3	037777	177747	040000	177775	100017	040216	000002	100017	037566
-3	037777	177747	040000	177775	100012	040154	000002	100012	037615
-4	037777	177757	137764	177776	000012	040106	000002	100012	037615
-4	037777	177760	040000	177776	000012	040106	000007	000012	037660
-4	037777	177760	040000	177776	000005	040066	000002	000006	037770
-5	037777	177763	137734	177776	100006	040031	000002	000006	037770
-5	037777	177767	040000	177775	100006	040031	000001	100036	037737
-5	037777	177767	040000	177776	100003	040017	000001	100003	037777
-6	037777	177771	137772	177777	000003	037777	000001	100003	037737
-6	037777	177774	040000	177777	000003	037777	000001	000003	037751
-6	037777	177774	040000	177777	000001	037773	000001	000001	037755
-7	037777	177775	137776	177777	100001	037764	000001	000001	037750
-7	037777	177777	040000	177777	100001	037764	000000	100001	037760
-7	037777	177777	040000	177777	100000	037762	000000	100000	037760
-8	037777	177777	140000	000000	000000	037760	000000	100000	037760
-8	040000	000000	040000	000000	000000	037760	000000	000000	037760
-8	040000	000000	040000	000000	000000	037760	000000	000000	037760

Three axis $\Delta x, \Delta y, \Delta z$ incrementally - 0 to 8 pulses and return to 0 again.

1	1	0							
16	128	1							
0	040000	000000	040000	000000	000000	040000	000000	000000	040000
16	040000	000000	040000	000000	000000	040000	000000	000000	040000
32	040000	000000	040000	000000	000000	040000	000000	000000	040000
48	040000	000000	040000	000000	000000	040000	000000	000000	040000
64	040000	000000	040000	000000	000000	040000	000000	000000	040000
80	040000	000000	040000	000000	000000	040000	000000	000000	040000
96	040000	000000	040000	000000	000000	040000	000000	000000	040000
112	040000	000000	040000	000000	000000	040000	000000	000000	040000
128	040000	000000	040000	000000	000000	040000	000000	000000	040000
16	037777	177600	040000	000000	000000	040000	000010	000000	035240
32	037777	177000	040000	000000	000000	040000	000020	000000	012500
48	037777	175600	040013	000000	000000	040000	000027	177777	127754
64	037777	174000	040047	000000	000000	040000	000037	177776	165234
-32	037777	167000	040334	000000	000000	040000	000057	177773	140334
-64	037777	171600	040127	000000	000000	040000	000047	177775	122574
-54	037777	174000	040325	000000	000000	040000	000037	177776	165334
-80	037777	175600	037767	000000	000000	040000	000027	177777	170074
-96	037777	177000	037753	000000	000000	040000	000027	000000	012630
-112	037777	177600	037753	000000	000000	040000	000010	000000	035350
-128	040000	000000	037753	000000	000000	040000	000000	000000	04007
-16	040000	000000	037753	000000	000000	040000	000000	000000	04007
-32	040000	000000	037753	000000	000000	040000	000000	000000	040070
-48	040000	000000	037753	000000	000000	040000	000000	000000	040070
-64	040000	000000	037753	000000	000000	040000	000000	000000	040070
-80	040000	000000	037753	000000	000000	040000	000000	000000	04007
-96	040000	000000	037753	000000	000000	040000	000000	000000	040070
-112	040000	000000	037753	000000	000000	040000	000000	000000	040070
-128	040000	000000	037753	000000	000000	040000	000000	000000	040070

Two axis, x,y sequentially - 0 to 128 pulses and return to 0 again.

TABLE 5.2 (cont.)

0	1	1							
16	128	1							
0	040000	000000	040000	000000	000000	040000	000000	000000	040000
16	037777	177600	040000	000000	000000	040000	000010	000000	035247
32	037777	177000	040000	000000	000000	040000	000020	000000	012500
48	037777	175600	040013	000000	000000	040000	000027	177777	127754
64	037777	174000	040047	000000	000000	040000	000037	177776	165234
80	037777	171600	040143	000000	000000	040000	000047	177775	172514
96	037777	167000	040320	000000	000000	040000	000057	177773	177774
112	037777	163600	040625	000000	000000	040000	000067	177771	015254
128	037777	160000	041233	000000	000000	040000	000077	177765	112534
16	037777	157600	041333	177770	000004	042520	000077	177765	112534
32	037777	157000	041637	177760	000010	055240	000077	177765	112534
48	037777	155600	042341	177750	000014	147760	000077	177765	112534
64	037777	154000	043271	177740	000021	112300	000077	177765	112534
80	037777	151600	044461	177730	000026	155220	000077	177765	112534
96	037777	147000	046132	177720	000034	137740	000077	177765	112534
112	037777	143500	050114	177710	000043	062460	000077	177765	112534
128	037777	140000	052435	177700	000052	165200	000077	177765	112534
-16	037777	143500	050075	177710	000043	062440	000077	177765	112534
-32	037777	147000	045676	177720	000034	137700	000077	177765	112534
-48	037777	151600	044411	177730	000026	155140	000077	177765	112534
-64	037777	154000	043267	177740	000021	112400	000077	177765	112534
-80	037777	155600	042245	177750	000014	147640	000077	177765	112534
-96	037777	157000	041526	177760	000010	065100	000077	177765	112534
-112	037777	157600	041215	177770	000004	042340	000077	177765	112534
-128	037777	160000	041117	000000	000000	037600	000077	177765	112534
-16	037777	163600	040455	000000	000000	037600	000067	177771	015274
-32	037777	167000	040163	000000	000000	037600	000057	177773	140034
-48	037777	171600	040002	000000	000000	037600	000047	177775	172574
-64	037777	174000	037704	000000	000000	037600	000037	177776	165334
-80	037777	175600	037646	000000	000000	037600	000027	177777	170074
-96	037777	177000	037612	000000	000000	037600	000020	000000	01262[illegible]
-112	037777	177600	037637	000000	000000	037600	000010	000000	03535[illegible]
-128	040000	000000	037637	000000	000000	037600	000000	000000	04007[illegible]

Two axis, y,z sequentially - 0 to 128 pulses and return to 0 again.

1	0	1							
16	128	1							
0	040000	000000	040000	000000	000000	040000	000000	000000	040000
16	040000	000000	040000	000000	000000	040000	000000	000000	040000
32	040000	000000	040000	000000	000000	040000	000000	000000	040000
48	040000	000000	040000	000000	000000	040000	000000	000000	040000
64	040000	000000	040000	000000	000000	040000	000000	000000	040000
80	040000	000000	040000	000000	000000	040000	000000	000000	040000
96	040000	000000	040000	000000	000000	040000	000000	000000	040000
112	040000	000000	040000	000000	000000	040000	000000	000000	040000
128	040000	000000	040000	000000	000000	040000	000000	000000	040000
16	037777	177600	040000	177770	000000	042520	000000	000000	040000
32	037777	177000	040000	177760	000000	055240	000000	000000	040000
48	037777	175600	040013	177750	000000	147760	000000	000000	040000
64	037777	174000	040047	177740	000001	112300	000000	000000	040000
-48	037777	171600	040123	177730	000002	155140	000000	000000	040000
-64	037777	174000	040025	177740	000001	112400	000000	000000	040000
-80	037777	175600	037767	177750	000000	147640	000000	000000	040000
-96	037777	177000	037757	177760	000000	055100	000000	000000	040000
-112	037777	177600	037753	177770	000000	042340	000000	000000	040000
-128	040000	000000	037753	000000	000000	037600	000000	000000	040000
-16	040000	000000	037753	000000	000000	037600	000000	000000	040000
-32	040000	000000	037753	000000	000000	037600	000000	000000	040000
-48	040000	000000	037753	000000	000000	037600	000000	000000	040000
-64	040000	000000	037753	000000	000000	037600	000000	000000	040000
-80	040000	000000	037753	000000	000000	037600	000000	000000	040000
-96	040000	000000	037753	000000	000000	037600	000000	000000	040000
-112	040000	000000	037753	000000	000000	037600	000000	000000	040000
-128	040000	000000	037753	000000	000000	037600	000000	000000	040000

Two axis, x,z sequentially - 0 to 128 pulses and return to 0 again.

TABLE 5.2 (cont.)

1	1	1							
16	12-	1							
0	040000	000000	040000	000000	000000	040000	000000	000000	040000
16	040000	000000	040000	000000	000000	040000	000000	000000	04000[illegible]
32	040000	000000	040000	000000	000000	040000	000000	000000	040000
48	040000	000000	040000	000000	000000	040000	000000	000000	040000
64	040000	000000	040000	000000	000000	040000	000000	000000	040000
80	040000	000000	040000	000000	000000	040000	000000	000000	040000
96	040000	000000	040000	000000	000000	040000	000000	000000	040000
112	040000	000000	040000	000060	000000	040000	000000	000000	040000
128	040000	000000	040000	000000	000000	040000	000000	000000	040000
16	077777	177500	040000	000000	000000	040000	000010	000000	035240
32	077777	177200	040000	000000	000000	040000	000020	000000	012500
48	077777	175[illegible]00	040017	000000	000000	040000	000027	177777	1[illegible]77[illegible]4
64	077777	174000	040047	000000	000000	040000	000037	177776	165234
80	037777	171500	040143	000000	000000	040000	000047	177775	172514
96	037777	167000	040320	000000	000000	040000	000057	177773	137774
112	077777	163600	040[illegible]05	000000	000000	040000	000067	177771	015254
-112	077777	157300	041216	177770	000004	042340	000077	177765	112534
-128	037777	160000	041112	000000	000000	077600	000077	177765	112534
-16	037777	163600	040455	000000	000000	077600	000067	177771	015274
-32	037777	167300	040163	000000	000000	037600	000057	177773	140034
-48	037777	171500	040002	000000	000000	037500	000047	177775	122574
-64	037777	174000	037704	000000	000000	037600	000037	17777[illegible]	155374
-80	037777	175600	037646	000000	000000	037600	000027	177777	130074
-96	037777	177000	037632	000000	000000	037500	000020	000000	012630
-112	037777	177500	037632	000000	000000	037600	000010	000000	035350
-128	040000	000000	037632	000000	000000	037600	000000	000000	040070
-16	040000	000000	037632	000000	000000	037600	000000	000000	040070
-32	040000	000000	037632	000000	000000	037600	000000	000000	04007
-48	040000	000000	037632	000000	000000	037600	000000	000000	040070
-64	040000	000000	037632	000000	000000	037600	000000	000000	040070
-80	040000	000000	037632	000000	000000	037600	000000	000000	040070
-96	040000	000000	037632	000000	000000	037600	000000	000000	040070
-112	040000	000000	037632	000000	000000	037600	000000	000000	040070
-128	040000	000000	037632	000000	000000	037500	000000	000000	040070

Three axis, x,y,z sequentially - 0 to 128 pulses and return to 0 again.

1	1	0							
16	1	128							
0	040000	000000	040000	000000	000000	040000	000000	000000	040000
16	037777	177617	140000	000000	000170	040000	000007	100000	033261
16	037777	177600	040000	000000	000170	040000	000010	000000	032700
32	037777	177037	140002	000000	000760	037776	000017	077777	170145
32	037777	177000	040003	000000	000760	037776	000017	177777	166205
48	037777	175657	140027	000000	002150	037751	000027	077777	026465
48	037777	175600	040031	000000	002150	037751	000027	177777	022140
64	077777	174077	140115	000000	003740	03766[illegible]	000037	077775	126400
64	037777	174000	040127	000000	003740	037650	000037	177775	116505
80	037777	171717	140301	000000	006130	03747[illegible]	000047	077773	027722
80	037777	171600	040313	000000	006130	03747[illegible]	000047	177773	013445
96	037777	167137	140627	000000	010720	037135	000057	077767	072645
96	037777	167000	040651	000000	010720	037136	000057	177767	051000
-16	077777	163500	041[illegible]	000000	014110	035375	000057	177762	007300
-32	077777	167000	040536	000000	011060	037130	000057	177767	027100
-32	037777	167000	040536	000000	010720	037152	000057	177767	051100
-48	037777	171500	040274	000000	006250	037475	000047	177772	177200
-48	077777	171600	040274	000000	006130	037510	000047	177773	013600
-64	077777	174000	040077	000000	004040	037674	000037	177775	106700
-64	077777	174000	040077	000000	003740	037701	000037	177775	116700
-80	037777	175600	040033	000000	002230	03777[illegible]	000027	177777	0160[illegible]
-80	037777	175600	040003	000000	002150	03777[illegible]	000027	177777	022405
-96	037777	177000	037755	000000	001020	040017	000017	177777	164505
-96	037777	177300	037755	000000	000760	040020	000017	177777	165505
-112	037777	177603	037752	000000	000710	040[illegible]23	000010	000000	032554
-112	037777	177600	037750	000000	000170	040023	000010	000000	033154
-128	040000	000000	037752	000000	000000	040023	000000	000000	040214
-128	040000	000000	037752	000000	000000	040023	000000	000000	040214

Two axis $\Delta x, \Delta y$ incrementally - 0 to 128 pulses and return to 0 again.

TABLE 5.2 (cont.)

