N68-85123

TRW
SYSTEMS GROUP
ONE SPACE PARK • REDONDO BEACH, CALIFORNIA

303

05128-6001-R000

BODY-FIXED, THREE-AXIS REFERENCE SYSTEM STUDY

PHASE II FINAL REPORT

15 DECEMBER 1966

Prepared for

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
George C. Marshall Space Flight Center
Huntsville, Alabama

Under

Contract No. NAS8-20209

Approved CW Sarture

C. W. Sarture, Manager
Guidance and Navigation Laboratory

TRW SYSTEMS
AN OPERATING GROUP OF TRW INC.
ONE SPACE PARK • REDONDO BEACH, CALIFORNIA

CONTENTS

CONTENTS (Continued)

CONTENTS (Continued)

CONTENTS (Continued)

ILLUSTRATIONS

ILLUSTRATIONS (Continued)

ILLUSTRATIONS (Continued)

ILLUSTRATIONS (Continued)

TABLES

*See Confidential Annex

TABLES (Continued)

ACKNOWLEDGMENT

The study results documented herein were obtained through the combined efforts of several members of the technical staff at TRW Systems. It is therefore appropriate to acknowledge both their valuable contributions and their areas of responsibility.

G.T. Aldrich	Study Direction
K.L. Baker and J.C. Wilcox	Airborne Computer Analyses, Alignment and Software Studies
M.J. Laubenstein and J.D. Paul	Performance Analysis
L.F. Munn and W.H. Musser	Airborne Computer Implementation
R.C. Rountree	Instrument Analysis

Particularly important to the successful dispatch of the Phase II study were the assistance and guidance provided by NASA/MSFC personnel. Specifically, the efforts of F.P. Daniel, Contracting Officer's Representative, are cited with grateful appreciation.

1. INTRODUCTION

1.1 PURPOSE AND SCOPE

This Phase II Body-Fixed, Three-Axis Reference System Study Final Report is the result of a follow-on contract awarded by NASA/MSFC to TRW Systems in April 1966, under Contract No. NAS8-20209. The purpose of the Phase II Study was to refine and supplement the strapdown inertial reference assembly (SIRA), software, airborne computer, and alignment analyses performed during Phase I of the same contract.

Specifically, the Phase I results in many areas were those which could be achieved using simplifying assumptions (linearization) and hand analyses. It was the purpose of the Phase II studies to investigate the validity of the Phase I results by means of more sophisticated computer simulations, in which no simplifying assumptions were made, for each of the areas analyzed. The Phase II analyses were directed, therefore, at confirming the analytical feasibility asserted by the Phase I results.

The scope of the Phase II study was delineated and implemented as shown in Figure 1-1 and explained below.

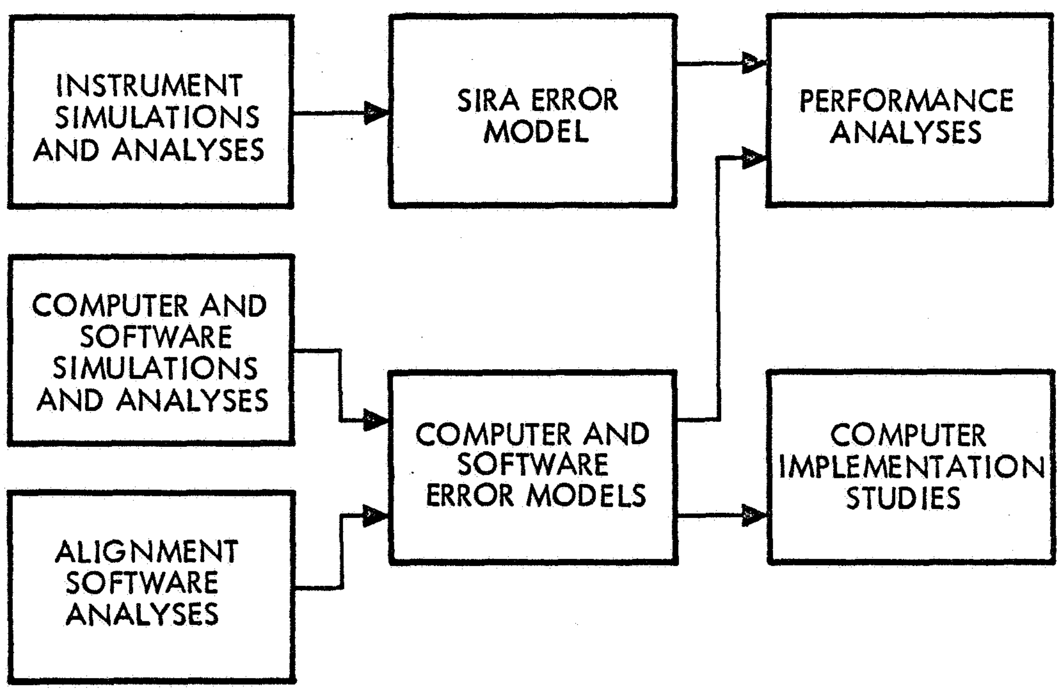


Figure 1-1. Phase II Study Implementation

1.1.1 Instrument Analyses

The full nonlinear equations for the single-axis platforms (SAP's) as derived in Phase I were simulated on both digital and analog computers for the purpose of verifying the Phase I results and investigating unmodeled error sources, if any. The PIGA's were not simulated.

1.1.2 Airborne Computer Analyses

The attitude reference and velocity transformation software investigated during Phase I was programmed on interpretive routines which simulated both digital differential analyzer (DDA) and general purpose (GP) airborne computers. The purpose of these simulations was to determine the effects of various word length and speed airborne digital computers on attitude and velocity error performance.

1.1.3 Alignment Analyses

Four methods of software implementation for initial SIRA alignment were simulated on a digital computer. The three methods completely simulated were constant gain, white-noise Kalman and variable gain filters. A fourth method, narrow-band Kalman Filter, was simulated to sufficient depth to allow evaluation of the filter convergence performance only.

1.1.4 SIRA, Computer and Software Error Models

Based upon the results of the simulations and analyses outlined in Paragraphs 1.1.1 through 1.1.3, refined models for the SIRA, software, and three airborne computers were formulated. The purpose of the refined error model formulation was to provide the input to system performance analyses.

1.1.5 System Performance Analyses

The error models developed as outlined in Paragraph 1.1.4 were input to the TRW generalized error analysis program in order to obtain the error performance of two SIRA configurations (differing instrument orientations) in combination with three different airborne computers (a DDA and two GP's) operating on an injection trajectory.

1.1.6 Airborne Computer Implementations

The software as refined in Phase II, together with the airborne computer errors as determined from simulation, were input to a study of possible airborne computer hardware implementations. Three possible implementations and their hardware implications were considered. These implementations were a GP, a DDA, and a hybrid DDA-GP.

1.2 FINAL REPORT ORGANIZATION

The Phase II Final Report is divided into seven main sections and two appendices. This division occurs naturally due to the method of study implementation shown in Figure 1-1.

Section 2 summarizes the conclusions and key results obtained in each analysis area. The study details leading to the results are documented in Sections 3 through 7 and in Appendices I and II.

Section 3 presents the system performance analyses and error models for various SIRA-airborne computer combinations operating on an injection trajectory. Section 4 details the simulation techniques and results from the instrument analyses. Section 5 shows the interpretive simulations and results of the airborne computer analyses. Section 6 presents the various methods of alignment software implementation which were simulated and evaluated. Section 7 discusses several possible airborne computer implementations and the hardware implications of each.

Two appendices complete the final report. Appendix I offers two alternate algorithms for use in an airborne computer. The first is an algorithm which reduces commutativity error in a GP computer, and the second is an algorithm intended specifically for an incremental hybrid computer. Evaluation of these alternate algorithms, other than by the hand analysis presented in the appendix, was considered beyond the scope of this study.

Appendix II presents the results of further study of the effects of coning on strapdown software beyond that done in Phase I.

2. SUMMARY

2.1 CONCLUSIONS

The Phase II study as conducted by TRW has both confirmed the Phase I study results and conclusively demonstrated the feasibility of a boost guidance system for Saturn composed of the MSFC SIRA and a suitable state-of-the-art airborne guidance computer.

The SIRA hardware error model, including the effects of a conservative evaluation of the strapdown instrument environmental errors, exhibits an injection inaccuracy of less than 9.8 fps (3σ) on the injection trajectory used for evaluation and using the Phase I SAP orientation along body axes. The Phase II SAP orientation (SAP's straddling thrust axis as well as PIGA's) gives an injection error of less than 11.4 fps (3σ) on the same trajectory, due to the excitation of additional mass unbalance terms.

A suitable state-of-the-art airborne computer may be designed for use with the SIRA which using the recommended software essentially contributes no additional injection error when the errors are combined in rss fashion. Such a suitable computer may be either a 16-bit, 10,000-cps DDA or a 24-bit, 10-msec GP. Shorter word length and/or slower speed computers may be used with increasing penalties in injection accuracy.

If self-contained alignment (in the airborne computer) is a requirement, it may be desirable to select either a GP or hybrid airborne computer rather than a pure DDA. These computer types are preferred due to the large increase in DDA capacity required to implement rapid alignment equations. Alignment software can be easily implemented (in a GP or GP portion of a hybrid) which allows alignment to better than 20 arc sec in less than 26 min time. Possibly, alignment software can be implemented which will allow the same alignment accuracy in less than 16 min time.

2.2 STUDY RESULTS

This subsection summarizes the study results obtained and developed in Sections 3 through 7. It is intended here to document the findings only, without method or justification. The analysis methods, simulation techniques used, and evaluation of results are presented in detail in the appropriate sections later in the report.

2.2.1 System Performance Analysis

The total system (SIRA, airborne computer and software) was evaluated for a 2 x 3 matrix of SIRA (PIGA's straddling thrust axis; SAP's on body axes; SAP's straddling thrust axis) and airborne computer (16 bit, 10,000-cps DDA; 24 bit, 10-msec GP; 21 bit, 20-msec GP) configuration combinations. The system performance was evaluated using the same injection trajectory as in Phase I for ease of comparison, and it includes the instrument errors due to a conservative strapdown environment. The results of these system performance analyses are shown in Table 2-1.

It is to be noted that the injection accuracy is relatively insensitive to SIRA instrument orientation (1.5 fps (3σ) injection error difference), and that either the 16-bit DDA or 24-bit GP performs equally well, including both software and computer hardware errors. The increased SIRA error in the Phase II instrument orientation (PIGA's and SAP's straddling thrust axis in pairs) is a result of the excitation of additional mass unbalance terms. This effect is both trajectory and g- level dependent, but it may be expected to be typically small.

Table 2-1. System Performance Analyses Results

	Phase I Instrument Configuration		Phase II Instrument Configuration	
Airborne Computer	Position, 3σ	Velocity, 3σ	Position, 3σ	Velocity, 3σ
16 bit, 10,000-cps	4105 ft	9.9 fps	4710 ft	11.5 fps
DDA*	1252 M	3.0 m/sec	1434 M	3.5 m/sec
24 bit, 10-msec	4138 ft	9.9 fps	4727 ft	11.6 fps
GP	1260 M	3.0 m/sec	1443 M	3.5 m/sec
21 bit, 20-msec	4868 ft	12.5 fps	5375 ft	13.9 fps
GP	1485 M	3.8 m/sec	1639 M	4.2 m/sec

*Assumes PIGA quantization of 0.0125 m/sec/pulse (see Section 5).

2.2.2 Instrument Analyses

The complete nonlinear mathematical model of the SAP as derived during Phase I was simulated on a digital computer. The objective of the digital simulation was to check the validity of linearization assumptions made to facilitate hand analysis during Phase I, and to investigate the existence of any unmodeled error sources.

The nonlinear SAP simulation was subjected to both step inputs of varying amplitude and sinusoidal inputs of varying amplitude and frequency. It was found that the nonlinear mathematical description reproduced the results of the linear mathematical description to four and sometimes five significant figures. This key result both validated the Phase I analysis and allowed the use of an existing TRW linear transfer function computer program for the evaluation of instrument loop rectification performance in the presence of sinusoidal or random inputs on multiple axes.

The rectification program was run for normalized power spectral density (PSD) inputs out to 2,000 cps, and the integrated error drift rate effect plotted versus frequency for each rectification error source. This valuable set of plots allows the determination of instrument rectification response to any shaped PSD without further computer runs.

Three new SAP error sources due to fictitious coning (unequal instrument frequency responses and phase shifts) were found and evaluated in Phase II. These sources do not appreciably affect SAP performance.

Due to the linearity validation in the SAP simulation, it was not necessary to simulate the similar PIGA. The Phase I results are sufficient and valid.

2.2.3 Airborne Computer Analyses

The attitude reference and velocity transformation software developed in Phase I was refined and programmed in interpretive routines which allowed the simulation of airborne computers of variable word length and/or speed. Both DDA and GP computers were simulated. The results of these simulations together with additional analyses allowed the separation and evaluation of error sources due to both software (truncation, commutativity, etc.) and computer hardware (roundoff, quantization, etc.)

The simulations of a GP airborne computer were evaluated and interpolated in order to allow the selection of two candidate airborne computers for use in the system performance analyses. One candidate machine (24 bit, 10 msec) was chosen as contributing negligible error compared to the SIRA, and the other (21 bit, 20 msec) was chosen as contributing approximately the same error as the SAP's. The performance of the two candidate GP's is presented in Table 2-2, and includes both software and computer hardware significant error contributions.

Table 2-2. Candidate GP Error Performance

Error Source	Equivalent Drift Error (deg/hr)	
	24-bit, 10-msec GP	21-bit, 20-msec GP
Slewing Truncation Error	0.0115	0.046
Coning Truncation and Commutativity	0.0100	0.040
Roundoff	0.0030	0.029
Quantization	0.0010	0.002
RSS	0.0156	0.0675

The simulations of a DDA computer showed that quantization of the PIGA outputs is a major error source. It was seen that increasing DDA word length and speed is not in itself sufficient to achieve adequate performance. The velocity quantization must be reduced as speed is increased in order to take advantage of the faster speed. Otherwise, the DDA is operating for many cycles between velocity pulses with no input.

The present MSFC PIGA quantization is 0.05 mps/pulse. In order to achieve reasonable performance (1.5 fps (3σ) total velocity injection error) it will be necessary to reduce PIGA quantization to 0.0125 mps/pulse at least.

This improvement yields satisfactory performance in conjunction with a 16-bit, 10,000-cps DDA, and it was incorporated in the system performance analyses.

During the Phase II computer analyses, it was decided that the Phase I DDA design capable of handling only a 1.4 deg/sec nominal vehicle rate was unrealistic. The Phase II results reflect an increase to 12 deg/sec or better nominal capability with input buffers to handle rates up to 60 deg/sec for short periods.

Simulations of three velocity transformation methods were evaluated during Phase II (using direction cosines present at beginning or end of velocity accumulation interval; direction cosines at middle of velocity accumulation interval; direction cosines averaged at beginning and end of velocity accumulation interval). It was found that significant improvement in error performance was gained with the use of either average direction cosines or the direction cosine values at the middle of the accumulation interval. Since a serial DDA must use the direction cosines at the end of a velocity accumulation interval, it is for precisely this reason that a DDA requires smaller velocity quantization than a GP. It cannot profit from the averaging improvement available to a GP.

2.2.4 Alignment Analyses

During Phase II, four methods of alignment software implementation were evaluated. Three of these methods (constant gain; white-noise Kalman; variable gain filters) were programmed in a complete alignment simulation involving direction cosine update, 99 percent wind sway, etc. Significant improvement was achieved over the Phase I constant gain alignment in the alignment time required in the presence of 99 percent winds by means of a software change. The Phase II software initiates alignment by setting in an initial average direction cosine matrix obtained by filtering the SAP and PIGA outputs for 3 to 4 min prior to initiation. In Phase I, alignment was initiated using the identity matrix for the starting direction cosines. This software refinement achieves better than 20 arc sec alignment accuracy in 26 min as opposed to one hr using the identity matrix method.

The results of the white-noise Kalman filter and variable gain filter simulations were unsatisfactory in that no significant improvement over the constant gain method in alignment time was seen. For this reason, a fourth method was investigated using a narrowband Kalman filter, but only the filter itself was simulated in order to examine convergence properties. The design of the narrowband Kalman filter assumes that the sway is approximately sinusoidal and is near a known frequency. This method offers an improvement to 16 min alignment time if the frequency and characteristic oscillation actually experienced do not differ greatly from those assumed in the Kalman filter design.

2.2.5 Airborne Computer Implementation Studies

Based upon the results of the airborne computer analyses and the alignment analyses, several hardware implemenations of the airborne computer were examined. These implementations consisted of DDA's employing several kinds of memories, GP's as discussed in the Phase I Final Report, and a hybrid DDA-GP in which the attitude reference equations are solved in the DDA portion, while the velocity transformation and alignment equations are solved in the GP portion.

The desire to implement the rapid alignment software developed in Phase II places stringent requirements upon a DDA. The number of computing elements is greatly increased by this software change. It is desirable, therefore, to select either a GP or hybrid DDA-GP for the airborne computer, where the alignment software may be simply implemented in the GP portion. The salient characteristics of the candidate computers studied are shown in Table 2-3.

Table 2-3. Airborne Computer Implementation Studies

Characteristic	Iteration Rate	Number of Flat Packs	Power	Flexibility	Remarks
System Type					
IC's	100 kHz	2600	122	Poor	Includes 129, 8-bit register packs.
MOS	33 kHz	510	25	Poor	Multi-function flat-packs.
Core DDA	10 kHz	675	52	Poor	Requires core stack and electronics.
Delay Line	62.5 kHz	940	116	Poor	Requires 27 read & 27 write amplifiers plus delay lines.
GP Computer (24-bit)	132 Hz	1060	37	Good	Requires core stack and electronics.
Hybrid (18-bit)	250Hz 100 kHz	850	36	Good	Requires core stack & electronics. Includes 36, 8-bit register packs

3. PERFORMANCE ANALYSES

3.1 INTRODUCTION AND SUMMARY

This section presents the results of the flight-error analyses performed during Phase II. These error analyses represent a refinement of those accomplished in Phase I (see Reference 1, Section 3) based upon the detailed instrument analyses (Section 4) and airborne-computer analyses (Section 5) performed during Phase I. In general, it is shown that the Phase I assertion, that computational and computer errors can be made small with respect to hardware errors, was valid.

The performance analyses performed during Phase II were accomplished for a 2 x 3 matrix of SIRA airborne-computer combinations. Both the Phase I instrument configuration (gyros on body axes, accelerometers straddling the thrust vector) and the Phase II instrument configuration (both gyros and accelerometers straddling the thrust vector in pairs) have been analyzed for performance in conjunction with three candidate-airborne computers (16-bit, 10,000-cps DDA; 24-bit, 10-msec GP; 21-bit, 20-msec GP). The total system error model for either instrument configuration operating with either the DDA or 24-bit GP gave a 3σ-velocity error of 3.5 m/sec or less. Either instrument configuration operating with the 21-bit GP gave a 3σ-velocity error of 4.2 m/sec or less.

3.2 TRAJECTORY

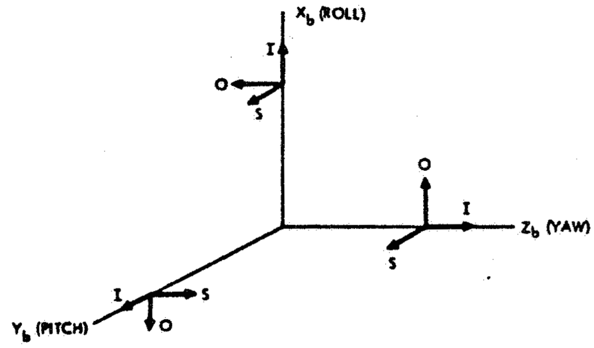
In order to provide ease of comparison with the Phase I results, the Phase I trajectory was used for the Phase II error analyses. This trajectory was generated by TRW based upon a reference trajectory supplied by MSFC during Phase I. The TRW trajectory is defined and compared to the MSFC trajectory in Reference 1, Subsection 3.1.

3.3 INSTRUMENT CONFIGURATION

The instrument orientations used in the error analyses are defined in Figures 3-1 and 3-2. While no attempt was made to determine optimal instrument orientation, it can be seen from inspection of Table 3-2* that

*Table 3-2 is in the Confidential Annex, "Annex of Error Models and Analyses (U)," TRW Document No. 05128-6002-R7000.

Gyro Orientation



Accelerometer Orientation

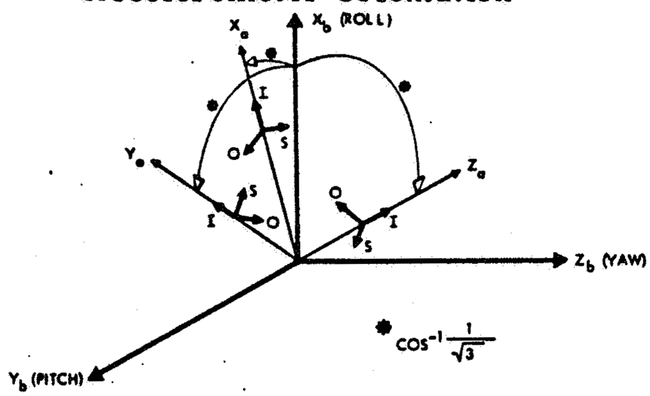


Figure 3-1. Phase I Instrument Orientation

Accelerometer and Gyro Orientation

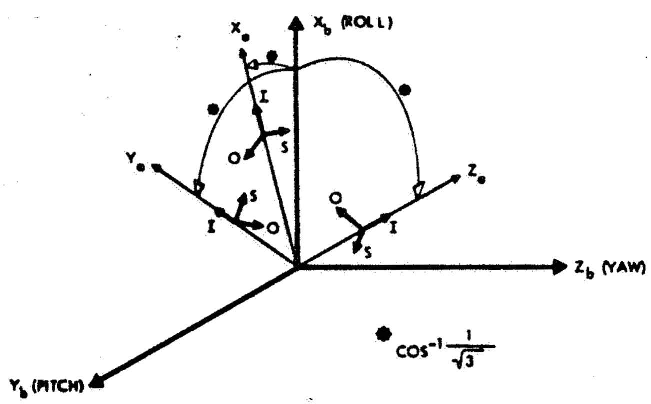


Figure 3-2. Phase II Instrument Orientation

there is little difference in the performance of either configuration with respect to velocity error (approximately 0.5 m/sec, 3σ) on this trajectory. This result is substantiated by the results of another study for MSFC by TRW in which several orientations were evaluated on another trajectory.

Primarily, the difference between instrument orientation performance arises due to gyro mass unbalance terms being excited in the straddle configuration. The magnitude of this effect would be trajectory and g-level dependent, but is probably small.

3.4 SYSTEM PERFORMANCE

The total system-error model, including instrument errors, software errors, airborne-computer errors and alignment errors, was input to the TRW generalized error-analysis program for evaluation on the selected trajectory. The results of these system-performance analyses are presented in this subsection.

3.4.1 Error Modeling

The instrument hardware error models used in these analyses are identical to those used in Phase I. The instrument error models are summarized in Table 3-1.* The instrument errors due to the strapdown environment are developed in Section 4 and applied to the hardware error model in Paragraph 3.4.2.* The software and airborne-computer errors are developed in Section 5, and summarized in Table 3-2.* The alignment errors are developed in Section 6 and summarized in Table 3-1.*

3.4.2 Numerical Error Models

This paragraph is contained in the Confidential Annex, "Annex of Error Models and Analyses (U)," TRW Document No. 05128-6002-R7000.

3.4.3 Results

The detailed injection errors obtained by propagating the numerical-error models of Paragraph 3.4.2 through the injection trajectory are presented in Table 3-2.* The total injection error is summarized in Table 3-3, however, for the 2 x 3 matrix of SIRA airborne-computer combinations.

*See Confidential Annex

Table 3-3. System Performance Analyses

Airborne Computer	Phase I Instrument Configuration		Phase II Instrument Configuration	
	Position, 3σ	Velocity, 3σ	Position, 3σ	Velocity, 3σ
16-bit, 10,000-cps DDA*	4105 ft	9.9 fps	4710 ft	11.5 fps
	1252 M	3.0 m/sec	1434 M	3.5 m/sec
24-bit, 10-msec GP	4138 ft	9.9 m/sec	4727 ft	11.6 fps
	1260 M	3.0 m/sec	1443 M	3.5 m/sec
21-bit, 20-msec GP	4868 ft	12.5 fps	5375 ft	13.9 fps
	1485 M	3.8 m/sec	1639 M	4.2 m/sec

*Assumes PIGA quantization of 0.0125 m/sec/pulse (see Section 5).

REFERENCE

1. D. D. Otten, "Body-Fixed, Three-Axis Reference System Study, Phase I Final Report," TRW Document No. 4499-6007-R0000, 2 May 1966

4. INSTRUMENT ANALYSES

4.1 INTRODUCTION

This section presents the results of analyses performed on a single-axis-platform (SAP) during Phase II of the Body-Fixed Three-Axis Reference System Study. The main objective during the Phase II SAP analyses was to digitally simulate the SAP equations of motion (derived during Phase I; see Reference 1, Section 7) in order to accurately determine rectification errors. A secondary effort was to inspect SAP loop operation and to supplement coning studies performed during Phase I.

Subsection 4.2 summarizes the Phase II work efforts; the remainder of this section is subdivided into four basic subsections:

- 4.3 SAP Digital Simulation (MIDAS)—Time Domain
- 4.4 Digitally Determined Rectification Errors—Frequency Domain
- 4.5 Transient Characteristics of SAP Loop
- 4.6 Fictitious Coning Phenomena

The first of these subsections presents the methods and results of digitally programming the complete nonlinear differential equations of the SAP. A description of the program is provided. This program is referred to as MIDAS (Modified Integration Digital Analog Simulator). By means of this simulation the SAP is shown to operate as a linear device. Both step and sinusoidal inputs are utilized to demonstrate this linear property.

The establishment of the SAP linear operational characteristics permitted use of another digital program known as the Drift Rectification Program. In the second subsection this program is described. Evaluation of each of the drift rectification errors that were determined in Phase I is repeated. In addition each of these errors is also presented in a normalized form (power spectral densities).

The third subsection presents some particular results of the MIDAS simulation that pertain to marginal loop stability. Graphs of SAP response time histories are shown that reflect root locus characteristics.

The overall system (SIRA) coning studies presented in Phase I are supplemented in the fourth subsection. This supplemental material pertains to SAP contributions to coning errors. These contributions are shown to arise when two separate SAP channels have different phase and gain characteristics.

4.2 SUMMARY AND CONCLUSIONS

A digital program was designed to simulate a time domain mode of operation for the SAP. A detailed description of the program is later presented. Two forms of excitation were employed, step and sinusoidal. The results of the program verify that MSFC-SAP operates in a linear mode. This linearization is demonstrated by two means: First, the responses to step inputs matched those of a linear system; and second, it was verified that the nonlinear SAP loops summing junction inputs and outputs compare numerically to those of the linearized summing junctions.

A second digital program was developed to calculate rectification errors that occur in a linear system. The program is later explained in detail. This program provides a means for determining error responses to either sinusoidal or random power spectral density excitations. Both responses are normalized; however, an explanation is provided to show how responses to shaped PSD inputs may be obtained. The sinusoidal response magnitudes are shown to have exact correlation to those obtained from the time domain solution (MIDAS).

The SAP error terms ϵ_3 through ϵ_8 from Phase I, as well as three new terms ϵ_{10}, ϵ_{11}, ϵ_{12}, have been calculated by the rectification error program, and are presented in Table 4-1. Numerical values are provided for both the drift coefficients (normalized sine responses) and the accumulated random integrations (described in Subsection 4.4). These latter responses pertain to power spectral densities (PSD) that are constant valued to 40 cps. Curves are presented in Subsections 4.4 and 4.6 for extensions of the PSD through 2000 cps.

A low magnitude oscillatory mode of operation of the MSFC-SAP is pointed out. Graphs are presented which demonstrate the SAP torque restoration associated with step error torques acting about the SAP gyro-output axis and SAP input axis. These restoration torques are shown to oscillate through several cycles with considerable relative overshoot. A root locus plot that was presented during Phase I serves to explain these characteristics.

An analysis is provided which clearly shows that fictitious coning effects attributed to the MSFC-SAP are negligible. This analysis includes a description of the fictitious coning phenomena, as well as two examples. First, two versions of a typical example involving torque motor back emf are presented. Second, a case induced by angular vibrations acting about the output axis of two different SAP's is provided. The errors that result are ϵ_{10}, ϵ_{11}, and ϵ_{12} shown in Table 4-1.

The analysis conducted during Phase II serves to partially upgrade the SIRA error model that was presented in Phase I. The upgrading reflects digital calculation of the SAP error model only. The PIGA error model from Phase I is assumed to remain intact. The SIRA package alignment is studied in Section 6.

4.3 SAP DIGITAL SIMULATION (MIDAS)—TIME DOMAIN

During the Phase I study activities, the equations of motion of a SAP were derived in detail. To determine performance errors these equations were linearized. The Phase II efforts are premised on utilizing the nonlinear version of these differential equations. Specifically, the version of the equations that is used reflects the nonlinearities that usually occur in Eulers' rigid body equations. These equations omit products of inertia.

The method selected to digitally simulate these nonlinear equations employs a standard TRW utility program commonly known as MIDAS (Modified Integration Digital Analog Simulator). MIDAS provides a means of obtaining digital solutions on an IBM 7094 computer for systems of differential equations by using conventional analog computer techniques. It features the use of floating point arithmetic and a variable step-size integration routine. These features eliminate the amplitude and time scaling problems usually associated with analog usage.

Table 4-1. Drift Rate Errors and Causes

ERROR	ERROR CAUSE	DRIFT COEFFICIENT	NUMERICAL VALUE, deg/hr (normalized)	RANDOM INDUCED ERROR	ACCUMULATED RANDOM INTEGRATION, deg/hr (normalized)**
ϵ_3	Rectification due to inputs of ω_2 and ω_3	$C_3 = \frac{1}{2}\left\lvert\frac{\theta}{\omega_2}\right\rvert \cos\left(\angle\frac{\theta}{\omega_2} - \phi_1\right)$	0.162	$\int_0^{f_{max}*} \phi_{23}(f)\ C_3(f)df$	3.1
ϵ_4	Rectification due to inputs of L_e' and ω_3 (L_e' are MUSA and MUIA activated by vibration inputs)	$C_4 = \frac{1}{2}\left\lvert\frac{\theta}{L_e'}\right\rvert \cos\left(\angle\frac{\theta}{L_e'} - \phi_2\right)$	0.0073	$\int_0^{f_{max}} \phi_{e3}(f)\ C_4(f)df$	0.13
ϵ_5	Rectification due to inputs of L_{pe}' and ω_3 (L_{pe}' are friction torques or mass unbalances excited by vibration)	$C_5 = \frac{1}{2}\left\lvert\frac{\theta}{L_{pe}}\right\rvert \cos\left(\angle\frac{\theta}{L_{pe}'} - \phi_3\right)$	0.00375	$\int_0^{f_{max}} \phi_{pe3}(f)\ C_5(f)df$	0.075
ϵ_6	Rectification due to inputs of ω_2 and ω_3 (different propagation through loop than ϵ_3)	$C_6 = \frac{1}{2}\frac{I_y}{H_r}\left\lvert\frac{\dot\theta_p}{\omega_2}\right\rvert \cos\left(\angle\frac{\dot\theta_p}{\omega_2} - \phi_1\right)$	0.0134	$\int_0^{f_{max}} \phi_{23}(f)\ C_6(f)dt$	0.01
ϵ_7	Rectification due to inputs of L_e' and ω_3 (different propagation through loop than ϵ_4)	$C_7 = \frac{1}{2}\frac{I_y}{H_r}\left\lvert\frac{\dot\theta_p}{L_e'}\right\rvert \cos\left(\angle\frac{\dot\theta_p}{L_e'} - \phi_2\right)$	7.5×10^{-4}	$\int_0^{f_{max}} \phi_{e3}(f)\ C_7(f)df$	0.02
ϵ_8	Rectification due to inputs of L_{pe}' and ω_3 (different propagation through loop than ϵ_5)	$C_8 = \frac{1}{2}\frac{I_y}{H_r}\left\lvert\frac{\dot\theta_p}{L_{pe}'}\right\rvert \cos\left(\angle\frac{\dot\theta_p}{L_{pe}'} - \phi_3\right)$	1.25×10^{-5}	$\int_0^{f_{max}} \phi_{pe3}(f)\ C_8(f)df$	-5.0×10^{-4}
ϵ_{10}	Fictitious coning error due to $\pm 11\%$ gain variation between SAP channels	$C_{10} = \frac{\left\lvert\frac{\dot\theta_p}{\omega_x}1\right\rvert\left\lvert\frac{\dot\theta_p}{\omega_x}2\right\rvert}{4\pi f} \sin\left(\angle\frac{\dot\theta_p}{\omega_x}1 - \frac{\dot\theta_p}{\omega_x}2\right)$	7.4×10^{-4}	$\int_0^{f_{max}} \phi_{23}(f)\ C_{10}(f)df$	0.03
ϵ_{11}	Fictitious coning error due to $\pm 11\%$ gain variation and $\pm 11\%$ damping between SAP channels	C_{11} = same as C_{10} different numerical values	7.4×10^{-4}	$\int_0^{f_{max}} \phi_{23}(f)\ C_{11}(f)df$	0.03
ϵ_{12}	Output axis fictitious coning error due to ± 5db gain variation	$C_{12} = \frac{\left\lvert\frac{\dot\theta_p}{\omega_2}1\right\rvert\left\lvert\frac{\dot\theta_p}{\omega_2}2\right\rvert}{4\pi f} \sin\left(\angle\frac{\dot\theta_p}{\omega_2}1 - \angle\frac{\dot\theta_p}{\omega_2}2\right)$	0.0143	$\int_0^{f_{max}} \phi_{23}(f)\ C_{12}(f)df$	0.0052

$^*f_{max}$ = 40 cps $\qquad \phi_{23} = \sqrt{\phi_2}\sqrt{\phi_3} \qquad \phi_{e3} = \sqrt{\phi_3}\sqrt{\phi_3} \qquad \phi_{pe3} = \sqrt{\phi_{pe}}\sqrt{\phi_3}$

$^{**}\phi_2 = \phi_3 = 1.0\ \frac{(deg/sec)^2}{cps} \qquad ^{**}\phi_e = \phi_{pe} = 1.0\ \frac{g^2}{cps}$ (multiplied times appropriate mass unbalance)

The program employs operational elements (adders, integrators, multipliers, etc.) that are familiar to analog computer programmers. These elements serve as building blocks. The basic programming procedure consists of essentially the following four steps:

(1) Draw analog block diagram

(2) Draw MIDAS block diagram using program operational elements

(3) Write program by listing operational element and its particular inputs

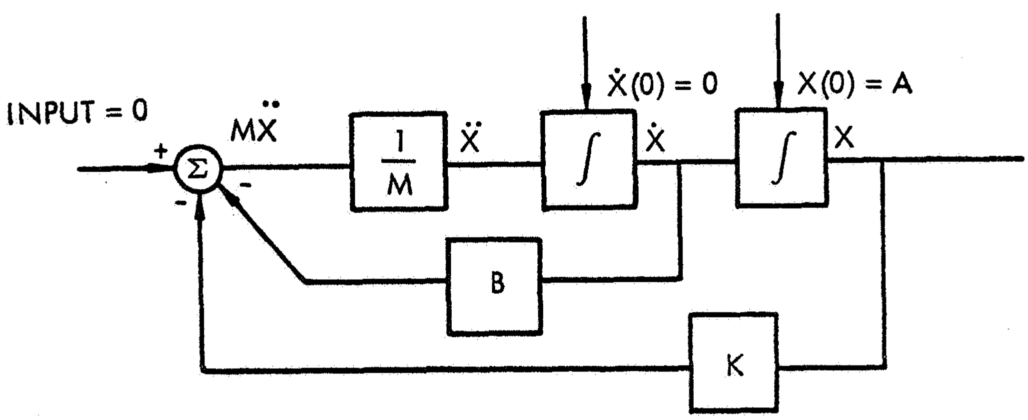
(4) Complete program by writing additional information pertaining to

- System constants specification
- Program system parameters (variations to "constant")
- Desired input-output information
- Plotting information.

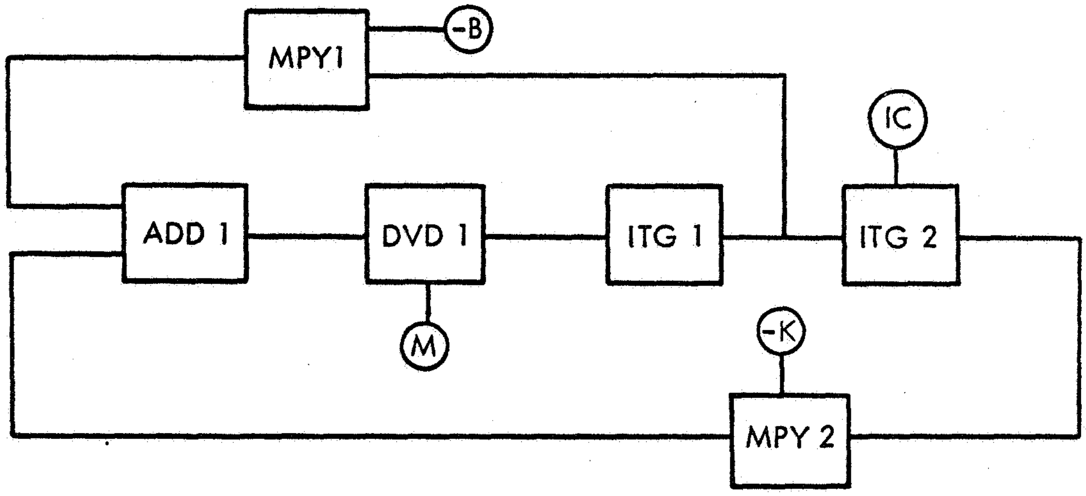
In order to furnish an insight into the programmed SAP equations a simple spring-mass-damper problem will be used to demonstrate the MIDAS techniques. The procedure prescribed by the above four steps are applied to the following system:

$$M\ddot{X} + B\dot{X} + KX = 0, \quad X(o) = A, \quad \dot{X}(o) = 0 \qquad (4\text{-}1)$$

Step 1



Step 2



Step 3

Element Number	Operational Element	Element Inputs
1	ADD 1	MPY 1, MPY2
2	DVD 1	ADD 1, M
3	ITG 1	DVD 1
4	ITG 2	ITG 1
5	MPY 1	-B, ITG 1
6	MPY 2	ITG 2, -K
7	FIN	IT, FT
8	HDR	T, XT, XDØT
9	END	IT, ITG 2, ITG 1

Step 4

CØN M, -B, -K, FT, A

PAR None

DATA (values corresponding to CØN; assumed for example)

-2.5 -15.0 2.0 1.0

PLØT None

The operational elements are appropriately chosen from a wide variety that are available. The total selection of elements is shown in Table 4-2. This table is reproduced from the programming guide (Reference 2). Each of the nine operational elements utilized in the above example are defined in this table.

Table 4-2. Summary of Elements

<table>
<tr><th>NAME</th><th>SYMBOL</th><th>INPUTS</th><th>OUTPUTS</th></tr>
<tr><td colspan="4">1. MATHEMATICAL OPERATIONS</td></tr>
<tr><td>INTEGRATE</td><td>Ij, ITGj</td><td>A</td><td>$\int_0^t A\,dt + IC$</td></tr>
<tr><td>HIGH GAIN SUMMER</td><td>HGSj</td><td>A_i(i=1,2..K < 6)</td><td>$\dfrac{-\sum_{i=2}^{K} A_i}{A_1}$</td></tr>
<tr><td>SUM</td><td>Sj, ADDj</td><td>A_i(i=1,2,..K ≤ 6)</td><td>$\sum_{i=1}^{K} A_i$</td></tr>
<tr><td>NEGATIVE</td><td>NEGj</td><td>A,B..F</td><td>$-(A + B + C + D + E + F)$</td></tr>
<tr><td>MULTIPLY, POTENTIOMETER</td><td>Mj, Pj, MPYj</td><td>A,B..F</td><td>$A \cdot B \cdot C \cdot D \cdot E \cdot F$</td></tr>
<tr><td>DIVIDE</td><td>Dj, DVDj</td><td>A,B</td><td>A/B</td></tr>
<tr><td>ABSOLUTE VALUE</td><td>ABSj</td><td>A</td><td>$|A|$</td></tr>
<tr><td>SQUARE ROOT</td><td>SQRj</td><td>A</td><td>$\sqrt{A}$ for $A \geq 0$</td></tr>
<tr><td>EXPONENTIAL</td><td>Ej</td><td>A</td><td>e^A</td></tr>
<tr><td>NATURAL LOGARITHM</td><td>LNj</td><td>A</td><td>ln A for A > 0</td></tr>
<tr><td>RESOLVER</td><td>RESj</td><td>A</td><td>B output = sin A
C output = cos A (Note: A must be in radians)</td></tr>
<tr><td>ARC TANGENT</td><td>ATj</td><td>A</td><td>$\tan^{-1} A$ where $-\frac{\pi}{2} < \tan^{-1} A < \frac{\pi}{2}$</td></tr>
<tr><td colspan="4">2. SWITCHING ELEMENTS</td></tr>
<tr><td>OUTPUT RELAY</td><td>ØRj</td><td>A,B</td><td><table><tr><th></th><th>B ≥ 0</th><th>B < 0</th></tr><tr><td>C output</td><td>A</td><td>0</td></tr><tr><td>D output</td><td>0</td><td>A</td></tr></table></td></tr>
<tr><td>INPUT RELAY</td><td>IRj</td><td>A,B,C,D</td><td><table><tr><th></th><th>C ≥ 0</th><th>C < 0</th></tr><tr><td>Output</td><td>A</td><td>B</td></tr></table>The D field may be used to change a PAR card value</td></tr>
<tr><td>FUNCTION SWITCH</td><td>FSWj</td><td>A,B,C,D</td><td><table><tr><th></th><th>D > 0</th><th>D = 0</th><th>D < 0</th></tr><tr><td>Output</td><td>A</td><td>B</td><td>C</td></tr></table></td></tr>
<tr><td>LIMITER</td><td>LIMj</td><td>A,B,C</td><td><table><tr><th></th><th>A > B</th><th>C ≤ A ≤ B</th><th>A < C</th></tr><tr><td>Output</td><td>B</td><td>A</td><td>C</td></tr></table></td></tr>
<tr><td>BANG-BANG</td><td>BBj</td><td>A,B</td><td><table><tr><th></th><th>A ≥ 0</th><th>A < 0</th></tr><tr><td>Output</td><td>B</td><td>-B</td></tr></table></td></tr>
</table>

Table 4-2. Summary of Elements (Continued)

<table>
<tr><th>NAME</th><th>SYMBOL</th><th>INPUTS</th><th>OUTPUTS</th></tr>
<tr><td>DEAD SPACE</td><td>DSJ</td><td>A,B,C</td><td><table><tr><td></td><td>A > B</td><td>C ≤ A ≤ B</td><td>A < C</td></tr><tr><td>Output</td><td>A-B</td><td>0</td><td>A-C</td></tr></table></td></tr>
<tr><td>AND GATE</td><td>AG_J</td><td>A,B</td><td><table><tr><td></td><td>A > 0 B > 0</td><td>A > 0 B < 0</td><td>A < 0 B < 0</td></tr><tr><td>Output</td><td>1.0</td><td>0.0</td><td>0.0</td></tr></table></td></tr>
<tr><td>ØR GATE</td><td>$ØG_J$</td><td>A,B</td><td><table><tr><td></td><td>A < 0 B < 0</td><td>A > 0</td><td>B > 0</td></tr><tr><td>Output</td><td>0.0</td><td>1.0</td><td>1.0</td></tr></table></td></tr>
<tr><td colspan="4">3. <u>ARBITRARY FUNCTIONS</u></td></tr>
<tr><td>FUNCTION</td><td>FJ</td><td>A, Data Cards</td><td>f(A). Linear interpolation; data for every case.</td></tr>
<tr><td>CONSTANT FUNCTION</td><td>GJ</td><td>A, Data Cards</td><td>f(A). Linear interpolation; data for first case only.</td></tr>
<tr><td>CURVE FOLLOWER</td><td>CFJ</td><td>A, Data Cards</td><td>f(A). Quadratic interpolation; data for every case.</td></tr>
<tr><td>CONSTANT CURVE FOLLOWER</td><td>CGJ</td><td>A, Data Cards</td><td>f(A). Quadratic interpolation; data for first case only.</td></tr>
<tr><td colspan="4">4. <u>ITERATIVE ELEMENT</u></td></tr>
<tr><td>IMPLICIT FUNCTION</td><td>IMPJ</td><td>A,B,C,D,E,F</td><td>See text.</td></tr>
<tr><td colspan="4">5. <u>RUN TERMINATION</u></td></tr>
<tr><td>FINISH</td><td>FIN</td><td>A,B,...,F</td><td>Stops computation when A ≥ B, post process data transfer on option.</td></tr>
<tr><td colspan="4">6. <u>INSERTION OF NUMERICAL ITEMS</u></td></tr>
<tr><td>CONSTANT</td><td>CØN</td><td>Data Cards</td><td>1 to 6 items/card. Data for first case only.</td></tr>
<tr><td>PARAMETER</td><td>PAR</td><td>Data Cards</td><td>1 to 6 items/card. Data for every case.</td></tr>
<tr><td>INITIAL CONDITION</td><td>IC</td><td>Data Cards</td><td>1 to 6 items/card. Non-zero IC's for every case.</td></tr>
<tr><td colspan="4">7. <u>SPECIAL STATEMENTS</u></td></tr>
<tr><td>HEADER</td><td>HDR</td><td>---</td><td>1 to 6 items/card. Provides titles for the readout.</td></tr>
<tr><td>READOUT</td><td>RØ</td><td>---</td><td>1 to 6 items/card. Specifies items to be printed.</td></tr>
<tr><td>END</td><td>END</td><td>---</td><td>Signifies the end of the symbolic program.</td></tr>
</table>

Table 4-2. Summary of Elements (Continued)

NAME	SYMBOL	INPUTS	OUTPUTS
8. SPECIAL NAMES			
INDEPENDENT VARIABLE	IT	---	Current value of the independent variable.
TIME BETWEEN READOUTS	TR	---	0.1 units of the independent variable, usually time, unless otherwise given on a CØN or PAR card.
MAXIMUM INTERVAL OF INTEGRATION	DINT	---	1.0 unless otherwise given on a CØN or PAR card.
MINIMUM INTERVAL OF INTEGRATION	MININT	---	10^{-6} unless otherwise given on a CØN or PAR card.
INTEGRATION OPTION	ØPTIØNJ	---	Specifies a non-standard integration option.

With these concepts in mind the programmed SAP configuration is introduced. The block diagram representation is shown in Figure 4-1. This diagram essentially reflects three differential equations: one that mathematically simulates torques acting about a SAP input axis, another simulating output axis torques, and a third representing the SAP servo function.