0	1	1							
16	1	128							
0	040000	000000	040000	000000	000000	040000	000000	000000	040000
16	037777	177417	[illegible]	177770	100000	044470	000010	000000	032710
16	037777	177400	040000	177770	000000	045450	000010	000000	032710
32	037777	176037	140004	177760	100000	107560	000017	177777	166227
32	037777	176000	040025	177750	000000	113520	000017	177777	156227
48	037777	173457	140016	177750	100001	051250	000027	177777	022152
48	037777	173400	040030	177740	000001	067170	000027	177777	022152
64	037777	170077	[illegible]	177740	100002	151340	000037	177775	116500
64	037777	170000	040242	177740	000002	171240	000037	177775	116500
80	037777	163517	140611	177730	100005	050030	000047	177773	013437
80	037777	163400	040623	177730	000005	100710	000047	177773	013430
96	037777	156137	141471	177720	100011	005120	000057	177767	050757
96	037777	156000	041517	177720	000011	050760	000057	177767	050760
112	037777	147537	143064	177710	100016	040610	000057	177762	006710
112	037777	147400	043040	177710	000016	121430	000067	177762	006710
128	037777	140177	145122	177700	100025	032700	000077	177753	075240
128	037777	140000	045174	177700	000025	132500	000077	177753	005242
-16	037777	147217	143042	177710	000016	121410	000070	077761	125551
-16	037777	147400	043065	177710	000016	121410	000057	177762	[illegible]
-32	037777	155637	141455	177720	000011	050720	000060	077767	004561
-32	037777	156000	041433	177720	000011	050720	000057	177767	051027
-48	037777	163257	140537	177730	000005	100530	000050	077772	162371
-48	037777	163400	040524	177730	000005	100530	000047	177773	013517
-64	037777	167577	140136	177740	000002	171140	000040	077775	076501
-64	037777	170000	040130	177740	000002	171140	000037	177775	116600
-80	037777	173317	137742	177750	000001	067050	000030	077777	011211
-80	037777	173400	037740	177750	000001	067050	000027	177777	022270
-96	037777	175737	137664	177760	000000	113360	000020	077777	152321
-96	037777	176000	037653	177760	000000	113350	000017	177777	156360
-112	037777	177257	137654	177770	000000	045270	000010	100000	072014
-112	037777	177400	037534	177770	000000	045270	000010	000000	033334
-128	037777	177777	137654	100000	000000	037600	000000	100000	040104
-128	040000	000000	037554	000000	000000	037600	000000	000000	040104

Two axis $\Delta y, \Delta z$ incrementally - 0 to 128 pulses and return to 0 again.

1	[illegible]	1							
16	1	128							
0	040000	000000	040000	000000	000000	040000	000000	000000	040000
16	037777	177317	140000	177770	100002	044460	000000	000170	040000
16	037777	177300	040000	177770	000000	045040	000000	000170	040000
32	037777	177037	140002	177760	100000	107540	000000	000760	037776
32	037777	177000	040003	177760	000000	111590	000000	000760	037777
48	037777	175557	140077	177750	100001	051220	000000	002150	077751
48	037777	175400	040031	177750	000001	055540	000000	002150	037751
64	037777	174077	140115	177740	100002	151300	000000	003740	077660
64	037777	174000	040122	177740	000002	151200	000000	-003740	077660
80	037777	171717	140301	177730	100005	047760	000000	006130	037470
80	037777	171600	040313	177730	000005	064240	000000	006130	037470
96	037777	167137	140627	177720	100011	005240	000000	010720	037130
-16	037777	163500	041417	177710	000016	[illegible]	000000	014110	076375
-32	037777	167000	[illegible]	177720	000011	050500	000000	011760	037137
-32	037777	167000	040535	177720	000011	[illegible]	000000	010720	037152
-48	037777	171500	040273	177730	000000	100500	000000	006250	037475
-48	037777	171500	040273	177730	000005	054100	000000	006137	037517
-64	037777	174000	040070	177740	000002	171000	000000	004740	037674
-64	037777	174000	[illegible]	177740	000002	[illegible]	000000	[illegible]	077771
-80	037777	175300	040007	177750	000001	061700	000000	[illegible]	[illegible]
-80	037777	175400	040007	177750	000001	055300	000000	002150	037777
-96	037777	177000	037755	177760	000000	113200	000000	001020	040017
-96	037777	177000	037745	177760	000000	111200	000000	000760	040027
-112	037777	177300	037752	177770	000000	045100	000000	000710	040023
-112	[illegible]	[illegible]	[illegible]	177770	000000	[illegible]	[illegible]	000170	040027
-128	040000	000000	037752	000000	000000	037400	000000	000000	040027
-128	040000	000000	037752	000000	000000	037400	000000	000000	040027

Two axis $\Delta x, \Delta z$ incrementally - 0 to 128 pulses and return to 0 again.

TABLE 5.2 (cont.)

1	1	1							
16	1	128							
0	040000	000000	040000	000000	000000	040000	000000	000000	040000
16	037777	177437	040000	177770	100170	047010	000007	100170	071271
16	037777	177417	137400	177770	100170	047010	000010	000170	073357
16	037777	177403	040000	177770	000170	047770	000010	000170	030035
32	037777	176077	040012	177760	100760	134012	000017	100757	145554
32	037777	176037	170000	177700	100760	134012	000020	000757	141717
32	037777	176000	040014	177740	000760	137752	000020	000757	141717
48	037777	173537	040164	177750	102151	155763	000027	102146	170000
48	037777	173457	133567	177750	102151	156763	000030	002146	114227
48	037777	173400	040112	177750	002151	167733	000030	002146	114227
64	037777	170177	040340	177740	103744	017547	000037	103734	067147
64	037777	170077	130456	177740	103744	017547	000040	003734	047445
64	037777	170000	040365	177740	003744	037547	000040	003734	047445
80	037777	163537	041174	177730	106137	156375	000047	106120	134154
80	037777	163517	124643	177730	106137	156375	000050	006120	103537
-48	037777	167400	041744	177730	006140	007133	000057	006120	103417
-64	037777	167577	170170	177740	004044	047756	000040	104034	017170
-64	037777	170007	047250	177740	004044	047356	000040	004034	073727
-64	037777	170000	047260	177740	003744	077756	000040	003734	047300
-80	037777	173317	137320	177750	002231	174051	000030	102226	075337
-80	037777	173400	037774	177750	002231	174051	000030	002226	107417
-80	037777	173400	037774	177750	002151	167464	000030	002146	114017
-96	037777	175737	135633	177760	001020	141472	000020	101017	133404
-96	037777	175000	037677	177760	001020	141472	000020	001017	137444
-96	037777	176000	037572	177760	000750	137473	000020	000757	141445
-112	037777	177057	137234	177770	000210	050051	000010	100210	072427
-112	037777	177400	037654	177770	000210	050051	000010	000210	077447
-112	037777	177400	037654	177770	000170	047451	000010	000170	030043
-128	037777	177777	137654	000000	000000	037421	000000	100000	037437
-128	040000	000000	037654	000000	000000	037421	000000	000000	037437
-128	040000	000000	037654	000000	000000	037421	000000	000000	037437

Two axis $\Delta x, \Delta y, \Delta z$ incrementally - 0 to 128 pulses and return to 0 again.

1	1	0							
4096	32768	1							
0	040000	000000	040000	000000	000000	040000	000000	000000	040000
4096	040000	000000	040000	000000	000000	040000	000000	000000	040000
8192	040000	000000	040000	000000	000000	040000	000000	000000	040000
12288	040000	000000	040000	000000	000000	040000	000000	000000	040000
16384	040000	000000	040000	000000	000000	040000	000000	000000	040000
20480	040000	000000	040000	000000	000000	040000	000000	000000	040000
24576	040000	000000	040000	000000	000000	040000	000000	000000	040000
28672	040000	000000	040000	000000	000000	040000	000000	000000	040000
32768	040000	000000	040000	000000	000000	040000	000000	000000	040000
4096	037600	025245	035207	000000	000000	040000	003772	125573	150345
8192	037002	124477	036241	000000	000000	040000	007725	073552	133027
12288	035515	067727	072357	000000	000000	040000	013551	071125	130127
16384	034052	050014	175554	000000	000000	040000	017256	164164	101447
20480	031746	150733	064506	000000	000000	040000	022562	035725	043374
24576	027323	176614	172114	000000	000000	040000	025637	174055	132257
-16384	034052	050014	175747	000000	000000	040000	017256	164164	101357
-20480	035515	067727	072536	000000	000000	040000	013551	071125	130077
-24576	037002	124477	035440	000000	000000	040000	007725	073552	132645
-28672	037600	025245	035404	000000	000000	040000	003772	125573	150137
-32768	040000	000000	040031	000000	000000	040000	000000	000000	041461
-4096	040000	000000	040031	000000	000000	040000	000000	000000	041461
-8192	040000	000000	040031	000000	000000	040000	000000	000000	041461
-12288	040000	000000	040031	000000	000000	040000	000000	000000	041461
-16384	040000	000000	040031	000000	000000	040000	000000	000000	041461
-20480	040000	000000	040031	000000	000000	040000	000000	000000	041461
-24576	040000	000000	040031	000000	000000	040000	000000	000000	041461
-28672	040000	000000	040031	000000	000000	040000	000000	000000	041461
-32768	040000	000000	040031	000000	000000	040000	000000	000000	041461

Two axis x,y sequentially - 0 to 32,768 pulses and return to 0 again.

TABLE 5.2 (cont.)

0	1	1							
4096	32768	1							
0	040000	000000	040030	000000	000000	040033	000000	000000	040000
4096	037600	025245	035227	000000	000000	040000	003772	125673	160345
8192	037002	124477	036241	000000	000000	040000	007725	073552	133023
12288	035615	067727	072357	000000	000000	040000	013561	001125	130127
16384	034052	050014	175554	000000	000000	040000	017256	154164	101440
20480	031746	150733	064506	000000	000000	040000	022562	035725	043374
24576	027323	176614	172114	000000	000000	040000	025637	174055	132257
28672	024406	013657	173402	000000	000000	040000	030437	067710	046071
32768	021224	050037	135327	000000	000000	040000	032732	124436	043614
4096	021117	037211	172166	175660	053432	027050	032732	124436	043614
8192	020501	016662	166114	173561	163430	166273	032732	124436	043614
12288	020255	022413	046370	171525	122024	126302	032732	124436	043614
16384	017130	121260	122723	167553	174637	032745	032732	124436	043614
20480	015012	163074	003417	165704	104453	100765	032732	124436	043614
24576	014515	021515	055525	164155	165634	142755	032732	124436	043614
28672	013052	047010	025557	162565	073366	053514	032732	124436	043614
32768	011255	166324	124453	161347	004425	063060	032732	124436	043614
-4096	013352	047010	025600	162565	073366	053506	032732	124436	043614
-8192	014515	021615	055500	164155	165634	142777	032732	124436	043614
-12288	016012	163074	003461	165704	104453	101105	032732	124436	043614
-16384	017130	121260	122776	167553	174637	033047	032732	124436	043614
-20480	020055	022413	046372	171525	122024	126362	032732	124436	043614
-24576	020501	016652	166100	173561	163430	166366	032732	124436	043614
-28672	021117	037211	172157	175660	053432	027102	032732	124436	043614
-32768	021224	050037	135555	000000	000000	040027	032732	124436	043614
-4096	024406	013657	173660	000000	000000	040027	030437	067310	046051
-8192	027323	176614	172433	000000	000000	040027	025637	174055	132151
-12288	031746	150733	065175	000000	000000	040027	022562	035725	043303
-16384	034052	050014	175151	000000	000000	040027	017256	154164	101235
-20480	035615	067727	073003	000000	000000	040027	013561	001125	127657
-24576	037002	124477	036623	000000	000000	040027	007725	073552	132500
-28672	037600	025245	035553	000000	000000	040027	003772	125673	157735
-32768	040000	000000	040163	000000	000000	040027	000000	000000	041255

Two axis, y,z sequentially - 0 to 32,768 pulses and return to 0 again.

1	0	1							
4096	32768	1							
0	040000	000000	040000	000000	000000	040000	000000	000000	040000
4096	040000	000000	040000	000000	000000	040000	000000	000000	040000
8192	040000	000000	040000	000000	000000	040000	000000	000000	040000
12288	040000	000000	040000	000000	000000	040000	000000	000000	040000
16384	040000	000000	040000	000000	000000	040000	000000	000000	040000
20480	040000	000000	040000	000000	000000	040000	000000	000000	040000
24576	040000	000000	040000	000000	000000	040000	000000	000000	040000
28672	040000	000000	040000	000000	000000	040000	000000	000000	040000
32768	040000	000000	040000	000000	000000	040000	000000	000000	040000
4096	037600	025245	035223	174005	052104	117367	000000	000000	040000
8192	037002	124477	035227	170052	104225	144711	000000	000000	040000
12288	035615	067727	072344	164216	176652	147605	000000	000000	040000
16384	034052	050014	175535	160521	013613	176300	000000	000000	040000
20480	031746	150733	064465	155215	142053	034345	000000	000000	040000
24576	027323	176614	172071	152140	003722	145471	000000	000000	040000
-16384	034052	050014	175737	160521	013613	175374	000000	000000	040000
-20480	035615	067727	072604	164216	176652	147740	000000	000000	040000
-24576	037002	124477	036442	170052	104225	145575	000000	000000	040000
-28672	037600	025245	035414	174005	052104	117512	000000	000000	040000
-32768	040000	000000	040044	000000	000000	036213	000000	000000	040000
-4096	040000	000000	040044	000000	000000	035213	000000	000000	040000
-8192	040000	000000	040044	000000	000000	035213	000000	000000	040000
-12288	040000	000000	040044	000000	000000	035217	000000	000000	040000
-16384	040000	000000	040044	000000	000000	036213	000000	000000	040000
-20480	040000	000000	040044	000000	000000	035213	000000	000000	040000
-24576	040000	000000	040044	000000	000000	036213	000000	000000	040000
-28672	040000	000000	040044	000000	000000	035213	000000	000000	040000
-32768	040000	000000	040044	000000	000000	036213	000000	000000	040000

Two axis x,z sequentially - 0 to 32,768 pulses and return to 0 again.

TABLE 5.2 (cont.)

1	1	1							
4096	32768	1							
0	040000	000000	040000	000000	000000	040000	000000	000000	040000
4096	040000	000000	040000	000000	000000	040000	000000	000000	040000
8192	040000	000000	040000	000000	000000	040000	000000	000000	040000
12288	040000	000000	040000	000000	000000	040000	000000	000000	040000
16384	040000	000000	040000	000000	000000	040000	000000	000000	040000
20480	040000	000000	040000	000000	000000	040000	000000	000000	040000
24576	040000	000000	040000	000000	000000	040000	000000	000000	040000
28672	040000	000000	040000	000000	000000	040000	000000	000000	040000
32768	040000	000000	040000	000000	000000	040000	000000	000000	040000
4096	037600	025245	035227	000000	000000	040000	003772	125573	160345
8192	037002	124477	036241	000000	000000	040000	007725	073552	173027
12288	035615	067727	072357	000000	000000	040000	013561	001125	170127
16384	034052	050014	175554	000000	000000	040000	017256	164164	101447
20480	031746	150733	064306	000000	000000	040000	022562	035725	043374
24576	027323	176614	172114	000000	000000	040000	025637	174055	132257
-16384	034052	050014	176151	000000	000000	040027	017255	164164	101235
-20480	035615	067727	073033	000000	000000	040027	013561	001125	127653
-24576	037002	124477	035623	000000	000000	040027	007725	073552	132500
-28672	037600	025245	035553	000000	000000	040027	003772	125573	157735
-32768	040000	000000	040163	000000	000000	040027	000000	000000	041255
-4096	040000	000000	040163	000000	000000	040027	000000	000000	041255
-8192	040000	000000	040163	000000	000000	040027	000000	000000	041255
-12288	040000	000000	040163	000000	000000	040027	000000	000000	041255
-16384	040000	000000	040163	000000	000000	040027	000000	000000	041255
-20480	040000	000000	040163	000000	000000	040027	000000	000000	041255
-24576	040000	000000	040163	000000	000000	040027	000000	000000	041255
-28672	040000	000000	040163	000000	000000	040027	000000	000000	041255
-32768	040000	000000	040163	000000	000000	040027	000000	000000	041255

Three axis x,y,z sequentially - 0 to 32,768 pulses and return to 0 again.

1	1	0							
4096	1	32768							
0	040000	000000	040000	000000	000000	040000	000000	000000	040000
4096	037600	062451	006010	000177	121314	047357	003764	155427	173570
4096	037600	052476	153331	000177	121314	047357	003765	055030	130740
8192	037005	067376	063240	000772	120254	150745	007652	134273	102335
8192	037005	047650	074437	000772	120254	150745	007653	032706	041604
12288	035632	166516	017277	002145	024625	172041	013347	111422	021363
12288	035632	137606	100545	002145	024625	172041	013344	005107	147407
16384	034124	023505	070720	003653	173037	153720	016545	144703	051427
16384	034123	166172	066642	003653	173037	153720	016546	035153	073327
20480	032113	040642	147771	005664	160533	016626	021375	066706	172776
20480	032112	175647	105726	005664	160533	016626	021375	153135	031510
24576	027640	067021	044715	010137	134555	012732	023575	157710	033125
24576	027640	017425	025375	010137	134555	012732	023576	037410	141573
28672	025170	165127	122435	012607	040163	007237	025303	032063	011773
28672	025170	112311	163765	012607	040163	007237	025303	124444	111425
32768	022375	153410	150573	015402	052652	177760	026263	050721	052171
32768	022375	076641	161735	015402	052652	177760	026263	115714	124447
-4096	025170	112311	153732	012607	112771	172752	025303	057025	136267
-4096	025170	112311	163732	012607	040163	007260	025303	104444	111437
-8192	027640	017425	025342	010140	004151	071275	023575	017111	030605
-8192	027640	017425	025342	010137	134555	012554	023576	037410	141534
-12288	032112	175647	105732	005665	023526	171213	021375	137763	025102
-12288	032112	175647	105732	005664	160533	016473	021375	153135	031363
-16384	034123	166172	066507	003654	030354	035356	016545	025423	047630
-16384	034123	166172	066607	003653	173037	153560	016546	035153	073244
-20480	035632	137606	100516	002145	053535	177642	013344	005575	047045
-20480	035632	137606	100516	002145	024625	171736	013344	005107	147431
-24576	037005	047650	074463	000772	140003	033424	007653	030720	161345
-24576	037005	047650	074463	000772	120254	150575	007653	032706	041625
-28672	037600	052476	153311	000177	131267	000665	003765	054431	056102
-28672	037600	052476	153311	000177	121314	047203	003765	055030	130700
-32768	040000	000000	037607	000000	000000	037675	000000	000000	042717
-32768	040000	000000	037607	000000	000000	037675	000000	000000	042717

Two axis $\Delta x, \Delta y$ incrementally - 0 to 32,768 pulses and return to 0 again.

TABLE 5.2 (cont.)

0	1	1							
4096	1	32768							
0	040000	000000	040000	000000	000000	040000	000000	000000	040000
4096	037400	135147	121541	174013	022350	104712	003765	055030	130214
4096	037400	125175	067460	174012	123347	022345	003765	055030	130214
8192	036012	137245	120273	170125	043504	176160	007653	032306	041163
8192	036012	117520	133456	170124	147457	117171	007653	032306	041163
12288	033465	126324	060564	164434	066356	057443	013344	005137	146735
12288	033465	077414	146345	164433	177203	031503	013344	005137	146735
16384	030250	011700	120444	161232	033075	017135	016545	035153	072673
16384	030247	154364	126115	161231	152355	030650	016545	035153	072677
20480	024225	036512	016777	156402	111071	105501	021375	153135	031137
20480	024225	173516	170506	156402	040415	053250	021375	153135	031137
24576	017500	105446	033414	154202	020070	045234	023576	037410	141345
24576	017500	037057	034373	154201	160667	077514	023575	037410	141345
28672	012361	077437	047442	152474	145715	066255	025303	104444	111361
28672	012361	024623	136332	152474	120752	141777	025303	104444	111361
32768	004777	052252	073662	151514	127057	026042	026263	115714	124471
32768	004772	175503	140032	151514	115070	156064	026263	115714	124471
-4096	012360	152014	100323	152474	120752	142044	025303	131406	110304
-4096	012361	024623	136376	152474	120752	142044	025303	104444	111443
-8192	017477	167455	075757	154201	160667	077637	023576	076510	170235
-8192	017500	037057	034200	154201	160667	077637	023576	037410	141507
-12288	024225	130522	171332	156402	040415	057406	021376	023510	155537
-12288	024225	173516	170277	156402	040415	053406	021375	153135	031276
-16384	030247	117047	152736	161231	152355	030736	016546	115572	166537
-16384	030247	154364	126003	161231	152355	030736	016546	035153	073076
-20480	033465	050504	044723	164433	177203	031575	013344	074762	117405
-20480	033465	077414	146316	164433	177203	031575	013344	005137	147265
-24576	036012	077771	152725	170124	147457	117242	007653	126733	061211
-24576	036012	117520	133566	170124	147457	117242	007653	032306	041477
-28672	037400	115222	035532	174012	123347	022572	003765	154031	173230
-28672	037400	125175	067613	174012	123347	022572	003765	055030	130611
-32768	037777	177777	137772	000000	000000	036571	000000	100000	041770
-32768	040000	000000	037772	000000	000000	036571	000000	000000	041770

Two axis $\Delta y, \Delta z$ incrementally - 0 to 32,768 pulses and return to 0 again.

1	0	1							
4096	1	32768							
0	040000	000000	040000	000000	000000	040000	000000	000000	040000
4096	037600	062451	075775	174313	022356	104117	000177	121714	047364
4096	037600	052475	153316	174012	122747	145747	000177	121714	047364
8192	037005	067373	053215	170125	043504	175354	000772	120254	150767
8192	037000	047650	074415	170124	145472	075106	000772	120254	150765
12288	035532	165315	017247	164434	066356	033535	002145	074025	172054
12288	035532	137535	100515	164433	172670	130311	002145	074525	172054
16384	034124	022305	[illegible]	161232	033075	015002	003657	173037	153747
16384	034127	165177	006105	161231	142525	004402	003657	173037	153747
20480	032113	040542	147725	156402	111071	174741	005664	150533	016562
20480	032112	175647	105667	156402	024643	046027	005664	160533	016562
24576	027540	057021	044350	154202	020070	044525	010137	134555	012777
24573	027640	017425	025330	154201	140367	175157	010137	134555	012777
-4096	025170	112311	157531	152474	073333	155333	012507	040163	007327
-8192	027640	017425	[illegible]	154201	160567	077151	010140	004151	071376
-8192	027540	017425	025271	154201	140367	135223	010137	134555	012514
-12288	032112	175647	105563	156402	040415	052537	005665	073525	131245
-12288	[illegible]	175647	105663	156402	024643	046356	005664	160533	016526
-16384	034177	165177	[illegible]	161231	152355	037102	003654	030754	036400
-16384	034177	165177	056543	161231	142525	074460	003653	173037	153617
-20480	075577	137505	100454	164433	177203	030656	002145	053535	177665
-20480	075577	137505	100454	164433	172670	130273	002145	074525	171761
-24576	037005	047350	074430	170124	147457	115352	000772	140003	033447
-24576	077000	047650	074430	170124	145472	070072	000772	120254	150614
-28672	077500	052475	153265	174012	123347	001512	000177	131757	000577
-28672	037570	052477	153265	174012	122747	147006	000177	121314	047211
-32768	040000	000000	077577	000000	000000	074700	000000	000000	077665
-32768	040000	000000	077577	000000	000000	074700	000000	000000	077660

Two axis $\Delta x, \Delta z$ incrementally - 0 to 32,768 pulses and return to 0 again.

TABLE 5.2 (cont.)

1	1	1							
4096	1	32768							
0	040000	000000	040000	000000	000000	040000	000000	000000	040000
4096	037401	017531	143237	174217	166270	157502	004157	002314	010606
4096	037401	007173	037401	174217	166270	157502	004157	101316	037537
4096	037400	177532	115160	174217	067266	150455	004157	101316	037537
8192	036020	022217	172764	171167	041732	046042	010570	135714	147255
8192	036020	000631	004273	171167	041732	046042	010571	031754	172327
8192	036017	163207	014117	171166	145672	062002	010571	031754	172327
12288	033520	034056	031073	166777	062164	125334	015260	114276	156765
12288	033520	001314	111036	166777	062164	125334	015261	003537	014357
12288	033517	157312	166147	166776	172724	144505	015261	003537	014357
16384	030372	020225	035413	165535	133633	025555	021650	067440	167567
16384	030371	154503	175326	165535	133633	025555	021650	150424	164517
16384	030371	127777	004030	165535	052647	121053	021650	150424	164517
20480	024531	115442	147551	165261	130765	063547	025765	005322	042574
20480	024531	041470	063422	165261	130765	053547	025765	056505	021727
20480	024531	014233	074112	165261	057503	005624	025765	056505	021727
24576	020311	151363	024561	166003	055677	067767	031463	033564	027061
24576	020311	066214	074165	166003	055677	067767	031463	074507	066661
24576	020311	042222	157142	166003	015054	137331	031463	074507	066661
28672	013661	017430	023745	167502	170306	172664	034414	061071	171557
28672	013660	126377	002020	167502	170306	172664	034414	110633	137606
28672	013660	105604	163073	167502	140545	136460	034414	110633	137606
32768	007174	035513	072200	172111	116341	036764	036477	121110	153037
32768	007177	140026	011367	172111	116341	036764	036477	137500	151077
32768	007173	124551	027701	172111	077751	151665	036470	137500	151077
-4096	013660	014553	050222	167503	031577	000652	034414	151166	067564
-4096	013660	105604	163035	167503	031577	000652	034414	131425	145204
-4096	013660	105604	163035	167502	140545	136371	034414	110633	137557
-8192	020310	157053	065153	166003	100223	154447	031463	161322	172600
-8192	020311	042222	157176	166003	100223	154447	031463	120500	151307
-8192	020311	042222	157176	166003	015054	137237	031463	074507	056607
-12288	024537	140263	012521	165261	133455	170440	025765	155723	153275
-12288	024531	014233	074106	165261	133455	170440	025765	124042	036511
-12288	024531	014233	074105	165261	057503	005571	025765	056505	021671
-16384	030371	064254	130557	165535	116371	066121	021651	056115	007555
-16384	030371	127777	004121	165535	116371	066121	021650	175131	173307
-16384	030371	127777	004121	165535	052647	120343	021650	150424	164547
-20480	033517	124550	023641	166777	025466	175243	015261	115000	100001
-20480	033517	157312	166402	166777	025466	175243	015261	025540	173715
-20480	033517	157312	166402	166776	172724	143744	015261	003537	014331
-24576	036017	141624	002661	171166	167254	161543	010571	142736	162134
-24576	036017	163207	014516	171166	167254	161543	010571	046577	035101
-24576	036017	163207	014516	171166	145672	061367	010571	031754	172257
-28672	037400	167272	175300	174217	077625	162334	004160	007561	037365
-28672	037400	177632	115540	174217	077625	162334	004157	110557	050247
-28672	037400	177632	115540	174217	067266	150140	004157	101316	037353
-32768	037777	177777	140211	000000	000000	037025	000000	100000	037132
-32768	040000	000000	040211	000000	000000	037025	000000	000000	037132
-32768	040000	000000	040211	000000	000000	037025	000000	000000	037132

Three axis $\Delta x, \Delta y, \Delta z$ incrementally - 0 to 32,768 pulses and return to 0 again.