However, it is to be noted that the first two of these particular differential equations are not the linearized input-axis and output-axis equations derived in Reference 1. Those equations were shown in Reference 1 to be represented by a simplified block diagram that is repeated here in Figure 4-2. Instead of the linearized version of the torque equations, a form that was derived earlier in Reference 1 is used as the basis for the MIDAS simulation in Figure 4-1. In particular the input and output equations from Reference 1 that are employed in the simulation are the following:

$$L_y = L_e' + L_a = \frac{d}{dt}\left[I_y(\omega_2 - \dot{\theta})\right] + I_x(\omega_1 + \theta\omega_3)(\omega_3 - \theta\omega_1)$$

$$-I_z(\omega_3 - \theta\omega_1)(\omega_1 + \theta\omega_3) - H_r(\omega_1 + \theta\omega_3) \qquad (4\text{-}2)$$

$$L_{xp} = L_{pe}' + L_{pa} = \frac{d}{dt}\left\{I_{px}\omega_1 + I_x(\omega_1 + \theta\omega_3) - \theta\left[I_z(\omega_3 - \theta\omega_1) + H_r\right]\right\}$$

$$+ I_{pz}\omega_3\omega_2 + I_z(\omega_3 - \theta\omega_1)\omega_2 + H_r\omega_2 + \theta I_x(\omega_1 + \theta\omega_3)\omega_2$$

$$- I_{py}\omega_2\omega_3 - I_y(\omega_2 - \dot{\theta})\omega_3 \qquad (4\text{-}3)$$

$$L_{pa}(s) = F(s)\theta(s) \qquad (4\text{-}4)$$

$$\omega_1 = \omega_X - \dot{\theta}_p \qquad (4\text{-}5)$$

$$\omega_2 = \omega_y \cos\theta_p + \omega_z \sin\theta_p \qquad (4\text{-}6)$$

$$\omega_3 = \omega_y \sin\theta_p - \omega_z \cos\theta_p \qquad (4\text{-}7)$$

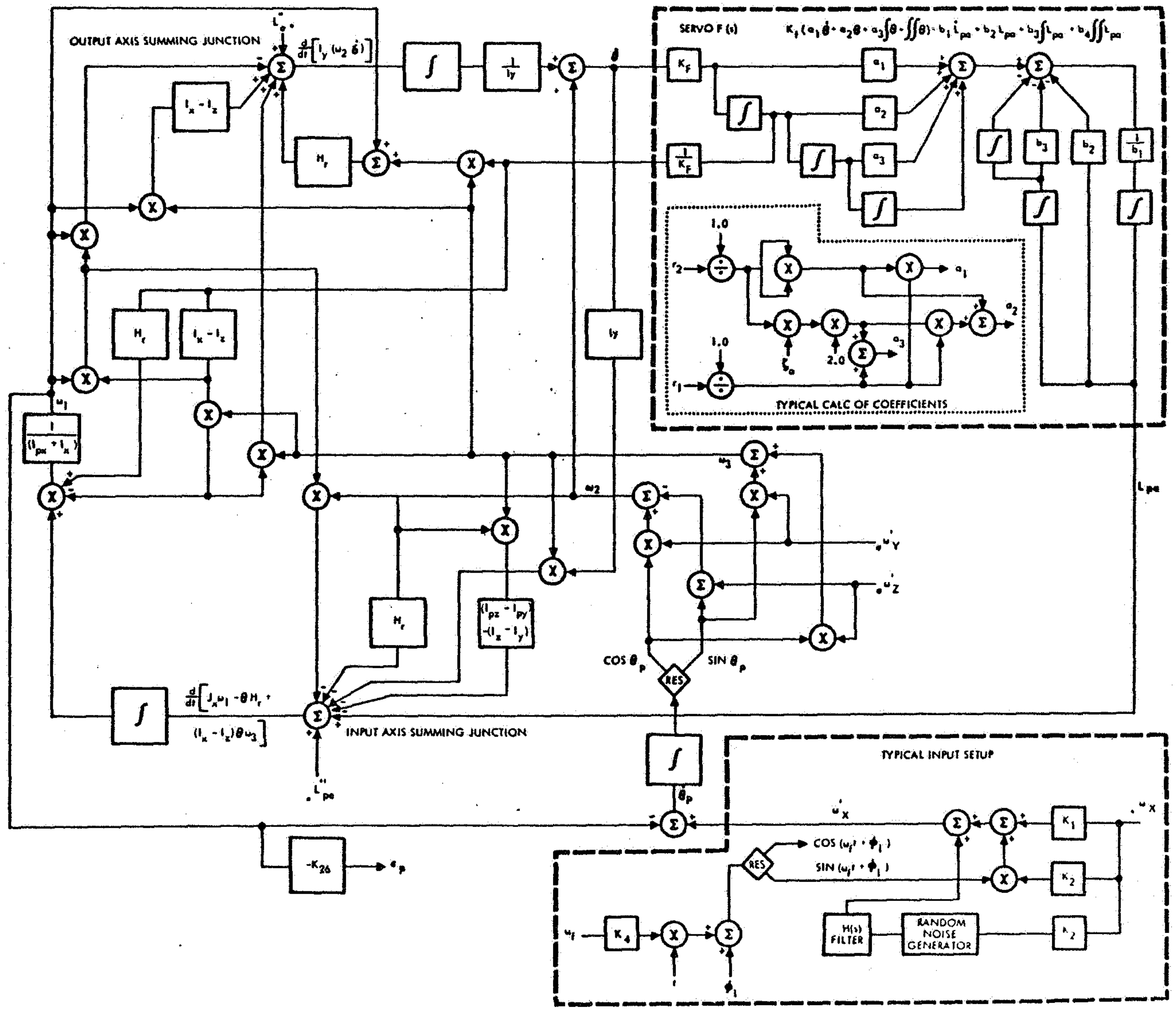


Figure 4-1. SAP Diagram, MSFC - Phase II MIDAS Simulation

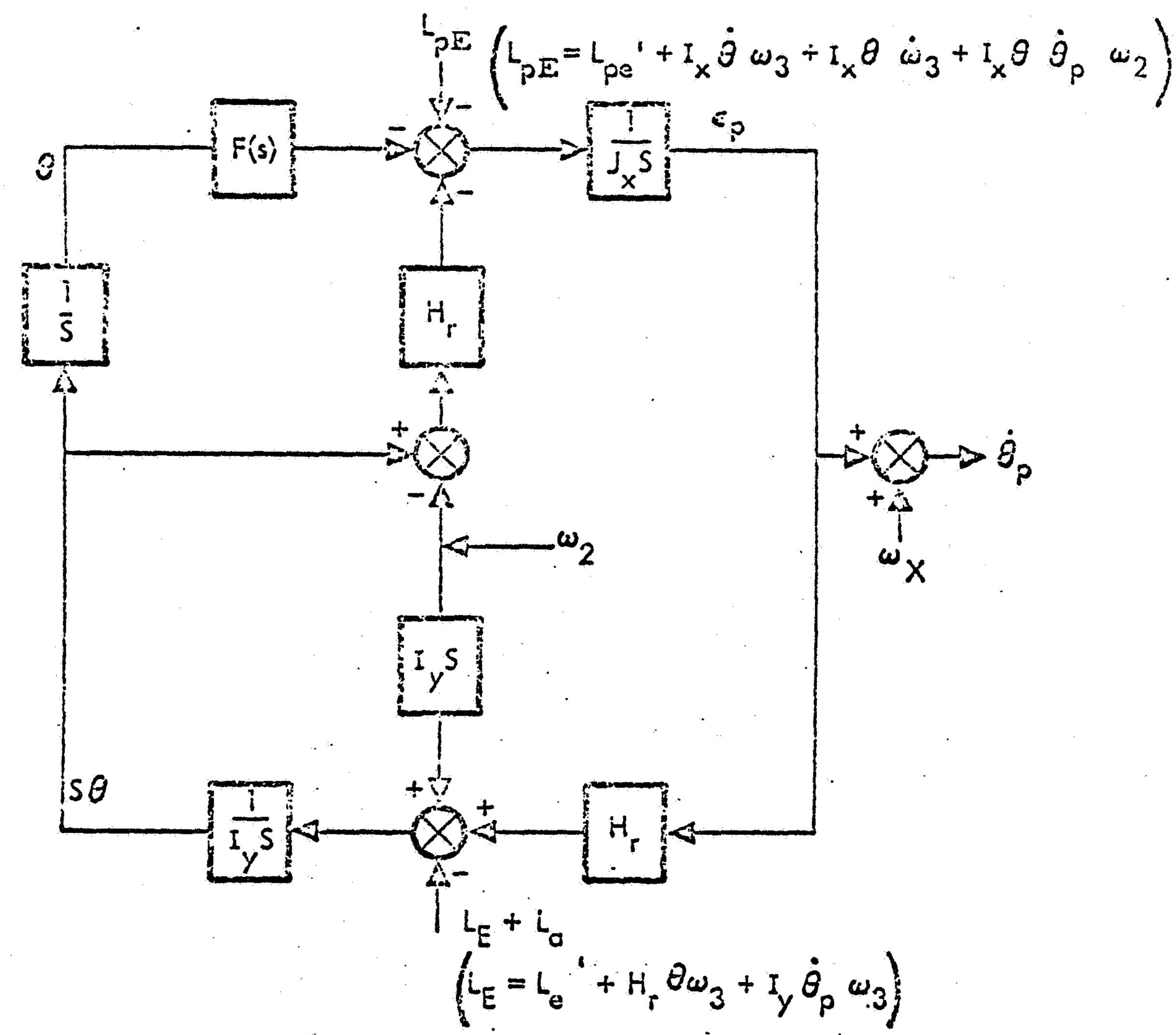


Figure 4-2. Simplified Block Diagram

The differential equation associated with the servo function, F(s), was reduced to a third-order system rather than the true fourth-order system that was designed by MSFC. A pole acting at 6000 rad/sec was deleted which has negligible effect on simulation. By lowering this order, one numerical integration is eliminated during computer operation. Since the primary expense of using the MIDAS program is integration routines, a considerable computer cost savings was obtained at essentially no loss of accuracy. The programmed equation (in frequency domain form) was

$$F'(s) = \frac{L_{pa}}{\theta}(s) = K_f \frac{\left(\frac{s}{r_1} + 1\right)\left(\frac{s^2}{r_2^2} + \frac{2\zeta_a}{r_2} s + 1\right)}{\left(\frac{s}{q_1} + 1\right)\left(\frac{s^2}{q_2^2} + \frac{2\zeta_b}{q_2} s + 1\right)} \quad (4\text{-}8)$$

In MIDAS differential equation form Equation (4-8) appears as

$$K_f\left[A_1\dot{\theta} + A_2\theta + A_3\int \theta \, dt + \iint \theta \, dt\right] = b_1\dot{L}_{pa} + b_2 L_{pa} + b_3 \int L_{pa} \, dt + b_4 \iint L_{pa} \, dt \quad (4\text{-}9)$$

where

$$A_1 = \frac{1}{r_1 r_2^2} \quad (4\text{-}9a)$$

$$A_2 = \frac{1}{r_2^2} + \frac{2\zeta_a}{r_1 r_2} \quad (4\text{-}9b)$$

$$A_3 = \frac{2\zeta_a}{r_2} + \frac{1}{r_1} \quad (4\text{-}9c)$$

$$b_1 = \frac{1}{q_1 q_2^2} \quad (4\text{-}9d)$$

$$b_2 = \frac{1}{q_2^2} + \frac{2\zeta_b}{q_2} \quad (4\text{-}9e)$$

$$b_3 = \frac{2\zeta_b}{q_2} + \frac{1}{q_1} \quad (4\text{-}9f)$$

The complete MIDAS program that is associated with the block diagram in Figure 4-1 is shown in Figure 4-3. Two decks of program cards have been prepared. One has input constants already programmed so as to allow only sinusoidal forms of the five forcing functions, ω_y, ω_z, L_{pe}, L_e, and L_a. The second deck is prepared to process only step function forms of these same inputs. Although Figure 4-1 indicates an additional form of excitation, i.e. random, the necessary subroutines to accomplish this have not been actually included in the program.

A typical computer output sheet for the SAP is reproduced in Figure 4-4. The program plotting information and particular input data that are associated with the SAP computations are shown in Figure 4-5. These plots were presented in Reference 3 and some of them will also be shown later in this report.

The step-input configuration served two basic purposes. It was originally used to check out the SAP/MIDAS program and later provided a means for obtaining transient responses of the SAP. In conjunction with these two functions the step input configuration aided in establishing the degree of linearity to which the SAP operates. The establishment of the linearity is essential to the verification of the assumptions and results of the rectification error analyses performed in Phase I.

To supplement the verification of a linear mode of operation, sinusoidal forcing functions were applied. These excitations were selected so as to represent the rectification inducement of the error term listed as ϵ_3 in Reference 1. The following Table 4-3 is extracted from Reference 3 and provides a summary of the step and sine input cases performed.

The linearity of the SAP can be demonstrated by inspecting the input and output magnitudes of the two major summing junctions in Figure 4-1. In each case the inputs consist of several signals, some of which are linear and the rest represent nonlinear signals. Also, in each case the output is a single signal. It remains to show that the output signal is identical to the sum of the linear signals at each summing junction. This not only establishes that the nonlinear signals are negligible, but further substantiates that the mathematical expressions representing these junctions are indeed the linearized input-axis and output-axis equations (see Reference 1). The following graphical and numerical comparisons will serve to demonstrate this linearity for both junctions.

```
?ИLNTFFF, ?.  X??7?? ??/1??? SINE INPUT DECK
PUNCH
SAP F??C?S, MSFC PH. ?
CONSTANTS
  1   CON     TU,HR,KE,R1,Q1,Q2
  2   CON     IX,IY,IZ,IPX,IPY,IPZ
  3   CON     K1,K2,K3,K4,K5,K6
  4   CON     K7,K8,K9,K10,K11,K12
  5   CON     K13,K14,K15,K16,K17,K18
  6   CON     K19,K20,K21,K22,K23,K24
  7   CON     K25,ONE,ZA,ZB,R2
  8   CON     PHI1,PHI2,PHI3,PHI4,PHI5
  9   PAR     WX,WY,WZ,LPE,LE
 10   PAR     WE,TR,FT,K26
 11   PAR     MININT
COMPUTE DRIFT RATE ERROR FOR INPUTS PER PAR DATA
OPTION?
 14   DVD1    ITG3,IY
 18   NEG7    DVD1
 16   ITG4    ITG6
 47   RES1    ITG2
 58   ITG6    ITG5
      DVD4    ONE,R2
 70   MPY22   DVD4,DVD4
 71   MPY23   DVD4,ZA
 73   DVD5    ONE,Q2
 83   MPY51   LPE,K13
 84   MPY52   LPE,K14
 85   MPY53   LPE,K15
 90   MPY55   WE,K16
 94   MPY57   LE,K17
 95   MPY58   LE,K18
 96   MPY59   LE,K19
101   MPY62   WE,K20
102   MPY63   MPY62,IT
103   ADD30   MPY63,PHI5
104   RES6    ADD30
105   MPY30   WX,K1
106   MPY31   WX,K2
107   MPY32   WX,K3
110   MPY34   MPY32,K21
      NEG11   ADD8
      MPY14   NEG11,IY
      MPY6    MPY14,ADD4
      MPY15   ADD10,HR
      ADD12   IPY,IY
      NEG13   ADD12
      ADD11   NEG13,IZ,IPZ
      MPY7    ADD4,ADD10
      MPY8    MPY7,ADD11
      NEG5    MPY8
      NEG2    MPY15
      MPY71   DVD7,HR
      NEG14   MPY16
      ADD2    ITG1,MPY71,NEG14
      ADD6    IPX,IX
      DVD2    ADD2,ADD6
```

Figure 4-3. SAP/MIDAS Program

```
 31  NEG3    DVD2
 24  ADD3    ADD17,NEG33
 22  ITG2    ADD3
     MPY9    DVD2,MPY70
     MPY1    MPY9,ADD10
     NEG1    MPY1
     ADD1    NEG1,NEG2,NE35,MPY6,ITG7,ADD26
 40  ITG1    ADD1
     MPY72   ADD4,DVD7
     ADD9    DVD2,MPY72
     MPY13   ADD9,HR
     MPY10   MPY9,DVD2
     NEG6    MPY10
     MPY12   DVD2,ADD4
     MPY11   MPY12,ADD5
     ADD7    NEG6,MPY11,MPY69,MPY13,ADD29
 13  ITG3    ADD7
138  MPY66   K26,NEG3
146  FIN     IT,FT
147  HDR     TIME,DR.RT.,PL.ANG,FL.ANG,PL.RT,LPA
148  HDR     SEC,DEG/HR,RAD,RAD,RA/SEC,DY-CM
149  HDR
     HDR     ,THETW3,HRW2,DJW1DT,DIW2DT,HRW1
     HDR     ,RA/SEC,DY-CM,DY-CM,DY-CM,DY-CM
     HDR
     HDR
 93  RES5    ADD27
 98  MPY60   RES6B,MPY58
 97  ADD28   MPY60,MPY57
 89  MPY65   RES5B,MPY52
 86  ADD25   MPY65,MPY51
 87  MPY54   MPY53,K24
 88  ADD26   ADD25,MPY54
100  ADD29   ADD28,MPY61
109  MPY33   RES2B,MPY31
108  ADD16   MPY33,MPY30
111  ADD17   MPY34,ADD16
119  ADD19   MPY37,MPY40
120  ADD20   MPY41,ADD19
 63  MPY64   MPY22,DVD8
 59  MPY19   ITG5,ADD32
 53  MPY5    ADD23,RES1B
 52  MPY3    ADD20,RES1B
 50  NEG4    MPY5
 49  MPY4    ADD20,RES1C
 48  MPY2    ADD23,RES1C
 51  ADD10   NEG4,MPY4
 46  ADD4    MPY2,MPY3
 15  ADD8    NEG7,ADD10
 54  MPY17   ADD8,KF
 55  MPY18   MPY17,MPY64
 56  ADD13   MPY18,MPY19,MPY67,ITG4
 57  ITG5    MPY17
 65  MPY20   ADD14,ITG7
 66  ITG8    ITG7
 68  ITG9    ITG8
 67  MPY21   ADD15,ITG8
 60  NEG8    MPY20
 61  NEG9    MPY21
```

Figure 4-3. SAP/MIDAS Program (Continued)

```
62  NEG10   ITG9
139 ADD31   ADD13,NEG8,NEG9,NEG10
76  DVD3    ADD31,MPY28
64  ITG7    DVD3
    NEG12   IZ
    ADD5    NEG12,IX
    MPY70   ADD5,DVD7
    MPY16   MPY70,ADD4
    MPY69   MPY16,ADD4
112 MPY35   WF,K4
113 MPY36   IT,MPY35
114 ADD18   MPY36,PHI1
115 RES2    ADD18
116 MPY37   WY,K5
117 MPY38   WY,K6
118 MPY39   WY,K7
124 MPY42   WF,K8
125 MPY43   MPY42,IT
126 ADD21   MPY43,PHI2
123 RES3    ADD21
121 MPY40   MPY38,RES3B
122 MPY41   MPY39,K22
127 MPY44   WZ,K9
128 MPY45   WZ,K10
129 MPY46   WZ,K11
82  DVD6    ONE,Q1
69  MPY61   K25,MPY59
77  MPY26   DVD5,Z9
78  MPY27   MPY26,TU
79  MPY29   DVD6,MPY27
81  ADD15   MPY27,DVD6
74  MPY25   DVD5,DVD5
75  MPY28   DVD6,MPY25
80  ADD14   MPY25,MPY29
141 DVD7    ITG5,KF
    DVD8    ONE,R1
72  MPY24   MPY23,TU
143 MPY68   MPY24,DVD8
144 ADD32   MPY22,MPY68
145 ADD33   MPY24,DVD8
146 MPY67   ITG6,ADD33
135 MPY49   WF,K12
136 MPY50   MPY49,IT
137 ADD24   PHI3,MPY50
134 RES4    ADD24
133 MPY48   K23,MPY46
132 MPY47   RES4B,MPY45
130 ADD22   MPY44,MPY47
131 ADD23   ADD22,MPY48
91  MPY56   MPY55,IT
92  ADD27   MPY56,PHI4
    HDR     ,W1,W1+QW3,JW1,JW1+NL,FL,RT
    HDR     ,RA/SEC,RA/SEC,G-C/SEC,G-C/SEC,RA/SEC
    HDR
150 RO      IT,MPY66,ITG2,DVD7,ADD3,ITG7
    RO      ,MPY72,NEG2,ADD1,ADD7,MPY13
    RO      ,DVD2,ADD9,ADD2,ITG1,ADD8
    RO
151 END
```

Figure 4-3. SAP/MIDAS Program (Continued)

TIME SEC	DR.RT. DEG/HR	PL.ANG RAD	FL.ANG RAD	PL.RT RA/SEC	LPA DY-CM
	THETW3 RA/SEC	HRW2 DY-CM	DJW1DT DY-CM	DIW2DT DY-CM	HRW1 DY-CM
	W1 RA/SEC	W1+OW3 RA/SEC	JW1 -C/SEC	JW1+NL -C/SEC	FL.RT RA/SEC
0.	-0.	0.	0.	0.	0.
	0.	0.	-0.	0.	0.
	0.	0.	0.	0.	-0.
UNDRFLOW AT 45022 IN MQ					
10.000000E-04	-8.299332E 01	-1.163907E-07	2.654404E-05	-4.026847E-04	3.055203E 04
	1.436034E-08	-1.347426E 04	1.707733E 04	1.006747E 03	1.006748E 03
	4.026847E-04	4.026991E-04	1.299454E 01	6.333625E 00	5.149219E-03
2.000000E-03	-3.263640E 02	-1.085458E-05	9.515817E-05	-1.583523E-03	$\boxed{4.484949E\ 04}$
	$\boxed{9.755942E-08}$	$\boxed{-2.563091E\ 04}$	$\boxed{1.921699E\ 04}$	3.959048E 03	3.959050E 03
	1.583523E-03	1.583620E-03	5.110027E 01	2.731072E 01	8.009577E-03
3.000000E-03	-5.101834E 02	-3.179855E-06	1.754845E-05	-2.475417E-03	3.099951E 04
	2.476508E-07	-3.528122E 04	-4.284647E 03	6.189156E 03	6.189162E 03
	2.475417E-03	2.475665E-03	7.988171E 01	3.601053E 01	7.542087E-03
4.000000E-03	-4.932673E 02	-5.702834E-06	2.383700E-05	-2.393339E-03	1.008057E 04
	3.955131E-07	-4.148148E 04	-3.140488E 04	5.984331E 03	5.984337E 03
	2.393339E-03	2.393735E-03	7.723307E 01	1.764050E 01	4.808756E-03
5.000000E-03	-2.864501E 02	-7.655247E-06	2.702594E-05	-1.389860E-03	-3.044392E 03
	4.715989E-07	-4.362532E 04	-4.667404E 04	$\boxed{3.475825E\ 03}$	$\boxed{3.475828E\ 03}$
	1.389860E-03	1.390331E-03	4.485078E 01	-2.271415E 01	1.631474E-03
6.000000E-03	1.122987E 01	-8.336192E-06	2.742118E-05	5.448747E-05	-4.514784E 03
	4.552175E-07	-4.150305E 04	-4.602171E 04	-1.350805E 02	-1.350806E 02
	-5.448747E-05	-5.403225E-05	-1.758311E 00	-7.031134E 01	-6.253165E-04
7.000000E-03	2.883610E 02	-7.582258E-06	2.623302E-05	1.399132E-03	1.944545E 03

Figure 4-4. Typical SAP/MIDAS Program Output

```
RO                   , DVD2     , ADD9     , ADD2     , ITG1     , ADD8
RO                   ,          ,          ,          ,          ,

PLOT        0.100000E 02  0.                  0.100000E 02
 1   SAP DRIFT RATE ERROR, DEG/HR, VS TIME, SEC.
IT          MPY66        0.              0.100000E 01
 2   SAP PLAT ANGLE, RAD, VS TIME, SEC.
IT          ITG2         0.              0.100000E 01
 3   SAP FLOAT ANGLE, RAD, VS TIME, SEC.
IT          DVD7         0.              0.100000E 01
 4   SAP PLAT RATE, RAD/SEC, VS TIME, SEC.
IT          ADD3         0.              0.100000E 01
 5   SAP PLAT TORQUE, DY-CM, VS TIME, SEC.
IT          ITG7         0.              0.100000E 01
 6   SAP THETA X W3, RAD/SEC, VS TIME, SEC.
IT          MPY72        0.              0.100000E 01
 7   SAP  HR X W2, DY-CM, VS TIME, SEC.
IT          NEG2         0.              0.100000E 01
 8   SAP  D(JW1+NL)/DT, DY-CM, VS TIME, SEC.
IT          ADD1         0.              0.100000E 01
 9   SAP  D(IW2+NL)/DT, DY-CM, VS TIME, SEC.
IT          ADD7         0.              0.100000E 01
10   SAP  HR(W1+OW3), DY-CM, VS TIME, SEC.
IT          MPY13        0.              0.100000E 01
        2.000000E 00    2.500000E 06    2.200000E 10    5.000000E 01    1.000000E 00    1.800000E 03
        1.670000E 03    1.210000E 03    1.820000E 03    3.060000E 04    1.700000E 04    3.050000E 04
        0.              1.000000E 00    0.              0.              0.              1.745000E-02
        0.              6.280000E 00    0.              1.745000E-02    0.              6.280000E 00
        0.              0.              0.              0.              0.              0.
        0.              0.              0.              0.              0.              0.
        0.              1.000000E 00    4.500000E-01    8.400000E-01    2.240000E 02
        0.              0.              0.              0.              0.
        0.              1.000000E 00    1.000000E 00    0.              0.
        1.500000E 01    3.300000E-03    2.000000E-01    2.061000E 05
        3.300000E-05
```

Figure 4-5. Plotting and Input Data Sheet, SAP/MIDAS Program

Table 4-3. Summary of MIDAS Work

Case	Figure	Plotted Parameters						Forcing Functions						
No.	No. (of Ref. 1)	ϵ_p	θ_p	θ	L_{pa}	$\theta\omega_3$	Other	L_{pe} lgr-cm	L_e lgr-cm	ω_X $\frac{1\ deg}{sec}$	ω_Y $\frac{1\ deg}{sec}$	ω_Z $\frac{1\ deg}{sec}$	Step Input	Freq cps
1	1	x						x					x	
1	2		x					x					x	
1	3			x				x					x	
1	4				x			x					x	
2	5	x							x				x	
2	6		x						x				x	
2	7			x					x				x	
2	8				x				x				x	
2	9		x							x			x	
3	10						$d(\theta_p)/dt$			x			x	
4	11	x									x		x	
4	12		x								x		x	
4	13			x							x		x	
4	14				x						x		x	
4	15					x					x		x	
4	16	x									x	x	x	
5	17		x								x	x	x	
5	18			x							x	x	x	
5	19				x						x	x	x	
5	20					x					x	x	x	
5	21						$H_r\omega_2$				x	x	x	
5	22						$d(J\omega_1)/dt$				x	x	x	
5	23						$d(I\omega_2)dt$				x	x	x	
5	24						$H_r(\omega_1 + \theta\omega_3)$				x	x	x	
6	25	x									x	x		15
6	26		x								x	x		15
6	27			x							x	x		15
6	28				x						x	x		15
6	29					x					x	x		15
7	30	x									x	x		50
7	31		x								x	x		50
7	32			x							x	x		50
7	33				x						x	x		50
7	34					x					x	x		50

4.3.1 Input Axis Summing Junction

The summing junction at the input axis from Figure 4-1 is magnified in Figure 4-6. It may be seen by inspection that the only linear input signals (neglecting error torques = L''_{pe}) are L_{pa} and $H_r\omega_2$. The output signal consists of a time rate of change of two linear signals and one non-linear signal $(I_x - I_z)\theta\omega_3$. To visually inspect these magnitudes note Figures 4-7 through 4-10. The approximate steady state values of Figures 4-7, 4-8, 4-9, and 4-10 are respectively identified as

$$H_r\omega_2 \approx 4.36 \times 10^4 \text{ dy-cm}$$

$$L_{pa} \approx 4.36 \times 10^4 \text{ dy-cm}$$

$$\frac{d}{dt} J\omega_1 + NL \approx 0 \text{ dy-cm}$$

and

$$\theta\omega_3 \approx 0 \text{ dy-cm}$$

Therefore, it can be determined that on an approximate basis

$$\frac{d}{dt}\left[J\omega_1 - \theta H_r + (I_x - I_z)\underbrace{\cancel{\theta\omega_3}}_{\approx 0}\right] \approx L_{pa} - H_r\omega_2 \tag{4-10}$$

or

$$J_x\dot{\omega}_1 - H_r\dot{\theta} \approx L_{pa} - H_r\omega_2 \tag{4-10a}$$

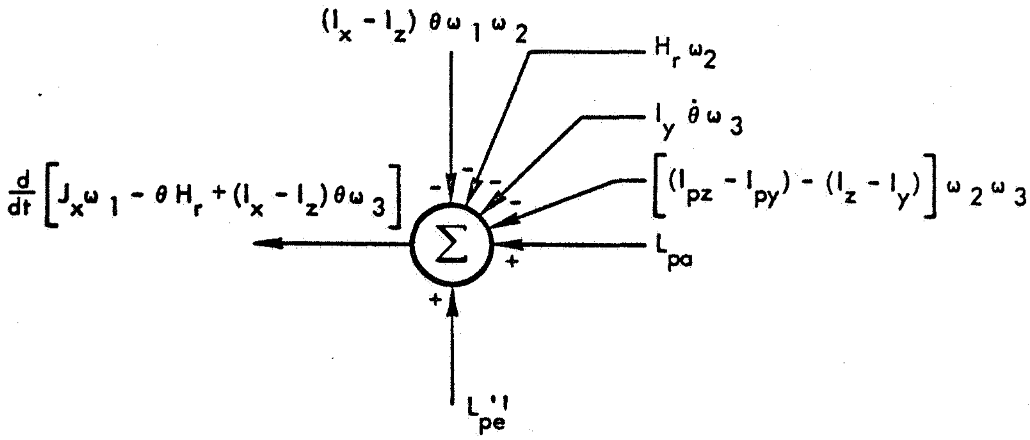
If Equation (4-5) is used to replace ω_1 and both sides of the resulting equation are multiplied by -1 the time domain equation appears as

$$J_x(\ddot{\theta}_p - \dot{\omega}_X) + H_r\dot{\theta} = H_r\omega_z - L_{pa} \tag{4-11}$$

or by use of Laplace transform notation

$$J_x\left[s^2\theta_p(s) - s\omega_X(s)\right] + H_r s\theta(s) = H_r\omega_2(s) - L_{pa}(s) \tag{4-12}$$

Equation (4-12) may be recognized as the linearized input-axis equation that was derived in Reference 1. In order to numerically verify the linearity, floating point numbers extracted from the computer run shown



$$\frac{d}{dt}\left[J_x\omega_1 - \theta H_r + \underbrace{(I_x - I_z)\,\theta\omega_3}_{\approx 0}\right] \approx L_{pa} - H_r\omega_2$$

OR

$$\frac{d}{dt}\left[J_x(\dot{\theta}_p - \omega_X) + \theta H_r\right] \approx H_r\omega_2 - L_{pa}$$

OR

$$\underbrace{J_x S\left[S\,\theta_p(s) - \omega_X(s)\right] + H_r S\,\theta(s) = H_r\,\omega_2(s) - L_{pa}(s)}_{\text{LINEAR "INPUT AXIS EQUATION"}}$$

Figure 4-6. Input Axis Summing Junction

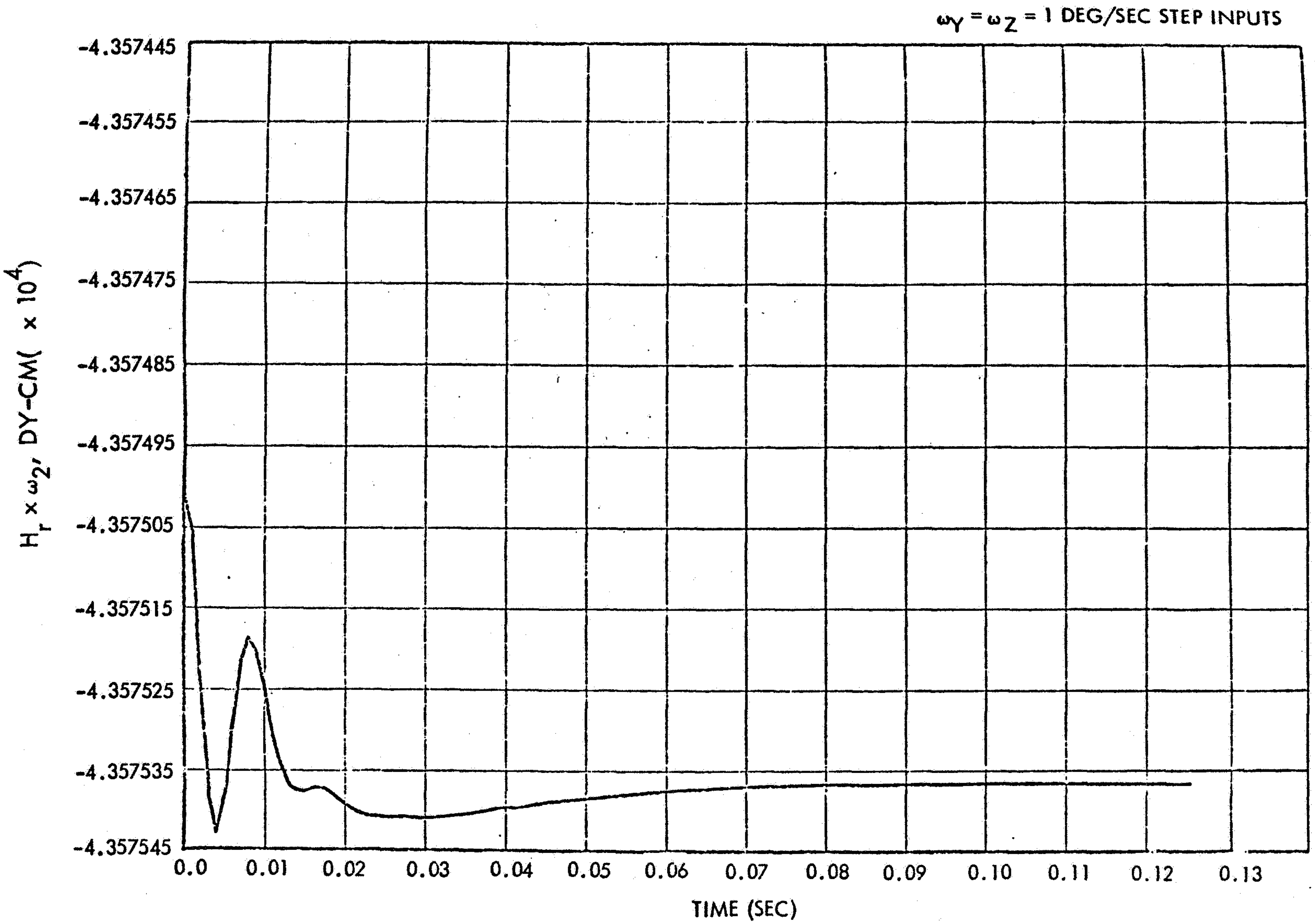


Figure 4-7. SAP- $H_r \times \omega_2$, DY-CM, versus Time (Sec)

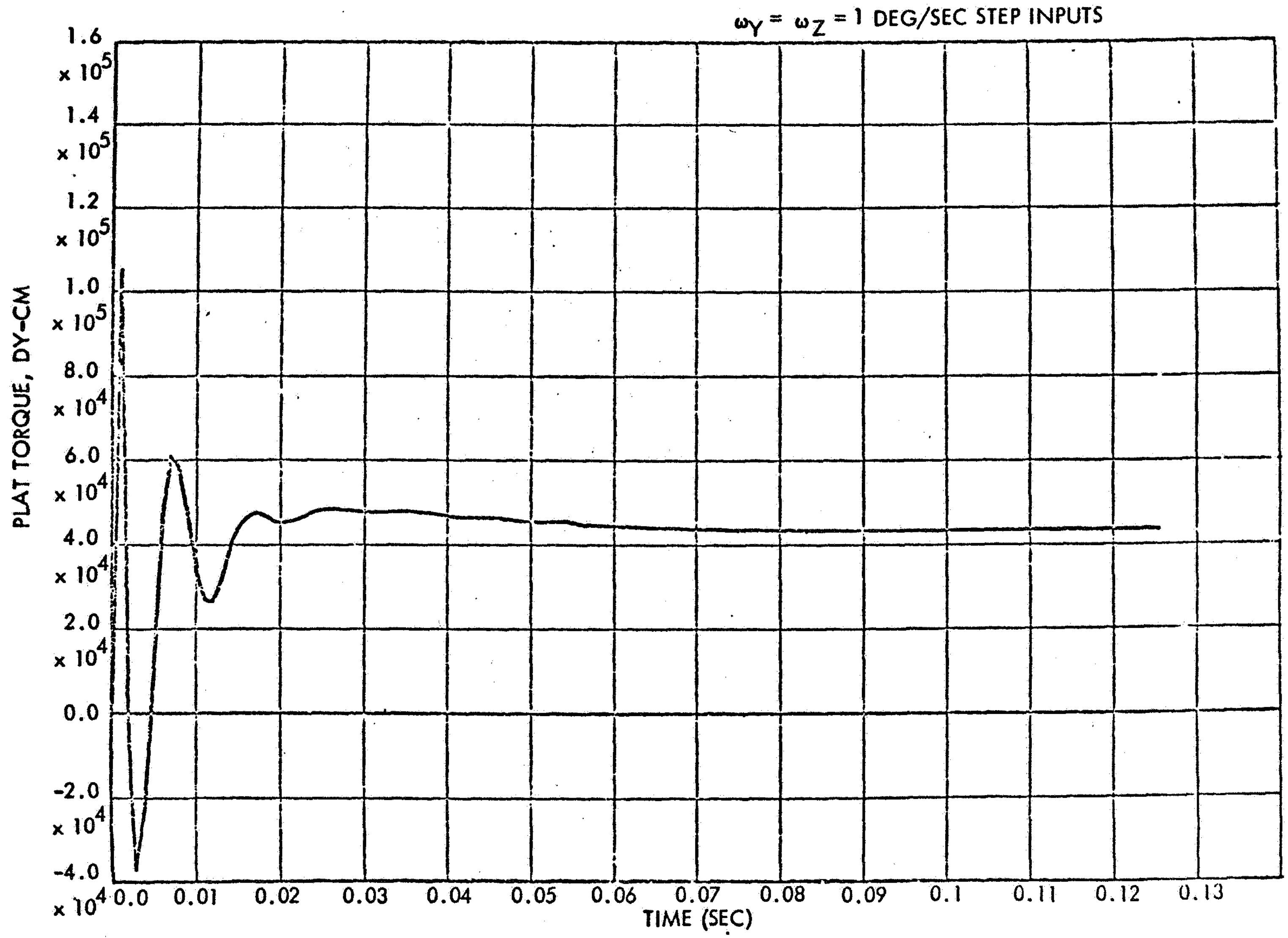


Figure 4-8. SAP Plat Torque, DY-CM

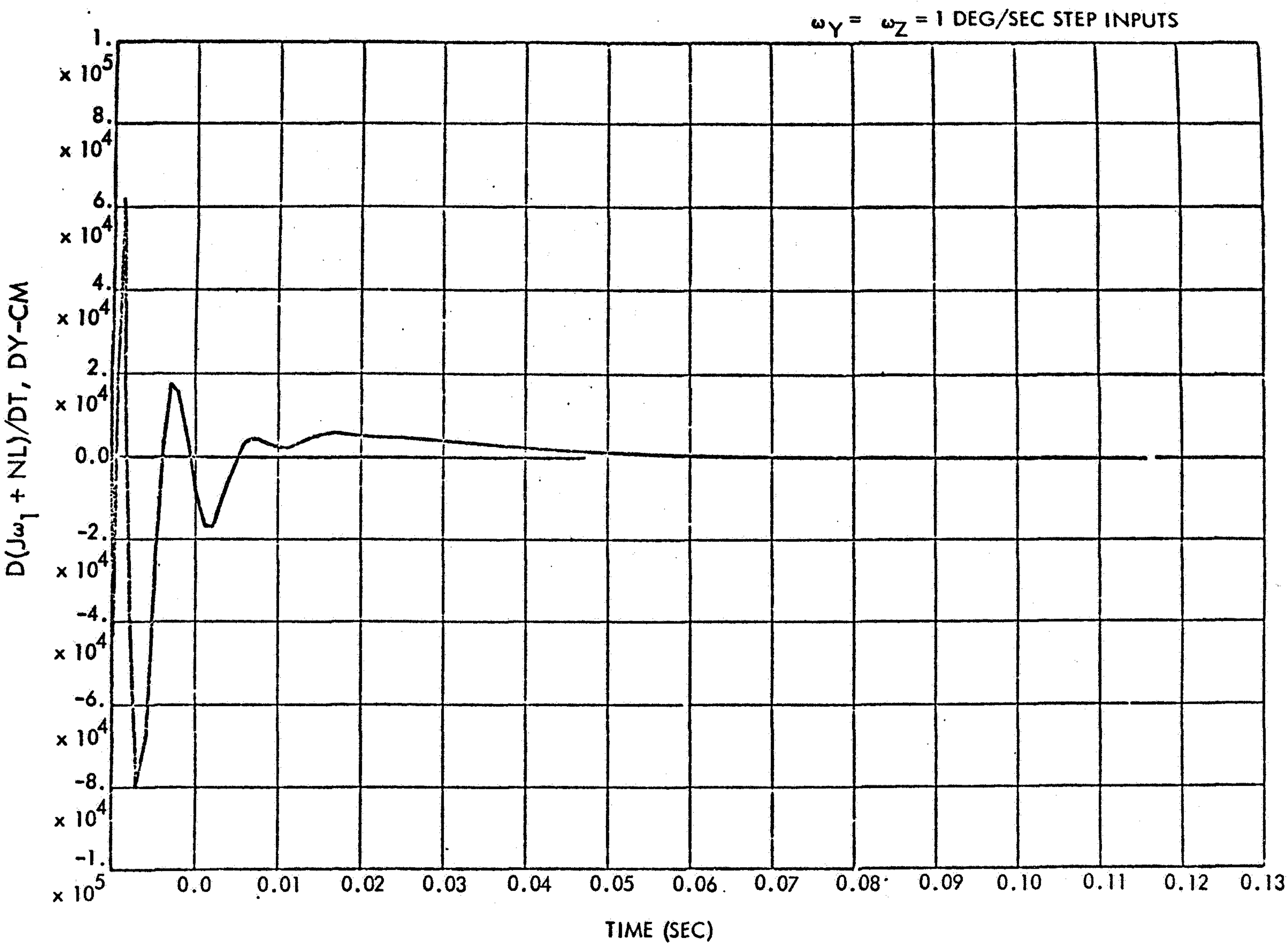


Figure 4-9. SAP $D(J\omega_1 + NL)/DT$, DY-CM, versus Time (Sec)

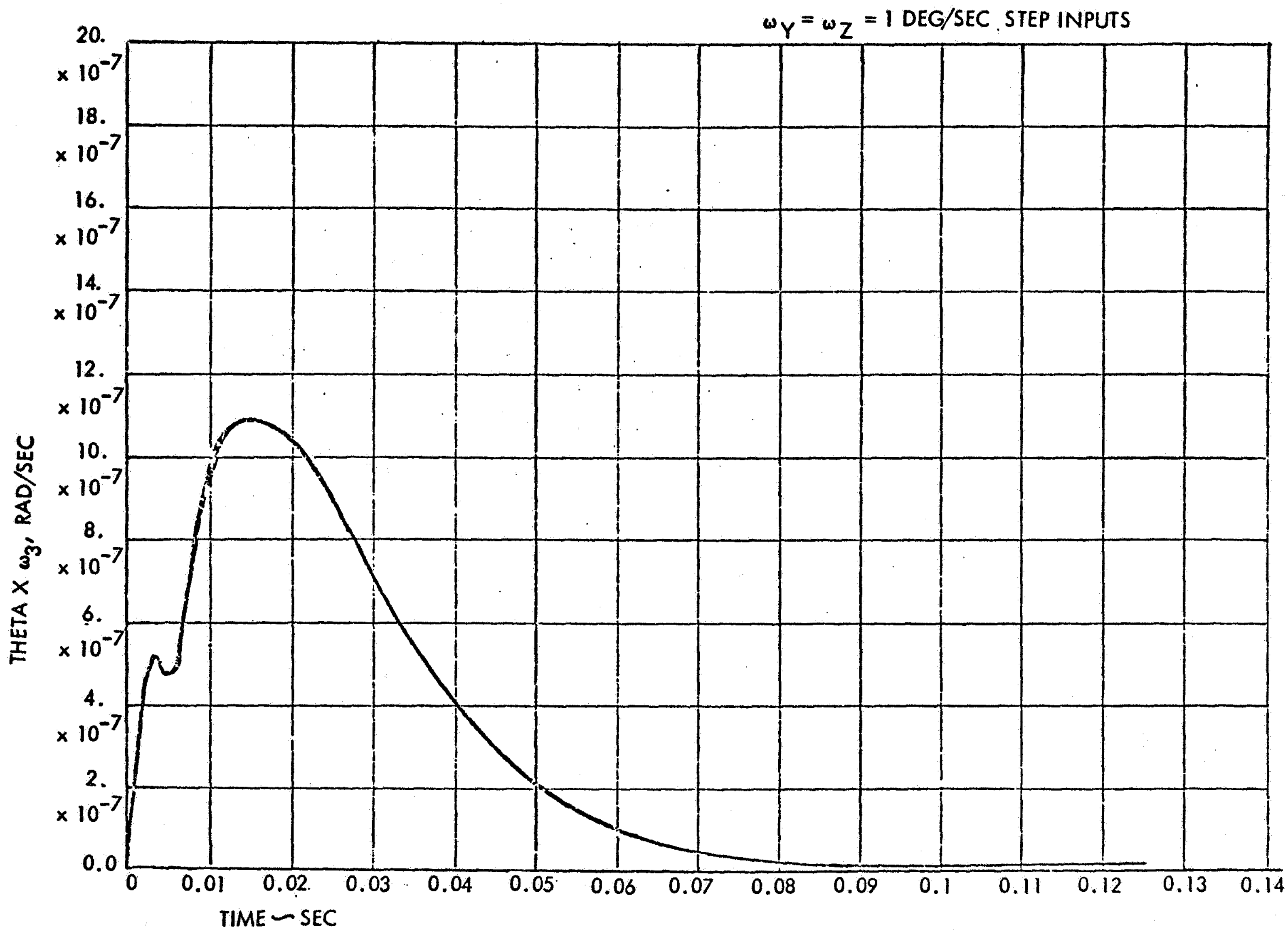


Figure 4-10. SAP $\theta \times \omega_3$, Rad/Sec, versus Time

in Figure 4-4 may be used. The second time interval, 0.002 sec, is arbitrarily selected for demonstrating purposes. The enclosed numbers in that figure are entitled in the headings above them. The difference between L_{pa} and $H_r\omega_2$ is

$$\begin{array}{r} 4.484949 \times 10^4 \\ - \underline{2.563091 \times 10^4} \\ 1.921858 \times 10^4 \end{array}$$

A simple comparison of this difference to that of "DJWIDT," 1.921699×10^4, indicates a direct relationship until the fifth significant figure. The percent of deviation of the linearity in this case is

$$\frac{(1.921858-1.921699) \times 10^4}{1.921699 \times 10^4} \times 100 = 0.0083\%$$

4.3.2 Output Axis Summing Junction

Linearity at the output axis is exhibited in a similar manner. In the case of the output axis summing junction, blown up in Figure 4-11, there is only one linear input signal $H_r\omega_1$. This signal is included in the expression $H_r(\omega_1 + \theta\omega_3)$. The magnitude of the nonlinear signal $\theta\omega_3$ has just been shown to be essentially a negligible quantity (when directly compared to linear signal magnitudes). Therefore it is necessary to show that

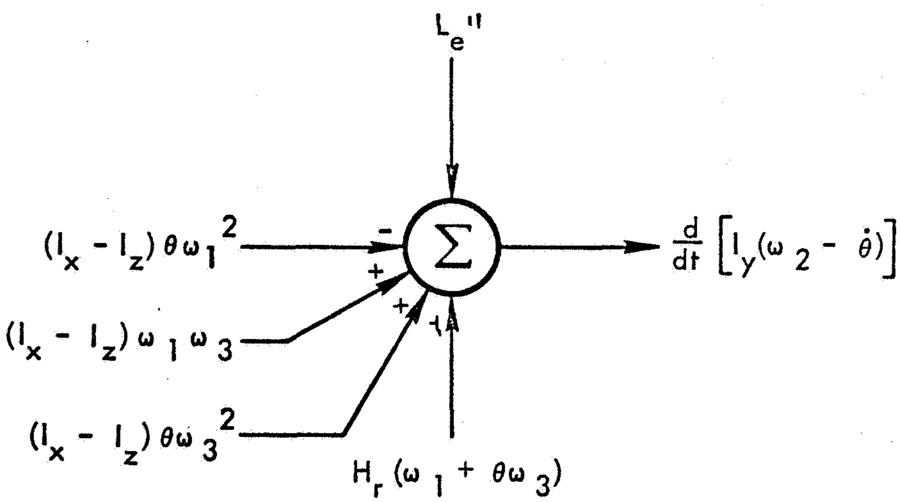
$$\frac{d}{dt}\left[I_y(\omega_2 - \dot{\theta})\right] \approx H_r\omega_1 \qquad (4\text{-}13)$$

This can be substantiated by visual inspection of Figures 4-12 and 4-13 which represent time traces of the two expressions involved. These two curves are essentially identical. The linearized output-axis equation results from substitution of Equation (4-5) for ω_1, and changing signs on each term, i.e.

$$I_y(\ddot{\theta} - \dot{\omega}_z) = H_r(\dot{\theta}_p - \omega_x) \qquad (4\text{-}14)$$

or in Laplace notation,

$$I_y s^2\theta(s) - H_r s\theta_p(s) = - H_r\omega_X(s) + I_y s\omega_2(s) \qquad (4\text{-}15)$$



$$\frac{d}{dt}\left[I_y(\omega_2 - \dot\theta)\right] \approx H_r(\omega_1 + \underbrace{\cancel{\theta\omega_3}}_{\approx 0}) = H_r(\omega_X - \dot\theta_p)$$

OR

$$\underbrace{I_y S^2\,\theta(s) - H_r S\,\theta_p(s) = -H_r\,\omega_X(s) + I_y S\,\omega_2(s)}_{\text{LINEAR "OUTPUT AXIS EQUATION"}}$$

Figure 4-11. Output Axis Summing Junction

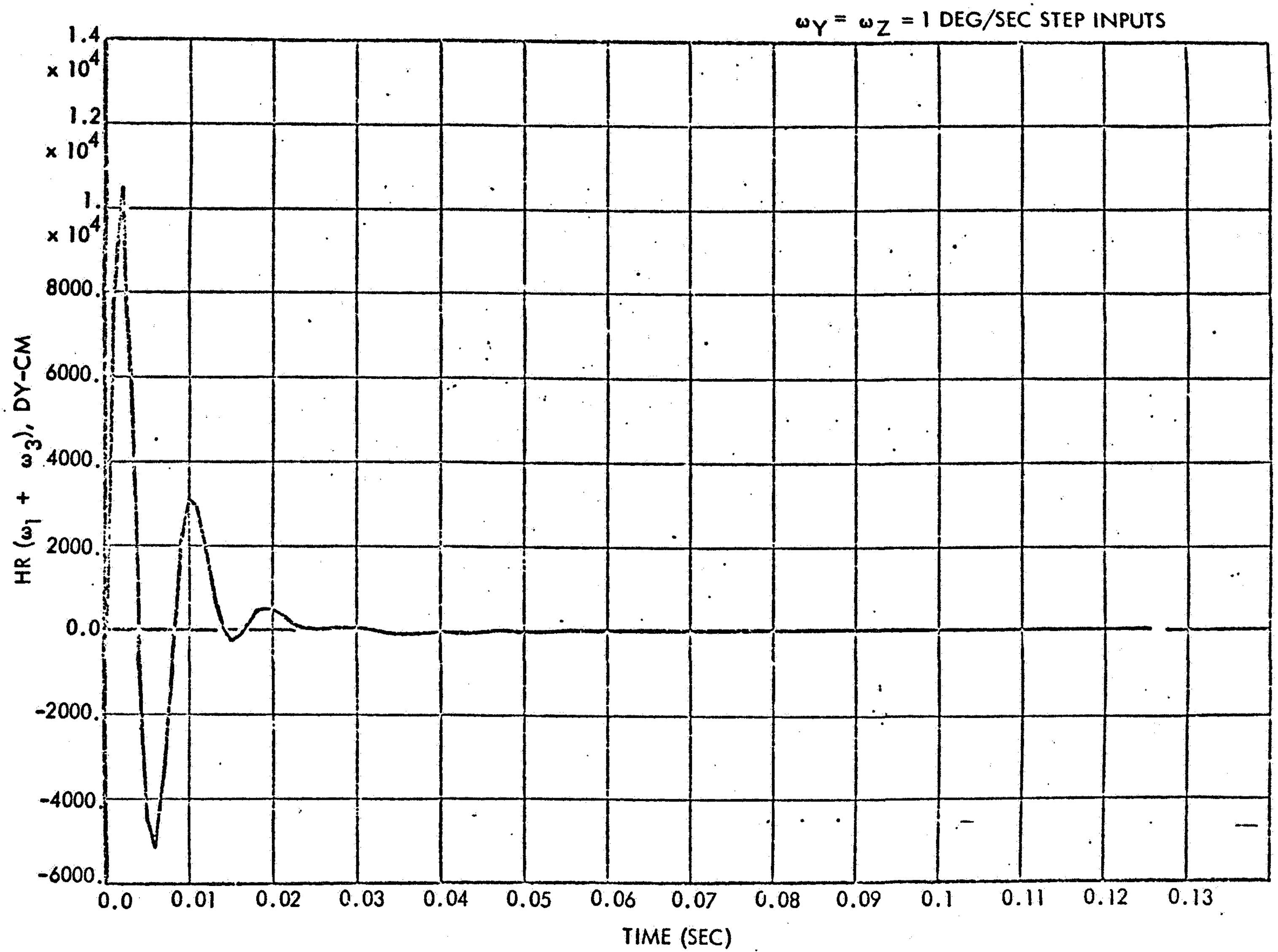


Figure 4-12. SAP HR($\omega_1 + \omega_3$), DY-CM, Versus Time (Sec)

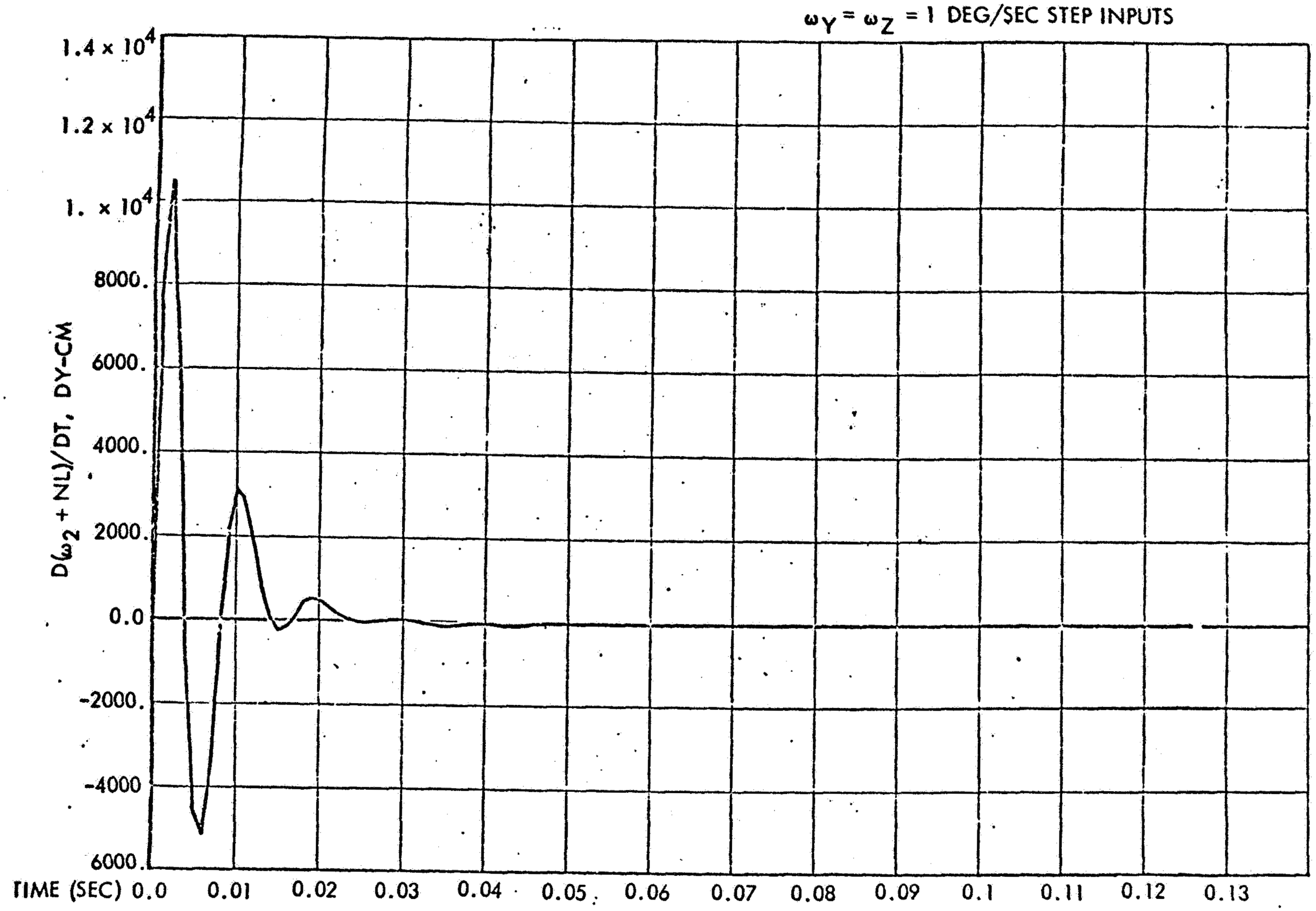


Figure 4-13. SAP $D(I\omega_2 + NL)/DT$, DY-CM, Versus Time (Sec)

The numerical illustration is based on the 0.005 second time interval. Note again on Figure 4-4 that the magnitudes of the two terms of interest are 3.475825×10^3 and 3.475828×10^3. Not until the last significant figure does any difference occur. The percent of linear deviation is

$$\frac{(3.475828 - 3.475825) \times 10^3}{3.475825 \times 10^3} \times 100 = 8.63 \times 10^{-5}\%$$

The establishment of the linear mode of SAP operation serves two important purposes:

(1) Verification of the Phase I assumptions of linearity. The simplified SAP block diagram from Reference 1 is therefor representative, see Figure 4-2.