TABLE 5.2 (cont.)

1	1	1							
1	1	9	1						
0	040000	000000	040000	000000	000000	040000	000000	000000	040000
1	037777	177777	140000	000000	000000	040000	000000	100000	037777
1	037777	177777	040000	177777	100000	040001	000000	100000	037777
2	037777	177777	040000	177777	100001	040001	000000	100001	037775
2	037777	177775	137776	177777	100001	040001	000001	000001	037771
2	037777	177774	040000	177777	000001	040007	000001	000001	037771
3	037777	177774	040000	177777	000003	040013	000001	000003	037765
3	037777	177771	137772	177777	000003	040013	000001	100003	037757
3	037777	177767	040000	177776	100003	040032	000001	100003	037757
4	037777	177767	040000	177776	100006	040042	000001	100006	037740
4	037777	177763	137764	177776	100006	040042	000002	000006	037712
4	037777	177760	040000	177776	000006	040076	000002	000006	037712
5	037777	177760	040000	177776	000012	040116	000002	000012	037672
5	037777	177753	137754	177776	000012	040116	000002	100012	037625
5	037777	177747	040000	177775	100012	040173	000002	100012	037625
6	037777	177747	040000	177775	100017	040223	000002	100017	037573
6	037777	177741	137742	177775	100017	040223	000003	000017	037503
6	037777	177734	040000	177775	000017	040325	000003	000017	037503
7	037777	177734	040000	177775	000025	040371	000003	000025	037437
7	037777	177725	137726	177775	000025	040371	000003	100025	037320
7	037777	177717	040000	177774	100025	040524	000003	100025	037320
8	037777	177717	040000	177774	100034	040604	000003	100034	037236
8	037777	177707	137710	177774	100034	040604	000004	000034	037064
8	037777	177700	040000	177774	000034	040774	000004	000034	037064
9	037777	177700	040000	177774	000044	041074	000004	000044	036764
-1	037777	177700	040000	177774	000034	040774	000004	000034	037064
-2	037777	177707	137710	177774	100034	040603	000004	000034	037064
-2	037777	177717	040000	177774	100034	040603	000003	100034	037235
-2	037777	177717	040000	177774	100025	040521	000003	100025	037315
-3	037777	177725	137726	177775	000025	040365	000003	100025	037315
-3	037777	177734	040000	177775	000025	040365	000003	000025	037433
-3	037777	177734	040000	177775	000017	040321	000003	000017	037477
-4	037777	177741	137742	177775	100017	040216	000003	000017	037477
-4	037777	177747	040000	177775	100017	040216	000002	100017	037566
-4	037777	177747	040000	177775	100012	040164	000002	100012	037616
-5	037777	177753	137754	177776	000012	040106	000002	100012	037616
-5	037777	177760	040000	177776	000012	040106	000002	000012	037662
-5	037777	177760	040000	177776	000006	040066	000002	000006	037702
-6	037777	177763	137764	177776	100006	040031	000002	000006	037702
-6	037777	177767	040000	177776	100006	040031	000001	100006	037727
-6	037777	177767	040000	177776	100003	040017	000001	100003	037737
-7	037777	177771	137772	177777	000003	037777	000001	100003	037737
-7	037777	177774	040000	177777	000003	037777	000001	000003	037751
-7	037777	177774	040000	177777	000001	037773	000001	000001	037755
-8	037777	177775	137776	177777	100001	037764	000001	000001	037755
-8	037777	177777	040000	177777	100001	037764	000000	100001	037760
-8	037777	177777	040000	177777	100000	037762	000000	100000	037760
-9	037777	177777	140000	000000	000000	037760	000000	100000	037760
-9	040000	000000	040000	000000	000000	037760	000000	000000	037760

Three axis $\Delta y, \Delta z, \Delta x$ incrementally - 0 to 8 pulses and return to 0 again.

TABLE 5.2 (cont.)

1	1	1							
1	1	9	1						
0	040000	000000	040000	000000	000000	040000	000000	000000	040000
1	037777	177777	140000	177777	100000	040000	000000	000000	040000
2	037777	177777	140000	177777	100000	040000	000000	000001	040000
2	037777	177777	037776	177777	100000	040000	000000	100001	037776
2	037777	177775	137776	177777	000000	040003	000000	100001	037776
3	037777	177775	137776	177777	000001	040007	000000	100003	037774
3	037777	177774	037770	177777	000001	040007	000001	000003	037765
3	037777	177771	137772	177775	100001	040021	000001	000003	037765
4	037777	177771	137772	177776	100003	040031	000001	000006	037761
4	037777	177767	037756	177776	100003	040031	000001	100006	037741
4	037777	177763	137764	177776	000003	040056	000001	100006	037741
5	037777	177763	137764	177776	000006	040076	000001	100012	037727
5	037777	177760	037740	177776	000006	040076	000002	000012	037672
5	037777	177753	137754	177775	100006	040142	000002	000012	037672
6	037777	177753	137754	177775	100012	040172	000002	000017	037652
6	037777	177747	037716	177775	100012	040172	000002	100017	037574
6	037777	177741	137742	177775	000012	040261	000002	100017	037574
7	037777	177741	137742	177775	000017	040325	000002	100025	037542
7	037777	177734	037670	177775	000017	040325	000003	000025	037437
7	037777	177725	137726	177774	100017	040443	000003	000025	037437
8	037777	177725	137726	177774	100025	040523	000003	000034	037373
8	037777	177717	037636	177774	100025	040523	000003	100034	037237
8	037777	177707	137710	177774	000025	040674	000003	100034	037237
9	037777	177707	137710	177774	000034	040774	000003	100044	037155
9	037777	177700	037600	177774	000034	040774	000004	000044	036764
-1	037777	177707	137710	177774	000034	040774	000003	100044	037154
-1	037777	177707	137710	177774	000025	040674	000003	100034	037234
-2	037777	177717	037636	177774	100025	040522	000003	100034	037234
-2	037777	177725	137726	177774	100025	040522	000003	000034	037367
-2	037777	177725	137726	177774	100017	040440	000003	000025	037433
-3	037777	177734	037670	177775	000017	040321	000003	000025	037433
-3	037777	177741	137742	177775	000017	040321	000002	100025	037535
-3	037777	177741	137742	177775	000012	040255	000002	100017	037565
-4	037777	177747	037716	177775	100012	040165	000002	100017	037565
-4	037777	177753	137754	177775	100012	040165	000002	000017	037642
-4	037777	177753	137754	177775	100006	040133	000002	000012	037662
-5	037777	177760	037740	177776	000006	040066	000002	000012	037662
-5	037777	177763	137764	177776	000006	040066	000001	100012	037716
-5	037777	177763	137764	177776	000003	040046	000001	100006	037726
-6	037777	177767	037756	177776	100003	040020	000001	100006	037726
-6	037777	177771	137772	177776	100003	040020	000001	000006	037745
-6	037777	177771	137772	177776	100001	040006	000001	000003	037751
-7	037777	177774	037770	177777	000001	037773	000001	000003	037751
-7	037777	177775	137776	177777	000001	037773	000000	100003	037757
-7	037777	177775	137776	177777	000000	037767	000000	100001	037757
-8	037777	177777	037776	177777	100000	037763	000000	100001	037757
-8	037777	177777	140000	177777	100000	037763	000000	000001	037760
-8	037777	177777	140000	177777	100000	037761	000000	000000	037760
-9	040000	000000	040000	000000	000000	037760	000000	000000	037760

Three axis $\Delta z, \Delta x, \Delta y$ incrementally - 0 to 8 pulses and return to 0 again.

TABLE 5.2 (cont.)

1	1	1							
4096	1	32769	1						
0	040000	000000	040000	000000	000000	040000	000000	000000	040000
4096	037401	017531	143233	174217	166270	157502	004157	002314	016606
4096	037401	007173	037401	174217	166270	157502	004157	101316	037532
4096	037400	177632	115160	174217	067266	150455	004157	101316	037532
8192	036020	022212	172764	171167	041732	046042	010570	135214	147255
8192	036020	000631	004273	171167	041732	046042	010571	031254	172327
8192	036017	163207	014117	171166	145672	062002	010571	031254	172320
12288	033520	034056	031073	166777	062164	125334	015260	114276	156765
12288	033520	001314	111036	166777	062164	125334	015261	003537	014357
12288	033517	157312	166147	166776	172724	144505	015261	003537	014357
16384	030372	020225	035413	165535	133633	025555	021650	057440	157562
16384	030371	154503	175326	165535	133633	025555	021650	150424	164517
16384	030371	127777	004030	165535	052647	121053	021650	150424	164517
20480	024531	115442	147551	165261	130765	063547	025765	075722	042574
20480	024531	041470	063422	165261	130765	063547	025765	056505	021727
20480	024531	014233	074112	165261	057503	005624	025765	056505	021727
24576	020311	151363	024561	166003	055677	067767	031463	033564	027061
24576	020311	066214	074166	166003	055677	067767	031463	074507	056661
24576	020311	042222	167142	166003	015054	137331	031463	074507	056661
28672	013661	017430	023745	167502	170306	172664	034414	061071	171557
28672	013660	126377	032020	167502	170306	172664	034414	110533	137606
28672	013660	105604	163073	167502	140545	136460	034414	110533	137605
32766	007174	035513	072200	172111	116341	036764	036472	121110	153037
32768	007173	140326	011367	172111	116341	036764	036472	137500	151077
32768	007173	124551	027731	172111	077751	151665	036472	137500	151077
-4096	013657	102727	050361	167503	073070	153263	034415	031517	071344
-4096	013657	173761	104057	167503	073070	153263	034415	001757	172434
-4096	013657	173761	104057	167503	002036	170116	034414	161166	067563
-8192	020310	047711	151327	166003	122552	031152	031464	046135	013415
-8192	020310	133362	025000	166003	122552	031152	031464	005314	010423
-8192	020310	133062	025000	166003	037402	042313	031463	161322	172577
-12288	024530	037046	151551	165261	136150	075242	025765	054041	114751
-12288	024530	113023	030373	165261	136150	075242	025766	002560	142660
-12288	024530	113023	030373	165261	062174	115136	025755	155323	163224
-16384	030370	174024	016220	165535	101131	123035	021651	163504	104610
-16384	030371	037547	104647	165535	101131	123035	021651	102522	051014
-16384	030371	037547	104647	165535	035406	142076	021651	056115	007554
-20480	033517	050002	022311	166776	170773	021731	015262	026741	123327
-20480	033517	102546	007555	166776	170773	021731	015261	137002	150557
-20480	033517	102546	007555	166776	136227	145726	015261	115000	100001
-24576	036017	102515	030346	171166	114601	017037	010572	054421	001135
-24576	036017	124201	065350	171166	114601	017037	010571	160361	152117
-24576	036017	124201	065350	171166	073215	073517	010571	142736	162135
-28672	037400	147370	140000	174217	011164	042215	004160	116224	154700
-28672	037400	157731	075447	174217	011164	042215	004160	017323	025356
-28672	037400	157731	075447	174217	000624	013111	004160	007561	037366
-32767	037777	177775	140206	177777	100001	037032	000001	000001	037137
-32768	037777	177777	040210	177777	100001	037032	000000	100001	037133
-32768	037777	177777	040210	177777	100000	037030	000000	100000	037133
-32769	037777	177777	140210	000000	000000	037026	000000	100000	037133
-32763	040000	000000	040210	000000	000000	037026	000000	000000	037133

Three axis $\Delta y, \Delta z, \Delta x$ incrementally - 0 to 32,768 pulses and return to 0 again.

TABLE 5.2 (cont.)