(2) Permits use of an inexpensive digital program that is designed to compute drift rectification errors of a linear system when excited by either sinusoidal or power spectral density forcing functions.

The following section explains this program and applies it to the MSFC designed SAP. In that section a correlation between time domain and frequency domain solutions further serves to verify linearity.

4.4 DIGITALLY DETERMINED RECTIFICATION ERRORS—FREQUENCY DOMAIN

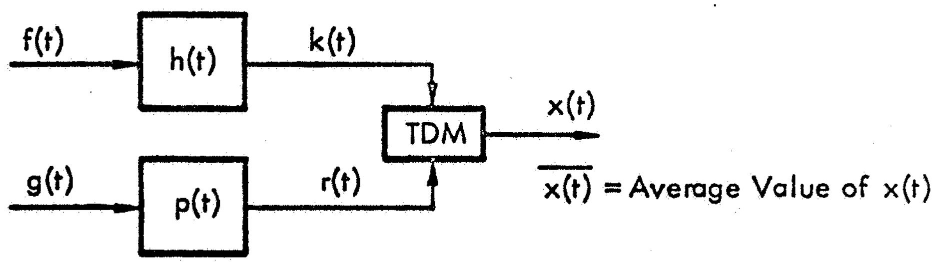
A digital computer program has been developed which determines drift rectification errors that occur in a linear system due to either sinusoidal or random forms of excitation. This program, identified as TRW No. AG040—Drift Rectification Program, operates in the frequency domain on system configurations expressed by transfer function pole-zero-gain characteristics. The forcing functions are either sinusoidal amplitudes and phase angle or power spectral density (PSD). The primary outputs are either a drift frequency sensitive coefficient (identical to a sine response) and an accumulated integration of random induced drift versus frequency. The type of calculations performed were those demonstrated in Section 7 of Reference 1. A general description of the program follows.

4.4.1 Program Description

The basic mathematical model that demonstrates a drift rectification error is shown in Figure 4-14. The actual rectification phenomenonoccurs as a result of a time-domain-multiplication (TDM) of two separate vibrational inputs after they have propogated through different inertial sensor servo paths. These paths might reflect alternate passages through a single sensor or might represent corresponding paths through two different sensors. In the case of SAP rectification, such as that treated in Reference 1, the case of an alternate passage through one sensor applies. Fictitious coning is an example of the case of different sensors (see subsection 4.6, Fictitious Coning and Reference 5).

Although the actual rectification phenomenon occurs in the time domain (i.e., TDM), it is often more convenient and useful to express the behavior in terms of frequency domain concepts (amplitude, phase, etc.). Such an approach was employed (using hand analyses techniques) to determine SAP rectification errors in Reference 1. To this extent the Drift Rectification Program was patterned after the calculations performed in that reference.

As described in Reference 1, the basic task consists of separately determining the frequency response of two independent inputs, as each propagates through a different transfer function, and then calculating



General Equation - Frequency Response

(in Rectification Program)

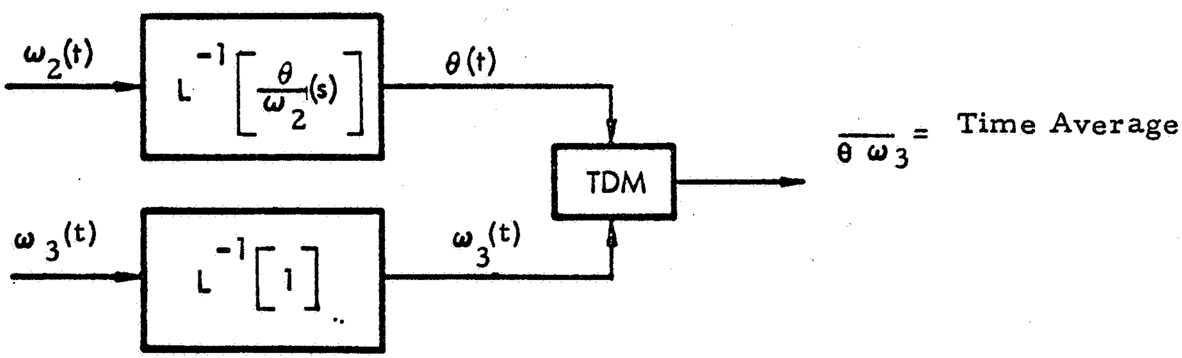
$$\overline{x(t)} = \int_0^{f_{max}} \Phi\ (f)\ C(f)df$$

$$\Phi\ (f) = \text{PSD} \left\{ \begin{array}{l} \text{Fourier Transform of} \\ f(t)\ g(t) \text{ correlation} \end{array} \right\}$$

$C\ (f)$ = Rectification Coefficient

$$= 1/2\,|H(f)|\ |P(f)\ \cos\left(\angle H(f) - \angle P(f)\right)$$

ϵ_3, SPIN-OUTPUT RECTIFICATION



$$\theta\omega_3 = \int_0^{f\ max} \left[\Phi_{23}\ (f)\right] \left[1/2 \left|\frac{\theta}{\omega_2}\ (f)\right| \cos \angle\frac{\theta}{\omega_2}\ (f)\right] df$$

Figure 4-14. Rectification Model

the product of the two responses. Two types of inputs are to be considered: both sinusoidal and random excitations. Sinusoidal inputs are merely amplitudes of two independent inputs and the phase angle between them. The random input will be expressed as a power spectral density. When random inputs are being considered the sinusoidal response in a normalized form serves as a frequency sensitive coefficient. The rectification error for each case described above is mathematically expressed by a single equation pertaining to that case.

Several portions of the Rectification Error Program have been borrowed from the TRW utility program, Frequency Response Subroutine. This routine was modified to allow the passage through two transfer functions per case. It was also modified to use either a root locus gain or a Bode gain in calculating the frequency response. A flow diagram on the program is shown in Figure 4-15.

The following subsections describe some of the significant blocks shown in Figure 4-15.

4.4.2 Built-in Frequencies

The frequency table will contain a set of built-in values of frequency at which frequency response will be computed. The first value, 0.1 rad/sec, will define the low end of the table. The first zero value will delimit the upper end of the table. This provision will allow additional frequency values to be input at the upper end of the table. Also, it allows any part of the built-in set to be used or a completely different set to be input over the built-in set. The range of the standard built-in set is 0.1 rad/sec to 12,600 rad/sec.

4.4.3 Additional Frequencies

At values of circular frequency, ω near the imaginary part of any of the poles or zeros, a more detailed description of a transfer function, G(s), may be desired, particularly if the real part of the pole or zero is small. For these frequencies a separate table will be constructed, which has two entries per additional frequency required. These requirements are either:

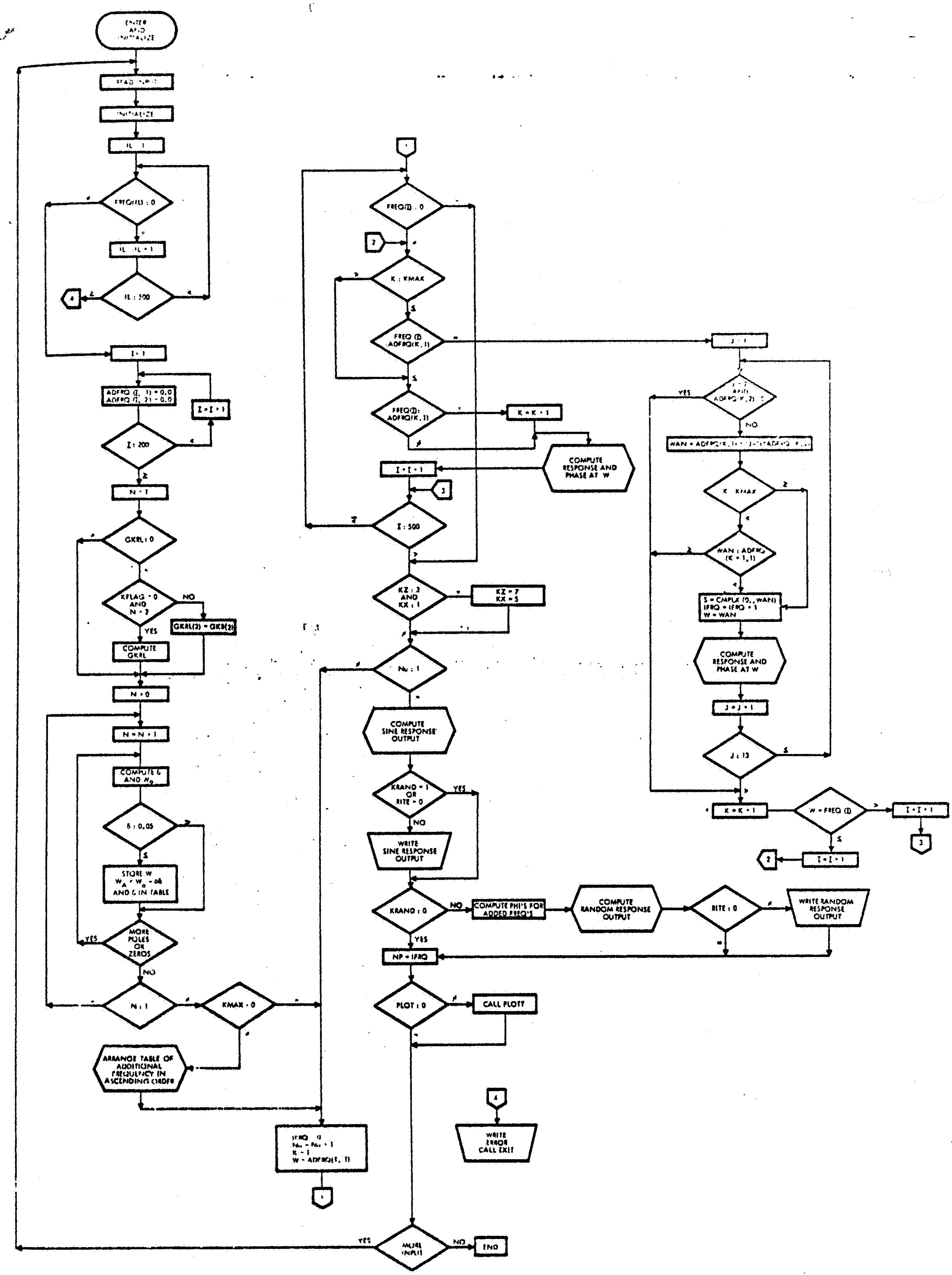


Figure 4-15. Computing Flow Diagram Drift Rectification Program

(1) A lower value of the range of additional frequencies

(2) An increment that will be added to produce 13 additional frequencies centered at the point of interest (resonant frequency).

This table must be constructed, arranged in ascending order, and the entries counted before inclusion with the built-in set begins.

This table will be

Engineering Symbol	Program Symbol (Dimension)
ω_A	ADFRQ(100, 2)

For all poles and zeros compute:

Center Frequency: (pole or zero = a + jb)

$$\omega_0 = \sqrt{a^2 + b^2} \tag{4-16}$$

Increment: (Note: An entry is made only if $\zeta \leq .05$)

$$\zeta = \frac{|a|}{\omega_0} \tag{4-17}$$

Lower Frequency:

$$\omega_A = \omega_0 - b\zeta \tag{4-18}$$

Thus the kth entry in the table will be ADFRQ(k, 1) = ω_A and ADFRQ(k, 2) = ζ. These frequencies define the behavior of $G(j\omega)$ in regions where phase and amplitudes are rapidly changing.

4.4.4 Modified "Frequency Response Subroutine"

The modified Frequency Response Subroutine computes the values of two transfer functions.

$$G_1(s) = \frac{N_1(s)}{D_1(s)} \tag{4-19a}$$

$$G_2(s') = \frac{N_2(s')}{D_2(s')} \tag{4-19b}$$

for given sets of values of s and s', where s and s' are Laplace transform representations of complex numbers. The modification also allows one of the transfer functions, say G_2, to be equal only to a pure gain. In other words, G_2 has no poles or zeros associated with it.

The transfer function is given as a ratio of two complex polynominals that have been factored into products of the roots of each of the polynomials. Thus, the transfer function in root locus form is given as

$$G(s) = \frac{N(s)}{D(s)} = K_{RL} \frac{(s - z_n)(s - z_{n-1})\ldots(s - z_1)}{(s - p_m)(s - p_{m-1})\ldots(s - p_1)} = K_{RL} \frac{\prod_{i=1}^{n} (s - z_i)}{\prod_{j=1}^{m} (s - p_j)} \quad (4\text{-}20a)$$

In more useful Bode form this equation appears as

$$G(s) = K_B \frac{\prod_{i=1}^{n} (s/z_i - 1)}{\prod_{j=1}^{m} (s/p_j - 1)} \quad (4\text{-}20b)$$

where p_i and z_i are complex numbers and are called the poles and zeros of the transfer function. K_{RL} and K_B are constants representing root locus and Bode gains.

Values of s that are pure imaginary and have a magnitude of ω, i.e., $s = 0 + j\omega$, are substituted into the equation for G and corresponding values of G are computed by another routine. The resulting values of G are complex numbers, a magnitude and an angle; the phase angle is represented by ϕ. These values of magnitudes and phase angles along with some of the input data are used to compute the sinusoidal response.

Responses

Case 1 - Sinusoidal Response

The equation for the sinusoidal induced error is

$$\epsilon_p(\omega_i) = \frac{1}{2} (K_0 X_1 X_2) |G_1| |G_2| \cos (\phi_1 - \phi_2 - \psi) \quad (4\text{-}21)$$

where $|G_i|$ is the magnitude of G_i at the frequency of interest, and ϕ is the corresponding phase angle. K_0, X_1, X_2, and ψ are data inputs. X_i is the amplitude of sinusoidal motion, ψ represents the phase angle between X_1 and X_2, and K_0 is an input gain used to convert the response to proper units.

Case 2 - Random Response

The equation for calculating the random response is

$$\text{Random Response} = \sum_{i=1}^{N} \Phi(\omega_i) C_s(\omega_i) \Delta\omega_i \tag{4-22}$$

The random response is dependent upon the sinusoidal response. However, X_1 and X_2 must be normalized, i.e., set equal to one, in order to obtain the frequency sensitive rectification coefficient, $C_s(\omega)$, needed to calculate the random response. This coefficient is defined as

$$C_s(\omega) = \frac{\epsilon_p(\omega)}{X_1 X_2} \tag{4-23}$$

In place of the sinusoidal amplitudes, X_1 and X_2, the magnitude of the random forcing function, $\Phi(\omega)$ expressed as a power spectral density at each frequency must be specified. This requires the program to add values of $\Phi(\omega)$ that correspond to the added frequencies. These values of $\Phi(\omega)$ are calculated by interpolation as follows:

$$\Phi_A(\omega) = \frac{\Phi_U(\omega) - \Phi_L(\omega)}{\omega_U - \omega_L} \times (\omega_A - \omega_L) + \Phi_L(\omega) \tag{4-24}$$

where $\Phi(\omega)$ represents the value of the PSD forcing function, and ω represents the circular frequency.

The subscripts are as follows:

A - An added value

L - The last input value used

U - The next input value to be used.

Also needed are frequency bandwidths which must be established for purpose of integration. The bandwidths $\Delta\omega$ are found as follows

$$\Delta\omega_i = \frac{\omega_{i+1} - \omega_{i-1}}{2} \tag{4-25a}$$

where ω is neither the first frequency nor the last. By this method the bandwidth will tend to center around ω_i. The first and last bandwidths are the exceptions. The first bandwidth is calculated as follows

$$\Delta\omega = \frac{\omega_2 - \omega_1}{2} \tag{4-25b}$$

while the last is

$$\Delta\omega = \frac{\omega_n - \omega_{n-1}}{2} \tag{4-25c}$$

The random response shown in Equation (4-22) will also be referred to as an accumulated random integration. It provides random-induced rectification errors attributed to a power spectrum existing through frequency ω_i. Also desired as output is a supplementary set of necessary calculations to show the contribution of each element of area to the overall area. This is shown as a percent of the total area.

$$\text{Percent of Area} = \frac{\Phi(\omega_i)C_s(\omega_i)\,\Delta\omega_i}{\sum_{i=1}^{N} \Phi(\omega_i)\,C_s(\omega_i)\Delta\omega_i} \times 100 \tag{4-26}$$

In all cases the output is presented versus frequency in cycles per second (cps) rather than circular frequency. Tables 4-4 and 4-5 show the programmed symbols for input and output data respectively. Figures 4-16 and 4-17 show schematically the binary and source program decks.

4.4.5 SAP Rectification Errors

A special case capability of the program is to perform the same type of calculations that were done by hand in Reference 1, Section 7. This capability refers to the setting of the second transfer function G_2 equal to a pure gain. These calculations involved the computation of sinusoidal

*IF PLOTTING IS NOT DESIRED

(1) DON'T USE NOTE CARD

(2) TAPE CARD INDICATES TAPE NUMBER ONLY

*, , , , , , TAPE 50

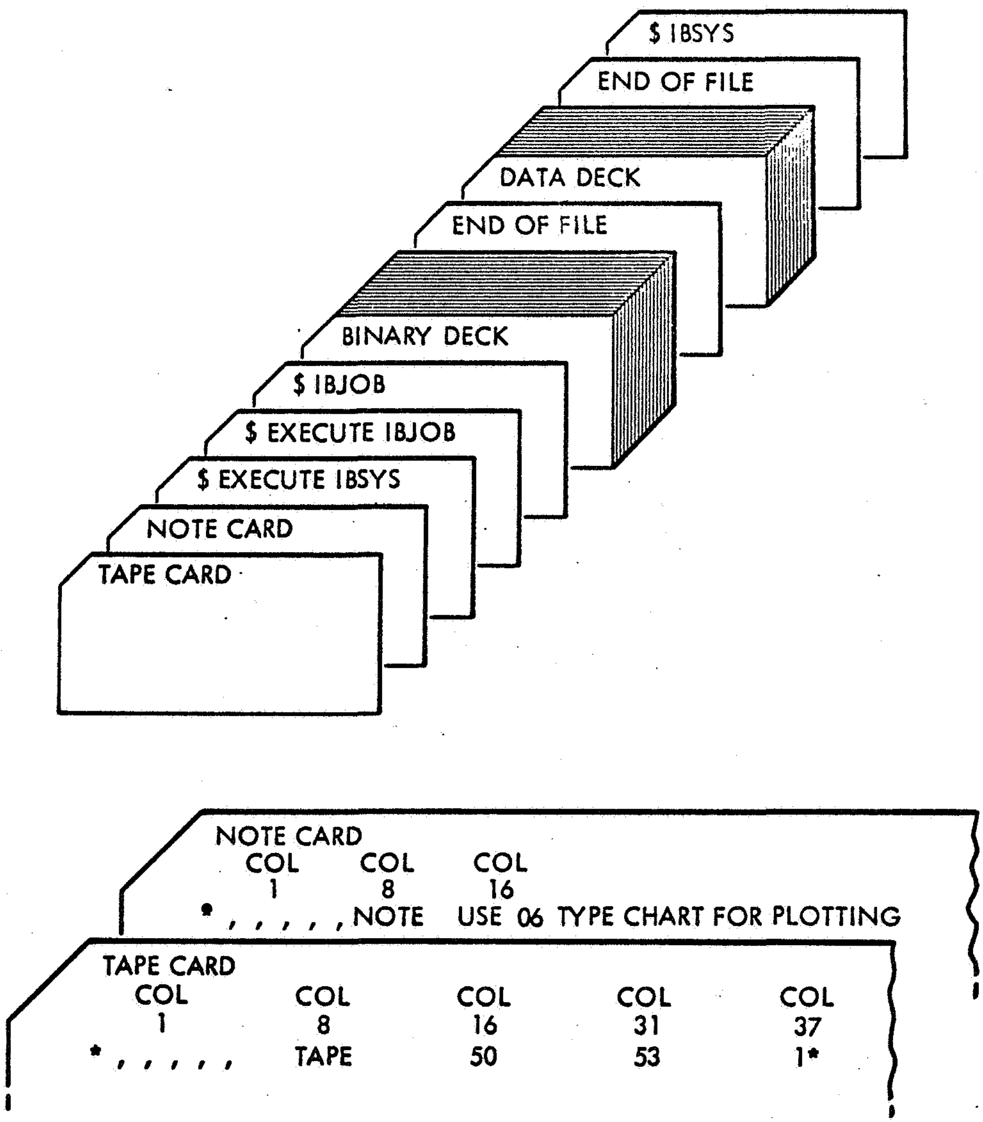


Figure 4-16. Binary Program Deck

*IF PLOTTING IS NOT DESIRED

(1) DON'T USE NOTE CARD

(2) TAPE CARD INDICATES TAPE NUMBER ONLY

*, , , , , TAPE 50

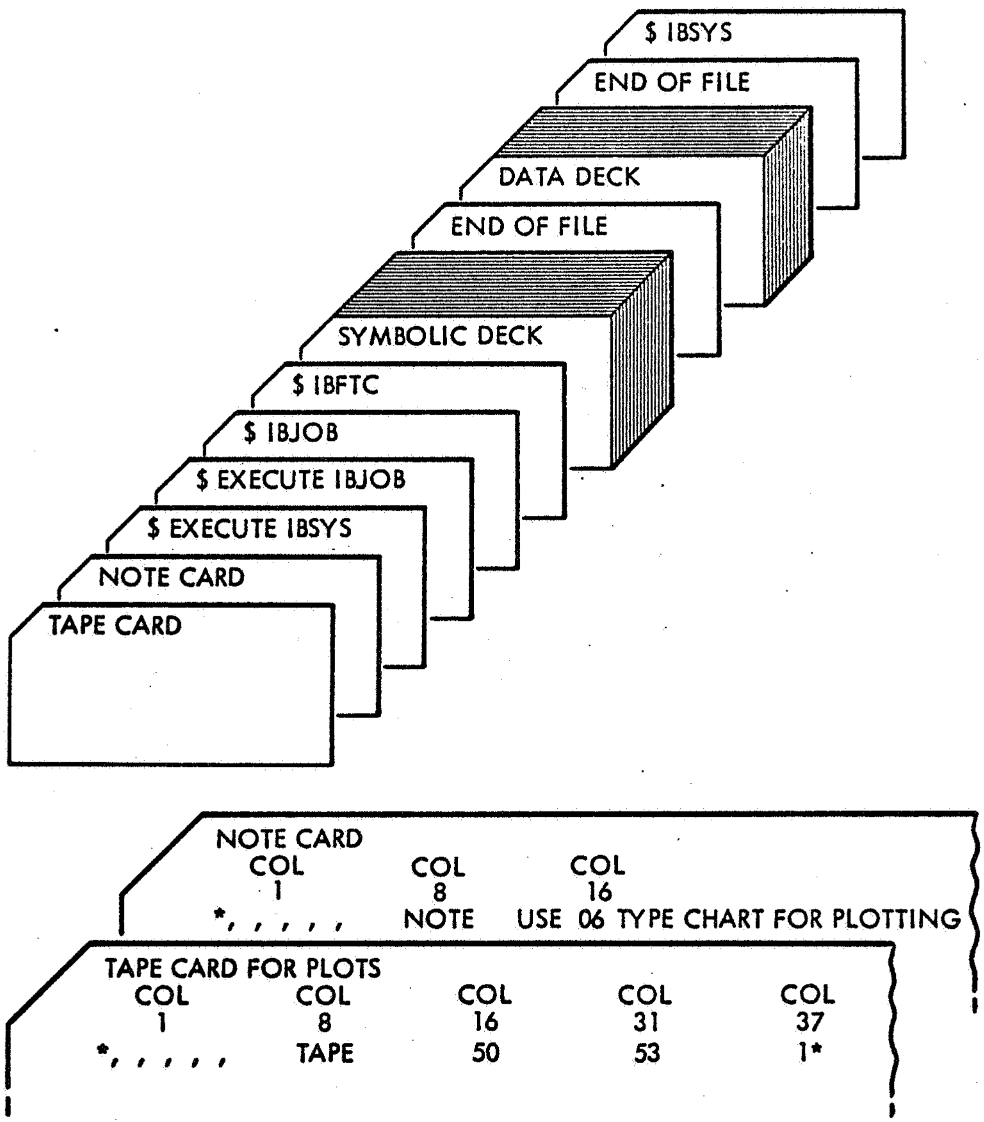


Figure 4-17. Source Program Deck

Table 4-4. Input Symbols of Rectification Program

PLS (1, 1)	Poles for the first transfer function
ZRS (1, 1)	Zeros for the first transfer function
NPL (1)	The number of poles in the first transfer function
NZER (1)	The number of zeros in the first transfer function
GKRL (1)	Root locus gain for the first transfer function
GKB (1)	Bode gain for the first transfer function
KFLAG	If KFLAG is input equal to zero, then the second transfer function is considered to be a constant
PLS (1, 2)	Poles for the second transfer function
ZRS (1, 2)	Zeros for the second transfer function
NPL (2)	The number of poles in the second transfer function
NZER (2)	The number of zeros in the second transfer function
GKRL (2)	Root locus gain for the second transfer function
GKB (2)	Bode gain for the second transfer function
KRAND	If KRAND is input equal to one, the program will compute both the sinusoidal response and the random response. If KRAND is not an input or is an input not equal to one the program will compute only the sinusoidal response.
PSI	Phase angle between the transfer functions
X1	Amplitude of the first transfer function. If X1 is not an input, the amplitude is considered to be one
X2	Amplitude of the second transfer function. If X2 is not an input, the amplitude is considered to be one
KØ	A conversion factor
PHI	Power spectral density power/cps (i. e., g^2/cps, ω^2/cps, etc.) There should be a PHI input for each frequency used from the built-in set of frequencies.
RITE	If RITE is input equal to zero, there will be no printed output
PLØT	If PLOT is input equal to zero, there will be no plotting
$ REDEF1 HEAD = H	This card defines HEAD to be Hollerith information
HEAD	HEAD is the header used on the plots to indicate the case title
FREQ(α)	α represents the counting number of the frequencies in the built-in set of frequencies. If FREQ(α) = 0.0, this will stop all computations after the (α - 1) frequency. If FREQ(α) = β., this will make the frequency at α to be equal to β. (Note: Frequencies must be in ascending order.)

Table 4-5. Output Symbols of Rectification Program

Sinusoidal Response

FREQ (CPS)	Frequencies in cycles per second
RESPONSE	Sinuoidal response
ANGLE (DG)	Difference of phase angles for G_1 and G_2 in degrees (includes input phase angle Ψ)

If plot ≠ O in input data, there will be a plot of the sinusoidal response versus the log frequency.

Random Response

FREQ (CPS)	Frequencies in cycles per second
RESPONSE	Sinusoidal response
ANGLE (DG)	Phase angle in degrees
RAN SUM	Running sum of the random response
RANDOM	Random response within each bandwidth
PERCENT	The percent each random response contributes toward the total response
DELW	The bandwidth of the frequency
RANDOM RESPONSE	The total response, same as the last entry under running sum

If plot ≠ O in input data, there will be two plots.

(1) Sinusoidal response versus log frequency
(2) Running sum versus log frequency

Note: (1) If RITE = O in input data, results will not be printed
(2) Input data will always be reflected in printout.

drift rates normalized to excitation amplitudes. As previously indicated, these normalized drift rates will hereafter be referred to as drift coefficients $C_i(\omega)$ where the i is indexed to drift errors ϵ_3 through ϵ_8 in Reference 1. These coefficients were denoted in Reference 1 as $(\epsilon_3/\Omega_2\Omega_3)(\omega)$, etc.

Certain of these coefficients are further defined to associate them to the type of error that occurs. For example C_3 may be recognized as being induced by angular rates that occur about both the spin and output axis of the SAP gyro. Therefore, it is defined as SAP Spin-Output Rectification. Rectification associated with C_8 may be similarly defined as SAP Spin-Input Rectification.

For demonstration purposes the SAP Spin-Output Rectification is selected. The symbolic representation of this is shown in Figure 4-14. The transfer function $G_1(s)$ is $\theta/\omega_2(s)$ while the second transfer function $G_2(s)$ is simply unity. The input data sheet is reproduced in Figure 4-18. One page of output data is shown in Figure 4-18A. The various headings are defined in Tables 4-4 and 4-5. A graph of the coefficient C_3 versus frequency is shown in Figure 4-19. This figure was also obtained from a computer operation. Figures 4-20 through 4-24 are computer plots of drift coefficients C_4 through C_8 respectively. To obtain a rectification error due to sinusoidal inputs merely multiply the value of the appropriate curve at the specific frequency by the amplitudes of the two forcing functions that are involved.

4.4.6 Linearity Verification

These curves of drift coefficients also serve to verify further the degree of linearity of the MSFC-SAP. Since the Rectification Program is to be used in conjunction with linear systems only, if its results match those of nonlinear MIDAS time domain simulation, then certainly the nonlinear effects must be negligible.

Two specific cases are used to verify this matching of linearity between frequency domain and time domain solutions. Both cases again refer to the ϵ_3, SAP Spin-Output Rectification condition. The frequency of interest for the two cases is arbitrarily selected to be 15 cps and 50 cps respectively. These frequencies compare to peak positive and peak negative values of ϵ_3 in Reference 1. Note on Figure 4-19 that the drift rectification error at these frequencies is 0.146 deg/hr and -0.0036 deg/hr, respectively.

FREQ(CPS)	RESPONSE	ANGLE(DG)	RAND SUM	RANDOM	PERCENT	DELW
0.015915	0.356785E-02	0.557867E 01	0.141960E-04	0.141960E-04	0.000181	0.025000
0.023873	0.356886E-02	0.833288E 01	0.425961E-04	0.284001E-04	0.000362	0.050000
0.031831	0.357028E-02	0.110461E 02	0.710074E-04	0.284113E-04	0.000362	0.050000
0.039789	0.357210E-02	0.137064E 02	0.994333E-04	0.284258E-04	0.000363	0.050000
0.047746	0.357432E-02	0.163035E 02	0.142099E-03	0.426653E-04	0.000544	0.075000
0.063662	0.357998E-02	0.212737E 02	0.199076E-03	0.569772E-04	0.000727	0.100000
0.079577	0.358726E-02	0.259054E 02	0.256169E-03	0.570930E-04	0.000728	0.100000
0.095493	0.359616E-02	0.301722E 02	0.313403E-03	0.572346E-04	0.000730	0.100000
0.111408	0.360667E-02	0.340685E 02	0.370805E-03	0.574020E-04	0.000732	0.100000
0.127324	0.361880E-02	0.376044E 02	0.428400E-03	0.575950E-04	0.000735	0.100000
0.143239	0.363255E-02	0.407999E 02	0.486214E-03	0.578138E-04	0.000738	0.100000
0.159155	0.364792E-02	0.436807E 02	0.660389E-03	0.174175E-03	0.002222	0.300000
0.238732	0.374897E-02	0.543311E 02	0.958723E-03	0.298334E-03	0.003805	0.500000
0.318310	0.389037E-02	0.607965E 02	0.125831E-02	0.309586E-03	0.003949	0.500000
0.397887	0.407205E-02	0.649005E 02	0.159235E-02	0.324043E-03	0.004134	0.500000
0.477465	0.429391E-02	0.676074E 02	0.193405E-02	0.341698E-03	0.004359	0.500000
0.557042	0.455583E-02	0.694373E 02	0.229659E-02	0.362541E-03	0.004625	0.500000
0.636620	0.485767E-02	0.706869E 02	0.268315E-02	0.386561E-03	0.004931	0.500000
0.716197	0.519928E-02	0.715348E 02	0.309690E-02	0.413745E-03	0.005278	0.500000
0.795774	0.558047E-02	0.720941E 02	0.354098E-02	0.444080E-03	0.005665	0.500000
0.875352	0.600106E-02	0.724397E 02	0.401853E-02	0.477549E-03	0.006092	0.500000
0.954929	0.646080E-02	0.726227E 02	0.453266E-02	0.514134E-03	0.006559	0.500000
1.034507	0.695947E-02	0.726794E 02	0.508648E-02	0.553817E-03	0.007065	0.500000
1.114084	0.749680E-02	0.726361E 02	0.568305E-02	0.596576E-03	0.007610	0.500000
1.193662	0.807251E-02	0.725121E 02	0.632544E-02	0.642389E-03	0.008195	0.500000
1.273239	0.868628E-02	0.723224E 02	0.701668E-02	0.691232E-03	0.008818	0.500000
1.352817	0.933780E-02	0.720783E 02	0.775975E-02	0.743078E-03	0.009479	0.500000
1.432394	0.100267E-01	0.717887E 02	0.855766E-02	0.797901E-03	0.010178	0.500000
1.511972	0.107527E-01	0.714606E 02	0.941333E-02	0.855671E-03	0.010915	0.500000
1.591549	0.115153E-01	0.710999E 02	0.107879E-01	0.137453E-02	0.017534	0.750000
1.750704	0.131488E-01	0.702980E 02	0.128806E-01	0.209269E-02	0.026696	1.000000
1.909859	0.149237E-01	0.694109E 02	0.152557E-01	0.237519E-02	0.030299	1.000000
2.069014	0.168363E-01	0.684584E 02	0.179353E-01	0.267958E-02	0.034182	1.000000
2.228169	0.188825E-01	0.674544E 02	0.209406E-01	0.300524E-02	0.038336	1.000000
2.387323	0.210577E-01	0.664094E 02	0.242920E-01	0.335144E-02	0.042753	1.000000
2.546478	0.233573E-01	0.653312E 02	0.280094E-01	0.371744E-02	0.047422	1.000000

Figure 4-18. Output Data for SAP Spin-Output Rectification

```
NPOL(1)=7,
PLS(1,1)=(-71.58,40.18),(-71.58,-40.18),(-654.6,0.0),
(-178.2,715.9),(-178.2,-715.9),(-1933.5,0.0),(-5913.3,0.0),
NZER(1)=6,
ZRS(1,1)=(-1.0,0.0),(-6000.0,0.0),(-1500.0,1000.0),
(-1500.0,-1000.0),(0.0,400.0),(0.0,-400.0),
GKB(1)=1.136E-4,GKB(2)=1.C,KFLAG=0,KO=62.8,
PSI=0.0,PHI=163*1.0,
$.REDEF HEAD=H
HEAD=MSFC PH II RECT ERROR EP3
KRAND=1 ,
$END
```

Figure 4-18A. Input Data for SAP Spin-Output Rectification

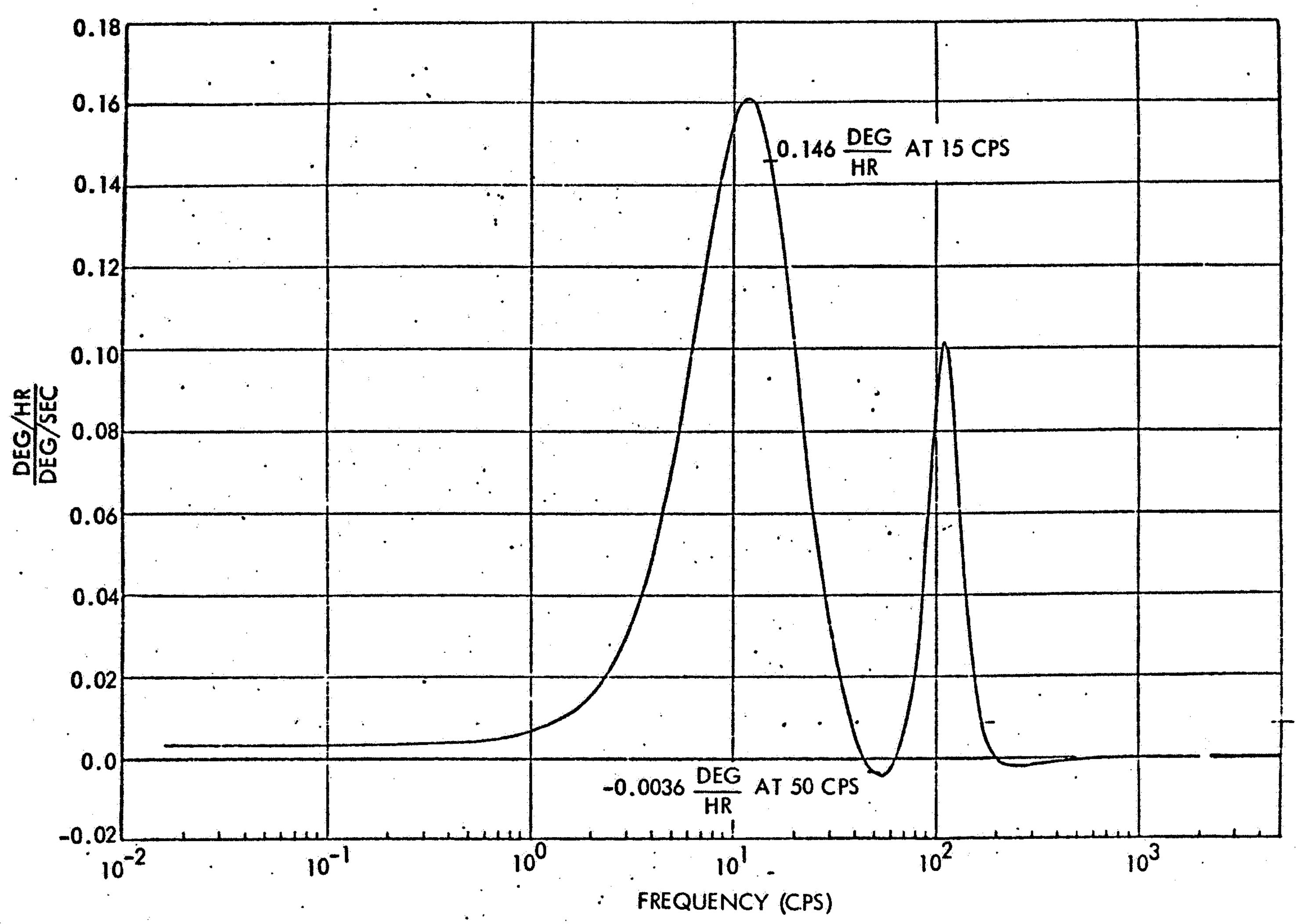


Figure 4-19. ϵ_3 Drift Coefficient

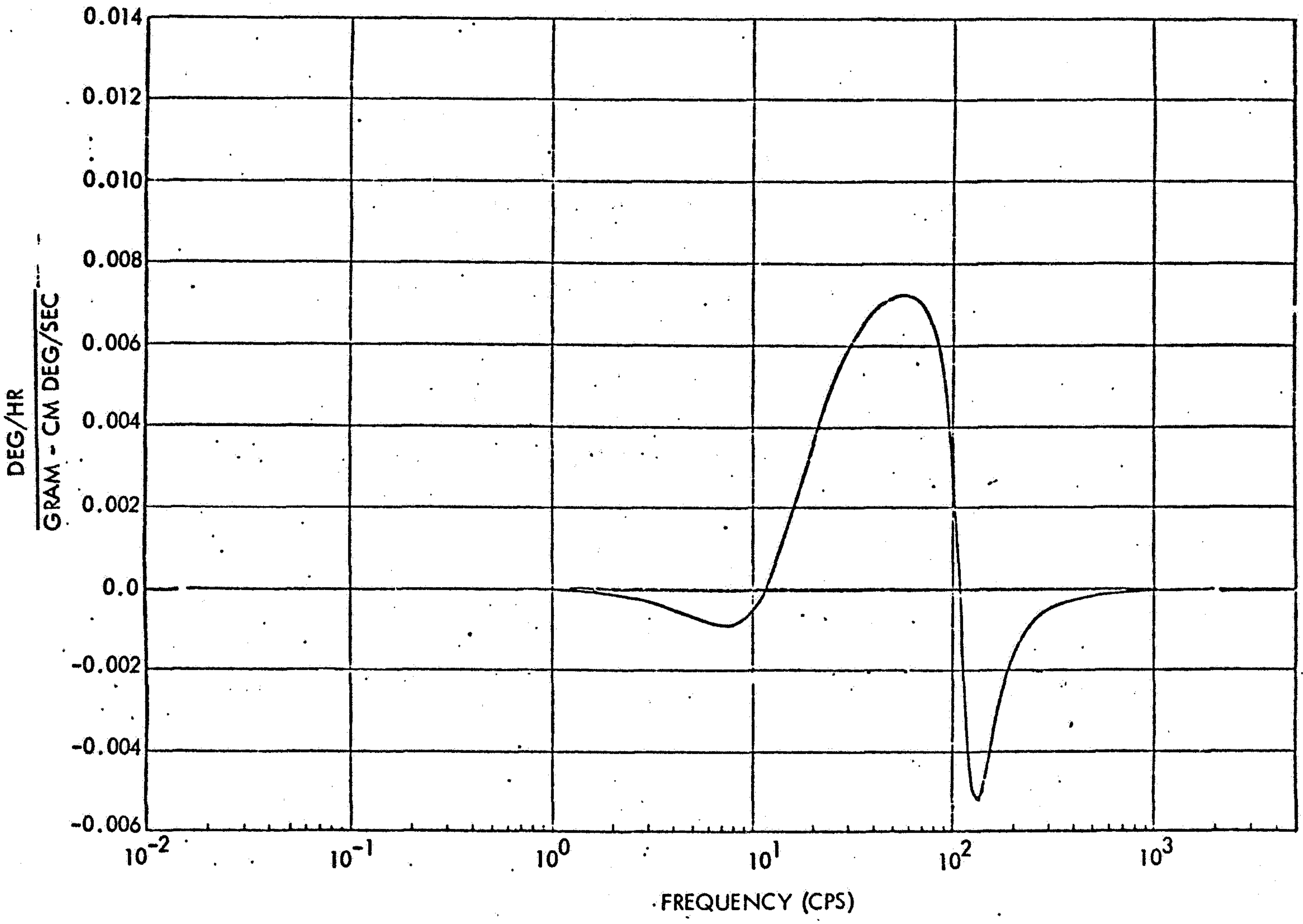


Figure 4-20. ϵ_4 Drift Coefficient

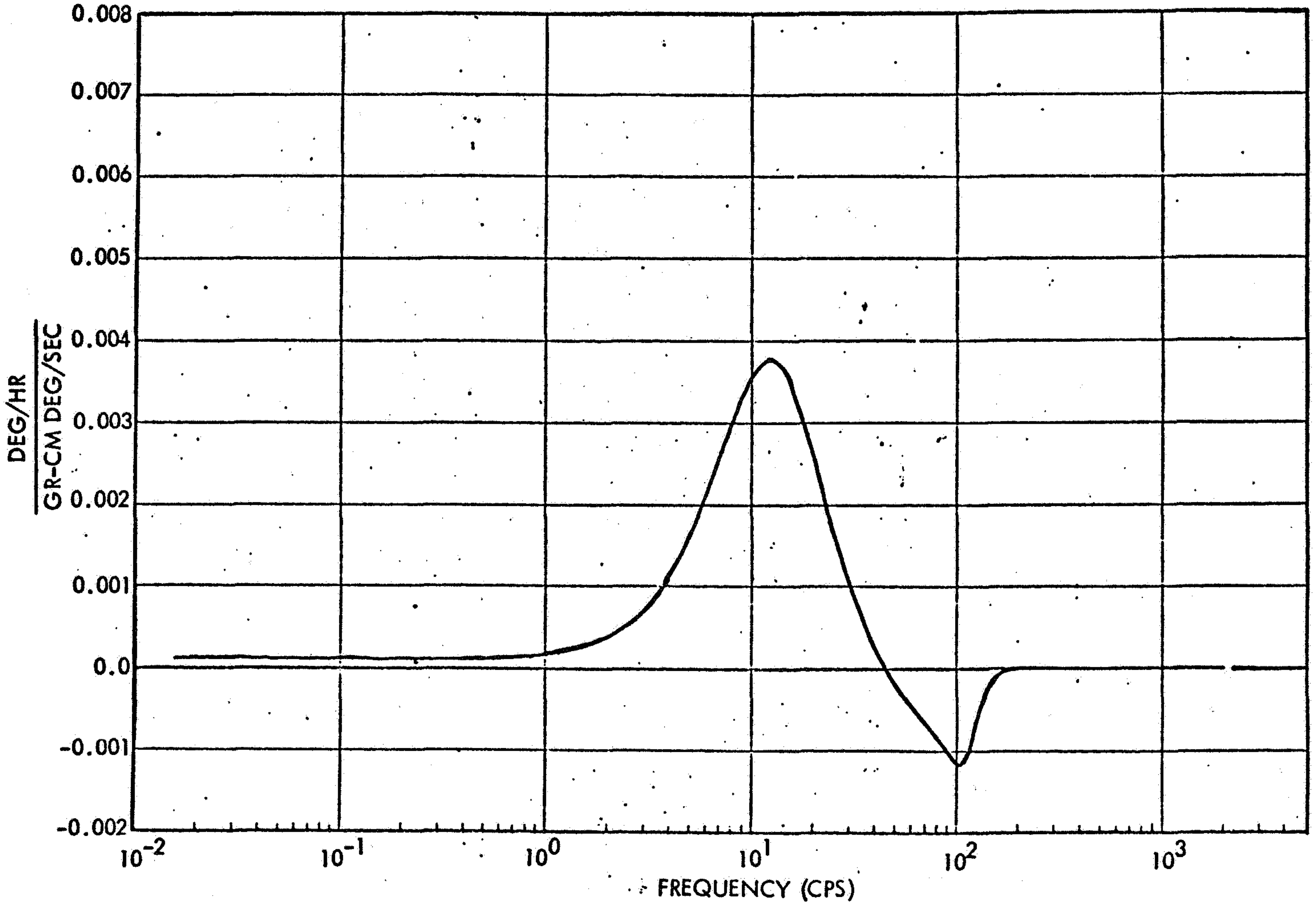


Figure 4-21. ϵ_5 Drift Coefficient

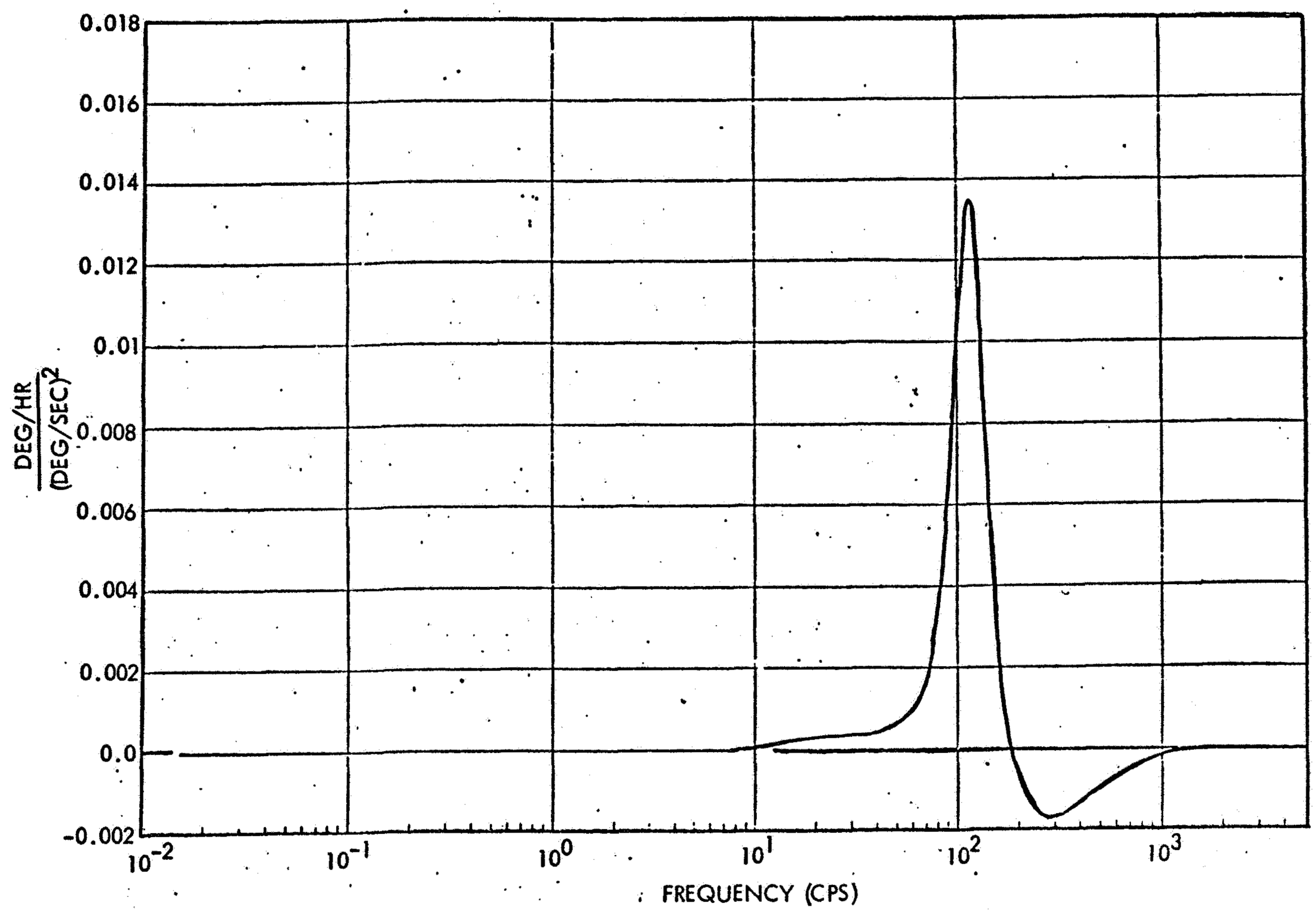


Figure 4-22. ϵ_6 Drift Coefficient

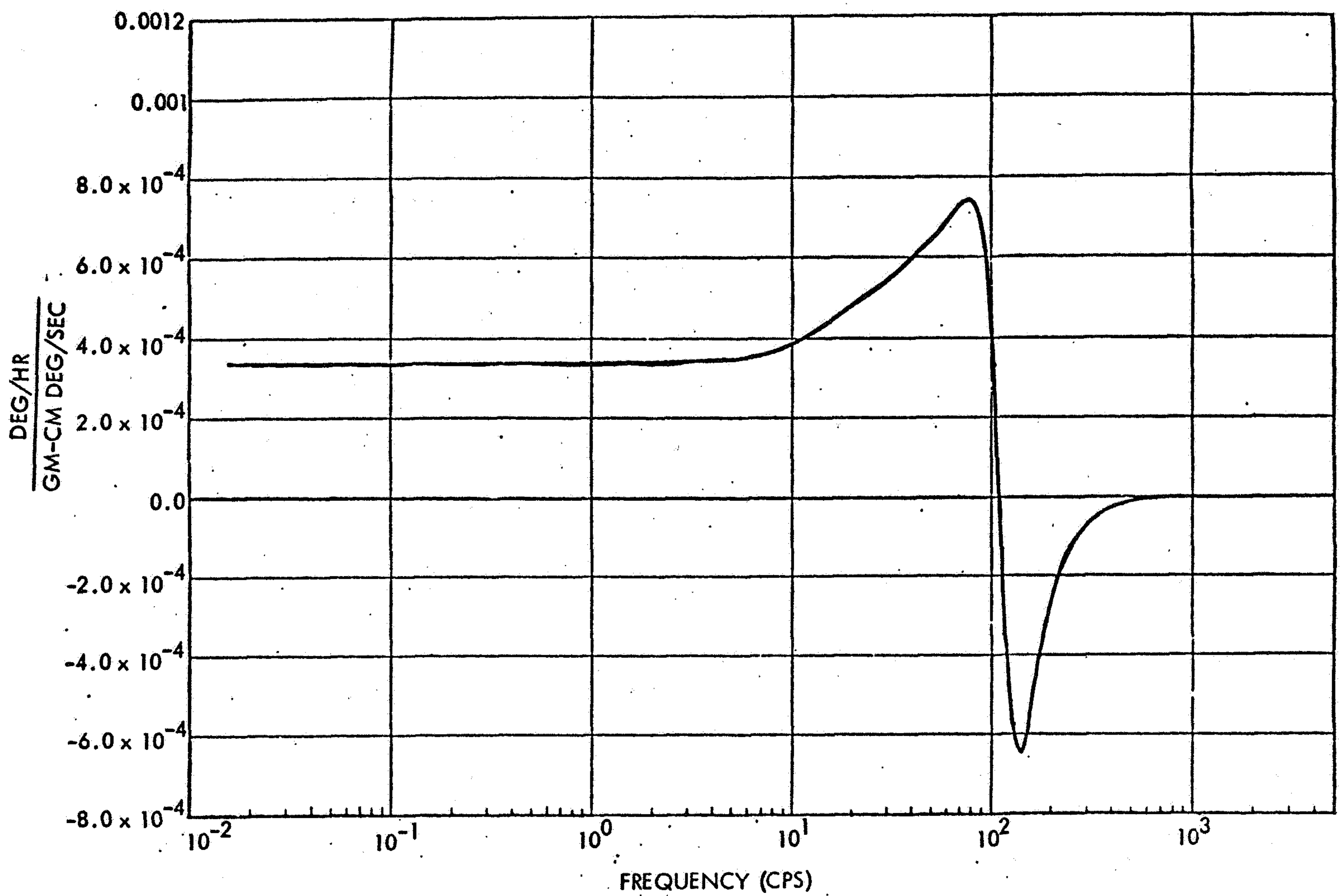


Figure 4-23. ϵ_7 Drift Coefficient

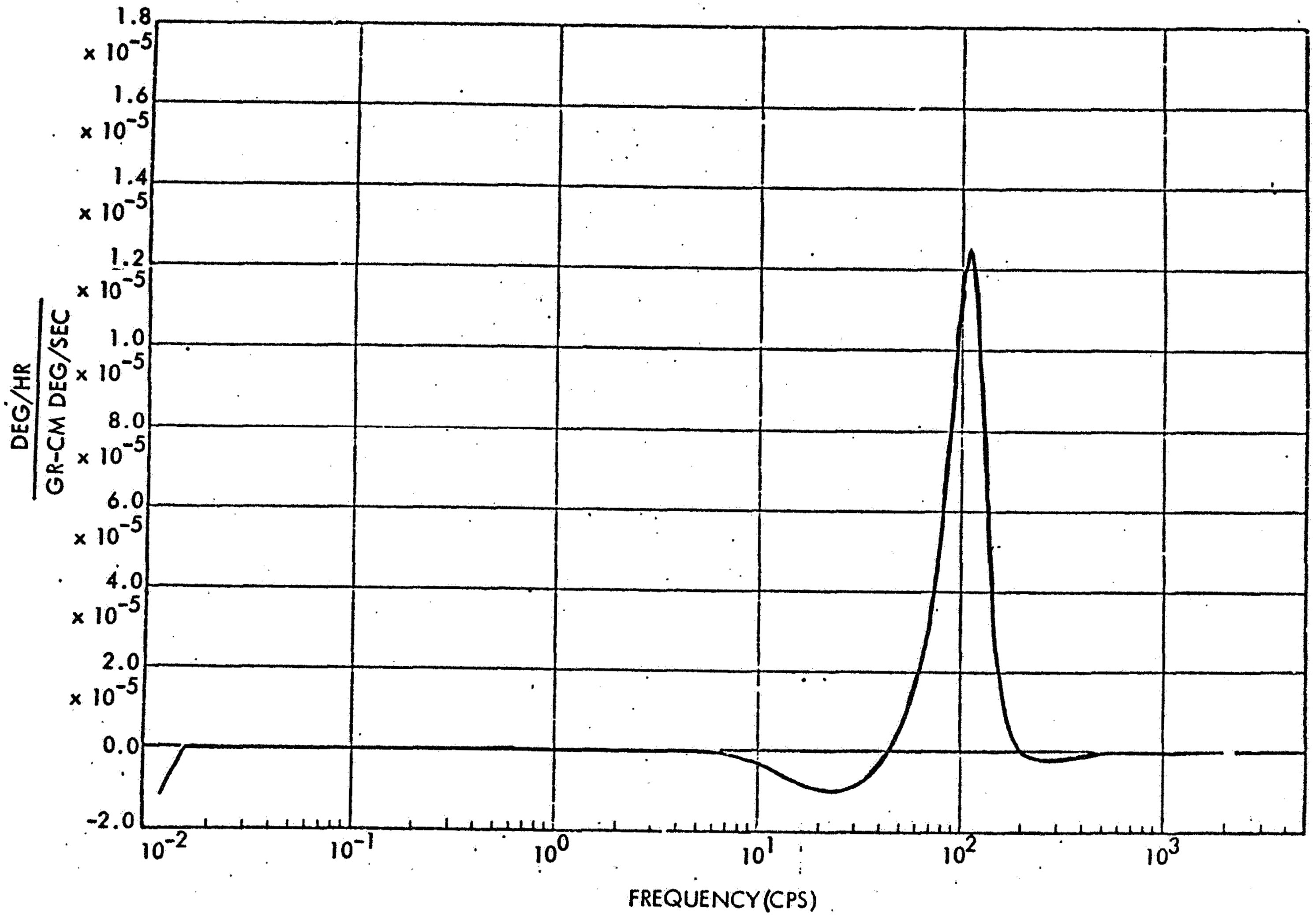


Figure 4-24. ϵ_8 Drift Coefficient

Figure 4-25 shows the SAP-MIDAS simulation results of the signal $\theta\omega_3$ for 15 cps inputs of ω_Y and ω_Z with 1 deg/sec amplitudes. It is easily ascertained geometrically that the average value between positive and negative peaks is 7.1×10^{-7} rad/sec or the same 0.146 deg/hr. Similarly, Figure 4-26 is indicative of a -0.0086 deg/hr error for the 50 cps inputs. This direct correlation obviously represents an exceptionally high degree of linearity.

4.4.7 PSD Inputs

The PSD inputs of vibration that were used in the Drift Rectification Program runs were all of flat unity magnitude. By this normalization, rectification errors for any magnitude of flat PSD inputs may be readily obtained by a simple scaling. However, the "accumulated random integration" performed by the program further allows these errors to be computed for any bandwidth (up to 2000 cps at present).

The accumulated random integration curve for ϵ_3 is shown in Figure 4-27. Note that if a 40 cps bandwidth is selected, the ϵ_3 rectification error for a 0.0005 $(\text{deg/sec})^2/\text{cps}$ PSD would be

$$0.0005 \times 3.1 \frac{\text{deg}}{\text{hr}}\Big|_{f=40 \text{ cps}} = 0.00155 \text{ deg/hr}$$

For an "extended environment" pertaining to a 300 cps bandwidth the ϵ_3 rectification error would be

$$0.0005 \times 8.2 \frac{\text{deg}}{\text{hr}}\Big|_{f=300 \text{ cps}} = 0.0041 \text{ deg/hr}$$

The accumulated random integration curve for each of the remaining rectification errors, ϵ_4 through ϵ_8, are shown in Figures 4-28 through 4-32 respectively. A summary of the random curves and their associated forcing functions is shown in Table 4-6.

If it is desired to obtain approximate rectification errors ϵ_3 through ϵ_8 for shaped PSD inputs these curves will also serve that purpose. To demonstrate the method assume that a PSD curve applicable to ϵ_3 appears as

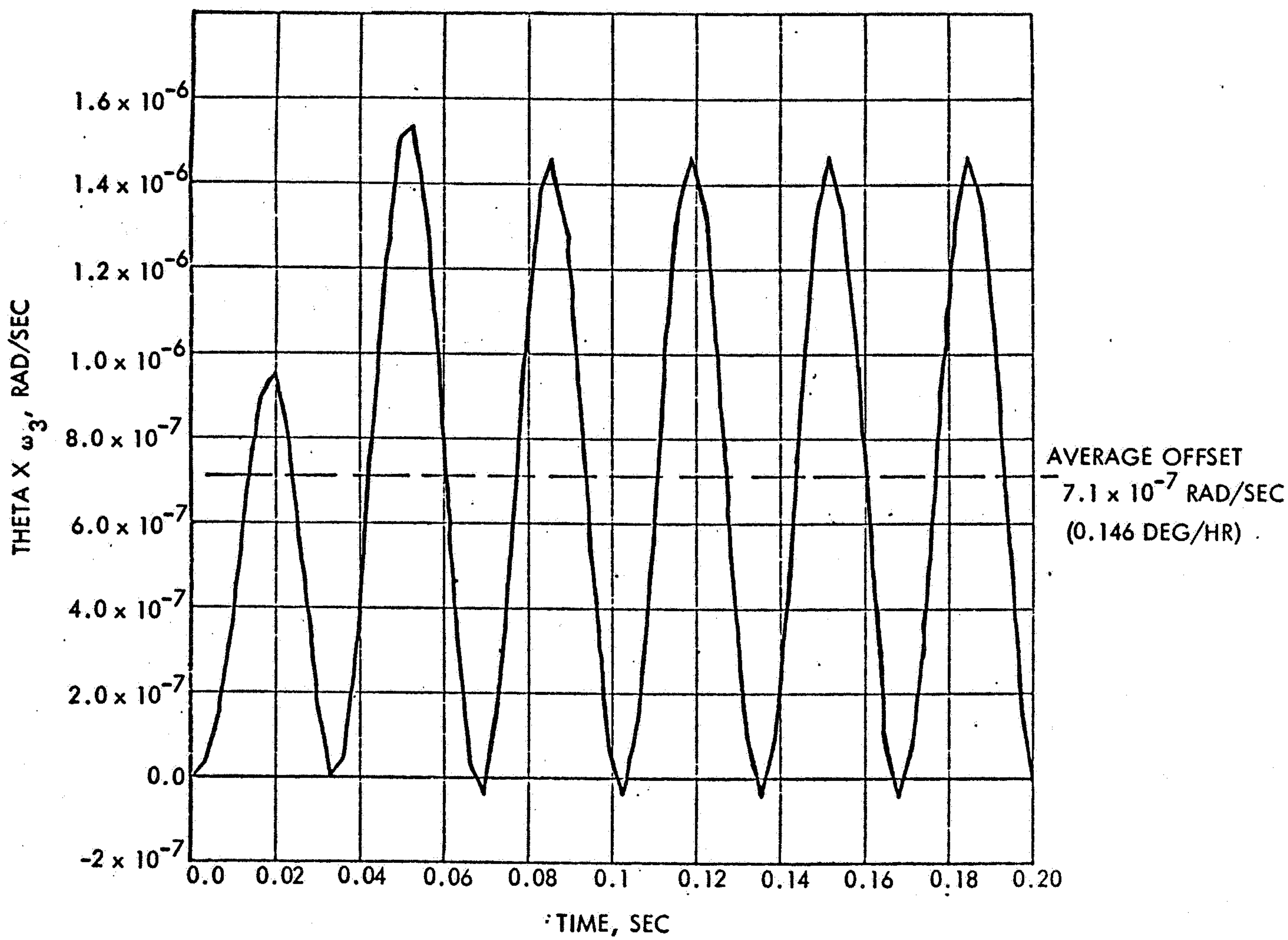


Figure 4-25. SAP $\theta\ \omega_3$, $\omega_Y = \omega_Z = 1$ Deg/Sec at 15 cps

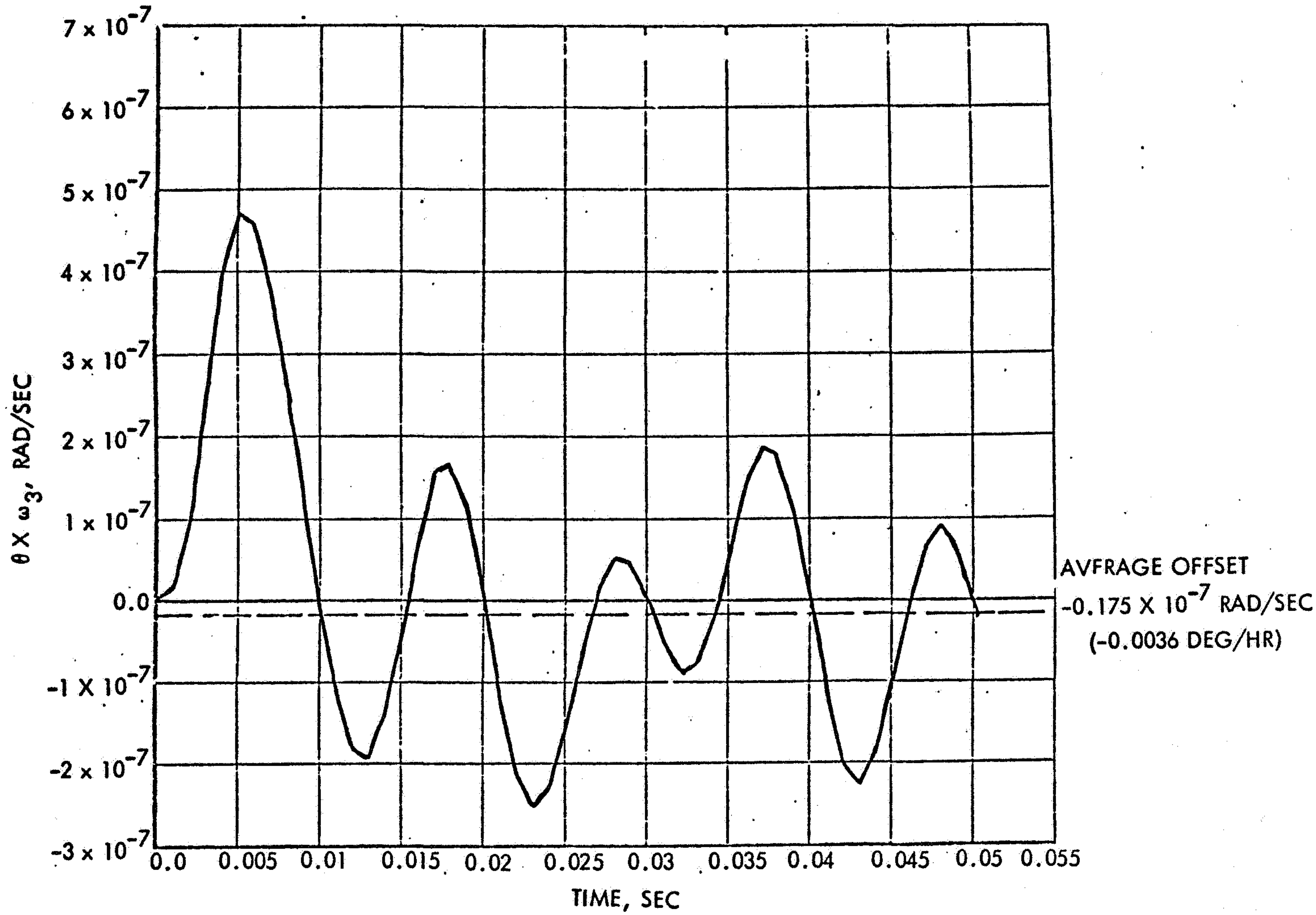


Figure 4-26. SAP $\theta\,\omega_3$ Signal $\omega_X = \omega_Y = 1$ Deg/Sec at 50 cps

Table 4-6. Summary of Rectification Error Work

Figure Number	Plotted Parameters	Random Forcing Function, PSD		
		$\Phi_{23} = \sqrt{\Phi_2}\sqrt{\Phi_3}$	$\Phi_{e3} = \sqrt{\Phi_e}\sqrt{\Phi_3}$	$\Phi_{pe3} = \sqrt{\Phi_{pe}}\sqrt{\Phi_3}$
4-27	$\Phi_{23} \int C_3 \, df$	X		
4-28	$\Phi_{e3} \int C_4 \, df$		X	
4-29	$\Phi_{pe3} \int C_5 \, df$			X
4-30	$\Phi_{23} \int C_6 \, df$	X		
4-31	$\Phi_{\epsilon 3} \int C_7 \, df$		X	
4-32	$\Phi_{pe3} \int C_8 \, df$			X

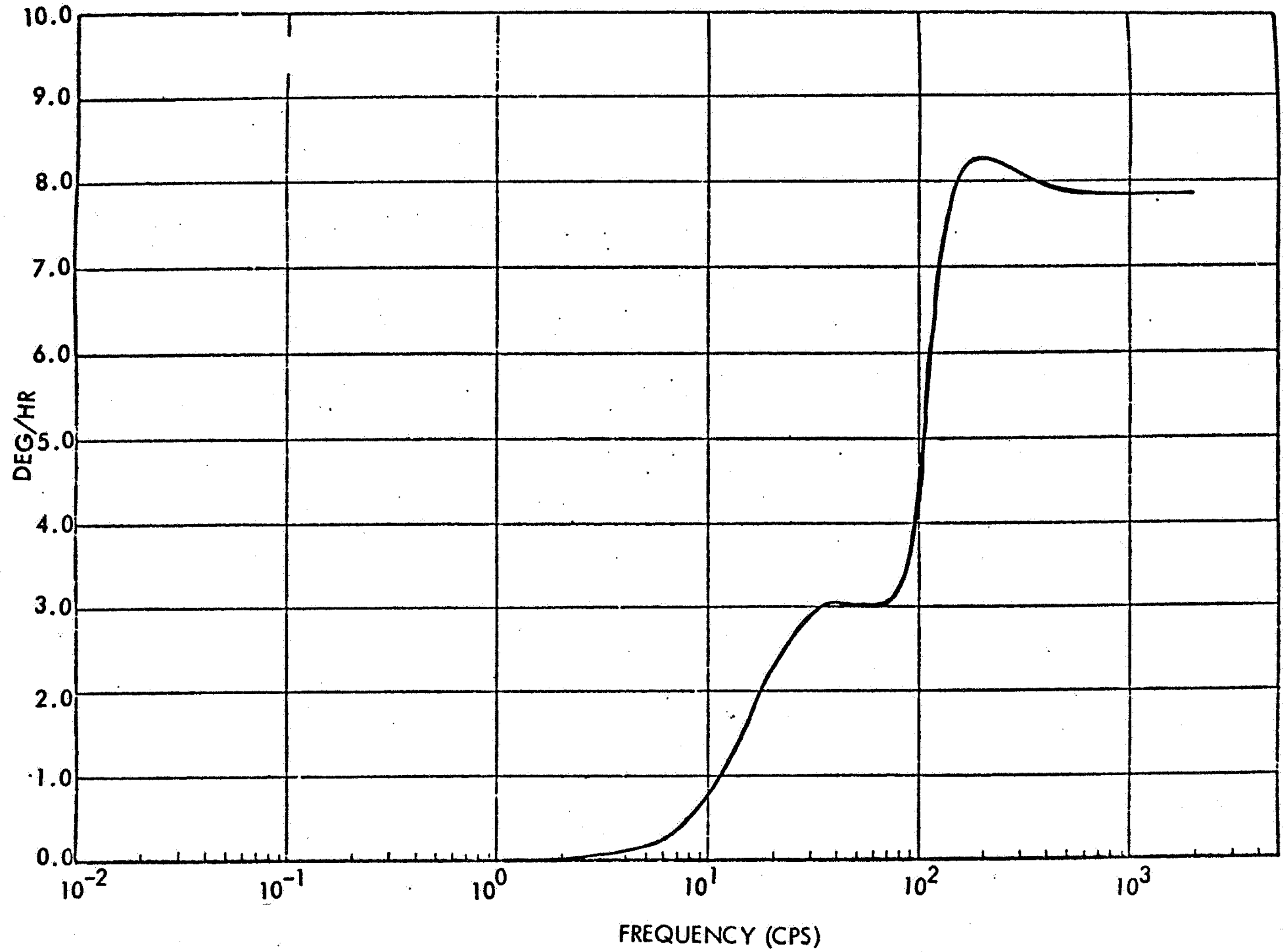


Figure 4-27. ϵ_3 Accumulated Random Integration Unity PSD Input

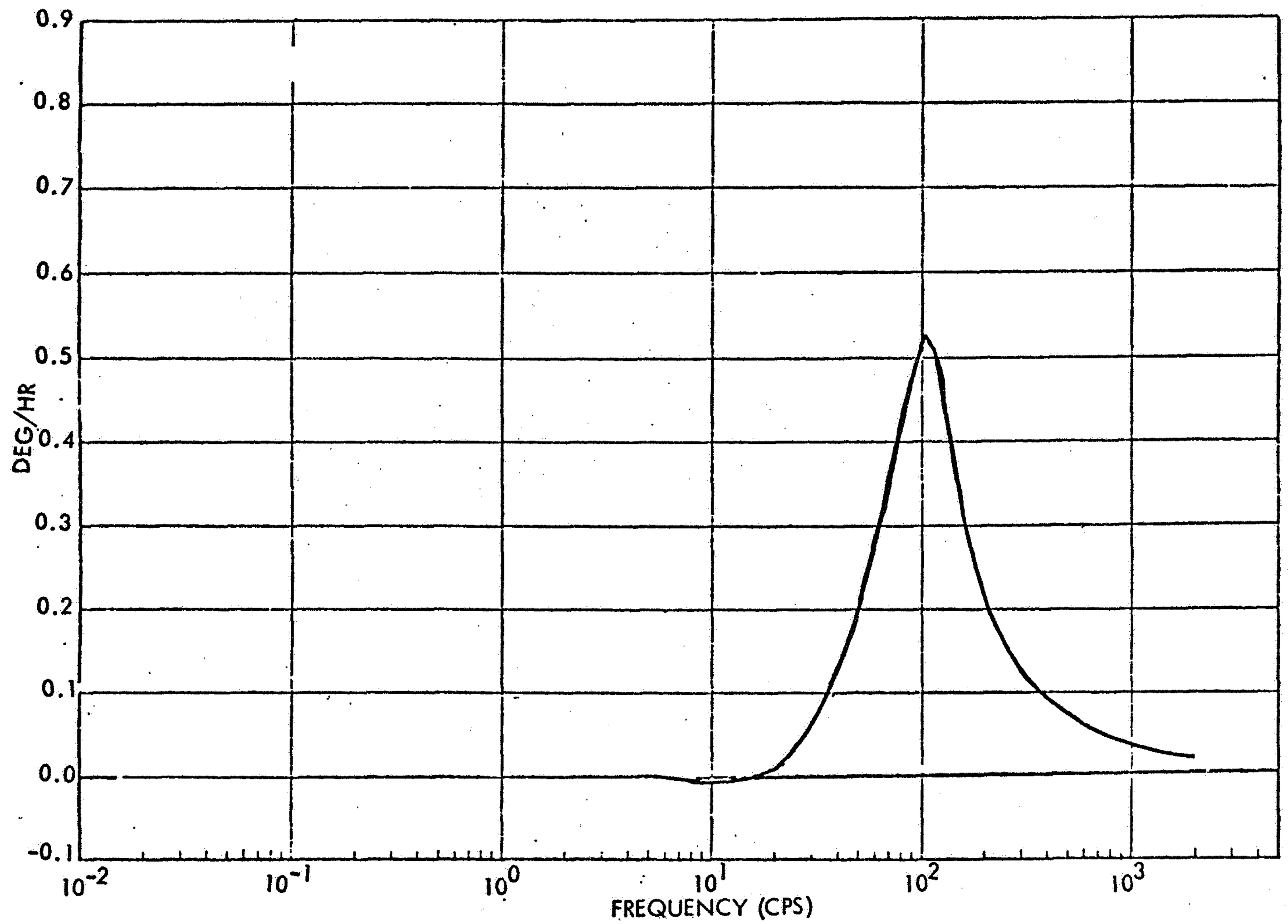


Figure 4-28. ϵ_4 Accumulated Random Integration Unity PSD Input

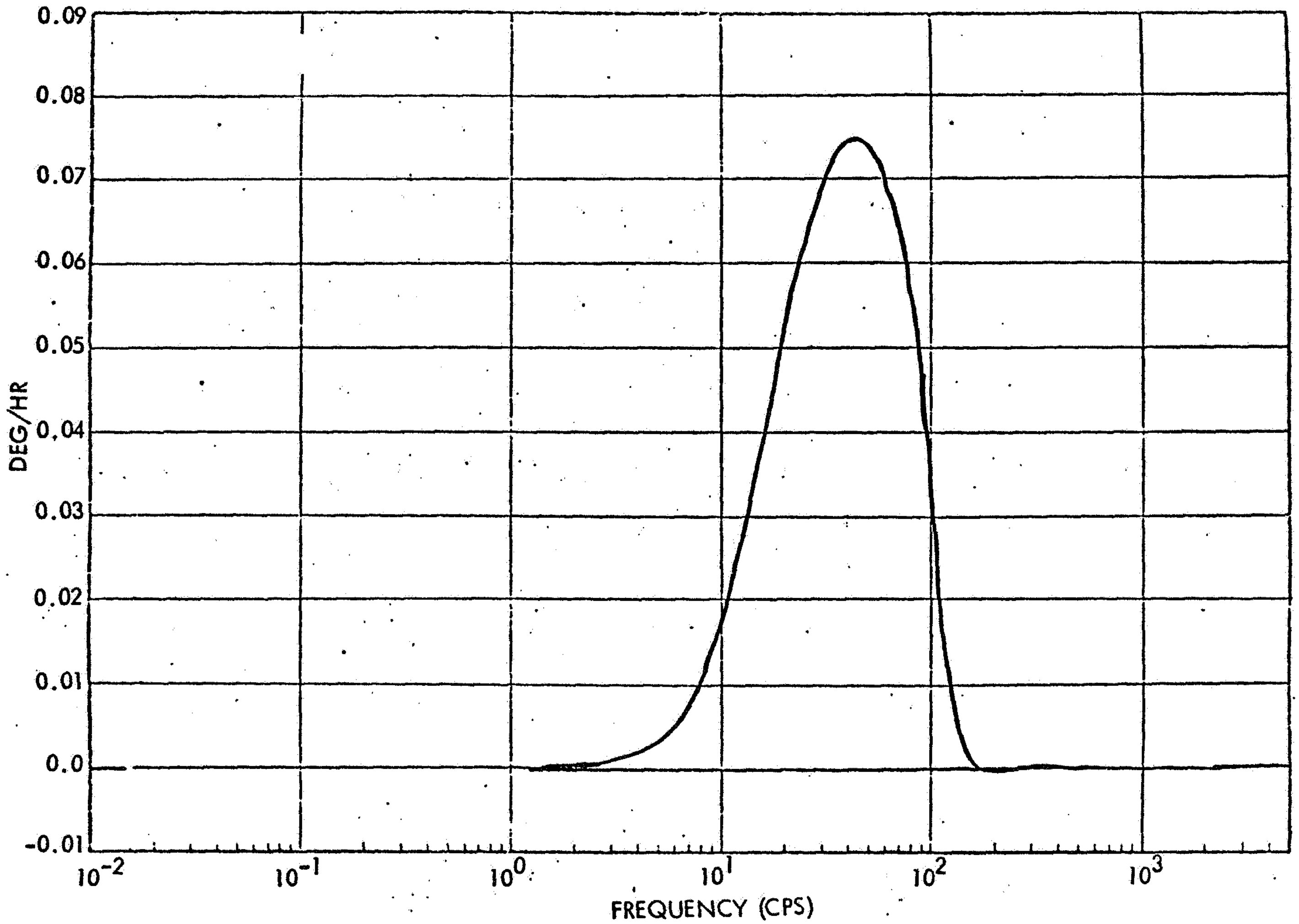


Figure 4-29. ϵ_5 Accumulated Random Integration Unity PSD Input

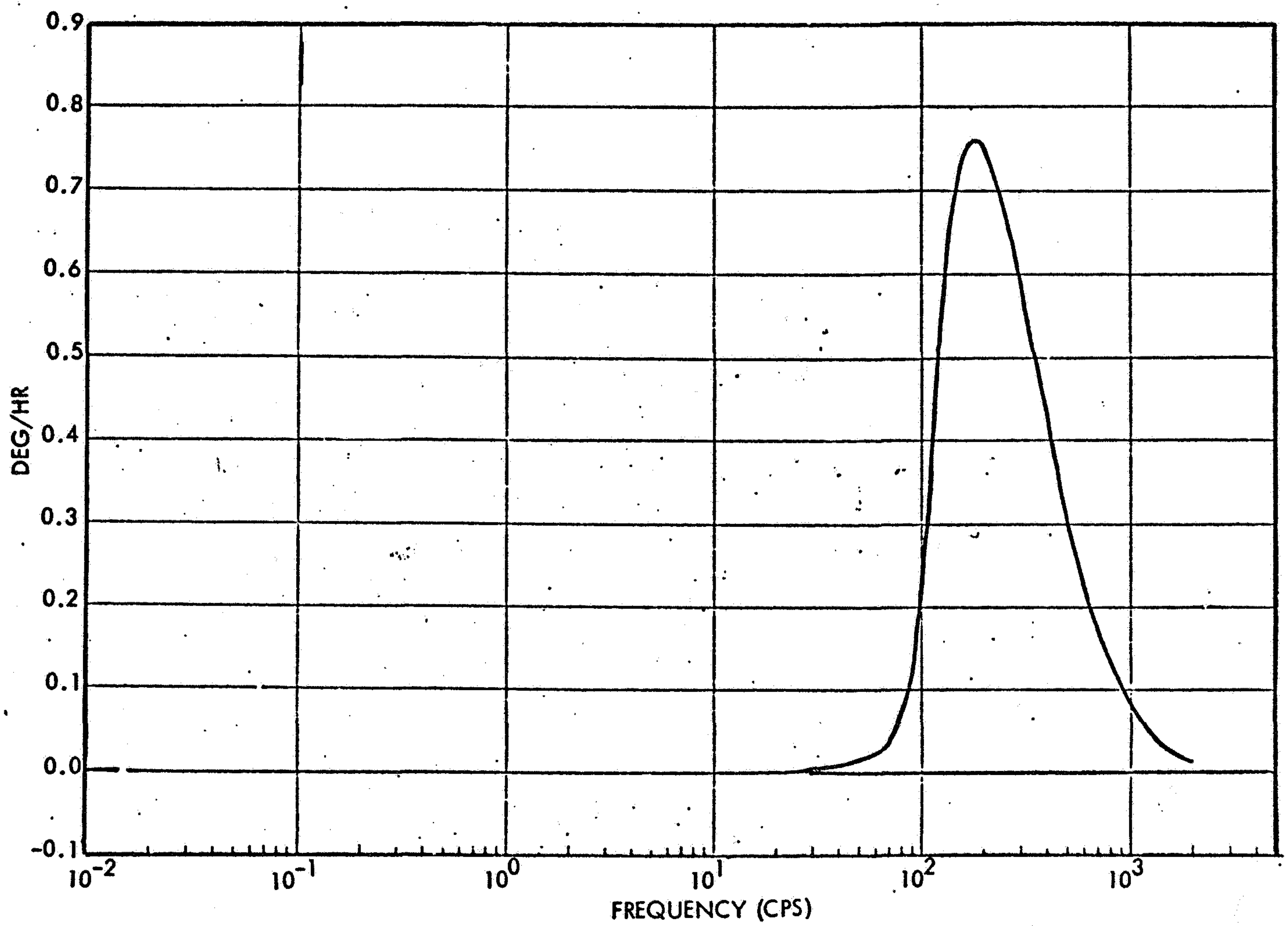


Figure 4-30. ϵ_6 Accumulated Random Integration Unity PSD Input

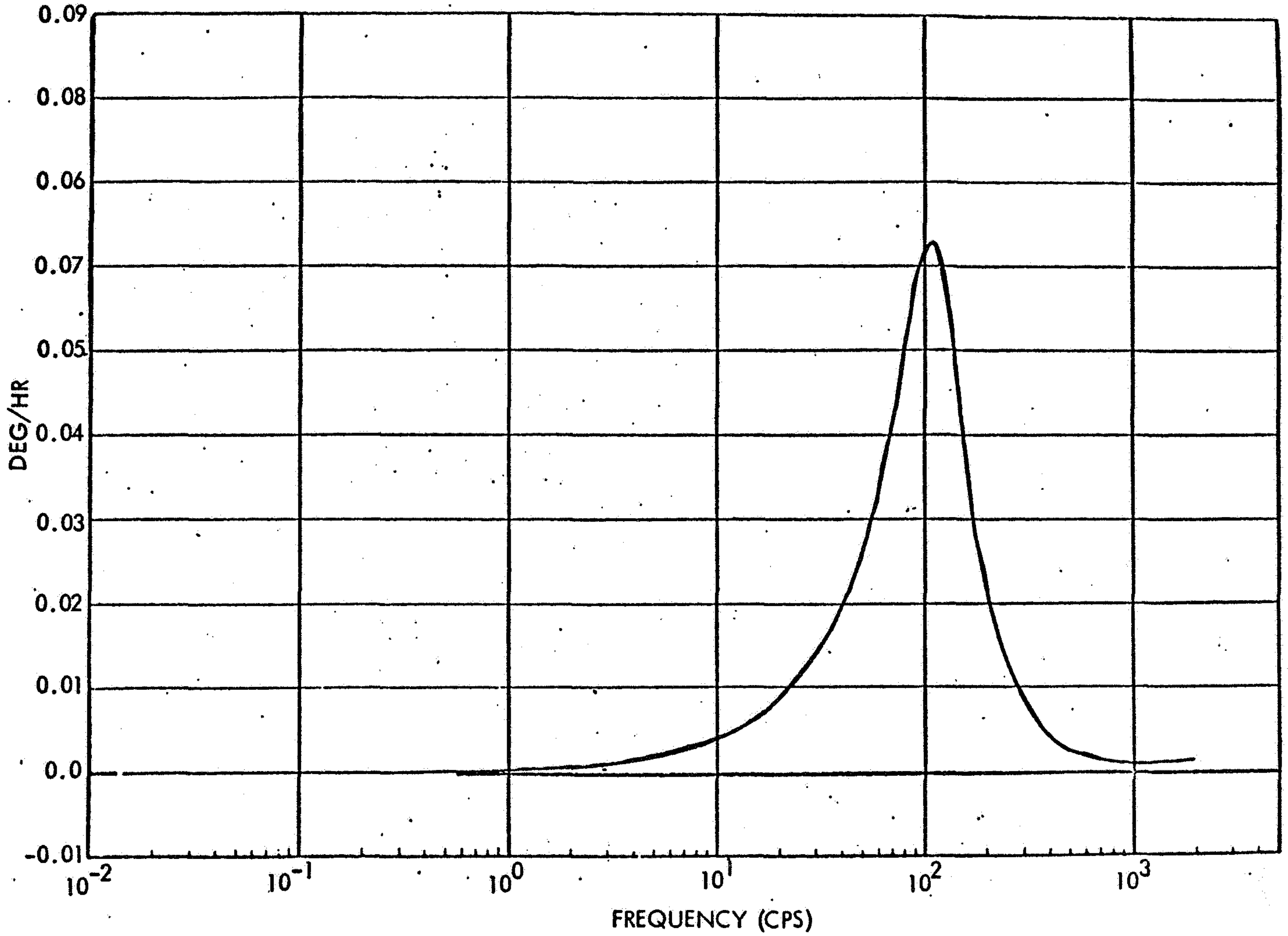


Figure 4-31. ϵ_7 Accumulated Random Integration Unity PSD Input

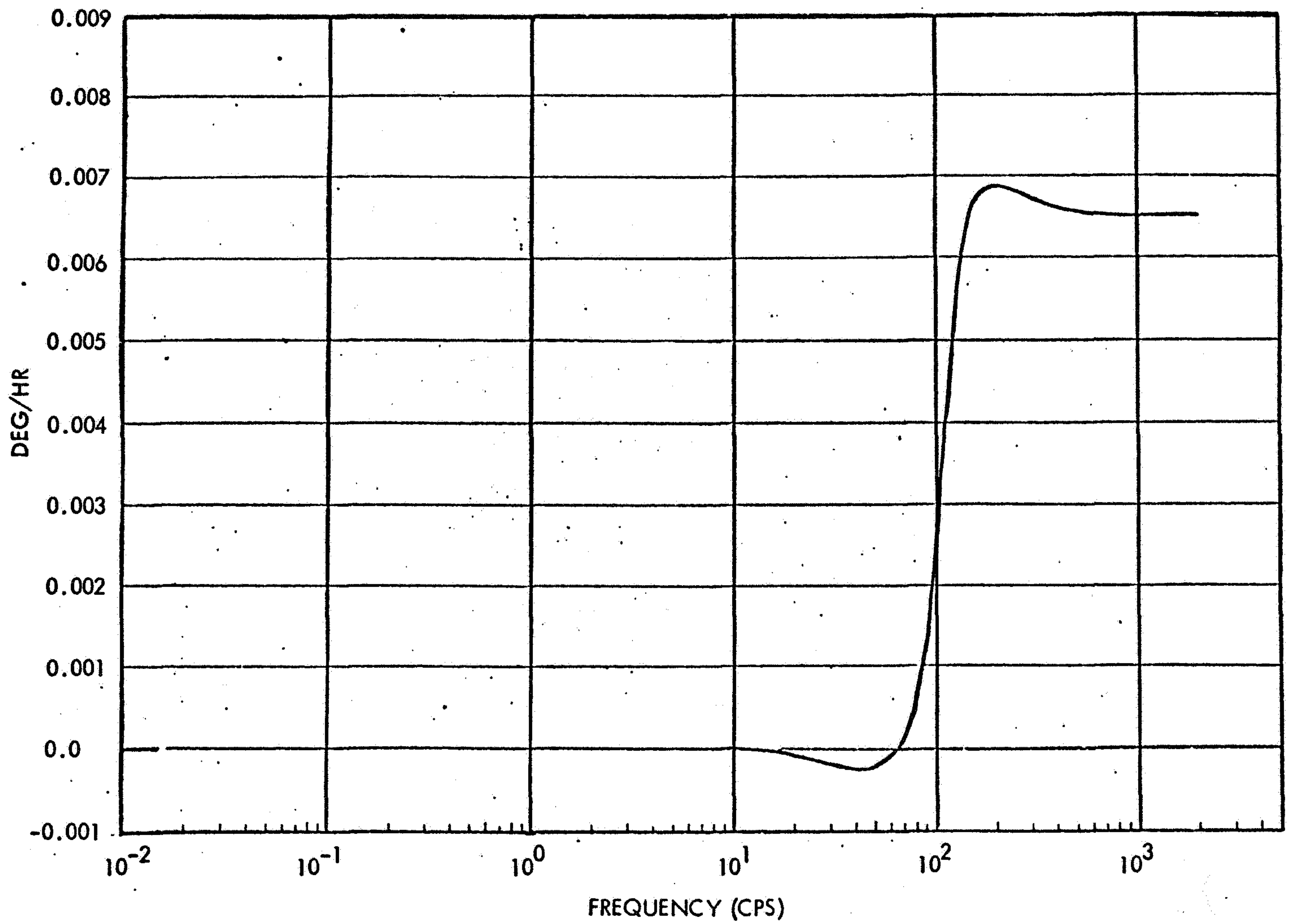
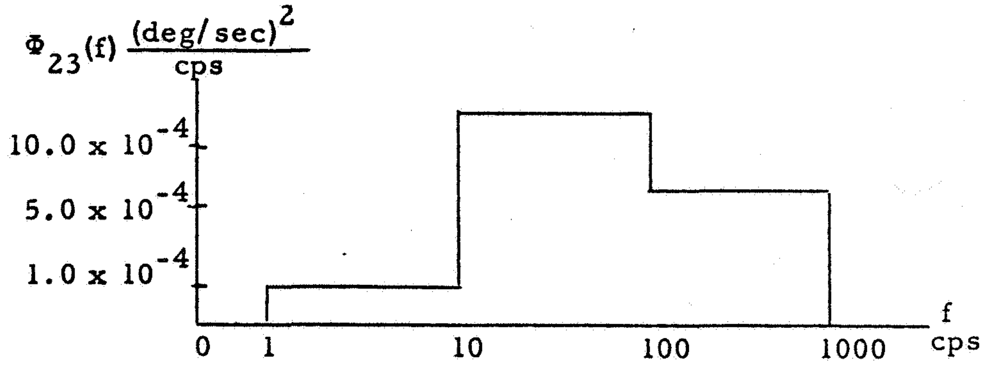


Figure 4-32. ϵ_8 Accumulated Random Integration Unity PSD Input



At the frequency, 1 cps the accumulated random integration to that point is 0.0 (from Figure 4-27). At 10 cps the normalized accumulated error is 0.75 deg/hr which corresponds to an error induced by the 1 to 10 cps bandwidth of

$$1.0 \times 10^{-4} \times (0.75-0.0) = 7.5 \times 10^{-5}\ \text{deg/hr.}$$

Similarly the error attributed to the 10 to 100 cps bandwidth is the PSD level times the difference of normalized error values,

$$10.0 \times 10^{-4} \times (4.25 - 0.75) = 3.5 \times 10^{-3}\ \text{deg/hr.}$$

Between the 100 to 1000 cps bandwidth the error is

$$5.0 \times 10^{-4} \times (7.85 - 4.25) = 1.8 \times 10^{-3}\ \text{deg/hr.}$$

The total error would therefore be the sum of the errors attributed to each bandwidth or 5.4×10^{-3} deg/hr.

Obviously the more narrow the bandwidths are made to approximate true PSD shaping, the more accurate the error evaluation will be. A good engineering approximation may be obtained by just such coarse bandwidths since the primary integration is reflected in the curves themselves.

4.5 TRANSIENT CHARACTERISTICS OF SAP LOOP

During the Phase I studies, a root locus determination was made of the MSFC-SAP configuration. The root loci were plotted in Reference 1 and that plot is repeated here in Figure 4-33. However, no attempts were made during Phase I to evaluate transient response or determine loop stability margins.

An inspection of Figure 4-33 reveals that a predominant closed loop pair of conjugate poles remains quite close to the imaginary axis. This position implies an oscillatory nature of the SAP operation. Such a mode is to be expected when employing a gas bearing gyro like the AB-5. The poles in question obviously propagate from the open loop pair of gyro poles.

As is indicated in Subsection 4.3, a series of step inputs were employed to check out the time-domain digital simulation program. These step inputs also serve to provide a series of transient responses. Three such responses are shown in Figures 4-34, 4-35, and 4-36. The first two figures represent SAP platform closed loop torquing required to overcome two types of input error torques. Figure 4-34 represents the electrically applied torque to error torques acting on the SAP gyro-output axis. Figure 4-35 represents torque resistance to error torques acting about the SAP input axis. Figure 4-36 portrays float angle response to error torques about the SAP gyro-output axis.