1	1	1							
4096	1	32769	1						
0	040000	000000	040000	000000	000000	040000	000000	000000	040000
4096	037401	027071	110261	174217	155732	162365	004156	103311	143021
4096	037401	016534	002434	174217	155732	162365	004157	002314	010646
4096	037401	007173	037317	174217	056730	134436	004157	002314	010646
8192	036020	037634	035727	171167	020350	171221	010570	041154	071217
8192	036020	016253	037337	171167	020350	171221	010570	135214	147321
8192	036020	000631	004223	171166	124310	152115	010570	135214	147321
12288	033520	056057	062357	166777	027423	116666	015260	025036	055361
12288	033520	023316	121016	166777	027423	116666	015260	114276	156756
12288	033520	001314	110625	166776	140163	073033	015260	114276	156755
16384	030372	044731	173654	165535	070112	073330	021650	006454	121403
16384	030372	001211	075537	165535	070112	073330	021657	067440	157546
16384	030371	154503	174777	165535	007126	115414	021650	067440	167546
20480	024531	142677	143646	165261	055013	076473	025764	134037	010554
20480	024531	066726	002265	165261	055013	076473	025765	005322	042401
20480	024531	041470	063031	165261	003530	146055	025765	005322	042401
24576	020311	175354	176157	166002	172531	024567	031462	173040	117007
24576	020311	112206	147232	166002	172531	024567	031462	033664	026572
24576	020311	066214	073671	166002	131706	074144	031463	033564	026577
28672	013661	040222	175246	167502	077256	051726	034414	031727	161660
28672	013660	147172	062625	167502	077256	051726	034414	051071	171275
28672	013660	126377	031617	167502	047514	154136	034414	061071	171275
32768	007174	051271	011163	172111	021154	010755	036472	102520	125205
32768	007173	154103	165332	172111	021154	010755	036472	121110	152602
32768	007173	140326	011272	172111	002564	074324	036472	121110	152602
-4096	013657	123521	073700	167503	002036	170616	034415	001757	172327
-4096	013660	014553	050077	167503	002036	170616	034414	152220	032033
-4096	013660	014553	050077	167502	111005	064750	034414	131425	145075
-8192	020310	073703	112771	166003	037402	043077	031464	005314	010441
-8192	020310	157353	065000	166003	037402	043077	031463	144472	135462
-8192	020310	157353	065000	166002	154232	155704	031463	120500	151316
-12286	024530	064304	056265	165261	062174	115625	025765	002560	142767
-12298	024530	140260	012345	165261	062174	115625	025765	131277	116203
-12288	024530	140260	012345	165261	006221	060264	025765	104042	006620
-16384	030371	020532	003627	165535	035406	142665	021651	102522	051145
-16384	030371	064254	130311	165535	035406	142665	021651	021537	144137
-16384	030371	064254	130311	165534	171664	123674	021650	175131	173432
-20480	033517	072005	014450	166776	136227	146406	015261	137002	150752
-20480	033517	124550	023216	166776	136227	146406	015261	047543	132205
-20480	033517	124550	023216	166776	103465	051101	015261	075540	174122
-24576	036017	120240	135232	171166	073215	073755	010571	160361	152356
-24576	036017	141624	002135	171166	073215	073755	010571	064322	070272
-24576	036017	141624	002135	171166	051632	140535	010571	046577	035342
-28672	037400	156733	034610	174217	000624	013211	004160	017223	076003
-28672	037400	167272	174254	174217	000624	013211	004157	120221	057555
-28672	037400	167272	174254	174216	170264	162111	004157	110657	050665
-32768	037777	177777	037523	177777	100000	037147	000000	100001	037652
-32768	037777	177777	137525	177777	100000	037147	000000	000001	037653
-32768	037777	177777	137525	177777	100000	037145	000000	000000	037653
-32769	040000	000000	037525	000000	000000	037144	000000	000000	037653

Three axis $\Delta z, \Delta x, \Delta y$ incrementally - 0 to 32,768 pulses and return to 0 again.

TABLE 5.2 (cont.)

AXQT
27 FEB 70 AT 10:49:02 THE COLLECTOR 1100-0013
[illegible] = 4254 / 1 61
[illegible]10221

0	1	0							
512	1024	1	10						
0	040000	000000	040000	000000	000000	040003	000000	000000	040000
512	037776	000002	165200	000000	000000	040000	000377	176525	113757
1024	037770	000052	165024	000000	000003	040000	000777	165753	027743
-512	037776	000002	165163	000000	000000	040000	000377	176525	114637
-1024	040000	000000	037670	000000	000000	040003	000000	000000	041541
0	040000	000000	037670	000000	000000	040000	000000	000000	041541
512	037775	000002	165365	000000	000000	040000	000377	176525	115517
1024	037770	000052	164567	000000	000000	040003	000777	165753	031477
-512	037776	000002	165040	000000	000000	040000	000377	176525	116371
-1024	040000	000000	037551	000000	000000	040000	000000	000000	043300
0	040000	000000	037551	000000	000000	040000	000000	000000	043300
512	037775	000002	164725	000000	000000	040000	000377	176525	117245
1024	037770	000052	164534	000000	000000	040000	000777	165753	033234
-512	037775	000002	164711	000000	000000	040003	000377	176525	120125
-1024	040000	000000	037444	000000	000000	040000	000000	000000	045037
0	040000	000000	037444	000000	000000	040003	000000	000000	045037
512	037775	000002	164601	000000	000000	040000	000377	176525	121304
1024	037770	000052	164371	000000	000000	040000	000777	165753	034772
-512	037775	000002	164565	000000	000000	040000	000377	176525	121664
-1024	040000	000000	037332	000000	000000	040000	000000	000000	046574
0	040000	000000	037332	000000	000000	040003	000000	000000	046574
512	037775	000002	164453	000000	000000	040003	000377	176525	122535
1024	037770	000052	164217	000000	000000	040000	000777	165753	036522
-512	037771	000002	164434	000000	000000	040000	000377	176525	123410
-1024	040000	000000	037217	000000	000000	040000	000000	000000	050327

Single Axis - Runs 1-5

TABLE 5.3

Oscillatory Runs-Ten Cycles 0 to 1024
Pulses and Return to 0 Again

0	040000	000000	037217	000000	000000	040000	000000	[illegible]	050000
512	037776	000032	164324	000000	000000	040000	000777	176525	124277
1024	037776	[illegible]	164255	000000	000000	040000	000777	165753	046254
-512	037770	[illegible]	164310	000000	000000	040000	000377	176525	125157
-1024	040000	000000	037104	000000	000000	040000	000000	070700	052360

0	040000	000000	037104	000000	000000	040000	000000	000000	052000
512	037774	000002	164177	000000	000000	040000	000377	176525	126327
1024	037770	000000	163707	000000	000000	040000	000777	165753	042000
-512	037775	000037	164136	000000	000000	040000	000377	176525	126700
-1024	040000	000000	036776	000000	000000	040000	000000	000000	053615

0	040000	000000	036776	000000	000000	040000	000000	000000	053615
512	037777	000002	164351	000000	000000	040000	000377	176525	127555
1024	037777	000000	163547	000000	000000	040000	000777	165753	040577
-512	037777	000002	164037	000000	000000	040000	000377	176525	127477
-1024	040000	000000	036671	000000	000000	040000	000000	000000	055345

0	040000	000000	036671	000000	000000	040000	000000	000000	055345
512	037776	000002	163723	000000	000000	040000	000377	176525	131303
1024	037776	000000	163377	000000	000000	040000	000777	165753	045266
-512	037776	000000	163707	000000	000000	040000	000377	176525	132157
-1024	040000	000000	036555	000000	000000	040000	000000	000000	057075

0	040000	000000	036555	000000	000000	040000	000000	000000	057075
512	037776	000002	163577	000000	000000	040000	000377	176525	133037
1024	037776	000000	163240	000000	000000	040000	000777	165753	047011
-512	037775	000002	163537	000000	000000	040000	000377	176525	133717
-1024	040000	000000	036435	000000	000000	040000	000000	000000	057674

Single Axis - Runs 6-10

TABLE 5.3 (cont.)

0	1	1							
12	1024	1	10						
0	040000	000030	040000	000000	000000	040000	000000	000000	040000
12	037776	000002	165400	000000	000000	040000	000377	176525	113757
1024	037777	000052	165024	000000	000000	040000	000777	165253	027743
12	037766	000155	111300	177400	021251	164056	000777	165253	027743
1024	037750	000525	104520	177000	052521	050415	000777	165253	027743
-12	037766	000155	111376	177400	021251	163776	000777	165253	027743
-1024	037770	000057	164711	000000	000000	037124	000777	165253	027743
-512	037775	000002	135300	000000	000000	037124	000377	176525	114637
-1024	040000	000000	037555	000000	000000	037124	000000	000000	041544

0	040000	000000	037555	000000	000000	037124	000000	000000	041544
12	037777	000000	164740	000000	000000	037124	000377	176525	115517
1024	037770	000057	164554	000000	000000	037124	000777	165253	031507
-12	037766	000155	111071	177400	021251	163227	000777	165253	031507
1024	037760	000525	104341	177000	052521	047553	000777	165253	031507
-12	037766	000155	111162	177400	021251	163130	000777	165253	031507
-1024	037777	000057	164427	000000	000000	035253	000777	165253	031507
-512	037775	000002	134505	000000	000000	037253	000377	176525	116377
-1024	040000	000000	037321	000000	000000	035253	000000	000000	043307

0	040000	000000	037521	000000	000000	035253	000000	000000	043307
512	037775	000007	164475	000000	000000	035253	000377	176525	117247
1024	037770	000057	164304	000000	000000	035253	000777	165253	073237
512	037766	000155	110617	177400	021251	162336	000777	165253	073237
1024	037760	000525	104055	177000	052521	046703	000777	165253	073237
-512	037766	000155	110504	177400	021251	162256	000777	165253	033237
-1024	037777	000057	164175	000000	000000	035401	000777	165253	073237
-512	037777	000000	[illegible]	000000	000000	035401	000377	176525	120127
-1024	[illegible]	[illegible]	[illegible]	000000	000000	035401	000000	000000	045047

0	040000	000000	037105	000000	000000	035401	000000	000000	045043
512	037776	000002	164242	000000	000000	035401	000377	176525	121005
1024	037770	000007	164032	000000	000000	035401	000777	165253	034771
512	037766	000155	110335	177400	021251	161466	000777	165253	034771
1024	037760	000525	103364	177000	052521	046037	000777	165253	034771
-512	037765	000155	110322	177400	021251	161406	000777	165253	034771
-1024	037770	000057	163716	000000	000000	034526	000777	165253	034771
-512	037775	000002	164112	000000	000000	034526	000377	176525	121665
-1024	040000	000000	036657	000000	000000	034526	000000	000000	046660

0	040000	000000	036657	000000	000000	034526	000000	000000	046600
512	037776	000002	164000	000000	000000	034526	000377	176525	122537
1024	037770	000052	163544	000000	000000	034526	000777	165253	036521
512	037766	000155	110044	177400	021251	160520	000777	165253	036521
1024	037760	000525	107226	177000	052521	045173	000777	165253	036521
-512	037766	000155	110031	177400	021251	160540	000777	165253	036521
-1024	037770	000057	163426	000000	000000	033653	000777	165253	036521
-512	037775	000002	153643	000000	000000	033653	000377	176525	123410
-1024	040000	000000	036426	000000	000000	033653	000000	000000	050337

Two Axis - Runs 1-5

TABLE 5.3 (cont.)

0	040000	000000	036426	000000	000000	033653	000000	000000	050333
512	037776	000002	163533	000000	000000	033653	000377	176525	124272
1024	037770	000052	163264	000000	000000	033653	000777	165753	040251
512	037766	000155	107562	177400	021251	157753	000777	165753	040251
1024	037760	000525	102765	177000	052521	044326	000777	165753	040251
-512	037765	000135	107547	177400	021251	157673	000777	165753	040251
-1024	037770	000052	163151	000000	000000	033002	000777	165253	040251
-512	037773	000000	163434	000000	000000	033002	000377	176525	125153
-1024	040000	000000	036400	000000	000000	033002	000000	000000	052070
0	040000	000000	036200	000000	000000	033002	000000	000000	052070
512	037776	000000	163273	000000	000000	033002	000377	176525	126025
1024	037770	000050	[illegible]	000000	000000	033002	000777	165753	042004
512	037766	000155	107372	177400	021251	157133	000777	165753	042004
1024	037760	000525	102465	177000	052521	043461	000777	165753	042004
-512	037756	000153	107236	177400	021251	157023	000777	165753	042004
-1024	037770	000052	162665	000000	000000	032130	000777	165753	042004
-512	037776	000002	163154	000000	000000	032130	000377	176525	126707
-1024	040000	000000	035754	000000	000000	032130	000000	000000	053625
0	040000	000000	035754	000000	000000	032130	000000	000000	053625
512	037770	000002	163027	000000	000000	032130	000377	176525	127557
1024	037770	000050	162525	000000	000000	032130	000777	165753	043537
512	037765	000155	107027	177400	021251	156233	000777	165753	043537
1024	037760	000525	102175	177000	052521	042616	000777	165753	043537
-512	037756	000155	106775	177400	021251	156153	000777	165753	043537
-1024	037770	000052	162415	000000	000000	031256	000777	165753	043537
-512	037776	000002	162706	000000	000000	031256	000377	176525	130437
-1024	040000	000000	035540	000000	000000	031256	000000	000000	055356
0	040000	000000	035540	000000	000000	031256	000000	000000	055356
512	037770	000002	162527	000000	000000	031256	000377	176525	131325
1024	037770	000050	162240	000000	000000	031256	000777	165753	045250
512	037760	000155	106522	177400	021251	155354	000777	165753	045256
1024	037760	000525	101677	177000	052521	041750	000777	165753	045256
-512	037760	000155	106507	177400	021251	155304	000777	165753	045256
-1024	037770	000052	162136	000000	000000	030403	000777	165753	045256
-512	037770	000002	162445	000000	000000	030403	000377	176525	132167
-1024	040000	000000	035314	000000	000000	030403	000000	000000	057100
0	040000	000000	035314	000000	000000	030403	000000	000000	057100
512	037770	000000	162226	000000	000000	030403	000377	176525	133035
1024	037770	000050	161777	000000	000000	030403	000777	165753	047000
512	037760	000150	106235	177400	021251	154512	000777	165753	047000
1024	037760	000525	101406	177000	052521	041101	000777	165753	047000
-512	037760	000155	106222	177400	021251	154432	000777	165753	047000
-1024	037770	000050	161664	000000	000000	027530	000777	165253	047000
-512	037770	000002	162203	000000	000000	027530	000377	176525	133717
-1024	040000	000000	035061	000000	000000	027530	000000	000000	060640

Two Axis - Runs 6-10

TABLE 5.3 (cont.)