Each of these plots serves to verify the critical location of the conjugate pair of closed loop poles mentioned above. In the first two figures the torque compensation was induced to travel through essentially three cycles of motion before damping out. The float angle motion is shown to travel through one complete cycle with a high transient peak. Although each of these plots reflects the lightly damped transients associated with the above mentioned pair of conjugate poles, it should be noted that for the assumed 1-gm-cm disturbance torque the maximum float angle that occurs is 3.3×10^{-6} rad (less than 1 arc sec). Such low level motion would have essentially no effect on system operation due to the apparent oscillatory mode of operation.

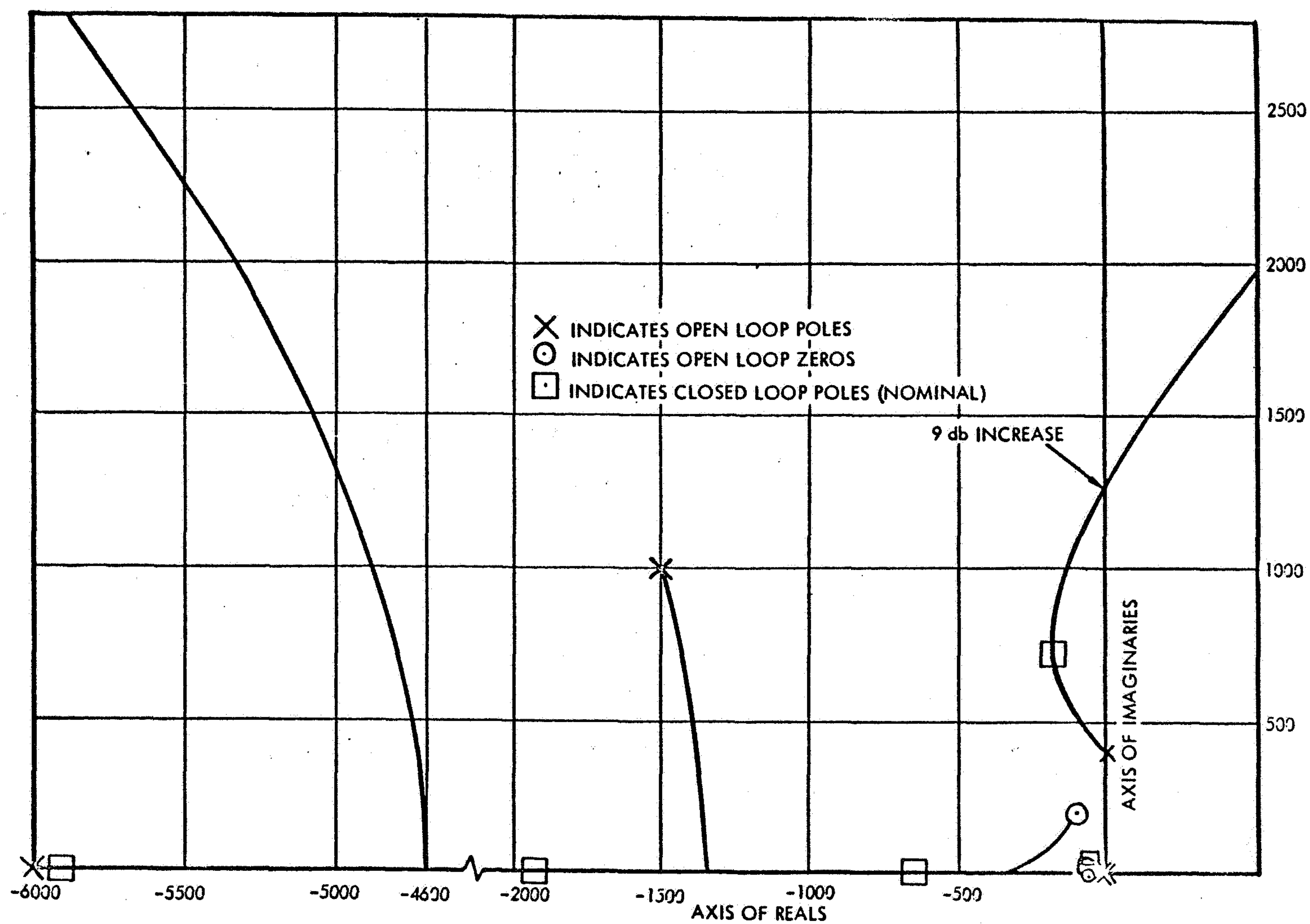


Figure 4-33. SAP-Root Locus Traces (Upper-half Plane Only)

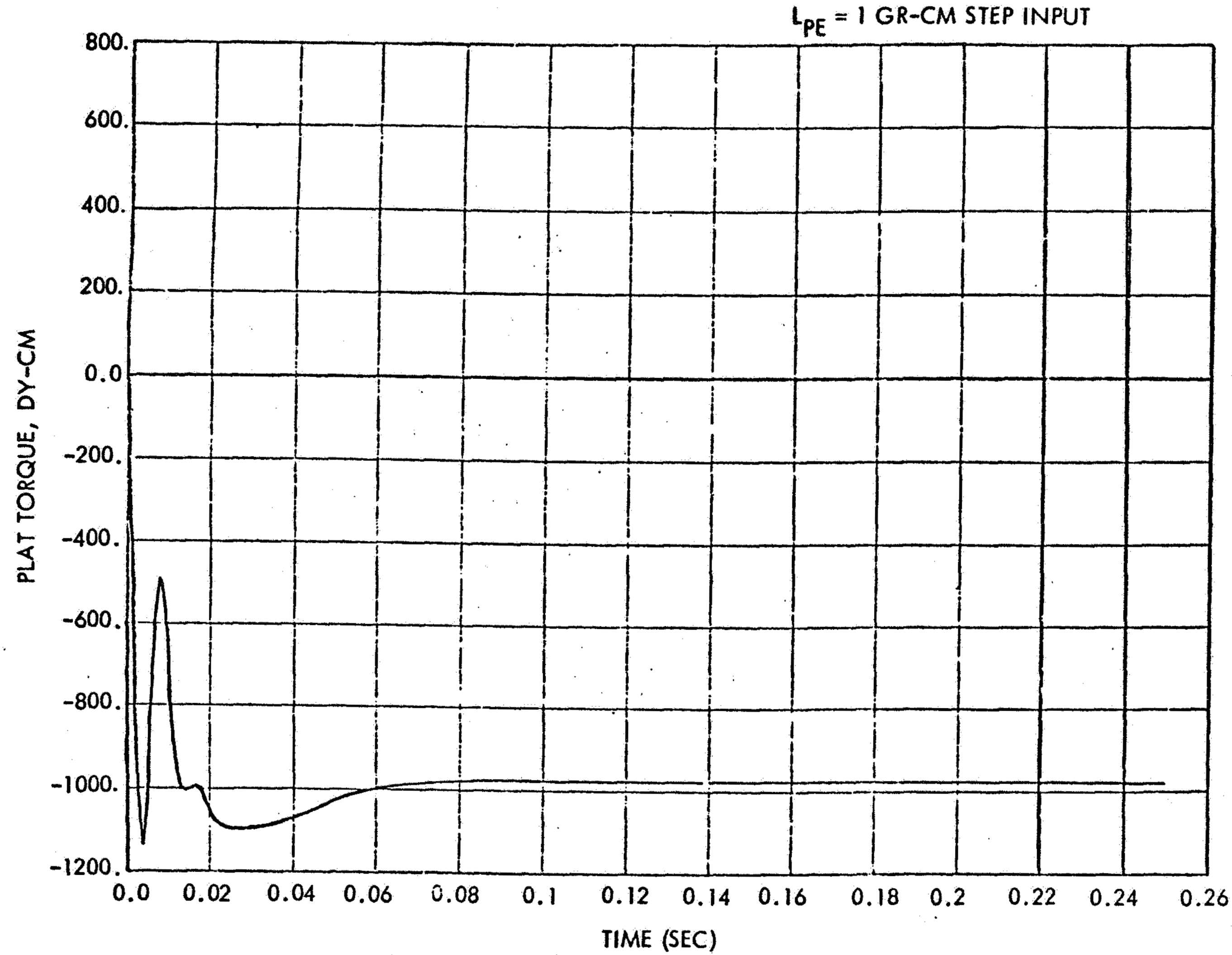


Figure 4-34. SAP Plat Torque, L_{pa}

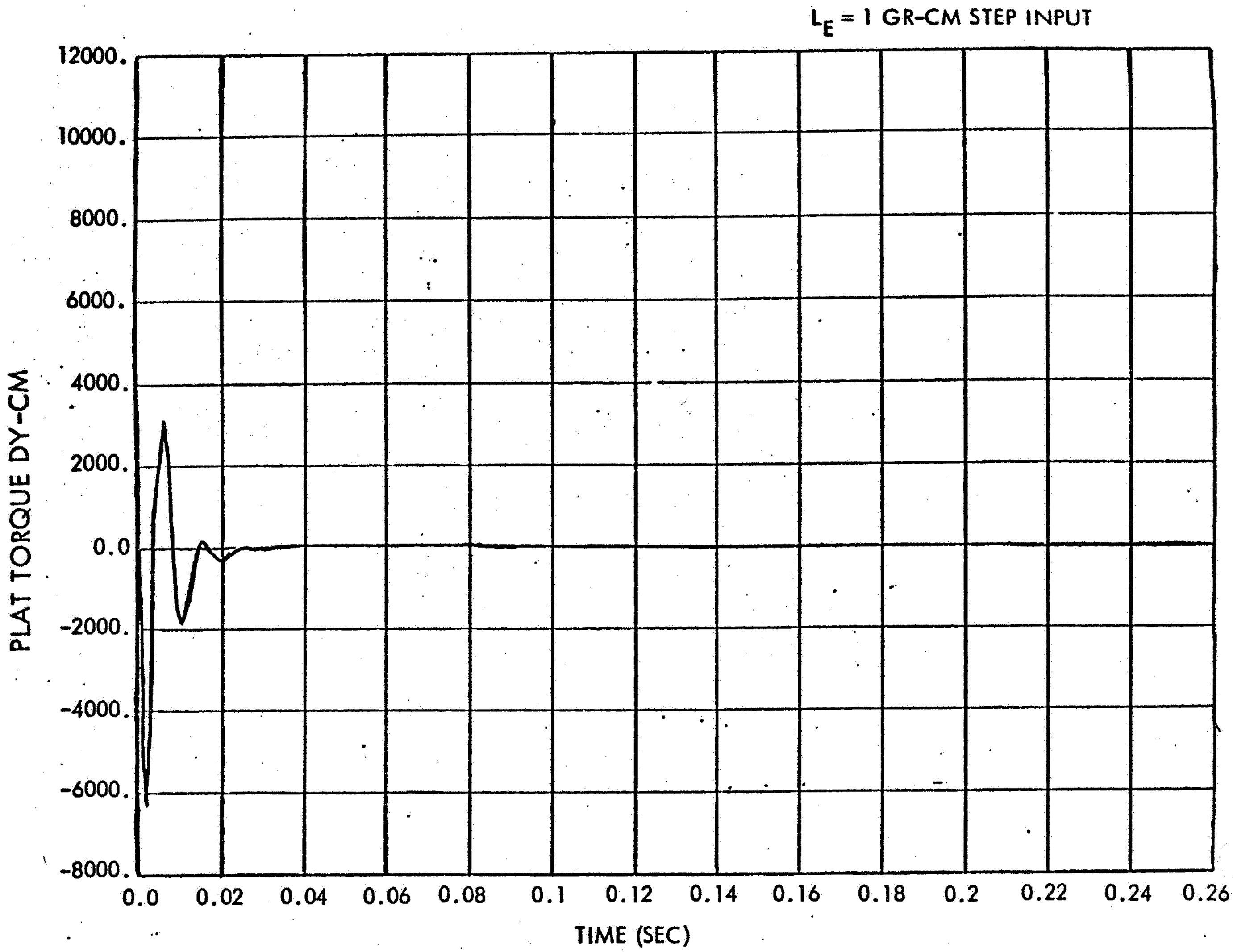


Figure 4-35. SAP Plat Torque, L_{pa}

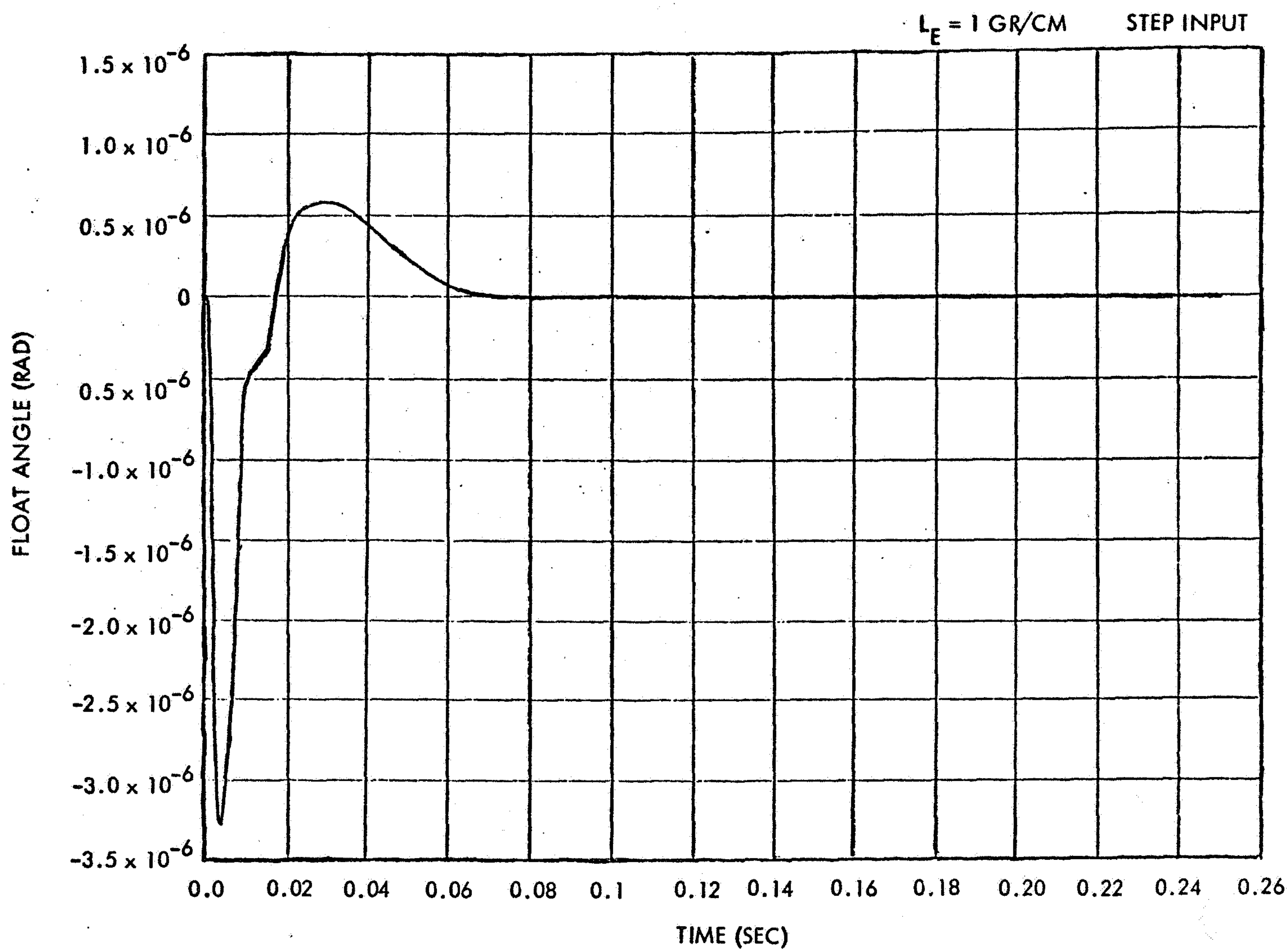


Figure 4-36. SAP Float Angle, Rad

The time domain program did not include such possible nonlinearities as float stops, amplifier saturation, or torque hysteresis. Considering the marginal location of the closed loop poles, a more extensive evaluation of the SAP should be made which would include the above mentioned nonlinearities. These conditions can be readily adopted by the MIDAS simulation program, once realistic values to represent them are established.

4.6 FICTITIOUS CONING

The phenomenon commonly referred to as coning is closely related to the basic technology of a strapdown guidance system. Therefore, a study was conducted of coning with regard to the MSFC-SAP. The aspect of coning that pertains specifically to SAP transfer of angular rate information is termed fictitious coning. Fictitious coning is, in essence, an error in the measurement of vibrational angular rotation of a vehicle about some fixed axis that is not coincident with either of the three SAP gyro-input axes. These imperfect measurements generally result from a combination of two error inducements:

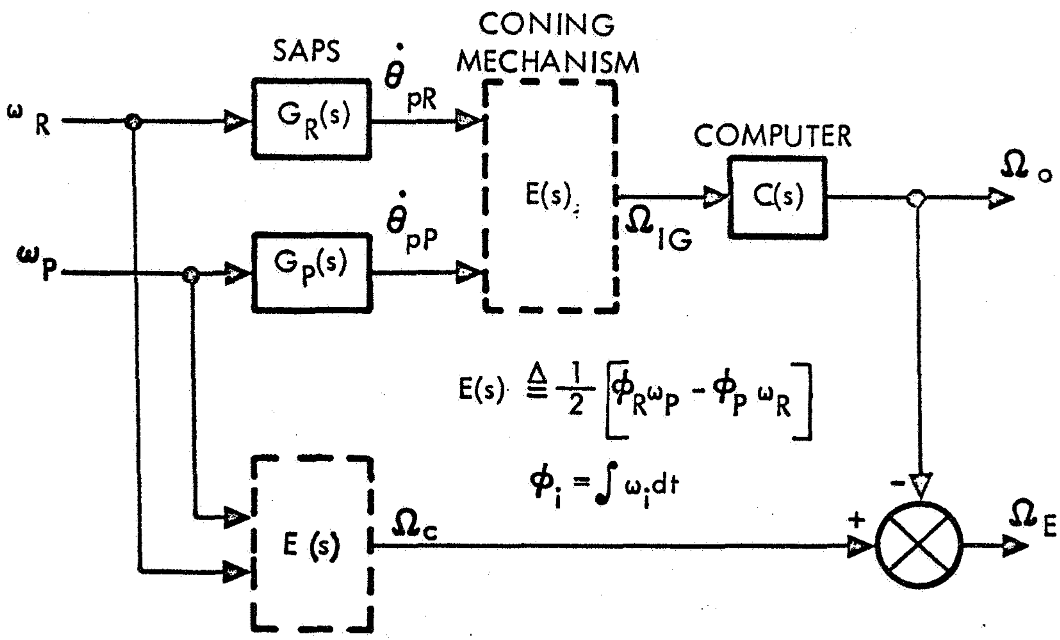
- Computer algorithm and sampling
- Gain and phase differences between SAP loops

The coning error induced by computer algorithms was studied in detail during Phase I (see Section 5 of Reference 1). The studies were based on perfect SAP loops with unity transfer functions.

If any two SAP transfer functions are not identical their outputs will also differ for identical inputs. This output difference would, in turn, be interpreted by a perfect computer as a measure of drift rate about the axis normal to the two axes in question for certain forms of excitation.

To pictorially demonstrate this occurrence, see Figure 4-37. The basic mechanism which induces a body to physically cone has been shown by Goodman and Robinson (Reference 5) to be oscillatory angular rates along two orthogonal axes acting 90 degrees out of phase with each other. Mathematically expressed, this true coning motion that occurs about a vehicle yaw axis (for example) for excitations about the roll and pitch axes is

$$\Omega_c(t) = 1/2 \left[\phi_R \omega_P - \phi_P \omega_R \right], \qquad (4\text{-}27)$$



Ω_c → TRUE VEHICLE CONING RATE

Ω_o → CONING RATE INDICATED BY COMPUTER

Ω_{IG} → CONING RATE INDICATED BY GYROS

Ω_E → ERROR IN SIRA CONING RATE INDICATION

$$\dot{\theta}_p - \omega_X = \frac{I_y FS\,\omega_2 + I_y S^2 L_{PE} - (H_r S + F)L_E}{J_x I_y S^3 + H_r{}^2 S + H_r F}$$

$$\therefore G_R(s) = G_P(s) = \frac{\dot{\theta}p}{\omega_X}(s) = 1.0$$

Figure 4-37. Fictitious Coning Mechanism

where

$$\phi_i = \int \omega_i \, dt \quad i = P(\text{pitch}),\ R(\text{roll})$$

ω_i = angular rate oscillations

If such motion was processed by perfect sensors and computer no yaw-axis fictitious coning error would be indicated. This is the lower path shown in Figure 4-37. However, any nonperfection of this processing through the upper path of Figure 4-37 yields a different indication of coning motion than the true motion. The difference between the readings of the two paths is the fictitious coning error.

There is an alternate way of interpreting fictitious coning. If the two input angular excitations are exactly in phase with each other, then it can be shown that no true coning motion of the vehicle exists. Without presenting mathematical verification, it is obvious that some greater degree of true coning will exist for excitations that are phased somewhere between zero and 90 degrees apart. It follows, therefore, that even when the excitations are precisely in phase, but the SAP loops that process these motions have phase differences, that an ideal computer would interpret these SAP signals to represent some degree of coning motion. This greater degree is again considered to be fictitious coning. In terms of frequency response to sinusoidal angular vibrations these relationships may be expressed as

$$\Omega_E(j\omega) = -\omega_{Ro}\omega_{Po} \left[C_{TC}(j\omega) \sin\psi + C_{FC}(j\omega) \cos\psi \right] \qquad (4\text{-}28a)$$

where

ω_{io} = amplitudes of angular vibrations about roll and pitch axes

ψ = phase angle between ω_R and ω_P,

$C_{TC}(j\omega)$ = true coning coefficient

$$= \frac{1 - C(j\omega)\left|G_R(j\omega)\right|\left|G_P(j\omega)\right| \cos\left[\angle G_R(j\omega) - \angle G_P(j\omega)\right]}{2\omega}, \qquad (4\text{-}28b)$$

$C_{FC}(j\omega)$ = fictitious coning coefficient

$$= \frac{C(j\omega)\left|G_R(j\omega)\right|\left|G_P(j\omega)\right| \sin\left[\angle G_R(j\omega) - \angle G_P(j\omega)\right]}{2\omega}, \qquad (4\text{-}28c)$$

$G_i(j\omega)$ and $C(j\omega)$ are SAP and computer transfer functions as defined by Figure 4-37.

The remaining work in this section will be based on an ideal computer where $C(j\omega) = 1.0$ and will only investigate the fictitious coning coefficient $C_{FC}(j\omega)$.

4.6.1 First Order Effects

In the case of the SAP it can be readily established that, in a first-order sense, no gain or phase differences exist between SAP loops. Therefore, no fictitious coning exists. An inspection of Equation (7-51) from Reference 1 will serve to verify this statement. That equation relating SAP output to four inputs, is presented here for convenience:

$$\dot{\theta}_P(s) = \omega_X(s) - \frac{I_y sF(s)\omega_2(s) + I_y s^2 L_{pE}(s) - \left[H_r s + F(s)\right]\left[L_E(s) + L_a(s)\right]}{J_x I_y s^3 + H_r^2 s + H_r F(s)} \qquad (4\text{-}29)$$

Note that if the forcing functions, ω_2, L_{pE}, L_E and L_a are independent of the input axis rate ω_X, then the transfer function between θ_p and ω_X is

$$\frac{\dot{\theta}_p}{\omega_X}(s) = 1.0 \qquad (4\text{-}30)$$

Since no explicit dynamical properties exist between input rate and output rate measurement then no explicit gain or phase differences exist between SAP loops. The $G_i(s)$ functions of Figure 4-37 do satisfy the idealistic conditions of unity SAP loops. Furthermore, no fictitious coning error contributions are attributed to the SAPs themselves.

This is not to conclude, however, that sensor-induced fictitious coning cannot exist in the SIRA. The existence of second-order error effects may well cause subtle differences between SAP loops. Such causes might be friction or torque motor back emf that act about a SAP input axis.

This phenomenon would appear in L_{pE} as a dynamical function of either the angular rate component acting along the SAP input axis or the platform rate, $\dot{\theta}_p$.

4.6.2 Torque Motor Induced Fictitious Coning

To demonstrate the approach necessary to determine fictitious coning errors, a case is assumed where back emf acts on the input axes of two different SAPs. In this event, the input axis error term, $L_{pE}(s)$ in Equation (4-29) assumes the form

$$L_{pE}(s) = D_E \dot{\theta}_p(s) \tag{4-31}$$

where

D_E = damping coefficient associated with back emf.

Neglecting excitations acting along the other two SAP axes, Equation (4-29) is rewritten as

$$\dot{\theta}_p(s) = \omega_X(s) - \frac{I_y s^2 D_E \dot{\theta}_p(s)}{J_x I_y s^3 + H_r^2 s + H_r F(s)} \tag{4-32a}$$

or, after rearranging

$$\left.\frac{\dot{\theta}_p(s)}{\omega_X}\right|_{D_E} = \frac{J_x I_y s^3 + H_r^2 s + H_r F(s)}{J_x I_y s^3 + I_y D_E s^2 + H_r^2 s + H_r F(s)} \tag{4-32b}$$

In order to evaluate Equation (4-32b) it is convenient to define a new closed loop transfer function. This step permits use of the work performed in Reference 1 and saves considerable computer time. The new function is

$$G_{CL_D}(s) = \frac{D_E \dfrac{\dot{\theta}_P(s)}{L_{pE}}}{1 + D_E \dfrac{\dot{\theta}_p(s)}{L_{pE}}} \tag{4-33}$$

where

$\frac{\dot{\theta}_P(s)}{L_{pE}}$ is the transfer function used to determine ϵ_8.

By simple algebraic manipulation it can be shown that the following equality (from Equations (4-32b) and (4-33) holds true:

$$\frac{\dot{\theta}_P(s)}{\omega_X} = G_{CL_D}(s) \cdot \frac{1}{D_E \frac{\dot{\theta}_P(s)}{L_{pE}}} . \qquad (4\text{-}34)$$

The numerators of G_{CL_D} and $D_E(\dot{\theta}_P/L_{pE})$ are identical. Therefore, the expression for $\dot{\theta}_P/\omega_X$ reduces to the ratio of the characteristic equations for $\dot{\theta}_P/L_{pE}$ and G_{CLD} (see Equation 4-32b). The characteristic equation for $\dot{\theta}_P/L_{pE}$ was determined during Phase 1 to be nominally that of

$$\left.\frac{\dot{\theta}_P(s)}{L_{pE}}\right|_{nom}$$

$$= \frac{1.235 \times 10^{-9}\, s^2 \left(\frac{s^2}{1.0} + 1\right)\left(\frac{s^2}{1500 \pm j1000} + 1\right)\left(\frac{s}{6000} + 1\right)\left(\frac{deg/sec}{gm\text{-}cm}\right)}{\left(\frac{s^2}{7.158 \pm j40.18} + 1\right)\left(\frac{s^2}{178.2 \pm j715.9} + 1\right)\left(\frac{s}{654.6} + 1\right)\left(\frac{s}{1933.5} + 1\right)\left(\frac{s}{5913.3} + 1\right)} \qquad (4\text{-}35)$$

The back emf damping coefficient is arbitrarily selected from standard catalogs of torque motors having essentially the same properties as that indicated for the MSFC-SAP. The nominal value used is

$$D_E = 1.356 \times 10^4 \frac{dy\text{-}cm}{rad/sec} .$$

The open loop gain variation that is used is ±11 percent from the nominal K_F/H_r. Such variation changed the location of the poles for $\dot{\theta}_P/L_{pE}(s)$ as follows:

$$
D_E \left. \frac{\dot{\theta}_P(s)}{L_{pE}} \right|_{+11 \text{ percent}}
$$

$$
= \frac{2.67 \times 10^{-10}\, s^2 \left(\frac{s}{1.0} + 1\right)\left(\frac{s^2}{1500 \pm j1000} + 1\right)\left(\frac{s}{6000} + 1\right)}{\left(\frac{s^2}{8.01 \pm j35.2} + 1\right)\left(\frac{s^2}{184.4 \pm j781.1} + 1\right)\left(\frac{s}{515.3} + 1\right)\left(\frac{s}{2054.5} + 1\right)\left(\frac{s}{5902.2} + 1\right)} \tag{4-36}
$$

and

$$
D_E \left. \frac{\dot{\theta}_P(s)}{L_{pE}} \right|_{-11 \text{ percent}}
$$

$$
= \frac{3.33 \times 10^{-10}\, s^2 \left(\frac{s}{1.0} + 1\right)\left(\frac{s^2}{1500 \pm j1000} + 1\right)\left(\frac{s}{6000} + 1\right)}{\left(\frac{s^2}{6.45 \pm 42.9} + 1\right)\left(\frac{s}{161.9 \pm j660.0} + 1\right)\left(\frac{s}{830.6} + 1\right)\left(\frac{s}{1794.3} + 1\right)\left(\frac{s}{5923.2} + 1\right)}
$$

Each of these equations is, in turn, used in conjunction with the TRW-Root Locus Utility Program to determine a $G_{CLD}(s)$ from Equation (4-33). The transfer functions for two channels of $\dot{\theta}_p/\omega_X(s)$ one with +11 percent open loop gain and the other with -11 percent gain, are then obtained by employing Equation (4-32b). These are the transfer functions that are to be used by the Rectification Program and they are shown in Figure 4-38. This figure is a reproduction of the input sheet to the Rectification Program.

The resulting drift coefficient C_{FC} (see Equation (4-52)) and accumulated random integration that is attributed to this fictitious coning condition are shown in Figures 4-39 and 4-40 respectively. Again, it is significant that the drift coefficient is a response that is normalized to rotational vibratory amplitudes that represent 1.0 deg/sec on each channel. Furthermore, the random curve has also been normalized to a $1.0(\text{deg/sec})^2/\text{cps}$ flat PSD environment.

```
NPOL(1)=7,NZER(1)=7,GKB(1)=1.0,PSI=0.0,KO=62.8,
KRAND=1,NPOL(2)=8,NZER(2)=7,GKB(2)=1.0E6,PHI=163*1.0,
PLS(1,1)=(-80.121914,35.194478),(-80.121914,-35.194478),
(-184.56316,781.24097),(-184.56316,-781.24097),
(-514.75811,0.0),(-2054.6969,0.0),(-5902.1927,0.0),
ZRS(1,1)=(-80.1,35.2),(-80.1,-35.2),(-184.2,781.1),
(-184.2,-781.1),(-515.3,0.0),(-2154.5,0.0),(-5902.2,0.0),
PLS(1,2)=(-64.521531,42.903023),(-64.521531,-42.903023),
(-162.30594,660.03873),(-162.30594,-660.03873),
(-829.89133,0.0),(-1794.6012,0.0),(-5923.1943,0.0),(-1.0E-6,0.0),
ZRS(1,2)=(-64.51,42.9),(-64.51,-42.9),(-161.9,660.0),
(-161.9,-660.0),(-830.60,0.0),(-1794.3,0.0),
(-5923.2,0.0),
$.REDEF HEAD=H
HEAD=MSFC FICT CONING  PSI=0
$END
```

Poles and zeros for gyro loop with +11% gain

Poles and zeros for gyro loop with -11% gain

Figure 4-38. Input MSFC SAP Fictitious Coning ± 11 P.C. Nom D.E.

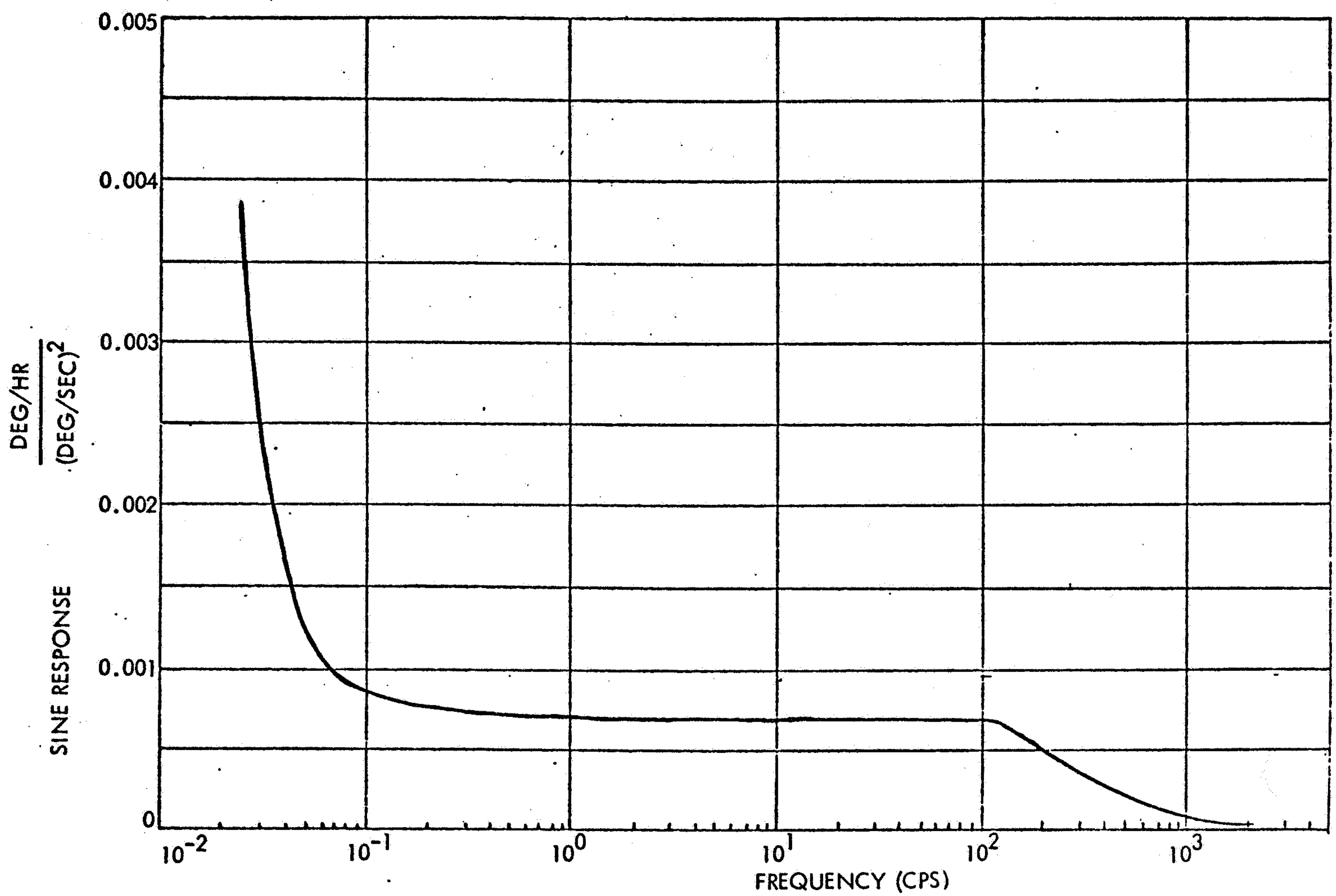


Figure 4-39. Frequency Coefficient for MSFC Fictitious Coning $\emptyset = 0 \pm 11$ Percent Gain

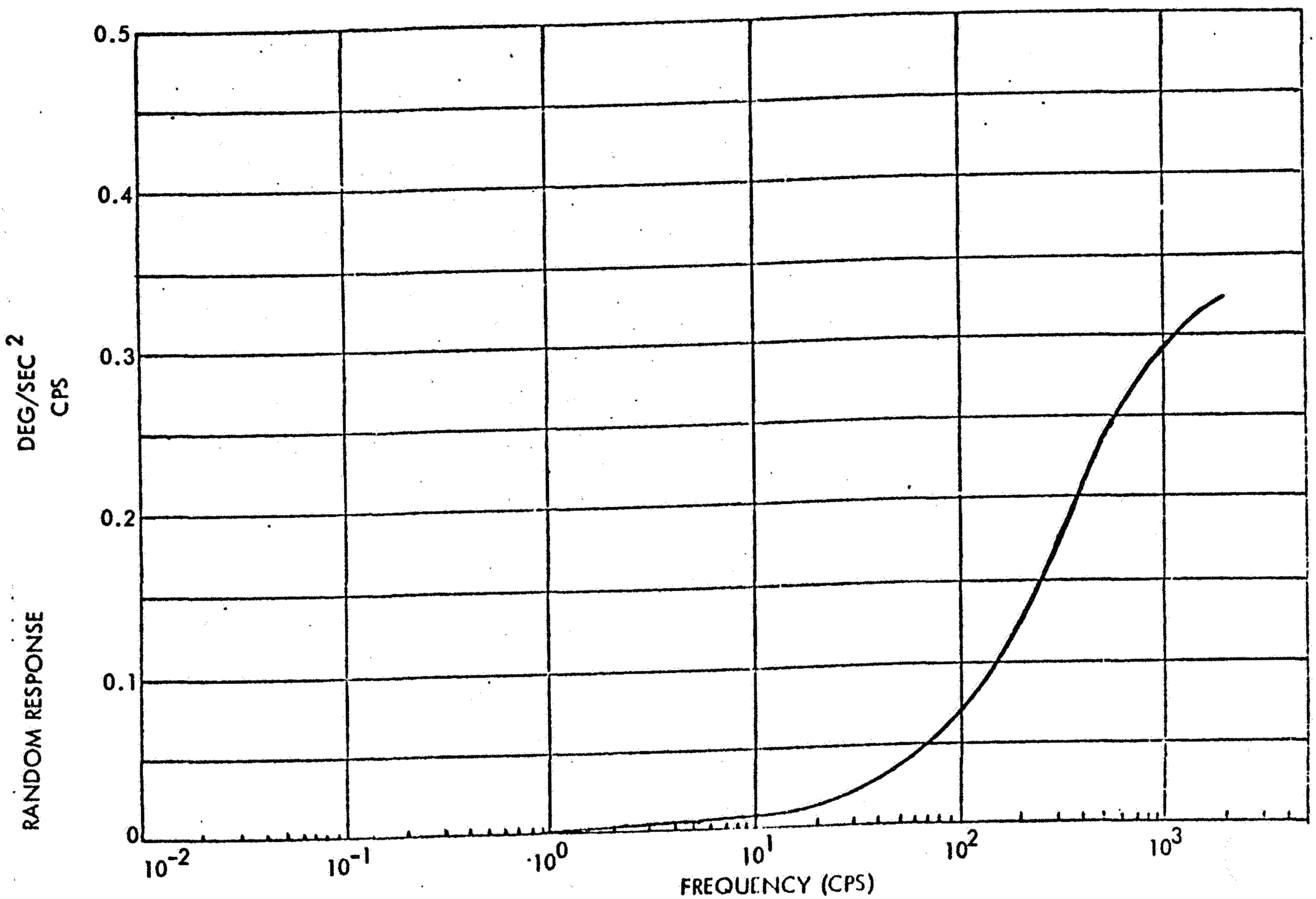


Figure 4-40. Random Integrated Error for MSFC Fictitious Coning Ø ± 11 Percent Gain

An inspection of these two curves establishes that the magnitudes of fictitious coning errors are negligible. From Figure 4-39 the maximum error in drift rate would be less than 0.004 deg/hr. From Figure 4-40 the worst-case random-induced error is calculated to be 1.63×10^{-4} deg/hr (0.35×0.0005).

The investigation of fictitious coning errors was also extended one step further. Instead of maintaining the nominal value of the damping coefficient, D_E it was also permuted ±11 percent. The combination of open loop gain variation with that of the damping variation was selected so as to provide an apparent worst-case occurrence. Such a condition occurs when gain increase is coupled with damping increase and vice-versa. The relocated pole positions for this case are shown in Figure 4-41. Note, by comparing Figure 4-38 to Figure 4-41, that this ±11 percent damping variation has insignificant effect on the pole relocation. The resulting plots (Figures 4-42 and 4-43) of drift coefficient and accumulated random integration further bear out this negligible change. These plots are nearly identical to those of Figures 4-39 and 4-40.

4.6.3 Output Axis Fictitious Coning

An entirely different type of fictitious coning than that described by the two previous cases can exist in the SIRA. This type is induced by angular vibrations that occur about the output reference axes of two different SAPs, i.e., $\dot{\omega}_2$. This phenomenon can be readily recognized by inspection of Equation (4-29). Although the transfer function between platform rate $\dot{\theta}_p$ and input rate ω_X is unity, there exists definite loop dynamics between $\dot{\theta}_p$ and the output axis rate ω_2. This appears in transfer function form as

$$\dot{\theta}_p(s) = \frac{I_y sF(s)}{J_x I_y s^3 + H_r^2 s + H_r F(s)} \cdot \omega_2(s) \tag{4-38}$$

An ideal computer would receive SAP pickoff signals without any knowledge of the cause. That is, coning motion of Equation (4-27) would only be interpreted by the computer to be of the form

$$\Omega_{IG}(t) = \frac{1}{2}\left[\theta_{p_R}\dot{\theta}_{p_P} - \theta_{p_P}\dot{\theta}_{p_R}\right] \tag{4-39}$$

```
$INPUT MSFC SAP F.C. +-11 P.C. DE
PSI=0.0,KRAND=1,
PLS(1,1)=(-80.124590,35.193802),(-80.124590,-35.193802),
(-184.60748,781.25819),(-184.60748,-781.25819),(-514.69199,0.0),
(-2054.6969,0.0),(-5902.1927,0.0),
PLS(1,2)=(-64.520276,42.902694),(-64.520276,-42.902694),
(-162.26178,660.03451),(-162.26178,-660.03451),
(-829.96841,0.0),(-1794.6012,0.0),(-5923.1943,0.0),(-1.0E-6,0.0)
$.REDEF HEAD=H
HEAD=MSFC SAP F.C. +-11 P.C. DE
$END
```

Relocated poles for gyro loop with +11% D_E increase and +11% gain increase

Relocated poles for gyro loop with 11% D_E decrease and 11% gain decrease

Figure 4-41. Input MSFC SAP Fictitious Coning ±11 P.C. DE

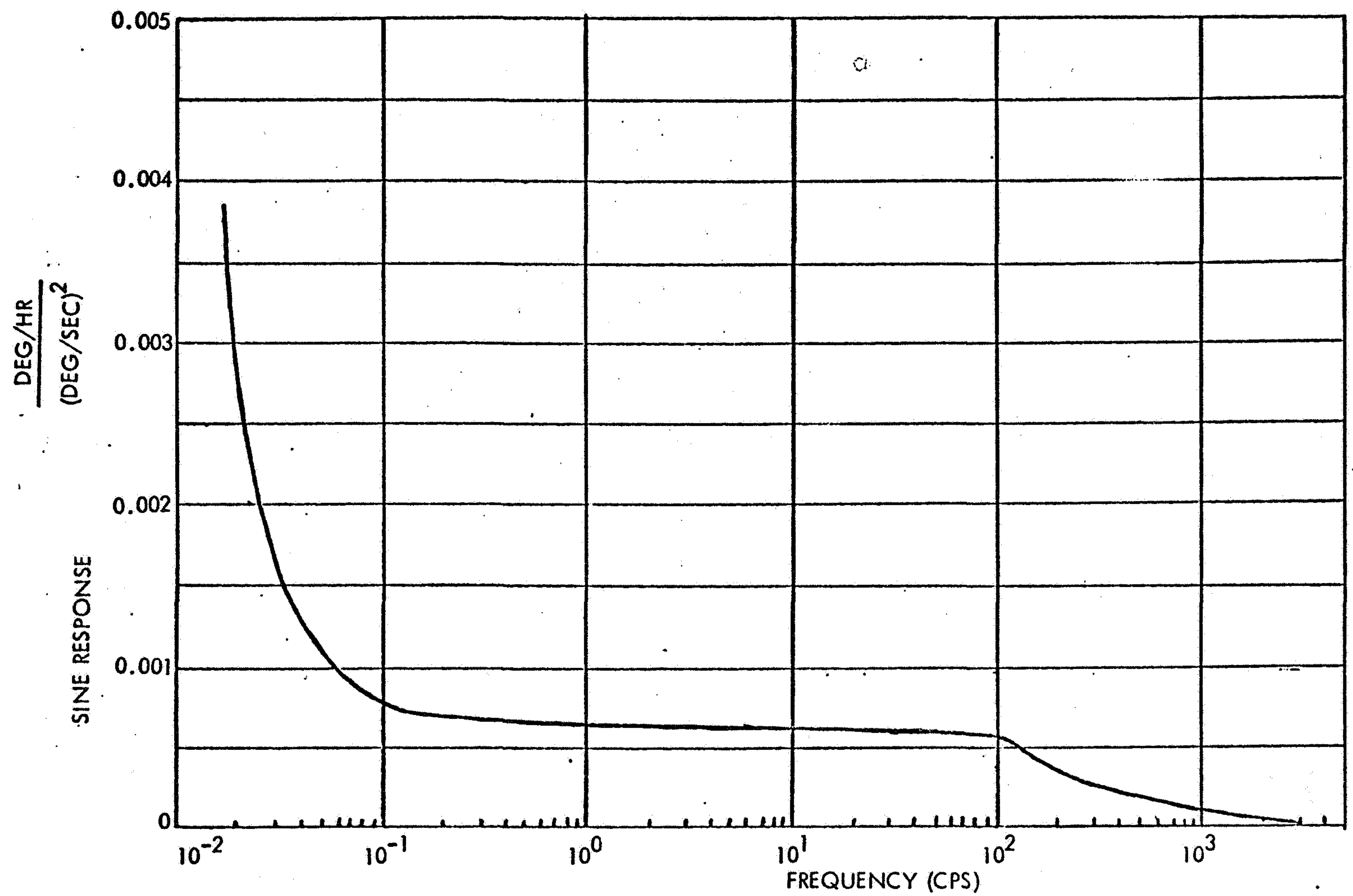


Figure 4-42. Frequency Coefficient for MSFC SAP F.C. ±11 P.C. DE

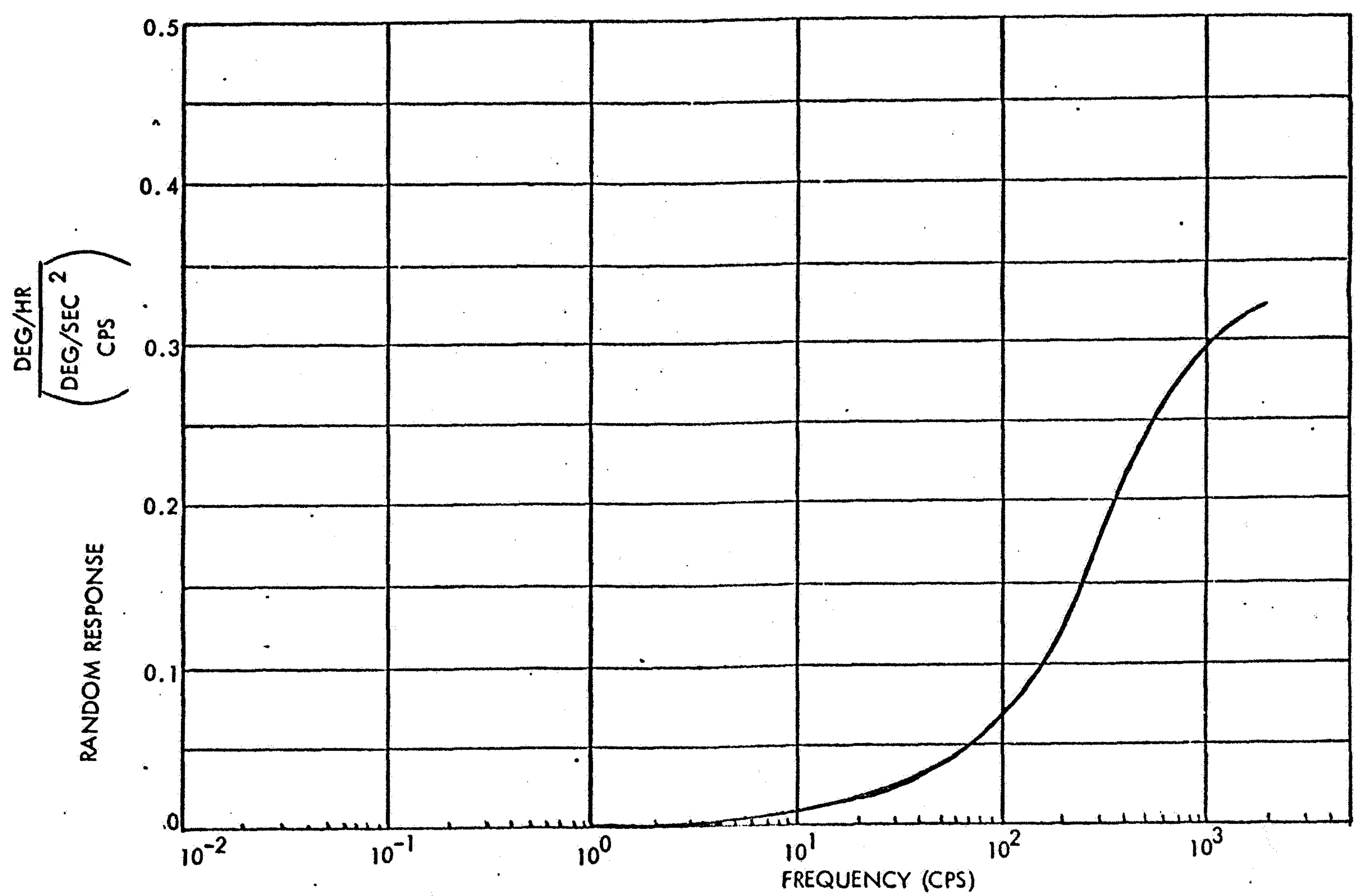


Figure 4-43. Random Integrated Error for MSFC SAP F.C. ±11 P.C. DE

The effects of fictitious coning due to input axis components of angular rate have been shown to be small. Therefore, consider the θ_p and $\dot{\theta}_p$ to be only a function of ω_2 as reflected by Equation (4-38). As discussed previously, if the two ω_2 components were exactly in phase and the roll and pitch axis transfer functions (Equation (4-35) were identical then again no fictitious coning would be indicated by Equation (4-39). However, if phase and gain differences do exist between loops then as before fictitious coning effects would be interpreted by the computer. To demonstrate the possible error magnitude the open loop gain associated with Equation (4-38) was perturbed ±5 db. Note that Equation (4-38) was also used to determine rectification error ϵ_8, so closed loop computer work had already been performed. The transfer functions associated with each SAP are shown in Figure 4-44. The drift coefficient associated with this "output axis" type of fictitious coning is shown in Figure 4-45. It produces a maximum value of 0.014 deg/hr normalized to 1 deg/sec amplitudes. The accumulated random integration is shown in Figure 4-46. At the 40 cps bandwidth cutoff the error is 0.0052 deg/hr (obtained from computer tape) normalized to a unity PSD of angular rate.