1	1	1							
512	1024	1	10						
0	040000	000000	040000	000000	000000	040000	000000	000000	040000
512	040000	000000	040000	000000	000000	040000	000000	000000	040000
1024	040000	000000	040000	000000	000000	040000	000000	000000	040000
512	037776	000002	165200	000000	000000	040000	000377	176525	113753
1024	037770	000057	165024	000000	000000	040000	000777	165253	027743
512	037766	000155	111350	177400	021251	164056	000777	165253	027743
1024	037760	000525	104520	177000	052521	050415	000777	165253	027743
-512	037766	000155	111336	177400	021251	163776	000777	165253	027743
-1024	037770	000057	164711	000000	000000	037124	000777	165253	027743
-512	037776	000002	1[illegible]050	000000	000000	037124	000377	176525	114635
-1024	040000	000000	037555	000000	000000	037124	000000	000000	041544
-512	040000	000000	037555	000000	000000	037124	000000	000000	041544
-1024	040000	000000	037555	000000	000000	037124	000000	000000	041544

0	040000	000000	037555	000000	000000	037124	000000	000000	041544
512	040000	000000	037555	000000	000000	037124	000000	000000	041544
1024	040000	000000	037555	000000	000000	037124	000000	000000	041544
512	037776	000002	164742	000000	000000	037124	000377	176525	115512
1024	037770	000052	164554	000000	000000	037124	000777	165253	031500
512	037766	000155	111075	177400	021251	163207	000777	165253	031500
1024	037760	000525	104341	177000	052521	047553	000777	165253	031500
-512	037766	000155	111062	177400	021251	163130	000777	165253	031500
-1024	037770	000052	154437	000000	000000	036253	000777	165253	031500
-512	037776	000002	164605	000000	000000	036253	000377	176525	116372
-1024	040000	000000	037321	000000	000000	036253	000000	000000	043303
-512	040000	000000	037321	000000	000000	036253	000000	000000	043303
-1024	040000	000000	037321	000000	000000	036253	000000	000000	043303

0	040000	000000	037321	000000	000000	036253	000000	000000	043303
512	040000	000000	037321	000000	000000	036253	000000	000000	043303
1024	040000	000000	037321	000000	000000	036253	000000	000000	043303
512	037776	000002	164475	000000	000000	036253	000377	176525	117247
1024	037770	000052	164304	000000	000000	036253	000777	165253	033233
512	037766	000155	110617	177400	021251	162336	000777	165253	033233
1024	037760	000525	104055	177000	052521	046703	000777	165253	033233
-512	037766	000155	110604	177400	021251	162256	000777	165253	033233
-1024	037770	000052	154175	000000	000000	035401	000777	165253	033233
-512	037776	000002	164352	000000	000000	035401	000377	176525	120127
-1024	040000	000000	037105	000000	000000	035401	000000	000000	045043
-512	040000	000000	037105	000000	000000	035401	000000	000000	045043
-1024	040000	000000	037105	000000	000000	035401	000000	000000	045043

0	040000	000000	037105	000000	000000	035401	000000	000000	045043
512	040000	000000	037105	000000	000000	035401	000000	000000	045043
1024	040000	000000	037105	000000	000000	035401	000000	000000	045043
512	037776	000002	164242	000000	000000	035401	000377	176525	1210[illegible]
1024	037770	000052	154032	000000	000000	035401	000777	165253	034771
512	037766	000155	110335	177400	021251	161466	000777	165253	034771
1024	037760	000525	103564	177000	052521	045037	000777	165253	034771
-512	037766	000155	110322	177400	021251	151406	000777	165253	034771
-1024	037770	000052	153715	000000	000000	034526	000777	165253	034771
-512	037776	000002	164112	000000	000000	034526	000377	176525	121665
-1024	040000	000000	036657	000000	000000	034526	000000	000000	046600
-512	040000	000000	036657	000000	000000	034526	000000	000000	046600
-1024	040000	000000	036657	000000	000000	034526	000000	000000	046600

Three Axis - Runs 1-4

TABLE 5.3 (cont.)

0	040000	000000	036657	000000	000000	034526	000000	000000	046600
512	040000	000000	036657	000000	000000	034526	000000	000000	046600
1024	040000	000000	036657	000000	000000	034526	000000	000000	046600
512	037776	000002	164000	000000	000000	034526	000377	176525	122537
1024	037770	000052	163544	000000	000000	034526	000777	165253	036521
512	037766	000155	110044	177400	021251	160520	000777	165253	036521
1024	037760	000525	103256	177000	052521	045173	000777	165253	036521
-512	037766	000155	110031	177400	021251	160540	000777	165253	036521
-1024	037770	000052	163426	000000	000000	033653	000777	165253	036521
-512	037776	000002	163643	000000	000000	033653	000377	176525	123417
-1024	040000	000000	036426	000000	000000	033653	000000	000000	050333
-512	040000	000000	036426	000000	000000	033653	000000	000000	050337
-1024	040000	000000	036426	000000	000000	033653	000000	000000	050333

0	040000	000000	036426	000000	000000	033653	000000	000000	050333
512	040000	000000	036426	000000	000000	033653	000000	000000	050337
1024	040000	000000	036426	000000	000000	033653	000000	000000	050337
512	037776	000002	163533	000000	000000	033653	000377	176525	124277
1024	037770	000052	163264	000000	000000	033653	000777	165253	040251
512	037766	000155	107332	177400	021251	157753	000777	165253	040251
1024	037760	000525	102765	177000	052521	044326	000777	165253	040251
-512	037766	000155	107547	177400	021251	157673	000777	165253	040251
-1024	037770	000052	163131	000000	000000	033002	000777	165253	040251
-512	037776	000002	163404	000000	000000	033002	000377	176525	125157
-1024	040000	000000	036200	000000	000000	033002	000000	000000	052070
-512	040000	000000	036200	000000	000000	033002	000000	000000	052070
-1024	040000	000000	036200	000000	000000	033002	000000	000000	052070

0	040000	000000	036200	000000	000000	033002	000000	000000	052070
512	040000	000000	036200	000000	000000	033002	000000	000000	052070
1024	040000	000000	036200	000000	000000	033002	000000	000000	052070
512	037776	000002	163273	000000	000000	033002	000377	176525	126026
1024	037770	000052	163003	000000	000000	033002	000777	165253	042004
512	037766	000155	107272	177400	021251	157103	000777	165253	042004
1024	037760	000525	107465	177000	052521	043451	000777	165253	042004
-512	037766	000155	107256	177400	021251	157023	000777	165253	042004
-1024	037770	000052	157655	000000	000000	032130	000777	165253	042004
-512	037776	000002	163104	000000	000000	032130	000377	176525	126707
-1024	040000	000000	035754	000000	000000	032130	000000	000000	053625
-512	040000	000000	035754	000000	000000	032130	000000	000000	053625
-1024	040000	000000	035754	000000	000000	032130	000000	000000	053625

0	040000	000000	035754	000000	000000	032130	000000	000000	053625
512	040000	000000	035754	000000	000000	032130	000000	000000	053625
1024	040000	000000	035754	000000	000000	032130	000000	000000	053625
512	037776	000002	163027	000000	000000	032130	000377	176525	127557
1024	037770	000052	162525	000000	000000	032130	000777	165253	043537
512	037766	000155	107007	177400	021251	156233	000777	165253	043537
1024	037760	000525	102175	177000	052521	042516	000777	165253	043537
-512	037766	000155	106775	177400	021251	156153	000777	165253	043537
-1024	037770	000052	157415	000000	000000	031256	000777	165253	043537
-512	037776	000002	162706	000000	000000	031256	000377	176525	130437
-1024	040000	000000	035540	000000	000000	031256	000000	000000	055356
-512	040000	000000	035540	000000	000000	031256	000000	000000	055356
-1024	040000	000000	035540	000000	000000	031256	000000	000000	055356

Three Axis - Runs 5-8

TABLE 5.3 (cont.)

0	040000	000000	035540	000000	000000	031256	000000	000000	055355
512	040000	000000	035540	000000	000000	031256	000000	000000	055355
1024	040000	000000	035540	000000	000000	031256	000000	000000	055355
512	037776	000002	162572	000000	000000	031256	000377	176525	131305
1024	037770	000052	162246	000000	000000	031256	000777	165253	045255
512	037766	000155	106522	177400	021251	155364	000777	165253	045255
1024	037760	000525	101677	177000	052521	041750	000777	165253	045255
-512	037766	000155	106507	177400	021251	155304	000777	165253	045255
-1024	037770	000052	152135	000000	000000	030403	000777	165253	045255
-512	037776	000002	162446	000000	000000	030403	000377	176525	132167
-1024	040000	000000	035314	000000	000000	030403	000000	000000	057106
-512	040000	000000	035314	000000	000000	030403	000000	000000	057106
-1024	040000	000000	035314	000000	000000	030403	000000	000000	057106

0	040000	000000	035314	000000	000000	030403	000000	000000	05710[illegible]
512	040000	000000	035314	000000	000000	030403	000000	000000	05710[illegible]
1024	040000	000000	035314	000000	000000	030403	000000	000000	05710[illegible]
512	037776	000002	162336	000000	000000	030403	000377	176525	13303[illegible]
1024	037770	000052	161777	000000	000000	030403	000777	165253	0470[illegible]
512	037766	000155	106235	177400	021251	154512	000777	165253	0470[illegible]
1024	037760	000525	101406	177000	052521	041101	000777	165253	0470[illegible]
-512	037766	000155	106222	177400	021251	154432	000777	165253	0470[illegible]
-1024	037770	000052	161664	000000	000000	027530	000777	165253	0470[illegible]
-512	037776	000002	162203	000000	000000	027530	000377	176525	133717
-1024	040000	000000	035061	000000	000000	027530	000000	000000	06064[illegible]
-512	040000	000000	035061	000000	000000	027530	000000	000000	06064[illegible]
-1024	040000	000000	035061	000000	000000	027530	000000	000000	06064[illegible]

Three Axis - Runs 9 and 10

TABLE 5.3 (cont.)

```
@FOR,IS JUPITR
CYCLE 000 COMPILED BY 1201 0057D ON 27 FEB 70 AT 10:48:52.

   MAIN PROGRAM

   STORAGE USED: CODE(1) 000015; DATA(0) 000042; BLANK COMMON(2) 000003

    0001      000275 100L       0001   000000F 12L       0001    00011C 150C      0001   000172 156C      0001   000134 164C
    0001     000141 171C        0001   000077 190L       0001    000277 199L      0001   000154 200C      0001   000273 270L
    0001     000425 290L        0001   000375 299L       0001    000471 300L      0001   000365 302G      0001   000403 313C
    0001     000445 326C        0001   000454 332G       0001    000463 336C      0001   000502 350G      0001   000473 350L
    0001     000674 435G        0001   000715 446C       0001    000756 461C      0001   000763 465G      0001   000770 471G
    0001     000646 490L        0001   000706 499L       0001    000702 500L      0001   000796 590L      0001   000774 600L
    0001     001000 650L        0000   000025 910OF      0000    000026 920OF     0001   000241 9GL       0000   000027 990OF
    0001     000023 910IF       0000   000032 920OF      0000  I 000004 C11       0000 I 000007 C12      0000 I 000012 C13
    0000 I 000001 DX            0000 I 000002 DY         0000  I 000003 D2        0000 I 000015 IN       0000 I 000021 IS
    0002 I 000002 J             0000 I 000024 K          0000  I 000023 L0        0000 I 000072 LIM      0000 I 000016 LIM1
    0000 I 000017 LIM2          0000 I 000020 LIM3       0000  I 000000 OUT       0002 I 000000 PR       0002 I 000001 Z
```

Main Program

TABLE 5.4

Program Listing

```
  200 FORMAT(1H1)
 8100 FORMAT(/)
      OUT = 6
      IN = 5
   10 READ(IN,8200) DX,DY,DZ
      IF(DX.EQ.-1) STOP
      WRITE(OUT,8200) DX,DY,DZ
      READ (IN ,8200) PR,LIM1,LIM2,LIM3
      WRITE(OUT,8200) PR,LIM1,LIM2,LIM3
       READ(IN,8100) C11
       READ(IN,8100) C12
      READ(IN,8100) C13
      DO 700 LIM=1,LIM3
      LB=0
      I=1
      WRITE(OUT,8000) LB,C11,C12,C13
      DO 350 K=1,LIM2
      LB=K*7
      IF(DX.EQ.0) GOTO 199
      DO 100 J=1,LIM1
      CALL ITER(C12,C13)
      IF((LIM1.EQ.1) .AND. (MOD(K,PR).EQ.0)) GOTO 90
      IF(MOD(J,PR).NE.0) GOTO 100
      LB=J*7
   90 WRITE(OUT,8000) LB,C11,C12,C13
  100 CONTINUE
  199 IF(DY.EQ.0) GOTO 299
      DO 200 J=1,LIM1
      CALL ITER(C13,C11)
      IF((LIM1.EQ.1) .AND. (MOD(K,PR).EQ.0)) GOTO 190
      IF(MOD(J,PR).NE.0) GOTO 200
      LB=J*7
  190 WRITE(OUT,8000) LB,C11,C12,C13
  200 CONTINUE
  299 IF(DZ.EQ.0) GOTO 350
      DO 300 J=1,LIM1
      CALL ITER(C11,C12)
      IF((LIM1.EQ.1) .AND. (MOD(K,PR).EQ.0)) GOTO 290
      IF(MOD(J,PR).NE.0) GOTO 300
      LB=J*7
  290 WRITE(OUT,8000) LB,C11,C12,C13
  300 CONTINUE
  350 CONTINUE
      I=-1
      DO 650 K=1,LIM2
      LB=K*7
      IF(DZ.EQ.0) GOTO 399
      DO 400 J=1,LIM1
      CALL ITER(C11,C12)
      IF((LIM1.EQ.1) .AND. (MOD(K,PR).EQ.0)) GOTO 390
      IF(MOD(J,PR).NE.0) GOTO 400
      LB=J*7
  390 WRITE(OUT,8000) LB,C11,C12,C13
  400 CONTINUE
```

Main Program (cont.)

TABLE 5.4 (cont.)

```
00407   490   IF(IY.EQ.0) GOTO 499
00411         DO 500 J=1,LIM1
00414         CALL ITER(C13,C11)
00415         IF((LIM1.EQ.1) .AND. (MOD(K,P ).EQ.0)) GOTO 490
00417         IF(MOD(J,PR).NE.0) GOTO 500
00421         LB=J*2
00422   490   WRITE(OUT,9000) LB,C11,C12,C13
00441   500   CONTINUE
00443   499   IF(IX.EQ.0) GOTO 650
00445         DO 600 J=1,LIM1
00450         CALL ITER(C12,C13)
00451         IF((LIM1.EQ.1) .AND. (MOD(K,PP).EQ.0)) GOTO 590
00453         IF(MOD(J,PR).NE.0) GOTO 600
00455         LB=J*2
00456   590   WRITE(OUT,9000) LB,C11,C12,C13
00475   600   CONTINUE
00477   650   CONTINUE
00501         WRITE(OUT,9200)
00503   700   CONTINUE
00505         GOTO 10
00505         END

      END OF COMPILATION:        NO DIAGNOSTICS.
```

Main Program (cont.)

TABLE 5.4 (cont.)

```
@FOR,IS MAR?
CYCLE ?I? COMPILED ?Y 12 1 CCS70 ON ?7 FEB 70 AT 1?:4?:5?.

   SU?ROUTINE ITER         EN?RY POINT CCC312

   STORAGE USED: CODE(1) 000322, DATA(0) 000061; BLANK COMMON(2) 000003

   EXTERNAL REFERENCES (BLOCK, NAME)

    0003   SCALE
    0004   NERR3$

   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0000    000035 9000F     0000 I 000010 IEC      0000 I 000005 IES      0000 I 000026 IFC      0000 I 000023 IFS
    0000 I 000025 IFC        0000 I 000022 IGS      0000 I 000024 IHC      0000 I 000021 IHS      0000 I 000032 IJC
    0000 I 000027 IJS        0000 I 000033 IKC      0000 I 000030 IKS      0000 I 000034 ILC      0000 I 000031 ILS
    0000    000040 INJP$     0000 I 000011 IRC      0000 I 000006 IRS      0000 I 000012 IYC      0000 I 000007 IYS
    0002 I 000002 J          0000 I 000016 JEC      0000 I 000013 JES      0000 I 000017 JRC      0000 I 000014 JFS
    0000 I 000020 JYC        0000 I 000015 JYS      0000 I 000002 MM       0000 I 000003 MN       0000 I 000001 PE
    0000 I 000004 NM         0000 I 000000 OUT      0002 I 000000 PR       0002 I 000001 Z
```

Subroutine ITER (performs one iteration on two direction cosines)

TABLE 5.4 (cont.)

```
      SUBROUTINE ITER(C,CY)
      INTEGER PR,Z,J,C,CY,OUT
      DIMENSION C(3),CY(3)
      COMMON PR,Z,J
      DATA MT/32768/,MB/[illegible]/,MN/[illegible]/,MD/65534/
      DATA OUT/6/
 1000 FORMAT(I10,6(4X,O6))
      IES=C(3)
      IRS=C(2)
      IYS=C(1)
      IEC=CY(3)
      IRC=CY(2)
      IYC=CY(1)
      JES=IES/MD
      JRS=IRS/MD
      JYS=IYS/MD
      JEC=IEC/MD
      JRC=IRC/MD
      JYC=IYC/MD
      IHS=IES-IYS-JRS
      IGS=IRS-MN*JYS
      IFS=IYS-MN*JYS
      IHC=IEC-IYC-JRC
      IGC=IRC-MN*JYC
      IFC=IYC-MN*JYC
      IJS=2*IRC+JEC
      IKS=2*IYS
      ILS=MN*JYS
      IJC=2*IRS+JES
      IKC=2*IYS
      ILC=MN*JYS
      CALL SCALE(IFS,IGS,IHS)
      CALL SCALE(IFC,IGC,IHC)
      CALL SCALE(ILS,IKS,IJS)
      CALL SCALE(ILC,IKC,IJC)
      IES=IHS+IJS*Z
      IRS=IGS+IKS*Z
      IYS=IFS+ILS*Z
      IEC=IHC-IJC*Z
      IRC=IGC-IKC*Z
      IYC=IFC-ILC*Z
      CALL SCALE(IYS,IRS,IES)
      CALL SCALE(IYC,IRC,IEC)
      JRS=IRS/MD
      JYS=IYS/MD
      JRC=IRC/MD
      JYC=IYC/MD
      C(3)=IES-IYS-JRS
      C(2)=IRS-MN*JYS
      C(1)=IYS-MN*JYS
      CY(3)=IEC-IYC-JRC
      CY(2)=IRC-MN*JYC
      CY(1)=IYC-MN*JYC
      CALL SCALE(C(1),C(2),C(3))
      CALL SCALE(CY(1),CY(2),CY(3))
      RETURN
      END

        END OF COMPILATION:        NO DIAGNOSTICS.
```

Subroutine ITER (cont.)

TABLE 5.4 (cont.)

```
RFOR,IS NEPTUN
CYCLE 000 COMPILED BY 1201 0057D ON 27 FEB 70 AT 10:49:59.

   SUBROUTINE SCALE       ENTRY POINT 000125

   STORAGE USED: CODE(1) 000166; DATA(0) 000007; BLANK COMMON(2) 000000

   EXTERNAL REFERENCES (BLOCK, NAME)

    0003   NERR3$

   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001   000001 1CL       0001   000015 10CL      0001   000030 20OL      0001   000034 21CL      0001   [illegible]
    0001   000063 40OL      0001   000067 41OL      0001 - 000100 50OL      0001   000110 60OL      0000   [illegible]
    0000 I 000001 K         0000 I 000000 MM
```

Subroutine SCALE

(scales one direction cosine by operating on each of the three integer parts)

TABLE 5.4 (cont.)

```
      SUBROUTINE SCALE(I1,I2,I3)
      DATA MM/65536/
      K=0
   10 IF(I3.LT.MM) GOTO 100
      I3=I3-MM
      K=K+1
      GO TO 10
  100 IF( I3.GE.0) GOTO 200
      I3=I3+MM
      K=K-1
      GOTO 100
  200 I2=I2+K
      K=0
  210 IF(I2.LT.MM) GOTO 300
      I2=I2-MM
      K=K+1
      GOTO 210
  300 IF(I2.GE.0) GOTO 400
      I2=I2+MM
      K=K-1
      GOTO 300
  400 I1=I1+K
      K=0
  410 IF(I1.LT.MM) GOTO 500
      I1=I1-MM
      GOTO 410
  500 IF(I1.GE.0) GOTO 600
      I1=I1+MM
      GOTO 500
  600 I1=IABS(I1)
      I2=IABS(I2)
      I3=IABS(I3)
      RETURN
```

Subroutine SCALE (cont.)

TABLE 5.4 (cont.)

APPENDIX A

SMOOTH PULSE SEQUENCES
(SRRC-RR-68-48)

SMOOTH PULSE SEQUENCES

A. J. Lincoln, M. Cohn and S. Even
Sperry Rand Research Center, Sudbury, Massachusetts

INTRODUCTION

The necessity sometimes arises in digital systems to generate external pulse rates whose frequency is related to the internal clock frequency by some proper rational fraction while still maintaining synchronism with the internal clock source. In many such cases, it is a requirement that pulses of the generated frequency be as uniformly spaced as possible within the constraint of synchronism. We define such pulse sequences as smooth pulse sequences in contrast with uniform sequences in which each pulse is separated from its predecessor by a fixed interval. The smooth sequences possess a number of interesting properties, both in their structure and in the ways they can be generated. Some of these properties will be treated in the following discussion.

Although it has not, to the authors' knowledge, been previously identified as such, a method has been in use for some time for generating smooth sequences. This is the constant multiplier used in digital differential analyzers (DDA's). Another technique often used for forming synchronous frequency multiples uses the "rate multiplier," and involves detecting the one-to-zero transitions of the stages of a binary counter chain. The rate multiplier technique, although capable of forming relatively smooth sequences, does not form a true smooth sequence, as can be easily demonstrated.

Other methods for generating smooth sequences can be described. Our interest here, however, is to provide a formal foundation for the subject and to prove several important properties of smooth pulse sequences.

PROPERTIES OF SMOOTH SEQUENCES

Choosing an arbitrary reference point in time, and assuming, for convenience, a unit-period clock, we define two types of pulse sequences: Uniform Sequences and Smooth Sequences. Each can be described by a doubly-infinite sequence of real numbers, the i^{th} number giving the arrival time, as counted by the clock, of the i^{th} pulse. Since our reference-time zero and the position of the index zero can be chosen arbitrarily and independently, we wish to consider equivalent those sequences which differ only by a shift in time or by reindexing. In other words, only the "pattern" of a sequence is important. Accordingly, we first make the following definition:

Definition 1:

Two arrival-time sequences $\{a_i\}$ and $\{b_i\}$ are equivalent if for some integers t and τ,

$$a_i = b_{i+\tau} + t \quad , \qquad -\infty \le i \le \infty \quad .$$

A uniform pulse sequence is an ideally smooth sequence with rate p/q times that of the clock. It consists of p uniformly spaced pulses during every q clock periods.

Definition 2:

A Uniform Pulse Sequence of rate p/q has pulses at times $\{u_i\}$, where $u_i = i\frac{q}{p}$, $-\infty \le i \le \infty$.

Unless q/p is an integer, the pulses of the uniform sequence do not all coincide in time with clock pulses, so the problem of synchronous pulse rate generation cannot generally be solved by a uniform sequence. The concept, though, is crucial in defining and characterizing smooth sequences, which, heuristically, are the best synchronous approximation to uniform sequences.

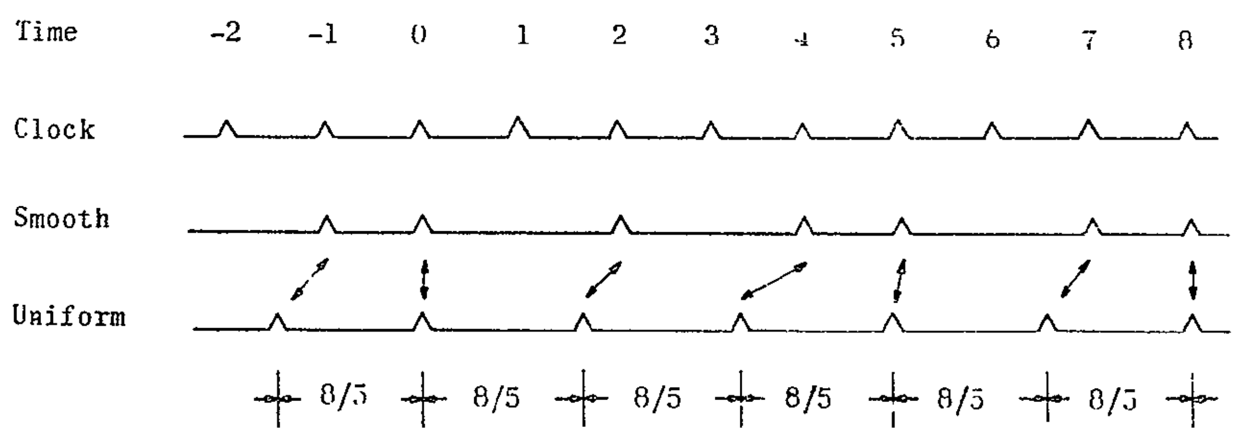
Definition 3:

A Smooth Pulse Sequence of rate p/q has pulses only at integral clock times $\{s_i\}$, $-\infty \le i \le \infty$; moreover, these pulses can be put into one-to-one correspondence with the pulses of the uniform sequence in such a way that

$\max_i(s_i - u_i) - \min_i(s_i - u_i) < 1$.

To construct a concrete example of a smooth pulse sequence for the fraction 5/8, we place pulses at those clock times which coincide with or immediately follow pulses of the uniform sequence.

Example 1



Here the arrival times of the smooth sequence are $\{\ldots, -1, 0, 2, 4, 5, 7, 8, \ldots\}$. If the pulse pattern between, say, time 0 through time 7 is repeated indefinitely, an infinite, smooth sequence of rate 5/8 results. Since this sequence was constructed to be periodic, it suffices to observe the smoothness of a single period. Under the natural correspondence (shown by arrows), the deviations between the smooth and the uniform arrival times range between 0 and 4/5 , satisfying Definition 3.

The last condition in Definition 3 bounds the range of time deviations between pulses of a smooth sequence and their correspondents in a uniform sequence. We next observe that the bound can be sharpened and that within the equivalence relation the deviations themselves can be bounded.