This error again causes no real concern. The drift coefficient (sinusoidal response) requires a specific high frequency before being excited. The random error would be negligible when multiplied by the 0.0005 PSD magnitude.

As has been indicated, the analysis presented here has assumed an ideal computer such that the coefficient C(s) is equal to 1.0. Actually, the effects of computer sampling have considerable consequences on the determination of true coning that is undergone by the vehicle. These effects are highly dependent on the frequency of the excitations. In general, if the excitation frequency is above half that of the sampling frequency (100 cps from Phase I) then the excitation effects are folded down. Excitation effects for frequencies below those of the half-sampling frequency are essentially passed through and contribute to the error content as previously explained.

```
PLS(1,1)=(-185.7,107.0),(-185.7,-107.0),(-65.8,0.0),
(-5833.0,0.0),(-117.2,1047.0),(-117.2,-1047.0),(-2494.0,0.0),
ZRS(1,1)=(-50.0,0.0),(-100.0,200.0),(-100.0,-200.0),
NPOL(1)=7,NZER(1)=3,GKB(1)=4.84E-4,
PSI=0.0,KO=62.8,
PLS(1,2)=(-43.9,-45.0),(-43.9,45.0),(-86.8,535.2),
(-86.8,-535.2),(-1394.0,537.0),(-1394.0,-537.0),(-5950.0,0.0),
ZRS(1,2)=(1.0E-6,0.0),(-50.0,0.0),(-100.0,200.0),
(-100.0,-200.0),
NPOL(2)=7,
NZER(2)=4,
GKB(2)=4.84E-10,
$.REDEF HEAD=|-
HEAD=SAP FICTITIOUS CONING ERROR OUTPUT AXIS +/-5DB
KRAND=1,
PHI=163*1.0,
$END
```

Poles and zeros for gyro loop with +5db gain

Poles and zeros for gyro loop with -5db gain

Figure 4-44. SAP Fictitious Coning Error - Output Axis ±5db Input Data Sheet

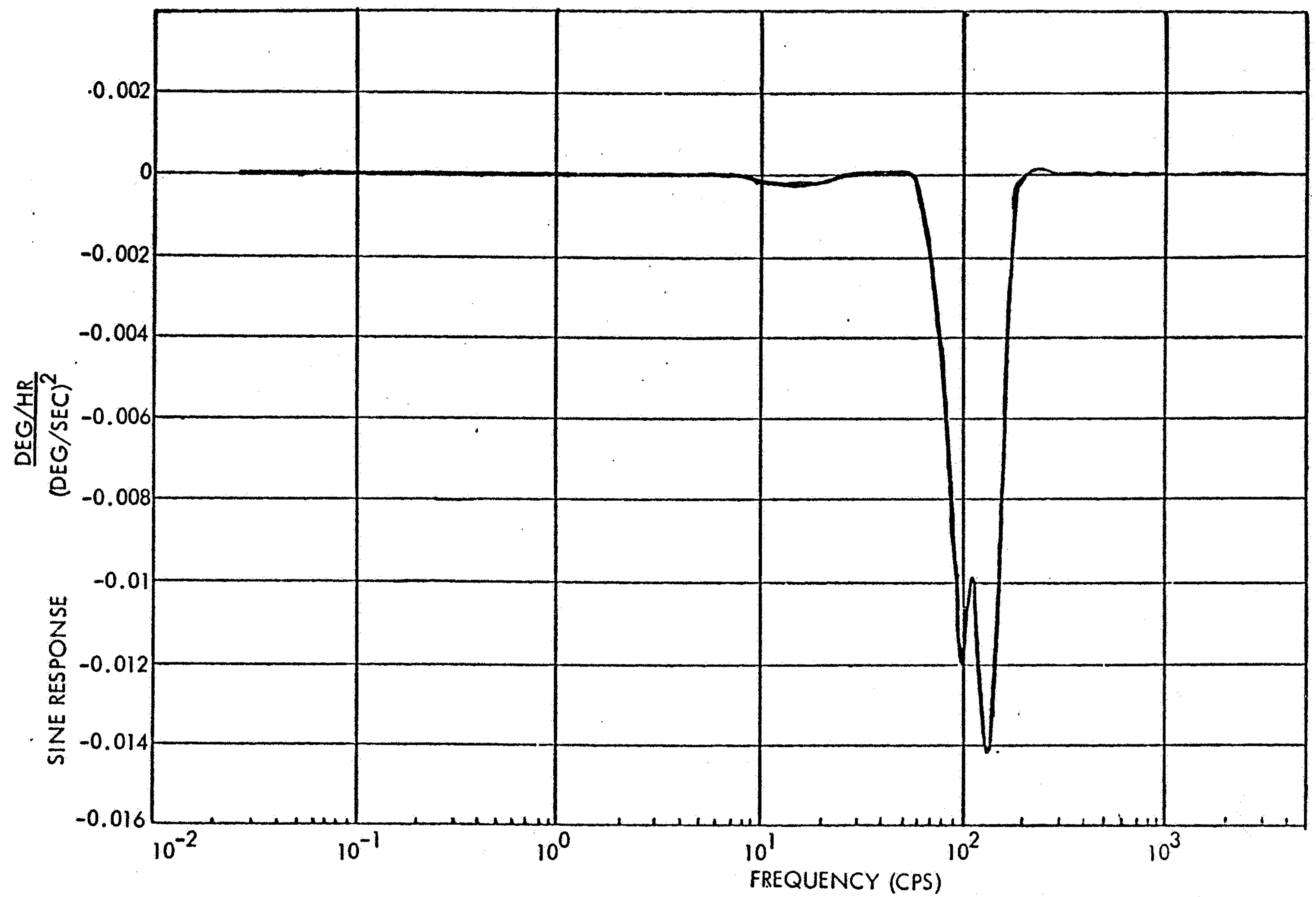


Figure 4-45. Frequency Coefficient for SAP Ficititious Coning Error — Output Axis ±5 db

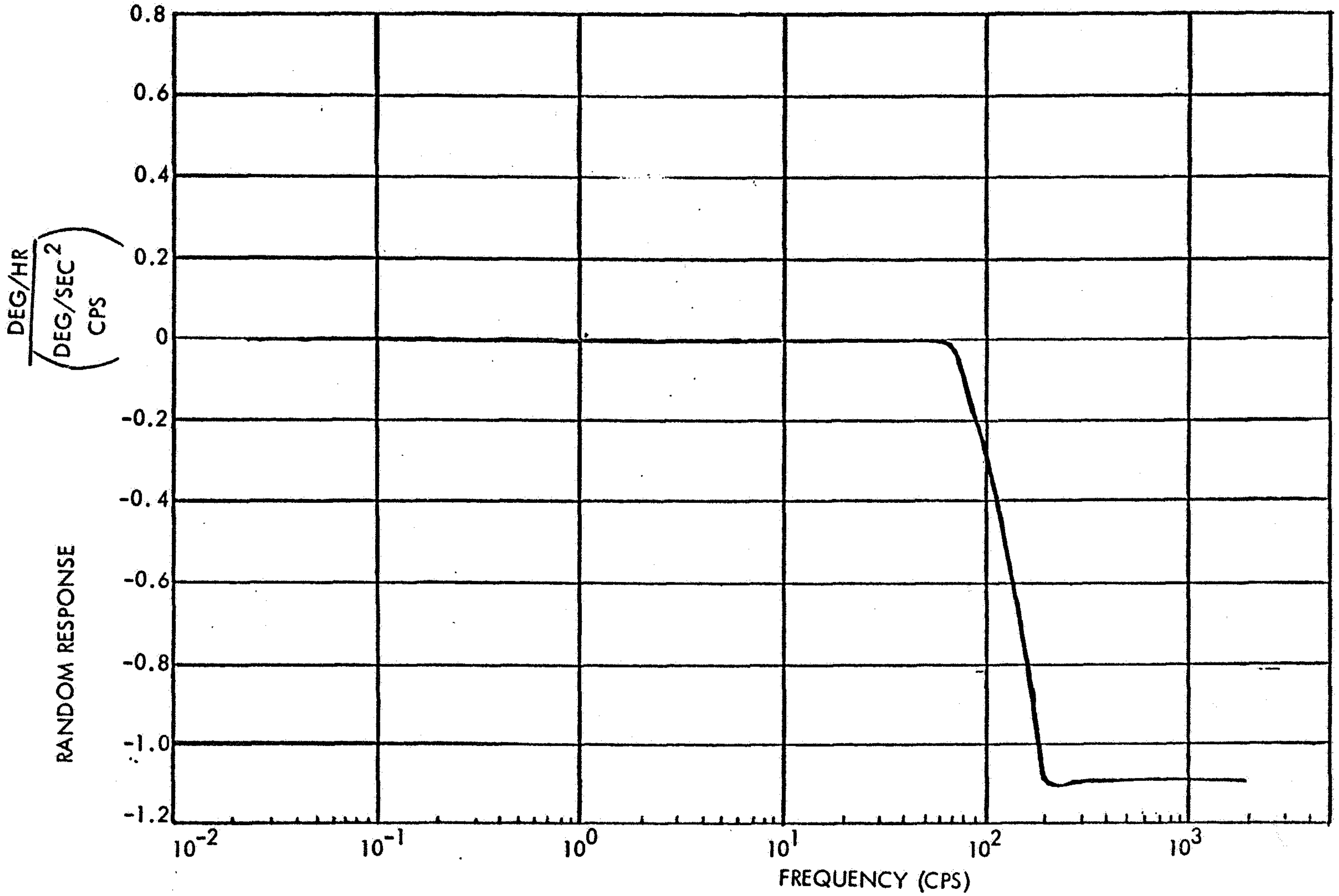


Figure 4-46. Random Integrated Error for SAP Fictitious Coning Error — Output Axis ±5 db

This condition may be both beneficial and detrimental. If only fictitious coning is induced, then high frequency contributions would be attenuated and system performance is assisted. However, if the vehicle is actually undergoing high frequency coning, then this true motion would also be lost to the computer. A rigorous treatment of the case of sampling has been accomplished for specific inputs only. The results of such analyses are presented in Reference 7. Included in that reference are some normalized curves to determine coning error rate in percent of true vehicle coning rate based on sample frequencies.

These results indicate that fictitious coning contributes no serious error effects in the MSFC-SIRA. However, these results should not be considered conclusive. Further studies should be conducted to determine the effects of all possible combinations of open loop gain and damping variations. Furthermore, realistic tolerances of these parameters should be established as should a true value for the back emf coefficient, D_E. Also the combined effects of sensor loop characteristics with computer sampling properties should be evaluated.

REFERENCES

1. D.D. Otten, "Body-Fixed, Three-Axis Reference System Study," TRW Report No. 4499-6007-R0000, TRW Systems, Redondo Beach, California, 2 May 1966.

2. H.W. Whitcomb, "Utility Program No. AZ013, Programming Guide for the Differential Equations Abstraction," TRW Systems, Redondo Beach, California, September 1965.

3. R.C. Rountree, "Phase II Progress on MSFC Strapdown Study—SAP Analysis," IOC No. 7222.6-25, TRW Systems, Redondo Beach, California, 1 July 1966.

4. R.C. Rountree, "Documentation of the Summary of SAP and PIGA Error Model Studies for MSFC, Phase I," IOC No. 7222.3-25, TRW Systems, Redondo Beach, California, 6 July 1966.

5. D. Sargent and R.C. Rountree, "Rectification Errors in Strapdown Inertial Components Due to Random Vibrations," Presented at 3rd Inertial Guidance Test Symposium, Holloman Air Force Base, New Mexico, 19-20 October 1966.

6. L. Goodman and A. Robinson, "Effect of Finite Rotations on Gyroscopic Sensing Devices," Journal of Applied Mechanics, pp. 210-213, June 1958.

7. K.L. Baker, "Analytic Platform Coning Study Results," IOC No. 7221.7-68, TRW Systems, Redondo Beach, California, 13 July 1966.

5. AIRBORNE COMPUTER ANALYSES

5.1 INTRODUCTION AND SUMMARY

During Phase I, the errors due to computer software were evaluated analytically (Reference 1, Section 5). During Phase II the software developed in Phase I was programmed in interpretive routines whereby the complete performance of the software in conjunction with various word length and speed airborne computers could be evaluated. The types of computers simulated included both General Purpose (GP) and Digital Differential Analyzer (DDA).

Subsection 5.2 summarizes the airborne Hypothetical Computer (Hypocomp) simulations of the attitude reference equations as developed in Phase I. The results of these simulations resulted in the choice of two airborne-computer word length and speed combinations for use in the performance analyses (Section 4) and the computer implementation studies (Section 7). These choices were a 24 bit (including sign), 10-msec machine achieving a total equivalent drift performance for both software and computer of 0.0156 deg/hr, and a less accurate 21 bit (including sign), 20-msec machine offering 0.067 deg/hr total equivalent drift performance.

Subsection 5.3 describes and presents the results of a simulation of the velocity transformation equations. This simulation evaluated three methods of transforming the velocity accumulated in body coordinates to inertial coordinates. The results of this simulation showed a definite improvement in accuracy can be obtained if either the direction cosines present at the middle of the body-velocity-accumulation interval, or the average of the direction cosines present at the beginning and end of the body-velocity-accumulation interval are used for the transformation.

Subsection 5.4 summarizes the interpretive simulation of various register length and speed DDA's for both attitude reference and velocity transformation performance. The results of this simulation resulted in the choice of a 16-bit register (not including sign), 10,000-cps (or faster) machine for the performance analyses (Section 4) and computer implementation studies (Section 7). This simulation also identified the need to improve the PIGA resolution by at least a factor of 4 from the present level of 0.05 mps if the airborne computer selected is a DDA.

5.2 HYPOCOMP SIMULATION OF THE ATTITUDE-REFERENCE EQUATIONS

5.2.1 Introduction

A simulation of the attitude-reference equations was performed using the Hypocomp computer program. The Hypocomp program simulates computations performed by a fixed-point digital computer of specified word length. No specific computer is represented, so that the results are only approximate for a particular computer.

5.2.2 Algorithms Compared

Two different basic algorithms are simulated. They are:

- Second-order Taylor series
- Modified second-order Taylor series

In addition, both algorithms have optional double-precision accumulation of the increments of the direction cosines, so that in actuality four different algorithms are studied. This choice of algorithms permits the comparison of a basic simple method (second-order Taylor series) with a more sophisticated method using cofactor calculations and orthogonality corrections.

5.2.2.1 Second-Order Taylor Series (Algorithm 1)

$$\Delta a_{1j} = -\tfrac{1}{2}(\Delta\alpha_2^{\,2} + \Delta\alpha_3^{\,2})a_{1j} + (\tfrac{1}{2}\Delta\alpha_1\Delta\alpha_2 + \Delta\alpha_3)a_{2j} + (\tfrac{1}{2}\Delta\alpha_3\Delta\alpha_1 - \Delta\alpha_2)a_{3j}$$

$$\Delta a_{2j} = (\tfrac{1}{2}\Delta\alpha_1\Delta\alpha_2 - \Delta\alpha_3)a_{1j} - \tfrac{1}{2}(\Delta\alpha_3^{\,2} + \Delta\alpha_1^{\,2})a_{2j} + (\tfrac{1}{2}\Delta\alpha_2\Delta\alpha_3 + \Delta\alpha_1)a_{3j}$$

$$\Delta a_{3j} = (\tfrac{1}{2}\Delta\alpha_3\Delta\alpha_1 + \Delta\alpha_2)a_{1j} + (\tfrac{1}{2}\Delta\alpha_2\Delta\alpha_3 - \Delta\alpha_1)a_{2j} - \tfrac{1}{2}(\Delta\alpha_1^{\,2} + \Delta\alpha_2^{\,2})a_{3j}$$

$$j = 1, 2, 3 \qquad (5\text{-}1)$$

$$a_{ij} = a_{ij} + \Delta a_{ij}; \quad i = 1, 2, 3; \quad j = 1, 2, 3 \qquad (5\text{-}2)$$

Equation (5-2) may optionally be performed in single or double precision. When double precision is used, the Δa_{ij} is calculated in single precision and then extended to double precision with zeros.

A double-right shift is performed to scale the Δa_{ij} the same as the a_{ij} and a double-precision add is performed. Only the most precise parts of the a_{ij} are used in Equation (5-1). In a computer without a double-precision add order, the same effect may be achieved by a single-precision series of operations in which the bits, which would be lost in shifting the Δa_{ij} right to scale them the same as the a_{ij}, are saved and accumulated in storage bins. Overflows from these bins are added to the least significant bits of the appropriate a_{ij}. A separate bin is maintained for each a_{ij}.

5.2.2.2 Modified Second-Order Taylor Series (Algorithm 2)

$$\begin{aligned} E_1 &= \tfrac{1}{2}(1 - a_{11}^2 - a_{12}^2 - a_{13}^2) \\ E_3 &= \tfrac{1}{2}(1 - a_{31}^2 - a_{32}^2 - a_{33}^2) \\ E_{13} &= \tfrac{1}{2}(a_{11}a_{31} + a_{12}a_{32} + a_{13}a_{33}) \end{aligned} \qquad (5\text{-}3)$$

$$\begin{aligned} \Delta a_{1j} &= -\tfrac{1}{2}(\Delta\alpha_2^2 + \Delta\alpha_3^2)a_{1j} + (\tfrac{1}{2}\Delta\alpha_1\Delta\alpha_2 + \Delta\alpha_3)a_{2j} + (\tfrac{1}{2}\Delta\alpha_3\Delta\alpha_1 - \Delta\alpha_2)a_{3j} \\ \Delta a_{3j} &= (\tfrac{1}{2}\Delta\alpha_3\Delta\alpha_1 + \Delta\alpha_2)a_{1j} + (\tfrac{1}{2}\Delta\alpha_2\Delta\alpha_3 - \Delta\alpha_1)a_{2j} - \tfrac{1}{2}(\Delta\alpha_1^2 + \Delta\alpha_2^2)a_{3j} \end{aligned} \qquad (5\text{-}4)$$

$$j = 1, 2, 3$$

$$\begin{aligned} a_{1j} &= a_{1j} + (\Delta a_{1j} + a_{1j}E_1 - a_{3j}E_{13}) \\ a_{3j} &= a_{3j} + (\Delta a_{3j} + a_{3j}E_3 - a_{1j}E_{13}) \end{aligned} \qquad j = 1, 2, 3 \qquad (5\text{-}5)$$

$$\begin{aligned} a_{21} &= a_{13}a_{32} - a_{12}a_{33} \\ a_{22} &= a_{11}a_{33} - a_{13}a_{31} \\ a_{23} &= a_{12}a_{31} - a_{11}a_{32} \end{aligned} \qquad (5\text{-}6)$$

The terms in the parentheses in Equation (5-5) are added together in single precision. The result is optionally added to a_{ij} by the same double-precision technique used in Equation (5-2).

5.2.3 Scaling

Three different sampling rates are studied: 8, 32, and 128 cps. For the same maximum input rates, the maximum values of the $\Delta\alpha_i$ and Δa_{ij} vary with sampling rate, so that the optimum scaling varies as well. If the higher sampling rates are not to be penalized by nonoptimum scaling, it is necessary to scale the equations for each rate. The scalings used are shown in the following table.

Variable	Units	Sampling Rate 8	32	128 cps
a_{ij}	-	1.01	1.01	1.01
Δa_{ij}	-	$1.01 \cdot 2^{-3}$	$1.01 \cdot 2^{-5}$	$1.01 \cdot 2^{-7}$
$\Delta\alpha_i$	rad	2^{-3}	2^{-5}	2^{-7}
E_{ij}	-	2^{-3}	2^{-5}	2^{-7}

These scalings allow for a maximum input rate of 1 rad/sec or 57.3 deg/sec and for a nonorthogonality error of about 1 percent.

5.2.4 Sensor Quantization

The inputs to the above algorithms are the incremental angles $\Delta\alpha_i$ received from the gyros. This information will, in general, be more coarsely quantized than the rest of the variables present in the computer and must be treated separately. The outputs of the Hypocomp drivers $\Delta\alpha'_i$ represent the outputs of ideal, unquantized sensors, which must be quantized to give the $\Delta\alpha_i$. Because the quantization process takes place in a closed loop containing an integrator (the gyro), the quantization errors, Q_i, are remembered and carried forward from one cycle to the next. The process is as follows.

Initially, before a run:

$$Q_i = 0 \qquad i = 1, 2, 3 \tag{5-7}$$

For each computation cycle

$$\Delta\alpha''_i = \Delta\alpha'_i + Q_i$$

$$\Delta\alpha_i = [\Delta\alpha''_i]_Q \qquad i = 1, 2, 3 \qquad (5\text{-}8)$$

$$Q_i = \Delta\alpha''_i - \Delta\alpha_i$$

The symbol $[\ ]_Q$ means roundoff to a quantum size Q. If $y = [x]_Q$, then y is Q times the largest (algebraically) integer less than or equal to $x/Q + 1/2$. Thus $|Q_i|$ is always less than or equal to $Q/2$. Q is specified for different runs. It is a negative integer power of 2 in rad measure.

5.2.5 Hypocomp Driver

5.2.5.1 Driver Equations

The angular velocity of the vehicle will be specified in the form

$$\omega_i = A_{O_i} + A_{A_i} t + A_i \Omega_i \cos(\Omega_i t + \varphi_i); \quad i = 1, 2, 3 \qquad (5\text{-}9)$$

The ideal gyro outputs at time t are:

$$\Delta\alpha'_i = \int_{t-\Delta t}^{t} \omega_i dt; \qquad i = 1, 2, 3 \qquad (5\text{-}10)$$

Thus

$$\Delta\alpha'_i = A_{O_i} \Delta t + A_{A_i} \Delta t (t - \frac{\Delta t}{2}) + A_i [\sin(\Omega_i t + \varphi_i) - \sin(\Omega_i t + \varphi_i - \Omega_i \Delta t)] \qquad i = 1, 2, 3 \qquad (5\text{-}11)$$

The coefficients, which are read in as inputs are:

A_{O_i}	constant-angular velocities (rad/sec)
A_{A_i}	constant-angular accelerations (rad/sec^2)
A_i	amplitude of sinusoidal oscillations (rad)
Ω_i	frequency of sinusoidal oscillations (rad/sec)

φ_i phase of sinusoidal oscillations (rad)

Δt computation interval (sec)

The driver outputs are computed in floating point. They are then converted to fixed point, scaled at 2^0. The quantizing operation in Equation (5-8) is next performed. Then the outputs are left shifted 3, 5, or 7 binary places to scale them at 2^{-3}, 2^{-5}, or 2^{-7} depending upon whether the sampling rate is 8, 32, or 128 cps, respectively.

5.2.5.2 Test Runs

The following driver outputs were used:

Test No.		
1	Three-axis slew	$A_{0_1} = A_{0_2} = A_{0_3} = 12\sqrt{3}$ deg/sec
2	Three-axis slew	$A_{0_1} = A_{0_2} = A_{0_3} = -6\sqrt{3}$ deg/sec
3	Three-axis vibration	$A_1 = A_2 = A_3 = 0.5$ deg $\Omega_1 = \Omega_2 = \Omega_3 = 10\pi$ rad/sec
4	Coning	$A_{0_1} = 2\pi(1 - \cos 0.5°)$ rad/sec $A_2 = A_3 = \sin 0.5°$ $\Omega_2 = \Omega_3 = 2\pi$ rad/sec $\varphi_3 = \pi/2$

The different runs made using these test inputs are tabulated in Paragraph 5.2.6.1.

5.2.5.3 Coning Test Runs

The slew and vibration tests have in common an angular-velocity vector with a constant direction in space and, therefore, must be considered as special cases. The simplest test case for which the direction of the angular-velocity vector is not constant and for which a closed form solution is obtainable is that in which an axis of the vehicle moves in a circular cone. Therefore, the largest group of runs investigating type of algorithm, precision of summation, sampling rate, word length, and quantization are made using circular coning about the x axis as the test motion.

If β is the coning half-angle and Ω is the coning angular frequency, then the angular velocity is

$$\{\omega\} = \begin{Bmatrix} \Omega(1 - \cos\beta) \\ \Omega \sin\beta \cos\Omega t \\ \Omega \sin\beta \cos(\Omega t + \pi/2) \end{Bmatrix} \tag{5-12}$$

Note that a small constant-angular rate is present on the x axis. It has been introduced to cause the body to return to its original orientation after each coning cycle. Therefore, any deviation of the direction cosine matrix from the identity matrix at the end of each cycle represents an error.

The driver coefficients for circular coning are

$$\begin{aligned}
A_{0_1} &= \Omega(1 - \cos\beta); \quad A_{0_2} = A_{0_3} = 0 \\
A_{A_1} &= A_{A_2} = A_{A_3} = 0 \\
A_1 &= 0; \quad A_2 = A_3 = \sin\beta \\
\Omega_1 &= 0; \quad \Omega_2 = \Omega_3 = \Omega \\
\varphi_1 &= \varphi_2 = 0; \quad \varphi_3 = \pi/2
\end{aligned} \tag{5-13}$$

The closed form solution of the direction cosine equation given Equation (5-12) is

$$\begin{aligned}
a_{11} &= \sin^2\beta \cos\Omega t + \cos^2\beta \\
a_{12} &= -\sin\beta \cos\beta(1 - \cos\Omega t) \\
a_{13} &= -\sin\beta \sin\Omega t \\
a_{21} &= -\sin\beta \cos\beta(1 - \cos\Omega t)\cos\Omega t + \sin\beta \sin^2\Omega t \\
a_{22} &= (\cos^2\beta \cos\Omega t + \sin^2\beta)\cos\Omega t + \cos\beta \sin^2\Omega t \\
a_{23} &= (1 - \cos\beta)\sin\Omega t \cos\Omega t \\
a_{31} &= \sin\beta \cos\beta(1 - \cos\Omega t)\sin\Omega t + \sin\beta \sin\Omega t \cos\Omega t
\end{aligned}$$

$$a_{32} = -(\cos^2\beta \cos\Omega t + \sin^2\beta)\sin\Omega t + \cos\beta \sin\Omega t \cos\Omega t$$
$$a_{33} = \cos\beta \sin^2\Omega t + \cos^2\Omega t$$

After each cycle, where $t = 2\pi n/\Omega$, $[a_{ij}]$ is the identity matrix.

One of the major errors affecting the attitude reference in the presence of coning motions is the commutativity error. This error is caused by the noncommutativity of rotations and the finite sampling rate of the system. It is independent of the choice of algorithm 1 or 2 since both algorithms assume a constant direction of the angular-velocity vector during the sampling period. Therefore, it is convenient to calculate the error analytically so that it may be separated from the other errors. As derived in Appendix I, the drift rate is

$$\omega_D = \frac{\Omega^3 T^2 \beta^2}{12} \tag{5-14}$$

For a typical case

$\Omega = 1$ cps $= 2\pi$ rad/sec

$T = 1/8,\ 1/32,\ 1/128$ sec

$\beta = 0.5$ deg $= 0.00873$ rad

$\omega_D = 0.246 \times 10^{-4},\ 0.154 \times 10^{-5},\ 0.960 \times 10^{-7}$, rad/sec

$= 5.07,\ 0.317,\ 0.0198$, deg/hr

5.2.5.4 Outputs

At each output printout time, the a_{ij} and Δa_{ij} are rescaled by multiplying by 1.01 and printed out. The scale, skew, and drift errors are also printed out. If an orthonormal vector triad is transformed by the matrix the resultant vector triad will not be orthonormal. The deviations of the vectors from unit lengths are the scale errors:

$$\epsilon_i \cong \tfrac{1}{2}(a_{i1}^2 + a_{i2}^2 + a_{i3}^2 - 1); \qquad i = 1, 2, 3 \tag{5-15}$$

The deviations of the angles between pairs of the vectors from $\pi/2$ rad are the skew errors:

$$\begin{aligned} e_1 &\cong a_{21}a_{31} + a_{22}a_{32} + a_{23}a_{33} \\ e_2 &\cong a_{31}a_{11} + a_{32}a_{12} + a_{33}a_{13} \\ e_3 &\cong a_{11}a_{21} + a_{12}a_{22} + a_{13}a_{23} \end{aligned} \tag{5-16}$$

The drift errors represent the rotation of the triad as a whole and are found by designing the inputs so that the ideal matrix is the identity matrix:

$$\begin{aligned} d_1 &\cong \tfrac{1}{2}(a_{23} - a_{32}) \\ d_2 &\cong \tfrac{1}{2}(a_{31} - a_{13}) \\ d_3 &\cong \tfrac{1}{2}(a_{12} - a_{21}) \end{aligned} \tag{5-17}$$

5.2.6 Results

5.2.6.1 Description of Runs

The computer runs made are cataloged in Table 5-1. Algorithm 1 refers to the second-order Taylor series algorithm and algorithm 2 refers to the second-order Taylor series with cofactor calculations and orthogonality corrections. "D" refers to double-precision accumulation and "S" refers to single precision. The word length is given in bits including the sign bit. Thus, if the word length is 18, the least significant bit has the value of 2^{-17}, and the maximum roundoff error is 2^{-17}. The quantization of the inputs is given in bits including the sign bit. Thus, if the quantization is 24, the least significant bit has the value of 2^{-17}, and the maximum quantization error is 2^{-18}. The remaining items in the table define the total number of cycles, the number of cycles which elapse between each printout, and the values assigned to the parameters of the Hypocomp driver.

5.2.6.2 Tabulation of Errors

The nine error components at the end of each run are tabulated in Table 5-2. They comprise the three components each of the skew, scale, and drift errors, in radian units.

Table 5-1. Description of Computer Runs

Run	1	2	3	4	5	6
Algorithm	1					
Prec	S					
Δt	1/32					
Scaling	2^{-5}					
Word Length	36	24	18	36	24	18
Quantization	36					
No. Int	6401					1281
Print Int	320					
A_{01}	0.36275991			0		
A_{02}	0.36275991			0		
A_{03}	0.36275991			0		
A_{A1}	0					
A_{A2}	0					
A_{A3}	0					
A_1	0			$0.87266462 \times 10^{-2}$		
A_2	0			$0.87266462 \times 10^{-2}$		
A_3	0			$0.87266462 \times 10^{-2}$		
Ω_1	0			31.415927		
Ω_2	0			31.415927		
Ω_3	0			31.415927		
Φ_1	0					
Φ_2	0					
Φ_3	0					
Test No.	1			3		

Table 5-1. Description of Computer Runs (Continued)

Run	7	8	9	10	11	12
Algorithm	1					
Prec	D					
Δt	1/32					
Scaling	2^{-5}					
Word Length	36	24	18	36	24	18
Quantization	23					
No. Int	6401			3201		
Print Int	320			160		
A_{01}	0.36275991					
A_{02}	0.36275991					
A_{03}	0.36275991					
A_{A1}	0					
A_{A2}	0					
A_{A3}	0					
A_1	0			$0.87266462 \times 10^{-2}$		
A_2	0			$0.87266462 \times 10^{-2}$		
A_3	0			$0.87266462 \times 10^{-2}$		
Ω_1	0			31.415927		
Ω_2	0			31.415927		
Ω_3	0			31.415927		
Φ_1	0					
Φ_2	0					
Φ_3	0					
Test No.	1			3		

Table 5-1. Description of Computer Runs (Continued)

Run	13	14	15	16	17	18
Algorithm	1					
Prec	D					
Δt	1/32					
Scaling	2^{-5}					
Word Length	36	24	18	36		
Quantization	23			18		
No. Int	6401					
Print Int	320			160		
A_{01}	-0.18137996			0	$0.23924439 \times 10^{-3}$	$-0.23924439 \times 10^{-3}$
A_{02}	-0.18137996			0		
A_{03}	-0.18137996			0		
A_{A1}	0					
A_{A2}	0					
A_{A3}	0					
A_1	0					
A_2	0			$0.87265346 \times 10^{-2}$		
A_3	0			$0.87265346 \times 10^{-2}$		
Ω_1	0					
Ω_2	0			6.2831853		
Ω_3	0			6.2831853		
Φ_1	0					
Φ_2	0					
Φ_3	0			1.5707963		
Test No.	2			4		

Table 5-1. Description of Computer Runs (Continued)

Run	19	20	21	22	23	24
Algorithm	2					
Prec	D					
Δt	1/8				1/32	
Scaling	2^{-3}				2^{-5}	
Word Length	15	18	21	24	15	18
Quantization	15					
No. Int	6401					
Print Int	160					
A_{01}	$0.23924439 \times 10^{-3}$					
A_{02}	0					
A_{03}	0					
A_{A1}	0					
A_{A2}	0					
A_{A3}	0					
A_1	0					
A_2	$0.87265346 \times 10^{-2}$					
A_3	$0.87265346 \times 10^{-2}$					
Ω_1	0					
Ω_2	6.2831853					
Ω_3	6.2831853					
Φ_1	0					
Φ_2	0					
Φ_3	1.5707963					
Test No.	4					

Table 5-1. Description of Computer Runs (Continued)

Run	25	26	27	28	29	30
Algorithm	2					
Prec	D					
Δt	1/32		1/128			
Scaling	2^{-5}		2^{-7}			
Word Length	21	24	15	18	21	24
Quantization	15					
No. Int	6401					
Print Int	160					
A_{01}	$0.23924439 \times 10^{-3}$					
A_{02}	0					
A_{03}	0					
A_{A1}	0					
A_{A2}	0					
A_{A3}	0					
A_1	0					
A_2	$0.87265346 \times 10^{-2}$					
A_3	$0.87265346 \times 10^{-2}$					
Ω_1	0					
Ω_2	6.2831853					
Ω_3	6.2831853					
Φ_1	0					
Φ_2	0					
Φ_3	1.5707963					
Test No.	4					

Table 5-1. Description of Computer Runs (Continued)

Run	31	32	33	34	35	36
Algorithm	2					
Prec	S					
Δt	1/8				1/32	
Scaling	2^{-3}				2^{-5}	
Word Length	15	18	21	24	15	18
Quantization	15					
No. Int	6401					
Print Int	160					
A_{01}	$0.23924439 \times 10^{-3}$					
A_{02}	0					
A_{03}	0					
A_{A1}	0					
A_{A2}	0					
A_{A3}	0					
A_1	0					
A_2	$0.87265346 \times 10^{-2}$					
A_3	$0.87265346 \times 10^{-2}$					
Ω_1	0					
Ω_2	6.2831853					
Ω_3	6.2831853					
Φ_1	0					
Φ_2	0					
Φ_3	1.5707963					
Test No.	4					

Table 5-.. Description of Computer Runs (Continued)

Run	37	38	39	40	41	42
Algorithm	2					
Prec	S					
Δt	1/32			1/128		
Scaling	2^{-5}			2^{-7}		
Word Length	21	24	15	18	21	24
Quantization	15					
No. Int	6401					
Print Int	160					
A_{01}	0.23924439×10^{-3}					
A_{02}	0					
A_{03}	0					
A_{A1}	0					
A_{A2}	0					
A_{A3}	0					
A_1	0					
A_2	0.87265346×10^{-2}					
A_3	0.87265346×10^{-2}					
Ω_1	0					
Ω_2	6.2831853					
Ω_3	6.2831853					
Φ_1	0					
Φ_2	0					
Φ_3	1.5707963					
Test No.	4					

Table 5-1. Description of Computer Runs (Continued)

Table 5-1. Description of Computer Runs (Continued)

Run	43	44	45	46	47	48
Algorithm	1					
Prec	D					
Δt	1/8		1/32		1/128	
Scaling	2^{-3}		2^{-5}		2^{-7}	
Word Length	18	24	18	24	18	24
Quantization	15					
No. Int	6401					
Print Int	160					
A_{01}	$0.23924439 \times 10^{-3}$					
A_{02}	0					
A_{03}	0					
A_{A1}	0					
A_{A2}	0					
A_{A3}	0					
A_1	0					
A_2	$0.87265346 \times 10^{-2}$					
A_3	$0.87265346 \times 10^{-2}$					
Ω_1	0					
Ω_2	6.2831853					
Ω_3	6.2831853					
Φ_1	0					
Φ_2	0					
Φ_3	1.5707963					
Test No.	4					

Table 5-1. Description of Computer Runs (Continued)

Run	49	50	51	52	53	54
Algorithm	2					
Prec	D					
Δt	1/8		1/32		1/32	
Scaling	2^{-3}		2^{-5}		2^{-7}	
Word Length	24	9	12	9	12	9
Quantization	12					
No. Int	6401					
Print Int	160					
A_{01}	$0.23924439 \times 10^{-3}$					
A_{02}	0					
A_{03}	0					
A_{A1}	0					
A_{A2}	0					
A_{A3}	0					
A_1	0					
A_2	$0.87265346 \times 10^{-2}$					
A_3	$0.87265346 \times 10^{-2}$					
Ω_1	0					
Ω_2	6.2831853					
Ω_3	6.2831853					
Φ_1	0					
Φ_2	0					
Φ_3	1.5707963					
Test No.	4					

Table 5-1. Description of Computer Runs (Continued)

Run	55	56	57	58	59	60
Algorithm	2			1		
Prec	D					
Δt	1/8	1/32	1/128	1/8	1/32	1/128
Scaling	2^{-3}	2^{-5}	2^{-7}	2^{-3}	2^{-5}	2^{-7}
Word Length	36					
Quantization	15					
No. Int	6401					
Print Int	160					
A_{01}	$0.23924439 \times 10^{-3}$					
A_{02}	0					
A_{03}	0					
A_{A1}	0					
A_{A2}	0					
A_{A3}	0					
A_1	0					
A_2	$0.87265346 \times 10^{-2}$					
A_3	$0.87265346 \times 10^{-2}$					
Ω_1	0					
Ω_2	6.2831853					
Ω_3	6.2831853					
Φ_1	0					
Φ_2	0					
Φ_3	1.5707963					
Test No.	4					

Table 5-2. Tabulation of Errors

RUN	SKEW			SCALE			DRIFT		
1	-.7929-4	-.7929-4	-.7929-4	.7928-4	.7928-4	.7928-4	.4668-2	.4668-2	.4668-2
2	-.8850-4	-.8850-4	-.8850-4	.8615-4	.8615-4	.8615-4	.4235-2	.4235-2	.4235-2
3	-.1804-2	-.1804-2	-.1804-2	.8083-3	.8083-3	.8083-3	-.2307-1	-.2307-1	-.2307-1
4	-.8252-5	-.8252-5	-.8252-5	.8247-5	.8247-5	.8247-5	-.1903-2	-.1903-2	-.1903-2
5	-.2144-4	-.2144-4	-.2144-4	.9825-5	.9825-5	.9825-5	-.1903-2	-.1903-2	-.1903-2
6	-.7707-4	-.7707-4	-.7707-4	.1444-3	.1444-3	.1444-3	-.8476-4	-.8476-4	-.8476-4
7	-.7929-4	-.7929-4	-.7929-4	.7928-4	.7928-4	.7928-4	.4668-2	.4668-2	.4668-2
8	-.1183-3	-.1183-3	-.1183-3	.9613-4	.9613-4	.9613-4	.4641-2	.4641-2	.4641-2
9	-.1929-2	-.1929-2	-.1929-2	.1187-2	.1187-2	.1187-2	.2924-2	.2924-2	.2924-2
10	-.4124-5	-.4124-5	-.4124-5	.4123-5	.4123-5	.4123-5	-.4816-3	-.4816-3	-.4816-3
11	-.1047-4	-.1047-4	-.1047-4	.4629-5	.4629-5	.4629-5	-.4815-3	-.4815-3	-.4815-3
12	-.4471-3	-.4471-3	-.4471-3	.5224-4	.5224-4	.5224-4	-.4778-3	-.4778-3	-.4778-3
13	-.4957-5	-.4957-5	-.4957-5	.4958-5	.4958-5	.4958-5	-.5869-3	-.5869-3	-.5869-3
14	-.2295-4	-.2295-4	-.2295-4	.2229-4	.2229-4	.2229-4	-.5634-3	-.5634-3	-.5634-3
15	-.1474-2	-.1474-2	-.1474-2	.1094-2	.1094-2	.1094-2	.9208-3	.9208-3	.9208-3
16	.3143-8	-.1164-9	0	.8556-8	.5239-8	.5472-8	-.4753-1	.3332-4	.1438-5
17	.3143-8	.2678-8	-.1164-9	.8615-8	.7625-8	.7800-8	.3039-3	-.3841-3	-.7761-5
18	.2910-8	-.2561-8	.3492-9	.8557-8	.7916-8	.8382-8	-.9525-1	.4502-3	.1063-4
19	-.6165-4	.1233-3	0	.1372-3	.1343-4	.1371-3	.3082-4	-.9247-3	0
20	-.8077-5	.9306-5	-.2449-5	.6438-5	.7993-5	.1632-4	.1894-1	-.1087-2	-.4893-3
21	-.1846-5	.2728-5	-.4089-6	.2424-5	.1011-5	.2257-5	.1898-1	-.1652-2	-.5495-3
22	-.3075-6	.3411-6	-.7555-7	.1731-6	.2424-6	.2041-6	.1904-1	-.1684-2	-.1822-3
23	.3038-7	.6165-4	.6163-4	.1368-3	.1344-4	.1368-3	0	-.2774-3	.1541-3
24	-.1541-4	.7706-5	.7974-7	.1348-4	.1346-4	.1351-4	.2543-3	-.3198-3	0
25	-.9916-6	.1951-5	-.1820-5	.1938-5	-.1053-7	.1978-5	.2952-3	-.3612-3	-.8187-4
26	-.2451-6	.1240-6	-.1257-6	.1310-6	-.8324-8	.5943-7	.3127-3	-.3682-3	-.1168-4
27		0	0	.1367-3	.1343-4	.1367-3	0	-.6165-4	0
28		.7705-5	.8848-8	.1343-4	-.1981-5	.5732-5	.8576-4	-.1117-3	0
29		0	-.1397-8	.1876-5	-.5716-7	.1877-5	-.1204-4	-.1214-3	0

Table 5-2. Tabulation of Errors (Continued)

RUN	SKEW			SCALE			DRIFT		
30		-.1204-7	-.2403-6	.7061-7	-.5722-7	.7061-7	.4635-5	-.1221-3	-.2288-5
31	.1148-3	-.1291-3	-.1326-3	.1034-3	-.1731-4	.1443-3	-.3347-1	-.3699-3	-.7459-2
32	-.1592-4	.8061-5	-.5306-5	-.1282-5	-.1327-7	.7653-5	.1580-1	-.6357-3	-.9940-3
33	-.1799-5	.1068-5	-.6403-8	.1500-5	-.1795-5	.1438-5	.1831-1	-.1682-2	-.5317-3
34	-.3033-6	.3173-6	-.2042-6	-.5437-7	-.1019-6	.1141-6	.1895-1	-.1691-2	-.1818-3
35	-.5937-4	.5737-4	.1297-3	.1365-4	-.4267-4	.1431-3	.1162-1	-.5240-3	.3082-3
36	-.7635-5	.7753-5	-.1621-4	-.9654-5	-.2815-4	-.4974-5	.3055-2	.2658-3	-.7706-5
37	.1923-5	.1925-5	-.9724-6	-.1003-5	-.5835-5	-.1002-5	-.5105-4	-.1859-3	-.1975-4
38	-.2446-6	.3641-6	-.1543-6	-.1195-6	-.5065-6	-.2077-6	.2544-3	-.3408-3	-.1168-4
39		.6834-7	-.6155-4	.7525-4	-.1097-3	.7510-4	.6165-4	-.1849-3	-.5240-3
40		-.2337-4	.9684-5	-.9672-5	-.3709-4	-.6161-5	.1071-1	-.1965-3	.1926-4
41		-.2889-5	.7101-8	-.4301-7	-.3909-5	-.4162-7	.4286-4	-.1700-3	0
42		-.1205-6	-.1183-6	-.1690-6	-.5387-6	-.1688-6	.1692-4	-.1321-3	-.1264-5
43	.2152-2	.8377-3	-.5489-3	.8329-3	.4675-2	.3156-2	.1916-1	-.1291-2	-.1052-2
44	.2761-4	.1709-4	-.3746-5	.2800-4	.7352-4	.4681-4	.1906-1	-.1683-2	-.1796-3
45	.1095-2	.8485-4	.2350-4	.4374-3	.5843-2	.5684-3	.4778-3	-.3275-3	-.3468-4
46	.1131-4	.3609-8	-.2650-5	.1181-4	.1252-4	.1210-4	.3188-3	.3689-3	-.1029-4
47	.3160-3	-.7916-8	.2573-7	.1984-3	.1290-3	.1290-3	.6550-4	-.1156-3	0
48	.3853-5	.1204-6	-.3602-6	.2358-5	.2592-5	.2479-5	.6622-5	-.1220-3	-.2227-5
49	-.2255-6	-.2240-6	-.4331-7	.1590-6	.1052-6	.8073-7	.1694-1	-.1611-2	-.6496-4
50	-.2708-6	.7893-7	-.1106-7	.9761-7	.1811-6	.2045-6	.2786-1	-.2242-3	-.1368-3
51	.2361-6	-.1513-8	-.2019-6	.6211-7	-.4331-7	.7602-7	-.1670-3	-.4886-3	-.9692-5
52	-.1205-6	-.9688-7	-.1122-6	.1838-6	.6548-7	.6548-7	.1099-2	-.7525-5	-.2149-4
53		-.1210-6	-.2407-6	.6316-7	-.1804-7	.1024-6	-.2802-3	.3009-6	-.2047-5
54		.2328-9	-.2414-6	.6316-7	-.1048-8	.1196-6	.1037-2	.8428-6	-.1204-6
55	0	0	-.1164-9	.2328-9	.4075-9	.1164-9	.1902-1	-.1691-2	-.1314-3
56	-.1164-9	0	-.1164-9	-.5821-10	-.5821-10	-.1164-9	.3138-3	-.3690-3	-.6839-7
57		0	-.1164-9	0	-.5821-10	0	.5307-5	-.1221-3	-.3900-8
58	.5821-8	.1630-8	-.1455-7	.1596-5	.8130-6	.8069-6	.1902-1	-.1691-2	-.1364-3
59	.3376-8	.2328-9	-.2328-9	.8498-8	.5646-8	.5413-8	.3138-3	-.3690-3	-.6822-7
60	.6985-9	0	0	-.1746	-.1164-9	0	.5308-5	-.1221-3	-.3842-8

5.2.6.3 Error Growth With Time

In order to study a wide variety of cases at a reasonable cost, it was necessary to restrict the length of the runs to 6400 cycles. This length corresponds to 800 sec at 8 samples/sec, but only 50 sec at 128 samples/sec. It, therefore, is necessary to find some means for extrapolating the results to the mission length by calculating a drift rate. An examination of the results shows some cases which exhibit a random walk characteristic ($\sim t^{1/2}$) and other cases which exhibit a ramp characteristic ($\sim t$). This phenomenon is understandable when the nature of the roundoff phenomenon is considered in detail as in Paragraph 5.2.6.5, where it is observed that the basic error probability density has a nonzero mean. With a given deterministic driving function two possibilities exist:

- The errors due to the nonzero mean average to zero and only the random walk errors are observed,
- The errors due to the nonzero mean do not average to zero and eventually grow to where they obscure the random walk error component.

Which of the two possibilities occurs depends upon the type of arithmetic used by the computer (such as sign-magnitude or 2's complement) as well as the type of driving function. Therefore, until the particular computer to be used is simulated exactly, the conservative assumption should be made that the errors have a ramp characteristic. Therefore, whenever a drift rate is calculated in this report, it is done under the assumption of linear error growth with time.

5.2.6.4 Interpretation of Results

In interpreting the results obtained, the following points should be kept in mind:

- The Hypocomp results may not be exactly the same as those of any existing computer.
- Only a small number of cases out of the infinity of possible ones have actually been studied so that sweeping general conclusions are difficult to come to.
- The four principal error sources (commutativity, quantization, truncation, and roundoff) interact with each other and sometimes cancel each other out so that the total error is zero or small.

- Previous studies have shown that slight changes in the inputs can cause large changes in the errors.

Therefore, the results should be interpreted with caution and applied with engineering judgment.

Table 5-3 compares single- and double-precision calculations for various word lengths and sampling rates. Algorithm 2 with a quantization of 15 bits and a coning input is used. The total extrapolated drift error in deg/hr is given. One of the major error sources is the commutativity error. As calculated in Subsection 5.3, it is 5.07, 0.317, and 0.0198 deg/hr for 8, 32, and 128 cps respectively.

In Table 5-4, the roundoff-drift error has been separated out by subtracting the double precision 36-bit word length runs from the others. The advantage of the double-precision accumulation technique is clearly seen.

In Table 5-5, the effects of quantization on the total extrapolated drift error are shown. It is seen that the MSFC quantization of about Q = 16 is adequate.

In Table 5-6, the roundoff-drift errors for algorithms 1 and 2 are compared. In 11 out of 12 cases, the error is less for algorithm 2.

In Table 5-7, the skew errors at the end of the run are compared for algorithms 1 and 2. The superiority of algorithm 2 is clearly seen.

In Table 5-8, the scale errors are compared, leading to the same conclusion.