Since the s_i are integers, and the u_i are integral multiples of q/p , the differences (s_i-u_i) between corresponding pulse times must be integral multiples of $1/p$, hence

Lemma 1: A smooth sequence satisfies the inequality

$$\max_i(s_i-u_i) - \min_i(s_i-u_i) \le \frac{p-1}{p} \quad .$$

Now suppose that i_0 is an index where the difference (s_i-u_i) achieves its minimum;† that is,

$$\min_i(s_i-u_i) = (s_{i_o}-u_{i_o}) \quad .$$

By an integral shift in time and a reindexing, we form the new sequence $\{s_i'\}$, where

$$s_i' = s_{i+i_o} - s_{i_o} \quad .$$

† To be rigorous here we should note that Lemma 1 and its preceding discussion imply that the differences (s_i-u_i) take on at most p distinct values, and that these values are bounded below by $s_j-u_j-(p-1)/p$, where j is an index chosen arbitrarily. Therefore, the sequence has a greatest lower bound, which is $s_{i_o}-u_{i_o}$ for some i_0 .

The minimum deviation of $\{s_i'\}$ from the uniform sequence is given by

$$\min_i (s_i' - u_i) = \min_i (s_{i+i_0} - s_{i_0} - u_i)$$

$$= \min_i (s_{i+i_0} - u_i - u_{i_0} - s_{i_0} + u_{i_0})$$

$$= \min_i (s_{i+i_0} - u_{i+i_0}) - (s_{i_0} - u_{i_0}) \quad .$$

But by definition of i_0 this quantity vanishes, so that all deviations of $\{s_i'\}$ from the uniform must be nonnegative. Invoking Lemma 1, we can now state:

Lemma 2: Within equivalence, every smooth sequence satisfies

$$0 \le (s_i - u_i) \le \frac{p-1}{p} \quad , \quad -\infty \le i \le \infty \quad .$$

The sequence in Example 1 was designed to satisfy this bound without an equivalence transformation.

We are now able to prove the fundamental theorem of smooth sequences.

Theorem 1: A smooth sequence of rate p/q is unique to within equivalence.

Proof: Given any two smooth sequences of rate p/q, consider their equivalent forms complying with Lemma 2. The i^{th} pulses of both smooth sequences must coincide with clock pulses, and must coincide with, or lag by less than a clock period, the same correspondent, u_i, in the uniform sequence of rate p/q. Therefore, the i^{th} pulses in the two smooth sequences must coincide with the same clock pulse, hence with each other. QED

This is an extremely strong result, following, as it does, only from the definitions of uniform and smooth sequences and the notion of equivalence.

So far we have used arrival times to describe pulse sequences. It is convenient now to introduce a dual description, the cumulative count. If $A(t)$ is the number of pulses in the sequence $\{a_i\}$ which have arrived by time t, we define the "count sequence" $\{\alpha(t)\}$, where

$$\alpha(t) = A(t) - A(a_0) \quad .$$

Thus $\alpha(t)$ is an integer-valued function of time, satisfying

Lemma 3: If $\{a_i\}$ and $\{\alpha(j)\}$ are, respectively, the arrival time sequence and the count sequence for a synchronous pulse sequence,

$$\alpha(j) = i \quad \text{for} \quad a_i \leq j < a_{i+1}$$

$$a_{\alpha(j)} \leq j < a_{\gamma(j)+1} \quad .$$

As done in Lemma 3, we will always use the lowercase Greek equivalent of the arrival-time character for the corresponding count sequence. Also, by analogy with the notation $\{u_i\}$ for the arrival times of pulses in the uniform sequence, we make the following definition.

Definition 4. A uniform sequence of rate p/q has a (rational) count sequence $\{\upsilon(j)\}$, where

$$\upsilon(j) = j\frac{p}{q} \quad , \qquad -\infty \leq j \leq \infty \quad .$$

We are now ready to prove the important relationship between the arrival time sequence and the count sequence.

Theorem 2 (Duality Theorem):

If $\{a_i\}$ and $\{\alpha(j)\}$ describe a synchronous pulse sequence of rate $\frac{p}{q}$,

$$\max_i(a_i-u_i) - \min_i(a_i-u_i) \le \frac{p-1}{p}$$

if and only if

$$\max_j(\upsilon(j) - \alpha(j)) - \min_j(\upsilon(j) - \alpha(j)) \le \frac{q-1}{q}$$

Proof: To prove the "if" part, observe that for any a_i ,

$$\frac{q}{p}\min_j(\upsilon(j) - \alpha(j)) = \frac{q}{p}\min_j\left(\frac{p}{q}j - \alpha(j)\right) \le \frac{q}{p}\left(\frac{p}{q}a_i - \alpha(a_i)\right) = (a_i - u_i) \ .$$

Also, since

$$\frac{q}{p}\max_j(\upsilon(j) - \alpha(j)) \ge \frac{q}{p}\left(\frac{p}{q}(a_i-1) - \alpha(a_i-1)\right) = \left(a_i - 1 - \frac{q}{p}(i-1)\right) \ ,$$

$$\left(1 - \frac{q}{p}\right) + \frac{q}{p}\max_j(\upsilon(j) - \alpha(j)) \ge (a_i-u_i) \ .$$

From these two results,

$$\left(1 - \frac{q}{p}\right) + \frac{q}{p}\max_j(\upsilon(j) - \alpha(j)) \ge \max_i(a_i-u_i) \ ,$$

$$\frac{q}{p}\min_j(\upsilon(j) - \alpha(j)) \le \min_i(a_i-u_i) \ .$$

By subtraction and hypothesis,

$$\left(1 - \frac{q}{p}\right) + \frac{q}{p}\left(\frac{q-1}{q}\right) = \frac{p-1}{p} \ge \max_i(a_i-u_i) - \min_i(a_i-u_i)$$

To prove the "only if" part, observe that for any clock time j, there is a pair of successive arrival times such that $a_i \leq j < a_{i+1}-1$, so that $\alpha(j) = i$. Therefore,

$$\frac{p}{q}\, a_i - i \leq \frac{p}{q}\, j - \alpha(j) \leq \frac{p}{q}\,(a_{i+1}-1) - i \quad ,$$

$$\frac{p}{q}\,(a_i-u_i) \leq \upsilon(j) - \alpha(j) \leq \frac{p}{q}\,(a_{i+1}-u_{i+1}) - \frac{p}{q} + 1 \quad .$$

But then

$$\left(1 - \frac{p}{q}\right) + \frac{p}{q} \max_i (a_{i+1}-u_{i+1}) \geq \max_j (\upsilon(j) - \alpha(j)) \quad ,$$

$$\frac{p}{q} \min_i (a_i - u_i) \leq \min_j (\upsilon(j) - \alpha(j)) \quad ;$$

again by subtraction and hypothesis,

$$\left(1 - \frac{p}{q}\right) + \frac{p}{q}\left(\frac{p-1}{p}\right) = \frac{q-1}{q} \geq \max_j (\upsilon(j) - \alpha(j)) - \min_j (\upsilon(j) - \alpha(j)) \;.$$

QED

An obvious consequence of Theorem 2 and Definition 3 is

Corollary 2.1: A synchronous pulse sequence described by the count sequence $\{\alpha(j)\}$ is smooth if and only if

$$\max_j (\upsilon(j) - \alpha(j)) - \min_j (\upsilon(j) - \alpha(j)) \leq \frac{q-1}{q} \quad .$$

In the discussion which follows, several of the results have analogous statements and analogous proofs in terms of arrival times as well as count sequences. In such cases, both statements, but only one proof, will be given.

Lemma 3: If, for all i, $a_{i+c} = a_i + d$, then for any integer k,

$$a_{i+kc} = a_i + kd .$$

Lemma 3′: If, for all j, $\alpha(j+d) = \alpha(j) + c$, then for any integer k,

$$\alpha(j + kd) = \alpha(j) + kc.$$

Proofs: Both proofs are simple inductions on k.

Lemma 4: If, for all i, $a_{i+c_1} = a_i + d_1$

and $a_{i+c_2} = a_i + d_2$

then $c_1/c_2 = d_1/d_2$

Lemma 4′. If, for all j, $\alpha(j+d_1) = \alpha(j) + c_1$

and $\alpha(j+d_2) = \alpha(j) + c_2$.

then $c_1/c_2 = d_1/d_2$.

Proofs: Consider $\alpha(j+d_1d_2)$; applying Lemma 3′ twice,

$$\alpha(j) + d_1c_2 = \alpha(j) + d_2 + c_1$$

for all j, so that

$$c_1/c_2 = d_1/d_2 .$$

Lemma 5: The arrival time sequence $\{s_i\}$ for a smooth sequence of rate $\frac{p'}{q'} = \frac{rp}{rq}$, p and q relatively prime, satisfies $s_{i+p} = s_i + q$ for all i.

Lemma 5′: The count sequence $\{\sigma(j)\}$ for a smooth sequence of rate $\frac{p'}{q'} = \frac{rp}{rq}$, p and q relatively prime, satisfies $\sigma(j+q) = \sigma(j) + p$ for all j.

Proofs: $\sigma(j+q) - \sigma(j) = \sigma(j+q) - \upsilon(j+q) - \sigma(j) + \upsilon(j) + q\frac{p'}{q'}$

$$= \{\sigma(j+q) - \upsilon(j+q) - \sigma(j) + \upsilon(j)\} + p .$$

The difference on the left-hand side of this equation is clearly an integer, while the bracketed quantity on the right is less than unity (by Cor. 2.1). Therefore, the bracketed quantity must be zero, proving the lemma.

We now define two new sequences, the first differences of the arrival times and the count sequence. The first, called the "gap" sequence, consists of the clock-time intervals between pulses. The second, called the "binary" sequence, is unity at those clock times when pulses are present, and is zero at all others. Thus, it depicts the pattern of the pulse sequence.

Definition 6: The gap sequence is given by

$$g_i = a_i - a_{i-1}, \quad -\infty \leq i \leq \infty .$$

Definition 7: The binary sequence is given by

$$\beta(j) = \alpha(j) - \alpha(j-1), \quad -\infty \leq j \leq \infty .$$

Theorem 3: A smooth pulse sequence is periodic. If its rate is $\frac{p'}{q'} = \frac{rp}{rq}$, p and q relatively prime, then the period is q clock times during which p pulses appear.

Proof:
$$\begin{aligned}\beta(j+q) - \beta(j) &= \sigma(j+q) - \sigma(j+q-1) - \sigma(j) + \sigma(j-1)\\ &= \sigma(j+q) - \sigma(j) - \sigma(j+q-1) + \sigma(j-1)\\ &= p - p = 0 .\end{aligned}$$

Lemma 5′ makes explicit the number of pulses in each period, and Lemma 4′ proves that if there were a shorter period than q, then p and q could not be relatively prime.

QED

These last results simplify the investigation of smooth sequences in two ways. First, only rates in reduced form, that is, with p and q relatively prime, need be considered. Second, a single period suffices to describe any smooth sequence. This permits the use of the following convenient notations for a smooth sequence of rate p/q :

(i) Arrival times: $(s_0, s_1, \ldots, s_{p-1})$;

(ii) Gaps $(g_0, g_1, \ldots, g_{p-1})$,

$$\left.\begin{aligned} &\text{where } g_i = s_i - s_{i-1} \\ &\text{and } \sum_{i=0}^{p-1} g_i = q \end{aligned}\right\} \text{indices modulo } p \text{ ;}$$

(iii) Counts $(\sigma(0), \sigma(1), \ldots, \sigma(q-1))$;

(iv) Binary $(\beta(0), \beta(1), \ldots, \beta(q-1))$,

$$\left.\begin{aligned} &\text{where } \beta(j) = \sigma(j) - \sigma(j-1) \\ &\text{and } \sum_{j=0}^{q-1} \beta(j) = p \end{aligned}\right\} \text{indices modulo } q \text{ .}$$

Finally, we consider the difference-sequences between smooth and uniform sequences.

Lemma 6. The sequence $(s_i - u_i)$ has period q .

Lemma 6': The sequence $(\upsilon(j) - \sigma(j))$ has period q .

Proofs: Since $u_i = \frac{q}{p} i$, $u_{i+p} = u_i + q$ for all i . By Lemma 5, $\{s_i\}$ satisfies the same recursion, so the difference $(s_i - u_i)$ has a period of q clock times (or p pulses). But since p and q are relatively prime, u_i and $(s_i - u_i)$ are integral only when i is a multiple of p . Therefore $(s_i - u_i)$ has no period less than q .

With this knowledge and the observation that all $(s_i - u_i)$ are proper fractions with denominator p, and all $(\upsilon(j)-\sigma(j))$ are proper fractions with denominator q, we can complete the refinement, begun in Lemmas 1 and 2, of the characterization of smooth sequences.

<u>Theorem 4:</u> A smooth sequence of rate $\frac{p}{q}$ satisfies

$$\max_i (s_i - u_i) - \min_i (s_i - u_i) = \frac{p-1}{p} ;$$

$$\max_j (\upsilon(j) - \sigma(j)) - \min_j (\upsilon(j) - \sigma(j)) = \frac{q-1}{q} .$$

<u>Proof:</u> Since the difference $(s_i - u_i)$ periodically assumes p different values, all of which are proper fractions with denominator p,

$$\max_i (s_i - u_i) - \min_i (s_i - u_i) \geq \frac{p-1}{p}$$

This inequality combined with Definition 3 proves the theorem for the arrival times. A similar argument can be made for the count sequence.

QED

SUMMARY

Starting from a definition of smooth sequences as synchronous pulse trains with bounded range of deviation from an ideal uniform rate, we have seen that the pattern of such a sequence is unique and periodic, and that the deviations from the ideal are quantized and less than unity. Dual descriptions have been given in terms of arrival times and cumulative counts, as well as in terms of their first differences. The deviations of smooth sequences from their uniform counterparts have been characterized.

In closing we note that from the uniqueness of these sequences, which holds even under our rather broad definition of smoothness, it follows that smooth sequences minimize all reasonable measures of deviation from uniform spacing.