5.2.6.5 Roundoff Analysis

An examination of the simulation results for the three-axis slew shows that the roundoff-drift error grows linearly with time. This result was rather unexpected, since previous simulations showed a roundoff-drift error with a random walk characteristic. The analysis to be described was undertaken to study this apparent discrepancy. It indicates that the results obtained are correct and come about because of the characteristics of the IBM 7094 computer which differ from the characteristics of the computers used in previous simulations. Therefore, although the Hypocomp results are of value in studying the roundoff

Table 5-3. Total Drift Error About Coning Axis - Deg/Hr
Q = 15, Algorithm 2, Coning

1/8	SP	DP
WL = 15	-8.65	0.00795
18	4.08	4.89
21	4.73	4.90
24	4.89	4.92
36		4.91
1/32	**SP**	**DP**
WL = 15	12.0	0.
18	3.15	0.262
21	-0.0526	0.304
24	0.262	0.322
36		0.323
1/128	**SP**	**DP**
WL = 15	0.254	0.
18	44.2	0.350
21	0.177	-0.0497
24	0.0698	0.0191
36		0.0219

Table 5-4. Round Off Drift Error About Coning Axis - Deg/Hr
Q = 15, Algorithm 2, Coning

1/8	SP	DP
WL = 15	-13.5	-4.89
18	- 0.829	-0.0197
21	- 0.183	-0.0111
24	- 0.0181	0.00622
1/32	**SP**	**DP**
WL = 15	11.7	-0.324
18	2.83	-0.0613
21	-0.377	-0.0191
24	-0.0582	-0.00112

Table 5-4. Round Off Drift Error About Coning Axis - Deg/Hr
Q = 15, Algorithm 2, Coning (Continued)

1/128	SP	DP
WL = 15	0.232	-0.0219
18	44.3	0.318
21	0.155	-0.0716
24	0.0480	-0.00277

Table 5-5. Total Drift Error About Coning Axis - Deg/Hr
WL = 24 Algorithm 2, Coning

1/8	SP	DP
Q = 9		7.19
12		4.37
15	4.89	4.92
1/32	**SP**	**DP**
Q = 9		1.13
12		-0.172
15	0.262	0.322
1/128	**SP**	**DP**
Q = 9		4.28
12		1.16
15	0.0698	0.0191

Table 5-6. Roundoff Drift Error About Coning Axis - Deg/Hr
Q = 15, Double Precision, Coning

1/8	ALG 1	ALG 2
WL = 18	0.0372	-0.0197
24	0.0154	0.00622
1/32	**ALG 1**	**ALG 2**
WL = 18	0.119	-0.0613
24	0.00521	-0.00112
1/128	**ALG 1**	**ALG 2**
WL = 18	0.207	0.318
24	0.00542	-0.00277

Table 5-7. Total Skew Error, 3 Component RSS, RAD
Q = 15, Double Precision, Coning

1/8	ALG 1	ALG 2
WL = 18	0.238-02	0.115-04
24	0.327-04	0.465-06
1/32	**ALG 1**	**ALG 2**
WL = 18	0.110-02	0.172-04
24	0.116-04	0.302-06
1/128	**ALG 1**	**ALG 2**
WL = 18	0.316-03	0.945-05
24	0.387-05	0.324-06

Table 5-8. Total Scale Error, 3 Component RSS, Dimensionless
Q = 15, Double Precision, Coning

1/8	ALG 1	ALG 2
WL = 18	0.570-02	0.193-04
24	0.915-04	0.361-06
1/32	**ALG 1**	**ALG 2**
WL = 18	0.101-02	0.234-04
24	0.210-04	0.144-06
1/128	**ALG 1**	**ALG 2**
WL = 18	0.270-03	0.148-04
24	0.429-05	0.115-06

errors in a general way, it is also necessary to perform an exact simulation of the particular computer to be used before the system design is frozen.

Roundoff error occurs whenever a multiplication or a right shift is performed. The amount of the error depends upon the type of arithmetic and the exact number involved. Two types of arithmetic will be studied: sign-magnitude and 2's complement.

The IBM 7094 Hypocomp simulation uses sign-magnitude arithmetic. Two's complement arithmetic, although not precisely what was used in the previous simulations, is studied as another case of interest. In the previous simulations, hardware and software procedures are used to round the results of multiplications and right shifts to the nearest value representable in the word length. Since this procedure produces a probability density with zero mean, it cannot cause a ramp-drift error, and need not be analyzed.

The effect of the different possible numbers involved will be treated by representing the errors as random processes. Figure 5-1 shows the roundoff characteristic of sign-magnitude arithmetic and Figure 5-2 shows it for 2's complement arithmetic. In both, the rounded

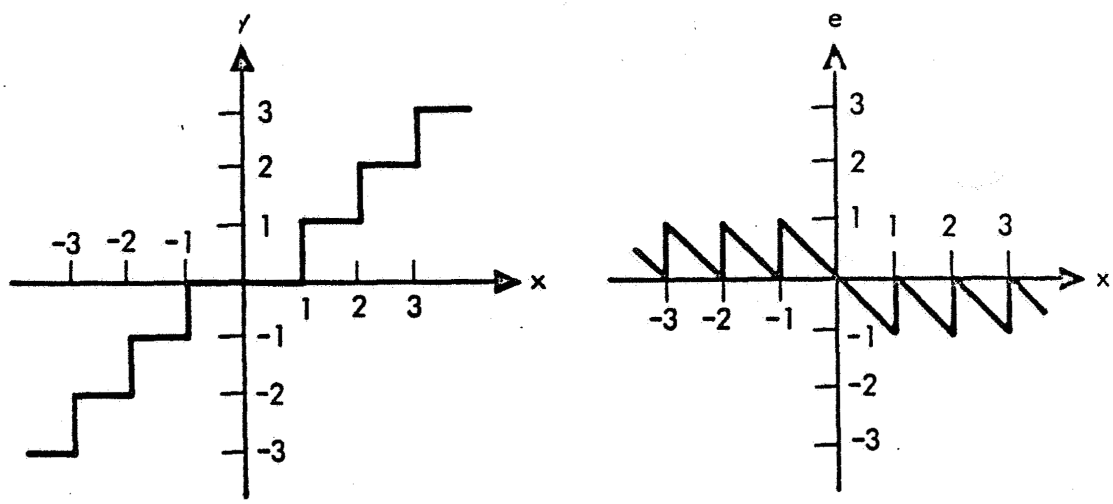


Figure 5-1. Sign-Magnitude Roundoff Error

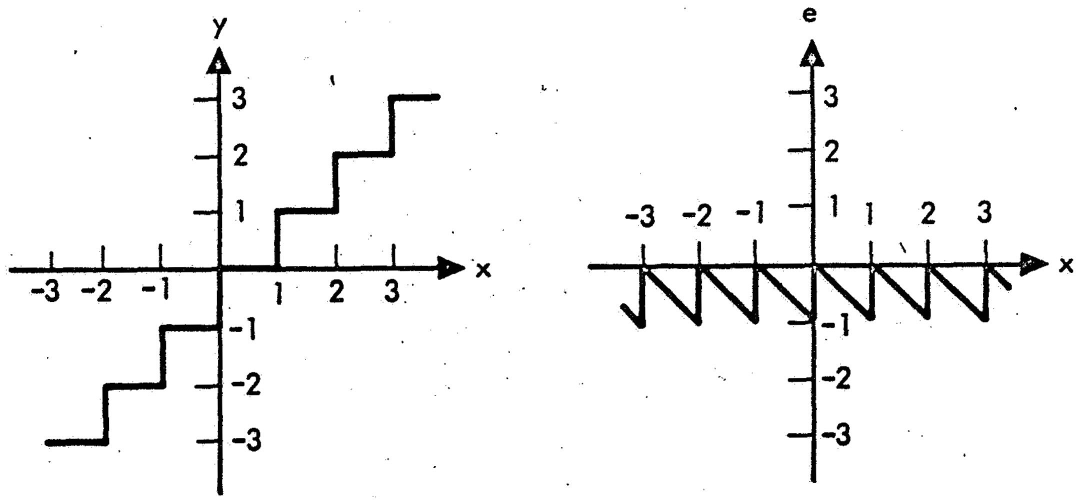


Figure 5-2. 2's Complement Roundoff Error

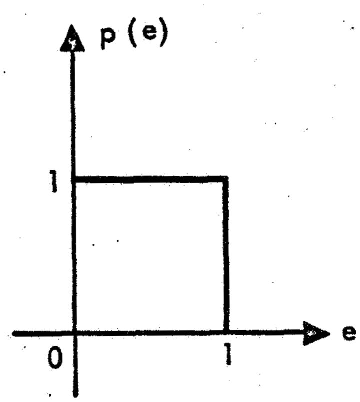


Figure 5-3. Probability Density of ϵ

off bits are simply truncated with no hardware or software procedure to modify the least precise retained bit. In both, figure x represents the original number, y the rounded number, and e the roundoff error, where

$$e = y - x \tag{5-18}$$

If x varies rapidly over its range of values, e will resemble a random process with a uniform probability density and a white power density spectrum. Let ϵ be a random process with the probability density shown in Figure 5-3. Then the sign-magnitude roundoff error may be approximated by

$$e = -2^{-(w-1-s)} \epsilon \, \mathrm{sgn}(x) \tag{5-19}$$

and the 2's complement error by

$$e = -2^{-(w-1-s)} \epsilon \tag{5-20}$$

where

$$\mathrm{sgn}(x) = \begin{matrix} -1 & x < 0 \\ 0 & x = 0 \\ 1 & x > 0 \end{matrix} \tag{5-21}$$

and

w = word length including sign bit
s = scaling of x.

The analysis will be performed for the second-order Taylor series algorithm with double-precision accumulation. The equations for the Δa_{ij} may be expressed as

$$[\Delta a] = [\Delta\alpha][a] \tag{5-22}$$

where

$$[\Delta\alpha] = \begin{bmatrix} -\frac{1}{2}(\Delta\alpha_2{}^2 + \Delta\alpha_3{}^2) & \frac{1}{2}\Delta\alpha_1\Delta\alpha_2 + \Delta\alpha_3 & \frac{1}{2}\Delta\alpha_3\Delta\alpha_1 - \Delta\alpha_2 \\ \frac{1}{2}\Delta\alpha_1\Delta\alpha_2 - \Delta\alpha_3 & -\frac{1}{2}(\Delta\alpha_3{}^2 + \Delta\alpha_1{}^2) & \frac{1}{2}\Delta\alpha_2\Delta\alpha_3 + \Delta\alpha_1 \\ \frac{1}{2}\Delta\alpha_3\Delta\alpha_1 + \Delta\alpha_2 & \frac{1}{2}\Delta\alpha_2\Delta\alpha_3 - \Delta\alpha_1 & -\frac{1}{2}(\Delta\alpha_1{}^2 + \Delta\alpha_2{}^2) \end{bmatrix} \tag{5-23}$$

All of the quadratic terms in Equation (5-23) are subject to roundoff error. Using the notation E(x) for the error in x gives

$$E(\tfrac{1}{2}\Delta\alpha_i\Delta\alpha_j) = e_{ij} = -2^{-(w-1-s)}\epsilon_{ij}\ \mathrm{sgn}(\Delta\alpha_i\Delta\alpha_j); \quad i = 1, 2, 3; \quad j = i, \ldots, 3 \tag{5-24}$$

for sign magnitude and

$$E(\tfrac{1}{2}\Delta\alpha_i\Delta\alpha_j) = e_{ij} = -2^{-(w\ 1\ s)}\epsilon_{ij}; \quad i = 1, 2, 3; \quad j = i, \ldots, 3 \tag{5-25}$$

for 2's complement roundoff; where the ϵ_{ij} are independent random processes similar to ϵ. The error in $[\Delta\alpha]$ is

$$E[\Delta\alpha] = \begin{bmatrix} -e_{22} - e_{33} & e_{12} & e_{13} \\ e_{12} & -e_{33} - e_{11} & e_{23} \\ e_{13} & e_{23} & -e_{11} - e_{22} \end{bmatrix} \tag{5-26}$$

It is seen at once that $E[\Delta\alpha]$ is a symmetric matrix. It, therefore, produces no drift error for either type of algorithm and need be considered no further.

The matrix multiplication in Equation (5-22) must now be analyzed. Let

$$E(\Delta\alpha_{ij}a_{jk}) = e_{ijk} = -2^{-(w-1-s)}\epsilon_{ijk}\,\mathrm{sgn}(\Delta\alpha_{ij}a_{jk});$$

$$i = 1, 2, 3; \quad j = 1, 2, 3; \quad k = 1, 2, 3 \tag{5-27}$$

for sign magnitude and

$$E(\Delta\alpha_{ij}a_{jk}) = e_{ijk} = -2^{-(2-1-s)}\epsilon_{ijk};$$

$$i = 1, 2, 3; \quad j = 1, 2, 3; \quad k = 1, 2, 3 \tag{5-28}$$

for 2's complement roundoff; where the ϵ_{ijk} are independent random processes similar to ϵ. The error in Δa_{ik} is

$$E(\Delta a_{ik}) = \sum_{j=1}^{3} e_{ijk}; \quad i = 1, 2, 3; \quad k = 1, 2, 3 \tag{5-29}$$

Since the double-precision case is being analyzed, no further roundoff errors need be considered and

$$E(a_{ik}) = E(\Delta a_{ik}); \quad i = 1, 2, 3; \quad k = 1, 2, 3 \tag{5-30}$$

Let the direction cosines matrix be considered to be the product of an ideal matrix $[\hat{a}]$ and an error matrix $[\theta]$.

$$[a] = [\hat{a}][\theta] \tag{5-31}$$

The drift errors then may be obtained from $[\theta]$.

$$d_1 = (\theta_{23} - \theta_{32})/2$$
$$d_2 = (\theta_{31} - \theta_{13})/2$$
$$d_3 = (\theta_{12} - \theta_{21})/2 \tag{5-32}$$

The direction cosine matrix may also be expressed by

$$[a] = [\hat{a}] + E[a] \tag{5-33}$$

where E[a] is given by Equation (5-30). In order to determine how the errors propagate with time, consider two update cycles. Ideally:

$$[\hat{a}]_2 = [I] + [\Delta\hat{\alpha}]\ [\hat{a}]_1 \tag{5-34}$$

Actually (ignoring truncation error)

$$[a]_1 = [a]_1 + E[a]_1 \tag{5-35}$$

$$[a]_2 = \left([I] + [\Delta\hat{\alpha}]\right)\left([\hat{a}]_1 + E[a]_1\right) + E[a]_2 \tag{5-36}$$

From Equation (5-31),

$$[\theta] = [\hat{a}]^T[a] \tag{5-37}$$

From Equations (5-34), (5-36), and (5-37)

$$[\theta]_2 = [\hat{a}]_2{}^T[a]_2 = [I] + [\hat{a}]_1{}^T\ E[a]_1 + [\hat{a}]_2{}^T\ E[a]_2 \tag{5-38}$$

Extending Equation (5-38) to the general case, the error after n iterations is

$$[\theta]_n = [I] + \sum_{m=1}^{n} [\hat{a}]_m{}^T\ E[a]_m \tag{5-39}$$

Let

$$[\theta]_n = [I] + \sum_{m=1}^{n} [\beta]_m \tag{5-40}$$

where

$$[\beta]_m = [\hat{a}]_m^T E[a]_m \tag{5-41}$$

Then

$$\beta_{1k} = \sum_{i=1}^{3} \hat{a}_{i1} E(a_{ik}) \tag{5-42}$$

from Equations (5-29) and (5-30),

$$\beta_{1k} = \sum_{i=1}^{3} \hat{a}_{i1} \sum_{j=1}^{3} e_{i,j,k} \tag{5-43}$$

Therefore

$$\beta_{1k} = -2^{-(w-1-s)} \sum_{i=1}^{3} \hat{a}_{i1} \sum_{j=1}^{3} \epsilon_{ijk} \operatorname{sgn}(\Delta\alpha_{ij} a_{jk}) \tag{5-44}$$

for sign-magnitude and

$$\beta_{1k} = -2^{-(w-1-s)} \sum_{i=1}^{3} \hat{a}_{i1} \sum_{j=1}^{3} \epsilon_{ijk} \tag{5-45}$$

for 2's complement. Now consider only the mean error

$$\overline{\epsilon}_{ijk} = \frac{1}{2}; \quad i = 1, 2, 3; \quad j = 1, 2, 3; \quad k = 1, 2, 3 \tag{5-46}$$

and approximate a by $\hat{a}$. Then

$$\overline{\beta}_{1k} = -2^{-(w-s)} \sum_{i=1}^{3} \hat{a}_{i1} \sum_{j=1}^{3} \operatorname{sgn}(\Delta\alpha_{ij} a_{jk}) \tag{5-47}$$

for sign-magnitude and

$$\beta_{1k} = -2^{-(w-s)} \cdot 3 \sum_{i=1}^{3} \hat{a}_{i1} \qquad (5\text{-}48)$$

for 2's complement.

Now consider a three axis ramp where

$$\omega_i = \omega; \qquad i = 1, 2, 3 \qquad (5\text{-}49)$$

and

$$\omega_0 = \left(\sum_{i=1}^{3} \omega_i^{\,2} \right)^{1/2} \qquad (5\text{-}50)$$

Then

$$\hat{a}_{11} = \hat{a}_{22} = \hat{a}_{33} = 1/2 + 2/3 \cos \omega_o t \qquad (5\text{-}51)$$

$$\hat{a}_{12} = \hat{a}_{23} = \hat{a}_{31} = 1/3 - 1/3 \cos \omega_o t + 1/\sqrt{3} \sin \omega_o t \qquad (5\text{-}52)$$

$$\hat{a}_{21} = \hat{a}_{32} = \hat{a}_{13} = 1/3 - 1/3 \cos \omega_o t - 1/\sqrt{3} \sin \omega_o t \qquad (5\text{-}53)$$

If Equations (5-51 through 5-53) are substituted into Equations (5-41) and (5-32) is applied to the result, the drift errors may be calculated. The summation is very tedious, however, so it will be approximated by assuming that there is a very large integer number of samples per cycle of the rotation. Equation (5-40) then becomes

$$[\theta]_n = [I] + n[\bar{\beta}] \qquad (5\text{-}54)$$

where $[\bar{\beta}]$ is the average value of $[\beta]$

$$[\bar{\beta}] = \frac{\omega_0}{2\pi} \int_0^{2\pi/\omega_0} [\beta] dt \qquad (5\text{-}55)$$

Performing the indicated calculations gives the average roundoff-drift error after n cycles as

$$\bar{d}_i = -\frac{7\sqrt{3}}{\pi} \cdot 2^{-(w-s+1)} n \, \mathrm{sgn}(\Delta\alpha_i); \quad i = 1, 2, 3 \qquad (5\text{-}56)$$

for sign magnitude and

$$\bar{d}_i = 0; \quad i = 1, 2, 3 \qquad (5\text{-}57)$$

for 2's complement.

Equation (5-56) is evaluated for several cases and compared to the Hypocomp results

Case	w	$\Delta\alpha$	n	s
I	18	0.01134	6400	-5
II	24	0.01134	6400	-5
III	18	-0.00567	6400	-5
IV	24	-0.00567	6400	-5

Case	Eq. (5-56)	Hypocomp
I	-0.148×10^{-2}	-0.174×10^{-2}
II	-0.231×10^{-4}	-0.272×10^{-4}
III	0.148×10^{-2}	0.151×10^{-2}
IV	0.231×10^{-4}	0.235×10^{-4}

Considering the approximations made in the derivation, the agreement seems quite good.

It should be noted that Equation (5-57) does not imply zero roundoff drift error for the 2's complement case, but merely zero average-drift error for an ensemble of runs. The error will have the character of a random walk, and the rms error for the ensemble will not be zero.

It should be borne in mind that the above results were obtained for a special case and have little, if any, general application.

5.2.7 Conclusions

1) Algorithm 2 offers considerable advantage over algorithm 1.

2) Double-precision accumulation provides a worthwhile decrease in the roundoff error.

3) MSFC quantization is adequate.

4) An algorithm for the control of cummutativity error such as that presented in Appendix I is desirable.

5) Two candidate computers can be selected from the results of Phase I and Phase II studies which offer different performance levels. The first has a sampling rate of 100 cps and a word length of 24 bits, including sign. The second has a sampling rate of 50 cps and a word length of 21 bits, including sign. MSFC quantization was assumed for both.

   The errors may be estimated by combining the results of Phase I (Table 5-11, page 5-51) with the results of this phase. The Phase I results are extrapolated to the 50-cps case, while the Phase II results are interpolated for both 50 and 100 cps. These errors and the rms are shown in Table 5-9.

Table 5-9. GP Computer Selections

Sampling Rate	100	50	cps
Word Length	24	21	bits
Slewing Truncation	0.0115	0.046	deg/hr
Coning Truncation and Commutativity	0.0100	0.040	deg/hr
Roundoff	0.0030	0.029	deg/hr
Quantization	0.0010	0.002	deg/hr
RMS	0.0156	0.068	deg/hr

5.3 VELOCITY TRANSFORMATION SIMULATIONS

5.3.1 Introduction and Summary

The purpose of this simulation was to provide a means for the comparison of several strapdown methods of obtaining the inertial velocity increments from velocity increments measured in body coordinates. Only three methods are simulated for comparison in this study, although other methods could easily be programmed for evaluation.

The velocity transformation methods programmed in this computer simulation are:

1) the velocity increments in vehicle coordinates are converted to inertial coordinates by multiplying by the direction cosines at the end of the velocity accumulation interval

2) the inertial velocity increments are obtained by multiplying the velocity increments in vehicle coordinates by the direction cosines present at the middle of the velocity accumulation interval

3) the inertial velocity increments are obtained by multiplying the velocity increments in vehicle coordinates by the average of the direction cosines present at the start and the end of the velocity accumulation interval.

The direction cosines may be maintained by either a first- or a second-order Taylor series algorithm, depending on the input variable IFC. In the next few sections, the complete simulation is described and includes the input and output format, as well as block diagrams of the acceleration drivers, the vehicle attitude drivers, and the strapdown equations mechanized in the simulation. Only the equation errors are presented. Neither computer quantization errors nor errors associated with the fact that the accelerometers are not all located at the same point were considered.

Of the three velocity transformation methods considered, the best method used the direction cosines present at the middle of the velocity accumulation interval to transform the velocity increments from vehicle coordinates to inertial coordinates. The second best method used the average of the direction cosines present at the start and end of the velocity accumulation interval. The least effective method used the direction cosines present at the start (or the end) of the velocity accumulation interval. The difference between the two best methods was very slight.

5.3.2 Velocity Transformation Simulation Description

The ideal method of computing the inertial velocity increments is to continuously evaluate the following integral

$$\Delta\overline{V}_I = \int_{t_1}^{t_2} \overline{A}_I dt = \int_{t_1}^{t_2} \left[a_{ij}^T\right] \overline{A}_b dt \qquad (5\text{-}58)$$

However, this is not possible as the $\left[a_{ij}\right]$ matrix, which relates the inertial and vehicle body axes, is available only at discrete intervals ($n\,\Delta t$ sec, where Δt is the computer cycle time). This problem is complicated more as the acceleration vector $\overline{A}_b$ is not available. The only information available about $\overline{A}_b$ is $\Delta\overline{V}_b$

$$\Delta\overline{V}_b = \int_{t_1}^{t_2} \overline{A}_b dt \qquad (5\text{-}59)$$

where:

$\Delta\overline{V}_b$ is a velocity vector equal to the velocity increments along the roll, pitch and yaw axes as measured by the strapdown accelerometers.

Thus, in order to design an accurate strapdown navigation system, two things must be accomplished. An accurate attitude reference algorithm must be used for maintaining the direction cosine matrix. Also, a method must be developed to accurately compute the correct answer to Equation (5-58). This simulation provides a means of comparing solutions to Equation (5-58), and mechanizes the three possible solutions shown below.

$$\Delta \overline{V}_I = \left[a_{ij}^T\right]_{(n\Delta t)} \Delta \overline{V}_b \tag{5-60}$$

$$\Delta \overline{V}_I = \left[a_{ij}^T\right]_{(n+.5)\Delta t} \Delta \overline{V}_b \tag{5-61}$$

$$\Delta V_I = \left[\left[a_{ij}^T\right]_{(n-1)\Delta t} + \left[a_{ij}^T\right]_{n\Delta t}\right] \frac{\Delta \overline{V}_b}{2} \tag{5-62}$$

A complete block diagram of the simulation is shown in Figure 5-4. In Paragraphs 5. 3. 1. 1 through 5. 3. 1. 6 the equations used in each block and the input variables required for their mechanization are described. The program's input format is described in Paragraph 5. 3. 1. 7, and the output format in Paragraph 5. 3. 1. 8.

5. 3. 2. 1 Variable Initialization

The first operation is to initialize the variables in the simulation which are not a part of the input format. These variables, such as time and M, may not have the correct values from the results of a previous run.

5. 3. 2. 2 Vehicle Attitude Driver

The vehicle attitude driver has two options; the option desired being designated by the fixed point input variable IFA. The two options are shown in Figure 5-5 for IFA = 1, and in Figure 5-6 for IFA = 2. When IFA = 1, the vehicle attitude may be (1) slewed at a constant angular velocity, (2) slewed at a constant angular acceleration, or (3) put under the influence of a sinusoidal vibration of any frequency, phase shift or amplitude. When IFA = 2, the vehicle attitude may be slewed at a constant angular velocity or put under the influence of square wave angular vibrations.

Thus, the attitude drivers may perform the following tests:

1) vehicle coning at any arbitrary amplitude, frequency or phasing

2) vibration tests - in phase sinusoidal motions

3) constant slewing tests

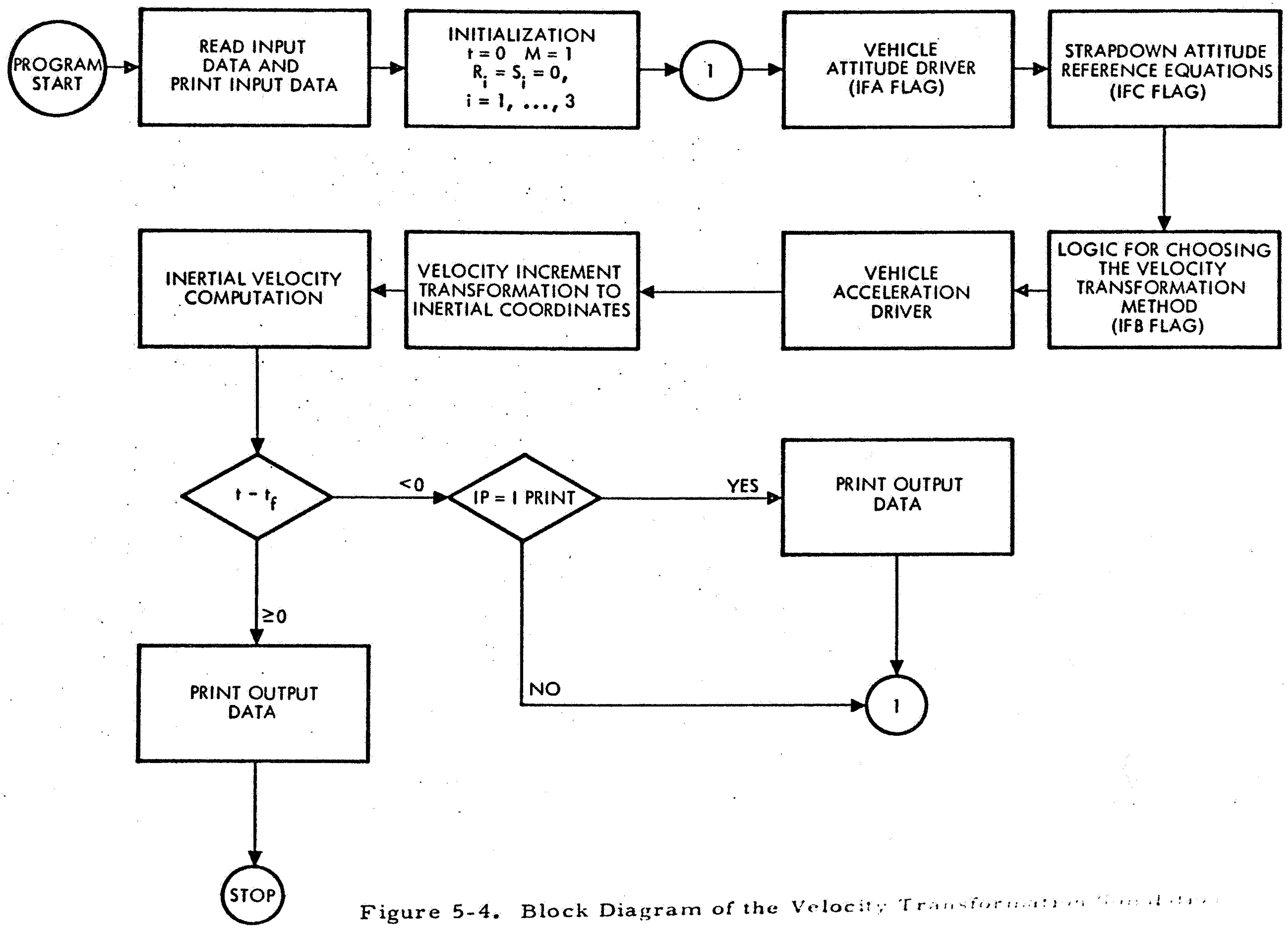


Figure 5-4. Block Diagram of the Velocity Transformation Simulation

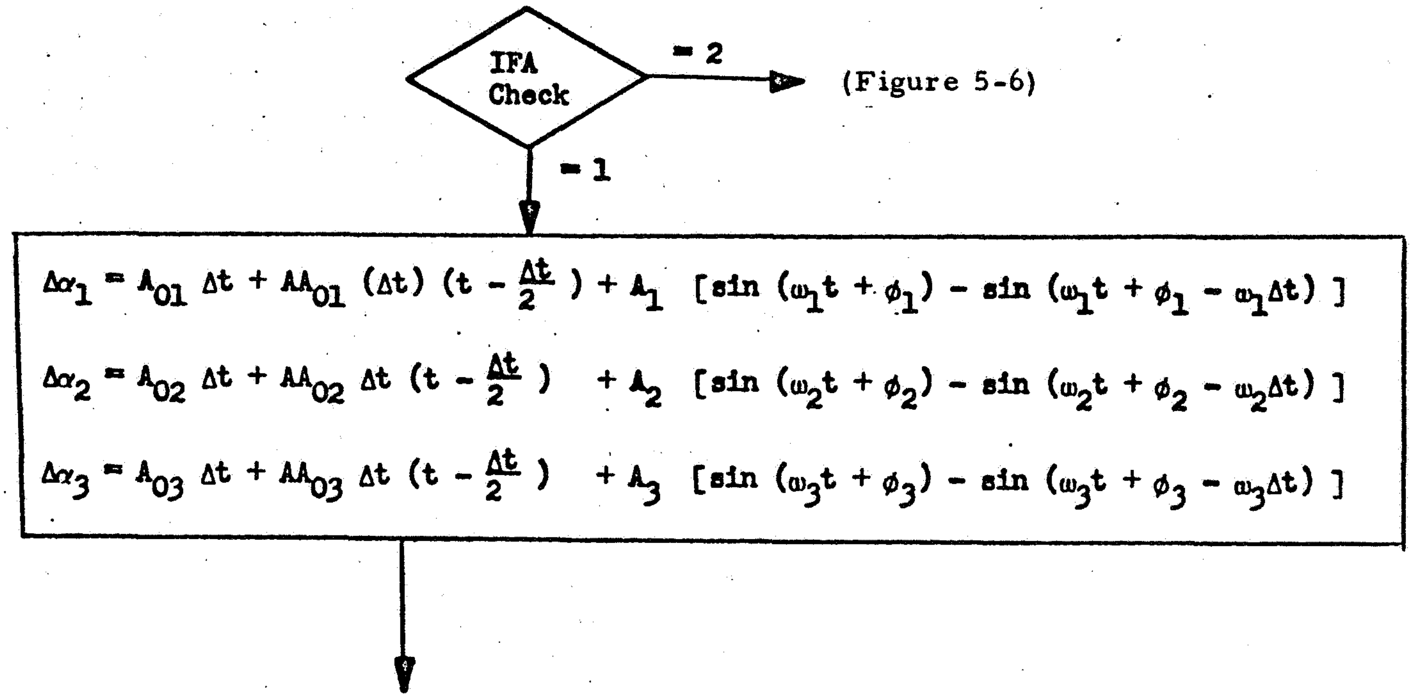


Figure 5-5. Vehicle Attitude Driver (IFA=1)

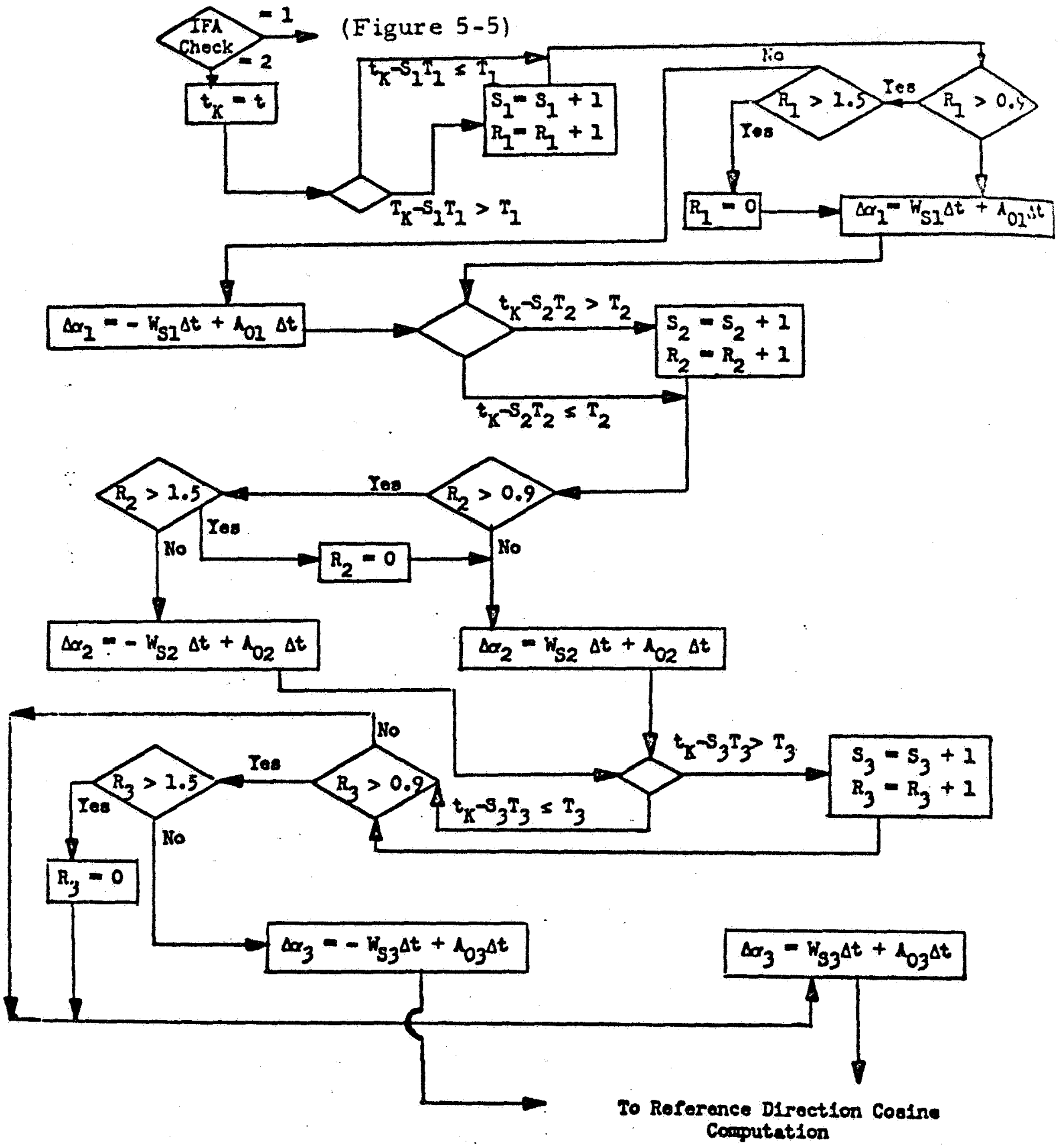


Figure 5-6. Vehicle Attitude Driver (IFA=2)

4) constant angular acceleration tests

5) any combination of (1) - (4)

6) square wave vibration tests

7) a combination of (3) and (6)

The outputs of the vehicle attitude driver are in the form of angular rotation increments ($\Delta\alpha_1$, $\Delta\alpha_2$, $\Delta\alpha_3$) the body mounted gyros would experience in each DDA computation interval Δt.

5.3.2.3 Reference Vehicle Attitude Computation

The reference vehicle attitude computation is performed using a second-order Taylor's series algorithm. This computation method is shown below in matrix notation as it is programmed in the scientific simulation, Equation (5-59). This mechanization is based on the $\left[a_{ij}\right]$ matrix relationship shown in Equation (5-63).

$$\begin{bmatrix} X_b \\ Y_b \\ Z_b \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} X_I \\ Y_I \\ Z_I \end{bmatrix} \tag{5-63}$$

where

X_b, Y_b, Z_b are the vehicle body axes

X_I, Y_I, Z_I are the inertial reference axes

$\left[a_{ij}\right]$ is an orthonormal matrix relating the two systems

$$\left[a_{ij}\right]_{n+1} = \begin{bmatrix} \left[1 - 0.5(\Delta\alpha_2^2 + \Delta\alpha_3^2)\right] & \left[\Delta\alpha_3 + 0.5\ \Delta\alpha_2\Delta\alpha_1\right] & \left[-\Delta\alpha_2 + 0.5\ \Delta\alpha_1\Delta\alpha_3\right] \\ \left[-\Delta\alpha_3 + 0.5\ \Delta\alpha_1\Delta\alpha_2\right] & \left[1 - 0.5(\Delta\alpha_1^2 + \Delta\alpha_3^2)\right] & \left[\Delta\alpha_1 + 0.5\ \Delta\alpha_2\Delta\alpha_3\right] \\ \left[\Delta\alpha_2 + 0.5\ \Delta\alpha_1\Delta\alpha_3\right] & \left[-\Delta\alpha_1 + 0.5\ \Delta\alpha_2\Delta\alpha_3\right] & \left[1 - 0.5(\Delta\alpha_1^2 + \Delta\alpha_2^2)\right] \end{bmatrix} \left[a_{ij}\right]_n \tag{5-64}$$

5.3.2.4 Velocity Transformation Logic

There are three methods for transforming the velocity increments measured by the accelerometers to inertial coordinates. These three methods are all similar in one respect; the output velocity increments are multiplied by a 3 x 3 orthonormal matrix. In the first method (IFB = 1) the orthonormal matrix used is the matrix present at the end of the velocity accumulation interval; in the second case (IFB = 2), the direction cosines present at the middle of the velocity accumulation interval are used; and in the third case (IFB = 3), the orthonormal matrix is obtained by averaging the direction cosines present at the start and the end of the velocity accumulation interval. The method used to obtain these matrices is shown in Figure 5-7. In order to use these three methods easily, a dummy matrix $\left[aP_{ij}\right]$ is defined, and it is this matrix which is used to multiply the velocity increments accumulated by the accelerometers in vehicle coordinates.

5.3.2.5 Vehicle Acceleration Driver

The vehicle thrust accelerations simulated in this computer simulation are the accelerations which would be experienced by the accelerometers mounted along the X_b, Y_b, and Z_b axes ($X_b \times Y_b = Z_b$). The accelerations simulated are shown in Equation (5-65). These accelerations include

$$
\begin{aligned}
A_{xb} &= A_{1x} + A_{2x}t + A_{3x}t^2 \\
A_{yb} &= A_{1y} + A_{2y}t + A_{3y}t^2 \\
A_{zb} &= A_{1z} + A_{2z}t + A_{3z}t^2
\end{aligned}
\tag{5-65}
$$

where

A_{xb}, A_{yb}, A_{zb} are the accelerations along the x_b, y_b, z_b vehicle axes

A_{1x}, A_{1y}, A_{1z} are constant accelerations along the x_b, y_b, z_b axes

A_{2x}, A_{2y}, A_{2z} are ramp accelerations along the x_b, y_b, z_b axes

A_{3x}, A_{3y}, A_{3z} are parabolic accelerations along the x_b, y_b, z_b axes

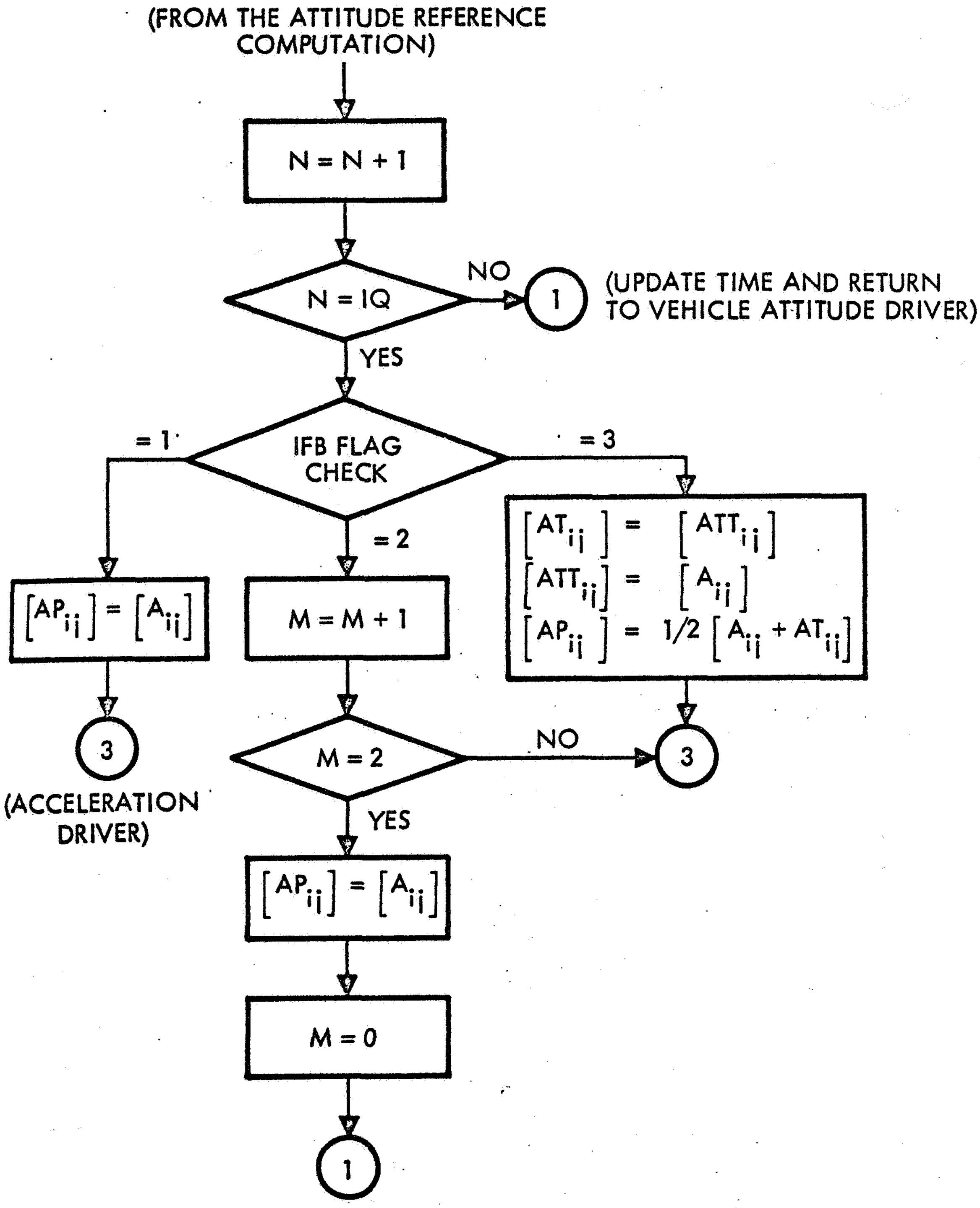


Figure 5-7. Velocity Transformation Logic

constant, ramp, and parabolic accelerations along any or all of the vehicle body axes. Since the accelerations are computed with a second order polynomial, simple acceleration profiles may be simulated. The equations mechanized in the simulation are shown in Figure 5-8.

5.3.2.6 Velocity Increment Transformation to Inertial Coordinates

The body mounted accelerations are simulated along the x_b, y_b, and z_b body axes, and experience the accelerations described by Equation (5-65). Thus, the velocity increments measured by the accelerometers are

$$\Delta\overline{V}_b = \Delta V_{xb}\hat{X}_b + \Delta V_{yb}\hat{Y}_b + \Delta V_{zb}\hat{Z}_b \qquad (5\text{-}66)$$

where

$\hat{X}_b, \hat{Y}_b, \hat{Z}_b$ are unit vectors along the X_b, Y_b, Z_b vehicle body axes.

$$\Delta V_{xb} = \int_{t_1}^{t_2} A_{xb}dt, \quad \Delta V_{yb} = \int_{t_1}^{t_2} A_{yb}dt, \quad \Delta V_{zb} = \int_{t_1}^{t_2} A_{zb}dt$$

$t_2 - t_1$ is the velocity update cycle time in sec.

Since the body axes are related to the inertial axes by the a_{ij} orthonormal matrix,

$$\begin{bmatrix} x_b \\ y_b \\ z_b \end{bmatrix} = \begin{bmatrix} a_{ij} \end{bmatrix} \begin{bmatrix} x_I \\ y_I \\ z_I \end{bmatrix} \qquad (5\text{-}67)$$

the inertial velocity increments are obtained by the following equations.

$$\begin{aligned} \Delta V_{xI} &= \Delta V_{xb}a_{11} + \Delta V_{yb}a_{21} + \Delta V_{zb}a_{31} \\ \Delta V_{yI} &= \Delta V_{xb}a_{12} + \Delta V_{yb}a_{22} + \Delta V_{zb}a_{32} \\ \Delta V_{zI} &= \Delta V_{xb}a_{13} + \Delta V_{yb}a_{23} + \Delta V_{zb}a_{33} \end{aligned} \qquad (5\text{-}68)$$

FROM THE VEHICLE ATTITUDE DRIVER

$$\Delta V_{xb} = \Delta t \left[A1X + A2X\,(t - \Delta t/2) + A3X\,(t^2 - t\Delta t + \Delta t^2/3) \right]$$

$$\Delta V_{yb} = \Delta t \left[A1Y + A2Y\,(t - \Delta t/2) + A3Y\,(t^2 - t\Delta t + \Delta t^2/3) \right]$$

$$\Delta V_{zb} = \Delta t \left[A1Z + A2Z\,(t - \Delta t/2) + A3Z\,(t^2 - t\Delta t + \Delta t^2/3) \right]$$

$$VXI = \Delta V_{xb}\, a_{11} + \Delta V_{yb}\, a_{21} + \Delta V_{zb}\, a_{31}$$

$$VYI = \Delta V_{xb}\, a_{12} + \Delta V_{yb}\, a_{22} + \Delta V_{zb}\, a_{32}$$

$$VZI = \Delta V_{xb}\, a_{13} + \Delta V_{yb}\, a_{23} + \Delta V_{zb}\, a_{33}$$

REFERENCE VELOCITY COMPUTATION

TO THE DDA MECHANIZATION
EQUATIONS

Figure 5-8. Vehicle Acceleration Driver

In the computer simulation, a dummy matrix aP_{ij} is used instead of the a_{ij} matrix so any of the three velocity transformation methods may be called out.

The inertial velocity is then computed by the below equations.

$$
\begin{aligned}
V_{xI} &= V_{xI} + \Delta V_{xI} \\
V_{yI} &= V_{yI} + \Delta V_{yI} \qquad (5\text{-}69) \\
V_{zI} &= V_{zI} + \Delta V_{zI}
\end{aligned}
$$

5.3.2.7 Program Input Variables

DELT	Attitude Reference Matrix update cycle time in sec.
IFA	Fixed-point flag for the attitude driver. When IFA = 1, a constant, ramp, and sinusoidal angular velocity may be applied to any or all of the three vehicle axes. When IFA = 2, a square wave angular velocity motion is applied to any of the vehicle axes.
A_{o1}, A_{o2}, A_{o3}	Constant angular velocities in rad/sec experienced by the vehicle about the x_b, y_b, and z_b vehicle axes.
AA_{o1}, AA_{o2}, AA_{o3}	Constant angular accelerations in rad/sec/sec experienced by the vehicle about the x_b, y_b, z_b vehicle axes.
A_1, A_2, A_3	The amplitudes in rad of the sinusoidal rotations experienced about the x_b, y_b, z_b vehicle axes, respectively.
W_1, W_2, W_3	The frequencies in rad/sec of the sinusoidal rotations about the x_b, y_b, z_b vehicle axes, respectively.
PHI1, PHI2, PHI3	The phase shift in rad of the sinusoidal rotations experienced about the x_b, y_b, z_b vehicle axes, respectively.
IFC	Fixed-point flag for the attitude reference equations used to maintain the direction cosine matrix. When IFC is 1, a second-order Taylor's series algorithm is used; when IFC is 2, a first-order Taylor's series algorithm is used.

IQ	A fixed-point input which is used in the logic for using the desired velocity transformation equations. This input allows the attitude reference equations to be updated faster than the velocity transformation equations are used. If IFB = 1 or 3, the velocity increments are transformed to inertial coordinates every (IQ) (DELT) sec and DELT1 (described below) must be input as this value. If IFB = 2, the velocity increments are transformed to inertial coordinates every 2 (IQ) (DELT) sec and DELT1 must be input as this value.
IFB	A fixed-point flag which calls out the desired velocity transformation driver. When IFB = 1, the velocity increments are transformed to inertial coordinates with the direction cosines present at the end of the velocity accumulation interval. When IFB = 2, the direction cosines present at the middle of the velocity accumulation interval are used for the velocity increment transformation. When IFB = 3, the velocity increments are transformed to inertial coordinates by using the average of the direction cosines present at the start and end of the velocity accumulation interval.
A1x, A1y, A1z	Constant accelerations in ft/ sec^2 along the x_b, y_b, z_b vehicle axes, respectively.
A2x, A2y, A2z	Ramp accelerations in ft/ sec^3 along the x_b, y_b, z_b vehicle axes, respectively.
A3x, A3y, A3z	Parabolic accelerations in ft/ sec^4 along the x_b, y_b, z_b vehicle axes, respectively.
DELT1	The integration interval for the vehicle accelerations and the time interval for the transformation of the velocity increments to inertial coordinates. As described in the description of the input variable IQ, the input DELT1 depends for its value on the input values of DELT and IQ.
IPRINT	IPRINT is a fixed-point input which causes the output data to be printed every (IPRINT) (DELT1) sec.
TF	The program stops when time is greater than TF sec.
WS1, WS2, WS3	The input rotational velocities for the square wave angular motions applied to the x_b, y_b, z_b vehicle axes, respectively. These inputs are required only when IFA = 2.
T1, T2, T3	One half the period in seconds of the square wave angular velocities applied to the x_b, y_b, z_b vehicle axes, respectively.

5.3.2.8 Program Output Variables

TIME	Time in seconds
A	The three by three orthonormal attitude reference matrix.
AP	A three by three orthonormal attitude reference matrix. When IFB = 1, A is identical with AP. When IFB = 2, AP lags A by DELT1/2 sec. When IFB = 3, AP is the average of the A matrix and the A matrix computed DELT1 seconds earlier.
VX1, VY1, VZ1	The inertial velocities computed by the strapdown system along the XI, YI, ZI inertial axes.
VxB, VyB, VzB	The integral of the vehicle accelerations along the x_b, y_b, z_b vehicle body axes. This output is included only to checkout the acceleration driver.

5.3.2.3 Velocity Transformation Results

Several tests were performed on the three sets of velocity transformation equations to determine which method was the most accurate. These tests were performed on the 7094 computer and were compared to results obtained from hand computed analysis. The errors of the three methods compared to hand computed results are shown in the next few tables.

From the tables, the best methods are apparently method B which uses the direction cosines present at the middle of the velocity accumulation interval, and method C which uses the average of the direction cosines present at the start and end of the velocity accumulation interval. The difference between the accuracy of the two methods is very slight, and is not involved in the decision as to which method should be used. The choice of method B or C depends entirely on the computer, the available computer space, the computer speed, etc.

Method A, which uses the direction cosines present at the end of the velocity accumulation interval, should only be used under extreme computer storage limitations, or in a very fast computer, as it is far inferior to method B or C. The effects of computer roundoff were not included in these results and must be considered before the final choice of the velocity transformation may be made.

Table 5-10. Velocity Transformation Results

$\bar{A} = (96 + 20t)\hat{Y}_b$ ft/sec^2, $\omega = 5$ arc deg/sec $\hat{X}_b$

t (sec)	Method	Δt (msec)	Δt_I (msec)	$\Delta \dot{Y}_I$ (fps)	$\Delta \dot{Z}_I$ (fps)
5	A	5	10	0.799	-0.305
	B	5	10	0.037	0.014
10	A	5	10	0.426	-0.709
	B	5	10	0.019	0.015
15	A	5	10	1.12	-1.0
	B	5	10	0.044	0.061
20	A	5	10	2.09	-0.947
	B	5	10	0.054	0.151

Table 5-11. Velocity Transformation Results

$\bar{A} = (96 + 10t^2)\hat{Y}_b$ ft/sec^2, $\bar{\omega} = 5\hat{X}_b$ arc deg/sec

t (sec)	Method	Δt (msec)	Δt_I (msec)	$\Delta \dot{Y}_I$ (fps)	$\Delta \dot{Z}_I$ (fps)
5	A	5	10	0.108	-0.372
	B	5	10	0.0525	0.0022
10	A	5	10	1.094	-1.46
	B	5	10	0.0488	0.0443
15	A	5	10	4.47	-2.82
	B	5	10	0.155	0.271
20	A	5	10	11.37	2.297
	B	5	10	0.174	0.951

Table 5-12. Velocity Transformation Results

$\bar{A} = 96\ \hat{Y}_b\ \text{ft/sec}^2,\ \bar{\omega} = 5\ \text{arc deg/sec}\ \hat{X}_b$

Time (sec)	Transformation Method	Δt (msec)	Δt_1 (msec)	$\Delta \dot{X}_I$ (fps)	$\Delta \dot{Y}_I$ (fps)	$\Delta \dot{Z}_I$ (fps)
5	A	5	10	0.0	0.0472	-0.202
		10	20	0.0	0.0918	-0.405
	B	10	20	0.0	0.00175	0.00055
	C	5	10	0.0	0.00222	0.00074
10	A	5	10	0.0	0.1802	-0.3618
		10	20	0.0	0.3497	-0.7309
		20	40	0.0	0.6897	-1.469
	B	10	20	0.0	0.00652	0.00433
	C	5	10	0.0	0.00873	0.00591
		10	10	0.0	0.015	0.0015
15	A	5	10	0.0	0.3721	-0.4451
		10	20	0.0	0.7236	-0.9135
	B	10	20	0.0	0.0118	0.0134
	C	5	10	0.0	0.0162	0.0185
		10	10	0.0	0.05	0.066
20	A	5	10	0.0	0.5842	-0.4352
		10	10	0.0	1.1412	-0.9174
		20	20	0.0	2.2614	-1.8777
	B	10	10	0.0	0.0143	0.0274
	C	5	5	0.0	0.0207	0.0377

5.4 DDA SIMULATION AND ANALYSIS

5.4.1 Introduction and Summary

This subsection describes the DDA interpretive simulation and presents the results of both the simulation runs and hand analysis. Hand analysis was required because of the prohibitive cost of running the DDA simulation for long intervals of time. Real time in seconds converted approximately to simulation time (7094 time) in minutes by virtue of the high iteration rate involved in a DDA.

The algorithm mechanized in the DDA simulation was a first-order Taylor's series for attitude reference maintenance and velocity transformation. Since the DDA is a serial device, it is necessary to use the direction cosines present at the end of a velocity accumulation interval for velocity transformation. It will be shown that this fact implies improved velocity resolution from the measuring instrument with increased computer speed in order to improve accuracy.

The accuracy results obtained for the DDA which mechanizes a simple navigation system are presented in this section for several velocity pulse quantizations, gyro pulse quantizations, and DDA computer speeds. These results were obtained both by computer simulation of the DDA and from hand analyses. The DDA selected, based on these results, uses a speed of 10,000 cps with gyro quantization of 2^{-16} rad, and velocity pulse quantization of 0.0125 M/sec. This design apparently will cause a 0.5 M/sec error (3σ) for a velocity change of 11,000 M/sec.

5.4.2 DDA Operation

DDA's are organized in a manner similar to numerical digital computers, that is, they contain input-output, memory, control, and arithmetic sections. The same type of equipment is used; the sequence of operations is governed by a program stored in the memory (or a program is permanently wired in instead of having a variable program capability)*, and the computer exercises but one order at a time. However,

*Some DDA's are also mechanized with a delay line technique. All serial DDA machines operate in the manner used in the simulation regardless of the DDA control method mechanized.

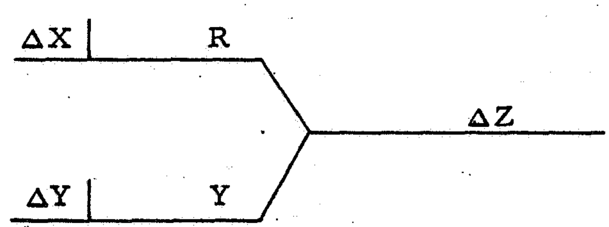
the only two orders available in a DDA are integrate and add. The variable memory is divided into words called integrators.

Each DDA integrator consists of two registers which are called the y register and the R register (see the DDA Integrator diagram below). Each integrator has two inputs, ΔX and ΔY, each of which accepts incremental signals in which a 1 represents a one increment increase in the variable, a minus 1 represents a one increment decrease in the variable and a 0 represents no change. The ΔY bits are summed in the Y register. The ΔX controls the operation of the integrator as follows: when ΔX is 1, the y register is added to the R register, when ΔX is minus 1, the y register is subtracted from the R register, and when ΔX is zero, nothing occurs. The output of the integrator is an incremental signal ΔZ, which has the value 1 whenever the R register overflows in the positive direction, minus 1 whenever the R register overflows in the negative direction, and zero if no overflow occurs. The ΔZ signal represents the results of integration to an accuracy consistent with the inputs. The ΔZ bits can be summed in another integrator, giving the results

$$\Delta Z = y\Delta X$$

$$Z = \Sigma\, y\Delta x \approx \int y\Delta X$$

A DDA integrator is shown below in the standard form, as will be used in this discussion.



DDA Integrator Symbol

For example, a DDA integrator operates in the following manner:

1) $Y = Y + \Delta Y$

2) if ΔX is a positive pulse,
$R = R + Y$

if ΔX is a negative pulse,
$R = R - Y$

if ΔX is zero,
R remains unchanged

3) if $R \geq 1.0$,
$\Delta Z = +1$ pulse
$R = R - 1.0$

if $R \leq -1.0$
$\Delta Z = -1$ pulse
$R = R + 1.0$

if $|R| < 1.0$,
$\Delta Z = 0$
R remains unchanged.

The DDA computer performs all of these operations for each integrator in a definite sequence. However, this sequence is always the same once the DDA is assembled. The primary advantage of the DDA is that it is capable of performing these operations on several integrators at a very fast speed.

5.4.3 DDA Scaling

As described in the previous section, the output of a DDA integrator is ΔZ where

$$\Delta Z = Y \Delta X$$

Thus, the scaling of the output of a DDA integrator is equal to the product of the scaling of the Y register and the scaling of the input variable ΔX. For instance, the scaling of a ΔZ pulse is 10 rad when the scaling of the Y register is 1.0 and the scaling of the ΔX input is 10 radians per bit.

In the DDA design characteristics shown in Figure 5-9, the scaling of the integrators used for the direction cosine maintenance is $\Delta X = 2^{-q}$ rad, $\Delta Z = 2^{-q}$ rad, and the Y register is scaled at 1.0 with the least significant bit scaled at 2^{-q} rad (q is the number of bits in the Y and R registers). The scaling of the integrators used for the velocity transformation is $\Delta X = K_1$ feet/second, $\Delta Z = K_1$ feet/second, and the scaling of the Y register is 1.0 with the least significant bit scaled at 2^{-q} rad. The scaling of the accelerometer and gyro output pulses is also K_1 ft/sec and 2^{-q} rad, corresponding to the scaling of the ΔX inputs used in the integrators.

The accuracy of incremental information is limited to the increment value or quantization of the input signal. This limitation is no different than the quantization of binary number representation. However, there is a link between accuracy and speed which results in a fundamental limitation to incremental information. The maximum rate of change which can occur in an incremental variable is equal to the bit rate times the incremental value. As an example, suppose each bit represents 0.1 ft/sec and the maximum rate at which bits can be transmitted is 1000 bits/sec. Then the maximum accelerations which can be tolerated is 100 ft/sec^2 or about 3 g's. If higher accelerations were experienced, this information would be lost.

The desired DDA design for the MSFC body mounted study is one which can maintain an accurate navigation system when under the influence of accelerations as high as 10 g's and rotation rates per axis as large as one rad/sec. In Figure 5-9, the quantization or scaling of the DDA's least significant bit is plotted as a function of the computer speed. As is evident from this graph, the computer speed increases inversely proportional to the DDA quantization in order to measure the same vehicle dynamics.

For the MSFC study, the high rates and accelerations are present only for a short interval of time, and so it appears as if the DDA could be designed for rotation rates and accelerations smaller than the expected maximum values if a buffer register is used to accumulate the outputs of the inertial instruments. In this manner, no information will be lost if the design rate is exceeded - provided this high rate is not continued long enough to completely fill the buffer registers. However, even though

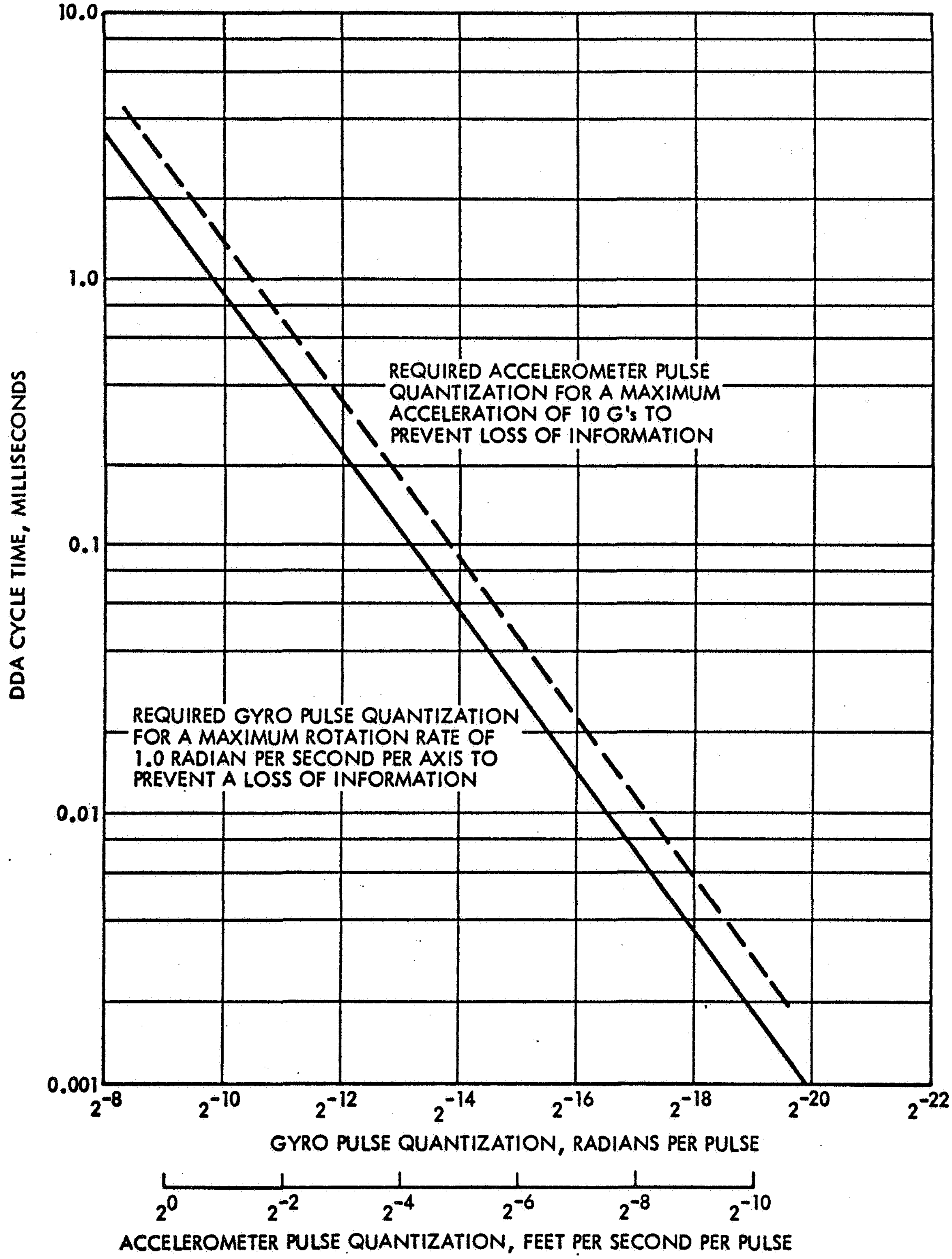


Figure 5-9. DDA Gyro and Accelerometer Pulse Quantization Versus DDA Computer Cycle Time to Prevent Loss of Information

information is not lost, errors will occur due to a time lag in processing the information.

The DDA designs tested on the scientific computer simulation are shown below in Table 5-13.

5.4.4 DDA Simulation Description

A block diagram of the complete simulation is shown in Figure 5-10. In paragraphs 5.4.4.1 through 5.4.4.5 the equations used in each block, and the input variables required for their mechanization are described.

5.4.4.1 Variable Initialization

The first operation of the program is to initialize the variables used in the simulation which are not a part of the input format. These variables, such as time, may not have the correct values from the results of a previous run.

5.4.4.2 Vehicle Attitude Driver

The vehicle attitude driver has two options: the option desired being designated by the fixed point input variable IFA. The two options are shown in Figure 5-11 for IFA = 1, and in Figure 5-12 for IFA = 2. When IFA = 1, the vehicle attitude may be (1) slewed at a constant angular velocity, (2) slewed at a constant angular acceleration, or (3) put under the influence of a sinusoidal vibration of any frequency, phase shift or amplitude. When IFA = 2, the vehicle attitude may be slewed at a constant angular velocity or put under the influence of square wave angular vibrations.

Thus, the attitude drivers may perform the following tests:

(1) Vehicle coning at any arbitrary amplitude, frequency, or phasing

(2) Vibration tests - in phase sinusoidal motions

(3) Constant slewing tests

(4) Constant angular acceleration tests

(5) Any combination of (1) through (4)

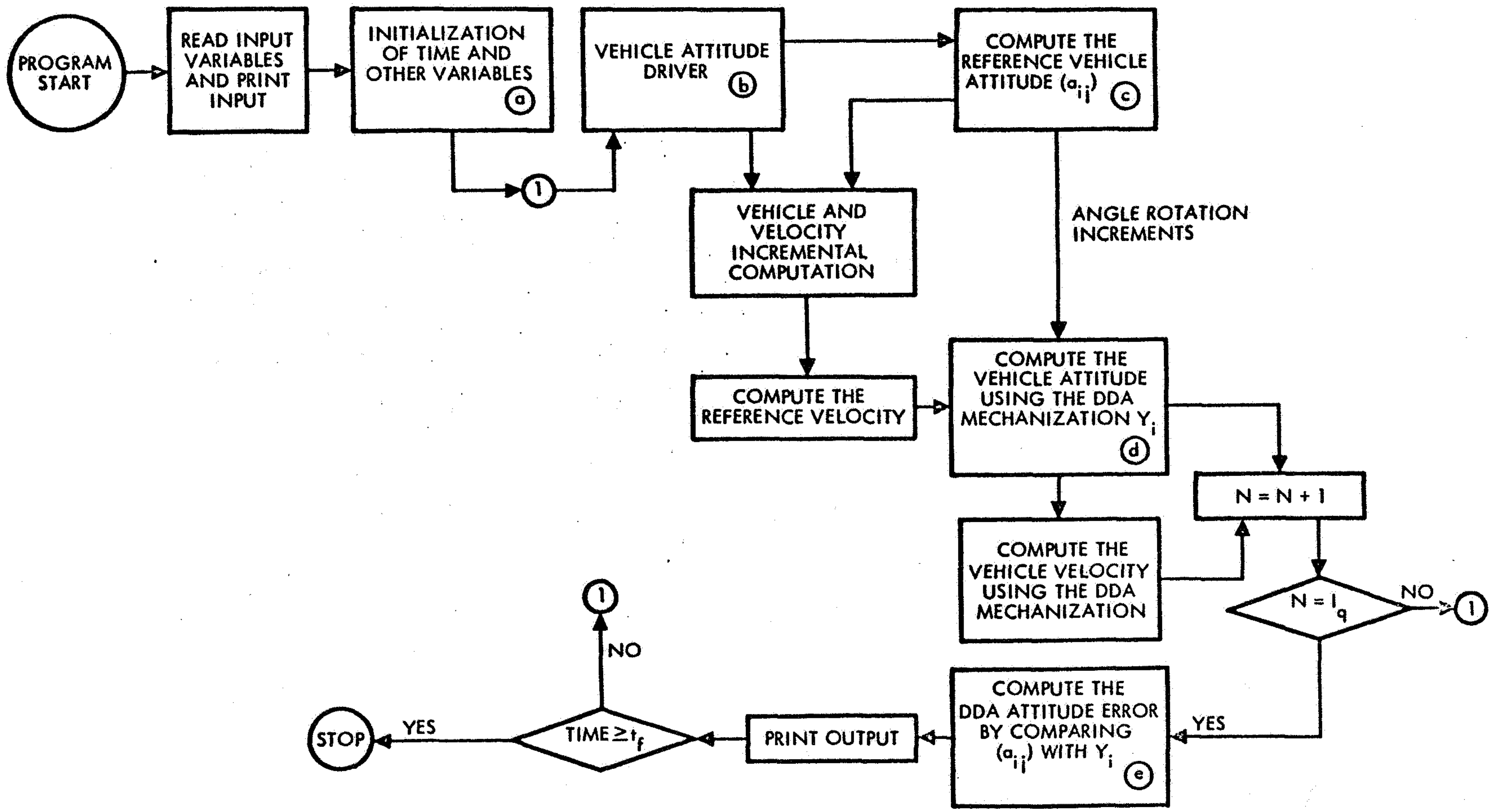


Figure 5-10. Block Diagram of the DDA Simulation

IFA Chock

= 2 → (Figure 5-12)

= 1 ↓

$$\Delta\alpha_1 = A_{01}\,\Delta t + AA_{01}\,(\Delta t)\,(t - \frac{\Delta t}{2}) + A_1\,[\sin(\omega_1 t + \phi_1) - \sin(\omega_1 t + \phi_1 - \omega_1 \Delta t)]$$

$$\Delta\alpha_2 = A_{02}\,\Delta t + AA_{02}\,\Delta t\,(t - \frac{\Delta t}{2}) + A_2\,[\sin(\omega_2 t + \phi_2) - \sin(\omega_2 t + \phi_2 - \omega_2 \Delta t)]$$

$$\Delta\alpha_3 = A_{03}\,\Delta t + AA_{03}\,\Delta t\,(t - \frac{\Delta t}{2}) + A_3\,[\sin(\omega_3 t + \phi_3) - \sin(\omega_3 t + \phi_3 - \omega_3 \Delta t)]$$

↓

To Reference Direction Cosine Computation

Figure 5-11. Vehicle Attitude Driver (IFA=1)

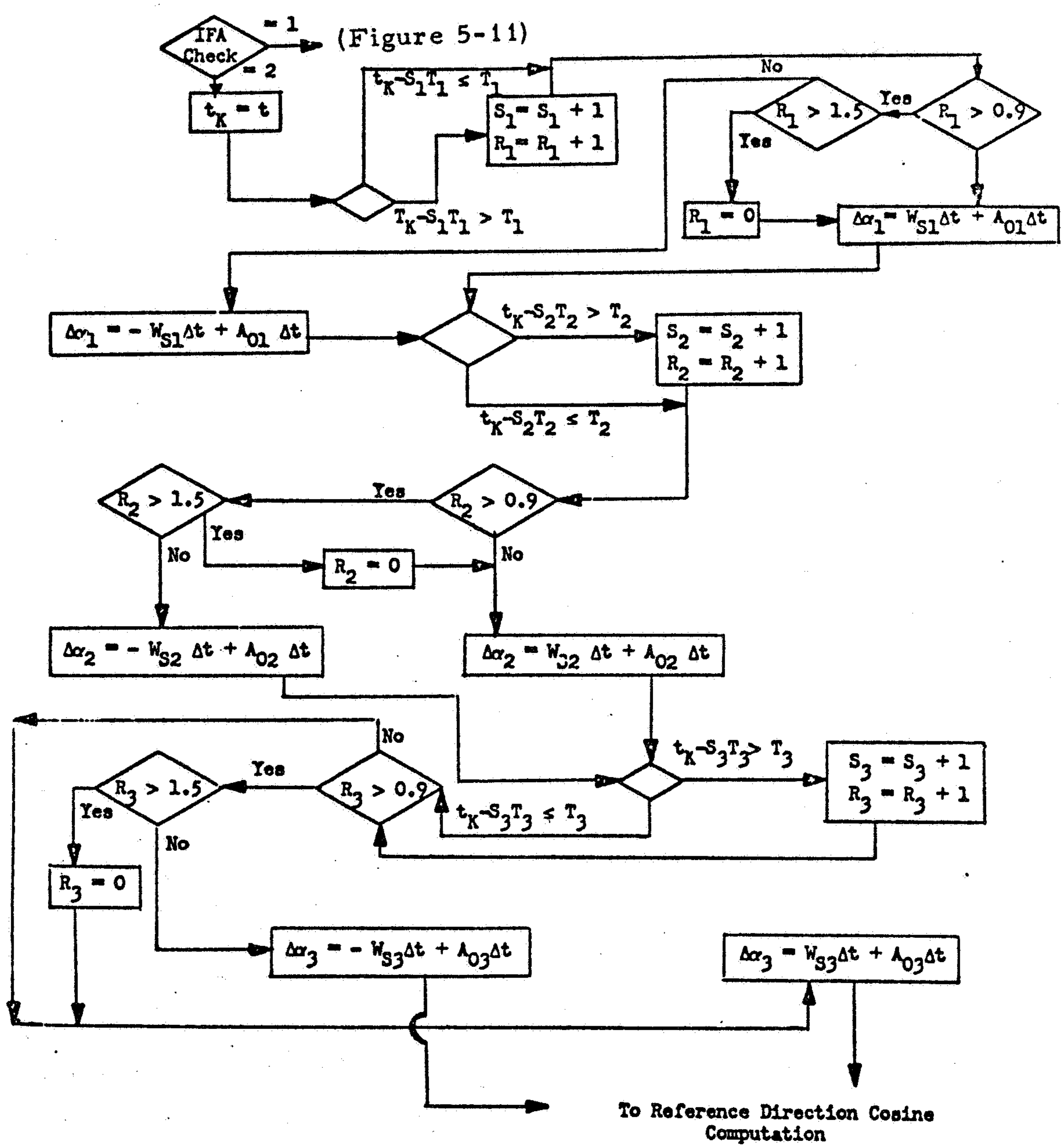


Figure 5-12. Vehicle Attitude Driver (IFA=2)

Table 5-13. DDA Designs Tested

Computer Quantization FK (rad)	Velocity Pulse Quantization CK2 (ft/sec)	Time per Computer Cycle (msec)	Max. Design Acceleration (ft/sec^2)	Max. Design Rotational Velocity (per axis) (deg/sec)	Equivalent* DDA Word Length (bits)
9.765625×10^{-4} (2^{-10})	0.322	1.0	322	55.95	12
9.765625×10^{-14} (2^{-10})	0.322	2.0	161	27.97	12
9.765625×10^{-4} (2^{-10})	0.322	5.0	80.5	11.19	12
$2.44140625 \times 10^{-4}$ (2^{-12})	0.0805	0.25	322	55.95	14
$2.44140625 \times 10^{-4}$ (2^{-12})	0.16404166	1.25	131	11.19	14
2.4414062×10^{-4} (2^{-12})	0.16404166	0.5	328	27.99	14
2.4414062×10^{-4} (2^{-12})	0.16404166	0.1	1640	140.0	14
$6.103515625 \times 10^{-5}$ (2^{-14})	0.16404166	0.25	656	13.99	16
$1.52587890625 \times 10^{-5}$ (2^{-16})	0.16404166	0.1	1640	8.74	18

*Including one bit for sign and one bit for round off problems.

(6) Square wave vibration tests

(7) A combination of (3) and (6).

The outputs of the vehicle attitude driver are in the form of angular rotation increments ($\Delta\alpha_1$, $\Delta\alpha_2$, $\Delta\alpha_3$) the body mounted gyros would experience in each DDA computation interval Δt.

5.4.4.3 Vehicle Acceleration Driver

The vehicle thrust accelerations simulated in this computer simulation are the accelerations which would be experienced by accelerometers mounted along the X_b, Y_b and Z_b body axes. The accelerations simulated are shown in Equation (2). These accelerations

$$
\begin{aligned}
A_{xb} &= A_{1x} + A_{2x}t + A_{3x}t^2 \\
A_{yb} &= A_{1y} + A_{2y}t + A_{3y}t^2 \\
A_{zb} &= A_{1z} + A_{2z}t + A_{3z}t^2
\end{aligned}
\qquad (5\text{-}71)
$$

where

A_{xb}, A_{yb}, A_{zb} are the accelerations along the X_b, Y_b, Z_b body axes
A_{1x}, A_{1y}, A_{1z} are constant accelerations along the X_b, Y_b, Z_b axes
A_{2x}, A_{2y}, A_{2z} are ramp accelerations along the X_b, Y_b, Z_b axes
A_{3x}, A_{3y}, A_{3z} are parabolic accelerations along the X_b, Y_b, Z_b axes

include constant, ramp, and parabolic accelerations along any or all of the vehicle body axes. Since the accelerations are computed with a second-order polynomial, simple acceleration profiles may be simulated. The equations mechanized in the simulation are shown in Figure 5-13.

FROM THE VELOCITY TRANSFORMATION LOGIC

3

$$\Delta V_{xb} = \Delta t \left[A1X + A2X\,(t - \Delta t/2) + A3X\,(t^2 - t\Delta t + \Delta t^2/3) \right]$$

$$\Delta V_{yb} = \Delta t \left[A1Y + A2Y\,(t - \Delta t/2) + A3Y\,(t^2 - t\Delta t + \Delta t^2/3) \right]$$

$$\Delta V_{zb} = \Delta t \left[A1Z + A2Z\,(t - \Delta t/2) + A3Z\,(t^2 - t\Delta t + \Delta t^2/3) \right]$$

$$VXI = \Delta V_{xb}\, AP_{11} + \Delta V_{yb}\, AP_{21} + \Delta V_{zb}\, AP_{31}$$

$$VYI = \Delta V_{xb}\, AP_{12} + \Delta V_{yb}\, AP_{22} + \Delta V_{zb}\, AP_{32}$$

$$VZI = \Delta V_{xb}\, AP_{13} + \Delta V_{yb}\, AP_{23} + \Delta V_{zb}\, AP_{33}$$

REFERENCE VELOCITY COMPUTATION

$$VXI = VXI + \Delta VXI$$

$$VYI = VYI + \Delta VYI$$

$$VZI = VZI + \Delta VZI$$

TO PROGRAM STOP AND PRINT OUT CHECK

Figure 5-13. Vehicle Velocity Increment Computation

5.4.4.4 Reference Vehicle Attitude Computation

The reference vehicle attitude computation is performed using a second-order Taylor's series algorithm. This computation method is shown below in matrix notation as it is programmed in the scientific simulation (Equation 5-73). This mechanization is based on the $\left[a_{ij} \right]$ matrix relationship shown in Equation (5-72).

$$\begin{bmatrix} X_b \\ Y_b \\ Z_b \end{bmatrix} = \begin{bmatrix} a_{11R} & a_{12R} & a_{13R} \\ a_{21R} & a_{22R} & a_{23R} \\ a_{31R} & a_{32R} & a_{33R} \end{bmatrix} \begin{bmatrix} X_I \\ Y_I \\ Z_I \end{bmatrix} \tag{5-72}$$

where

X_b, Y_b, Z_b are the vehicle body axes

X_I, Y_I, Z_I are the inertial reference axes

$\left[a_{ij} \right]$ is an orthonormal matrix relating the two systems

$$\left[a_{ijR}\right]_m = \begin{bmatrix} \left[1 - 0.5(\Delta\alpha_2^2 + \Delta\alpha_3^2)\right] & \left[\Delta\alpha_3 + 0.5\ \Delta\alpha_2 \Delta\alpha_1\right] & \left[-\Delta\alpha_2 + 0.5\ \Delta\alpha_1 \Delta\alpha_3\right] \\ \left[-\Delta\alpha_3 + 0.5\ \Delta\alpha_1 \Delta\alpha_2\right] & \left[1 - 0.5(\Delta\alpha_1^2 + \Delta\alpha_3^2)\right] & \left[\Delta\alpha_1 + 0.5\ \Delta\alpha_2 \Delta\alpha_3\right] \\ \left[\Delta\alpha_2 + 0.5\ \Delta\alpha_1 \Delta\alpha_3\right] & \left[-\Delta\alpha_1 + 0.5\ \Delta\alpha_2 \Delta\alpha_3\right] & \left[1 - 0.5(\Delta\alpha_1^2 + \Delta\alpha_2^2)\right] \end{bmatrix} \left[a_{ijR}\right]_n \tag{5-73}$$

5.4.4.5 DDA Gyro-Output Scaling

The gyro pulses received by the DDA computer are generated at the same frequency the DDA operates. That is, if the DDA operates at a 1000 cps rate, the pulses received by the DDA from the gyros must also be at a 1000 cps rate. The pulses which are received by the DDA computer each represent a definite value of angle rotation, and they are generated in either a binary or ternary pulse sequence. Both the binary or ternary pulse sequences may be generated in the simulation, depending on the fixed point input JFA. If JFA = 1, a ternary pulse sequence is used; if JFA = 2, a binary pulse sequence is used.

A binary pulse sequence is a sequence of pulses with either a positive K_1 (scaling of the gyro pulses in rad) a negative K_1, but never a zero value. Thus, a zero rate would be a sequence of alternating pulses, a maximum positive rate would be a sequence of all positive pulses, etc. The method used for generating the binary pulses is shown in Figure 5-14. For the binary sequence mechanization, a positive pulse is output when the gyro is off null in the positive direction, and a negative pulse when the gyro is off null in the negative direction. By doing this, the gyro will be torqued back and forth, like a pulse-torqued gyro.

The ternary pulse sequence is a sequence of pulses of value zero, $+K_1$, or $-K_1$. This pulse sequence is generated in the computer simulation as shown in Figure 5-15. The ternary pulse sequence is typical of the single axis platform gyro mechanization.

5.4.4.6 <u>DDA Accelerometer Output Scaling</u>

The accelerometer output pulses received by the DDA computer are generated at the same frequency the DDA operates. The pulses received by the DDA computer each represent a definite value of velocity, and are generated in either a binary or ternary sequence. Both the binary and ternary pulse sequences may be generated in the simulation, depending on the fixed-point input JFB. If JFB #1, a binary pulse sequence is simulated; if JFB = 1, a ternary pulse sequence is simulated.

A binary pulse sequence is a sequence of pulses with either a positive CK2 (scaling of the accelerometer pulses, ft/sec), a negative CK2 value, but never a zero value. Thus a zero accelerometer output would be a sequence of alternating positive and negative pulses, a maximum positive accelerometer output would be a sequence of all positive pulses, etc. The method used for generating the binary pulses is shown in Figure 5-16, and is the same method used for generating the gyro binary pulse sequence (Figure 5-14).

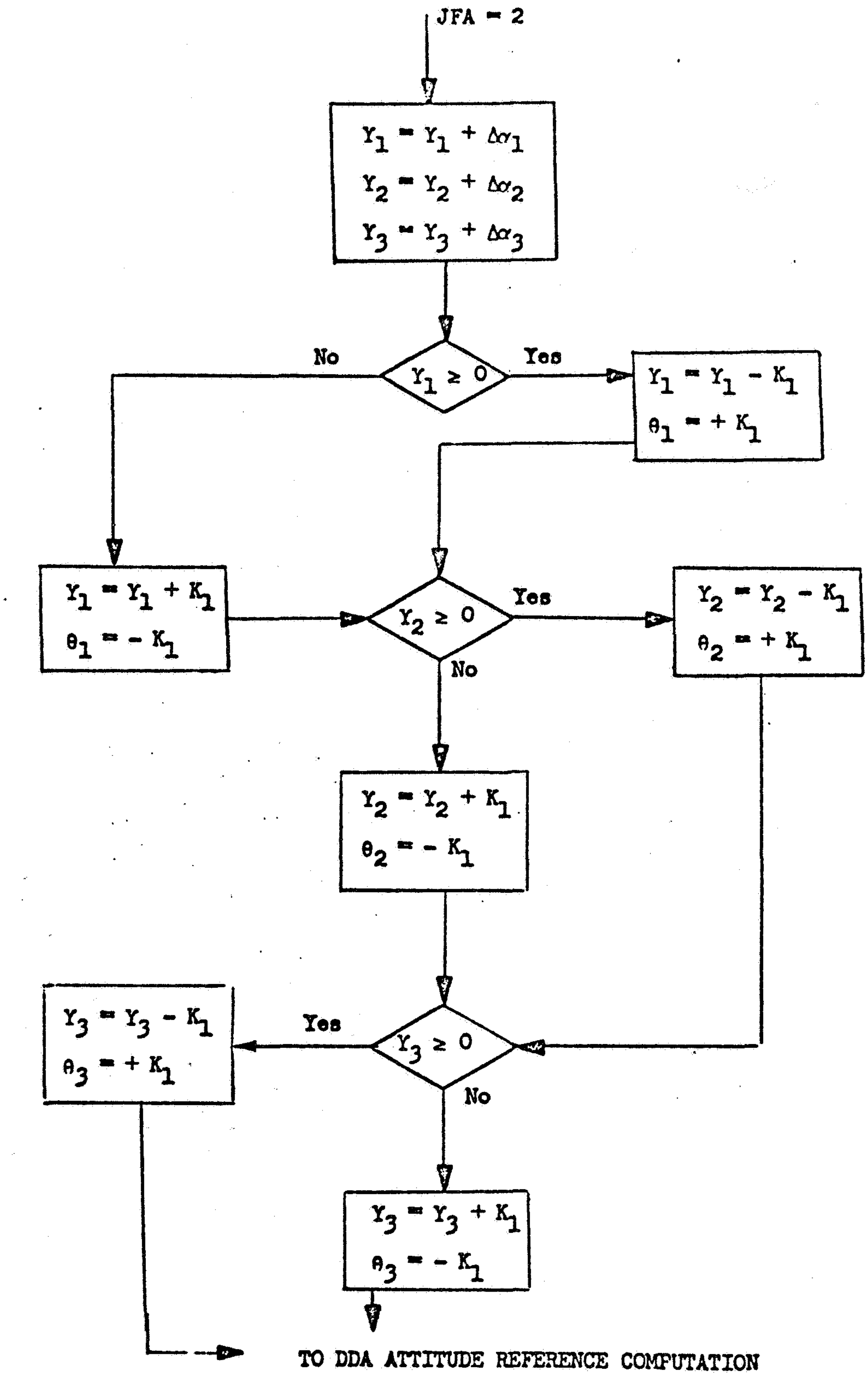


Figure 5-14. Binary Gyro Pulse Sequence Computation

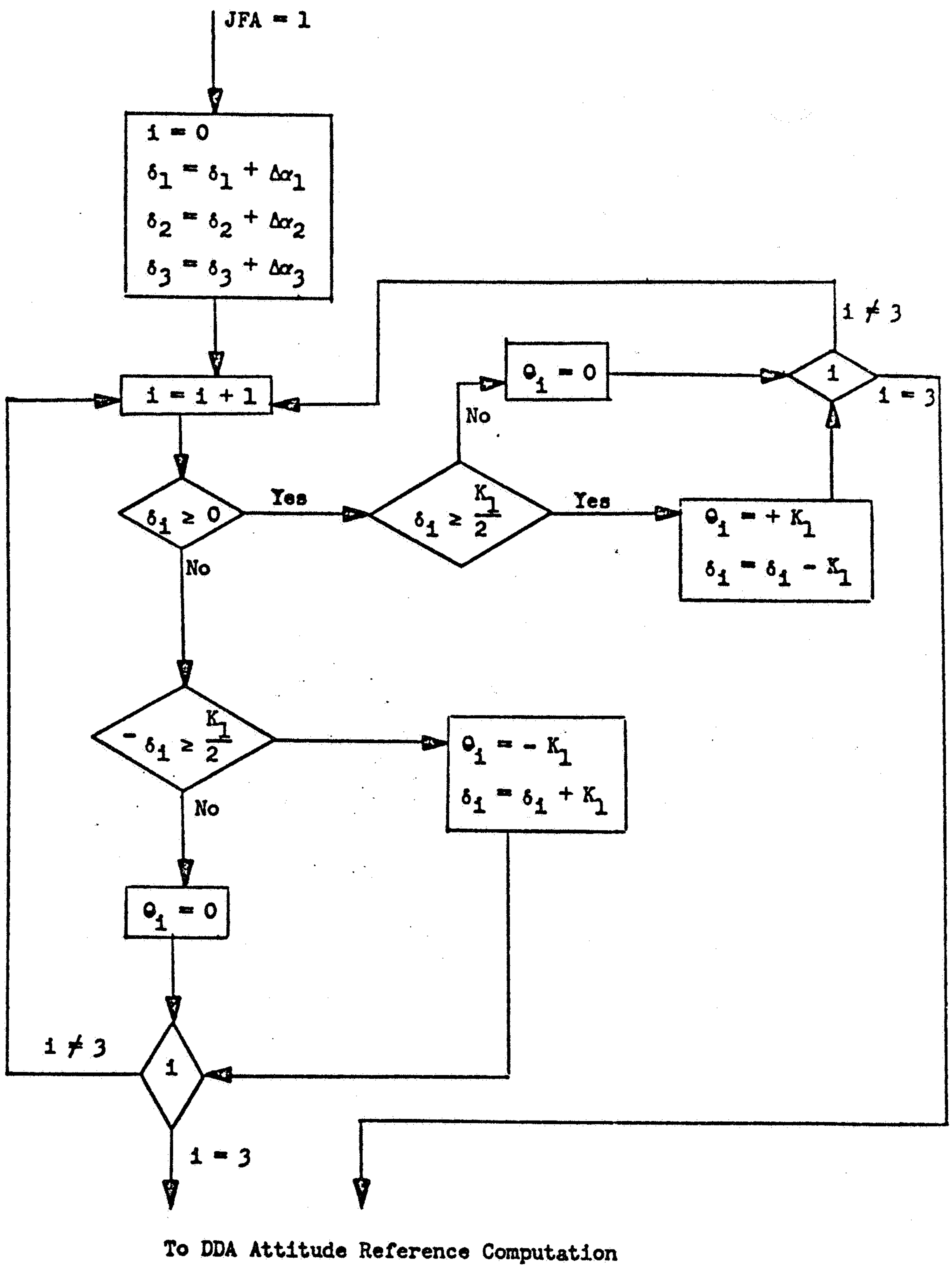


Figure 5-15. Ternary Gyro Pulse Sequence Computation

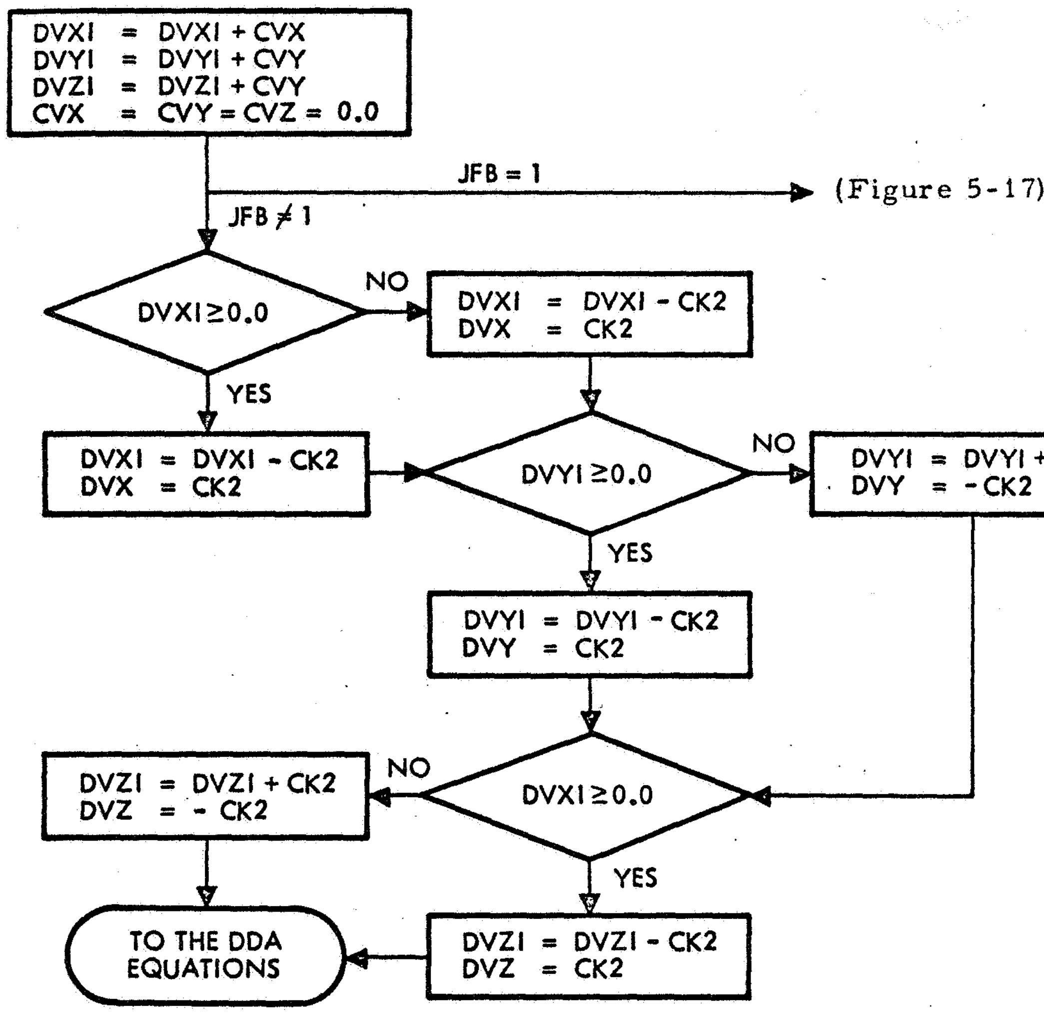


Figure 5-16. Binary Accelerometer Pulse Sequence Computation

The ternary pulse sequence is a sequence of pulses of value zero, + CK2, or - CK2. This pulse sequence is generated in the computer simulation as shown in Figure 5-17, and is the same method employed for generating the gyro ternary pulse sequence (Figure 5-15).

5.4.4.7 DDA Attitude Reference Computation

The DDA simulation mechanizes the attitude reference equations of a strapdown system using a first-order Taylor's series algorithm for computing the direction cosines. The relationship between the two coordinate systems is

$$\begin{bmatrix} X_b \\ Y_b \\ Z_b \end{bmatrix} = \left[a_{ij}\right] \begin{bmatrix} X_I \\ Y_I \\ Z_I \end{bmatrix} \qquad (5\text{-}74)$$

where

X_b, Y_b, Z_b are the vehicle roll, pitch, and yaw axes, respectively

X_I, Y_I, Z_I are the inertial coordinate axes for navigation and guidance purposes

$\left[a_{ij}\right]$ is the direction cosine matrix which relates the two systems

shown in Equation (5-74). The first-order Taylor's series algorithm used to update the a_{ij} direction cosine matrix is shown in matrix notation in Equation (5-75) and in equation form in Equation (5-76). The DDA simulation mechanizes the equations shown in Equation (5-76); the DDA mechanization of these equations is shown using DDA symbols in Figure 5-18.

$$\left[a_{ij}\right](n+1) = \begin{bmatrix} 1 & \Delta\alpha_3 & -\Delta\alpha_2 \\ -\Delta\alpha_3 & 1 & \Delta\alpha_1 \\ \Delta\alpha_2 & -\Delta\alpha_1 & 1 \end{bmatrix} \left[a_{ij}\right](n) \qquad (5\text{-}75)$$

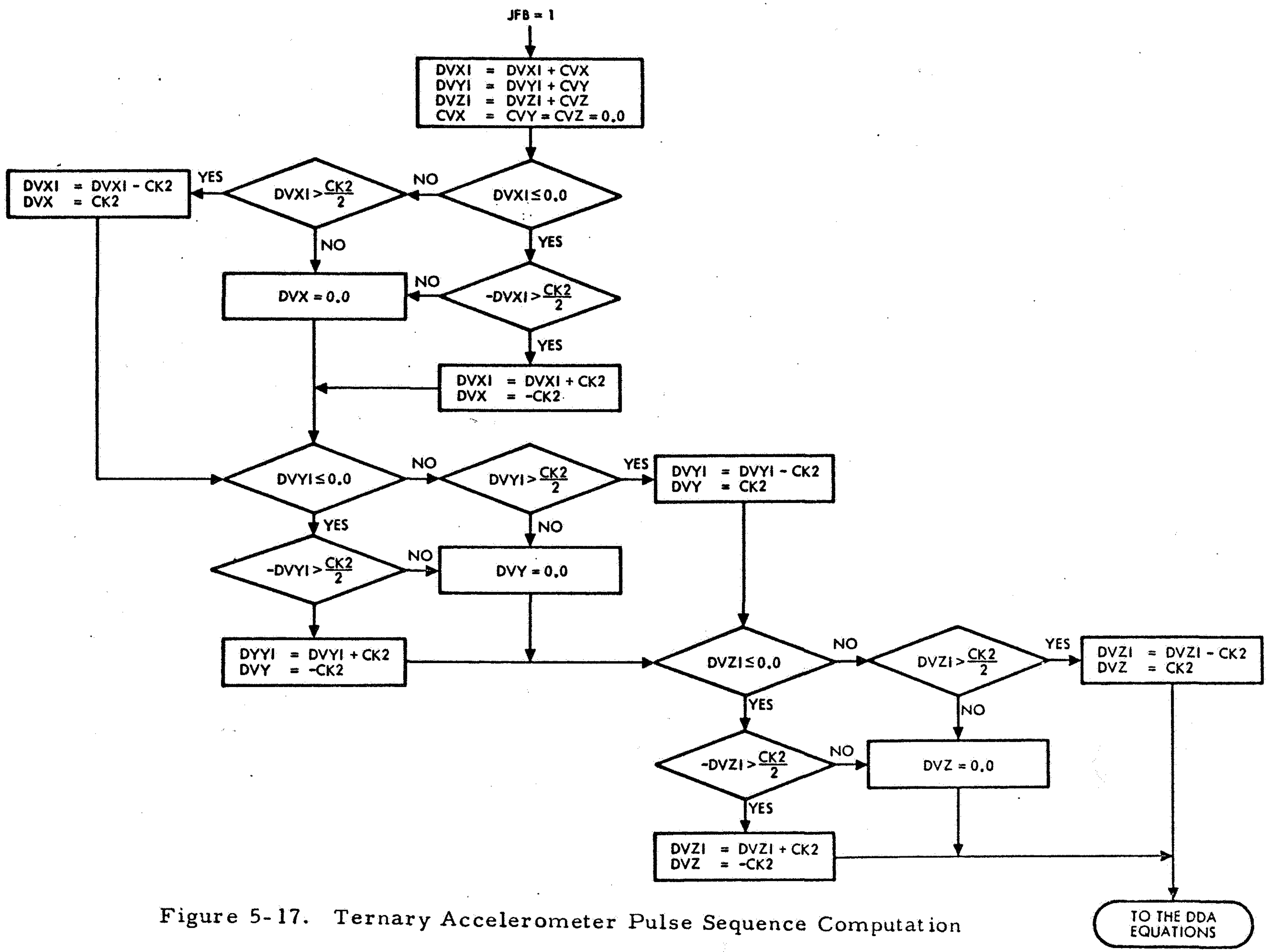


Figure 5-17. Ternary Accelerometer Pulse Sequence Computation

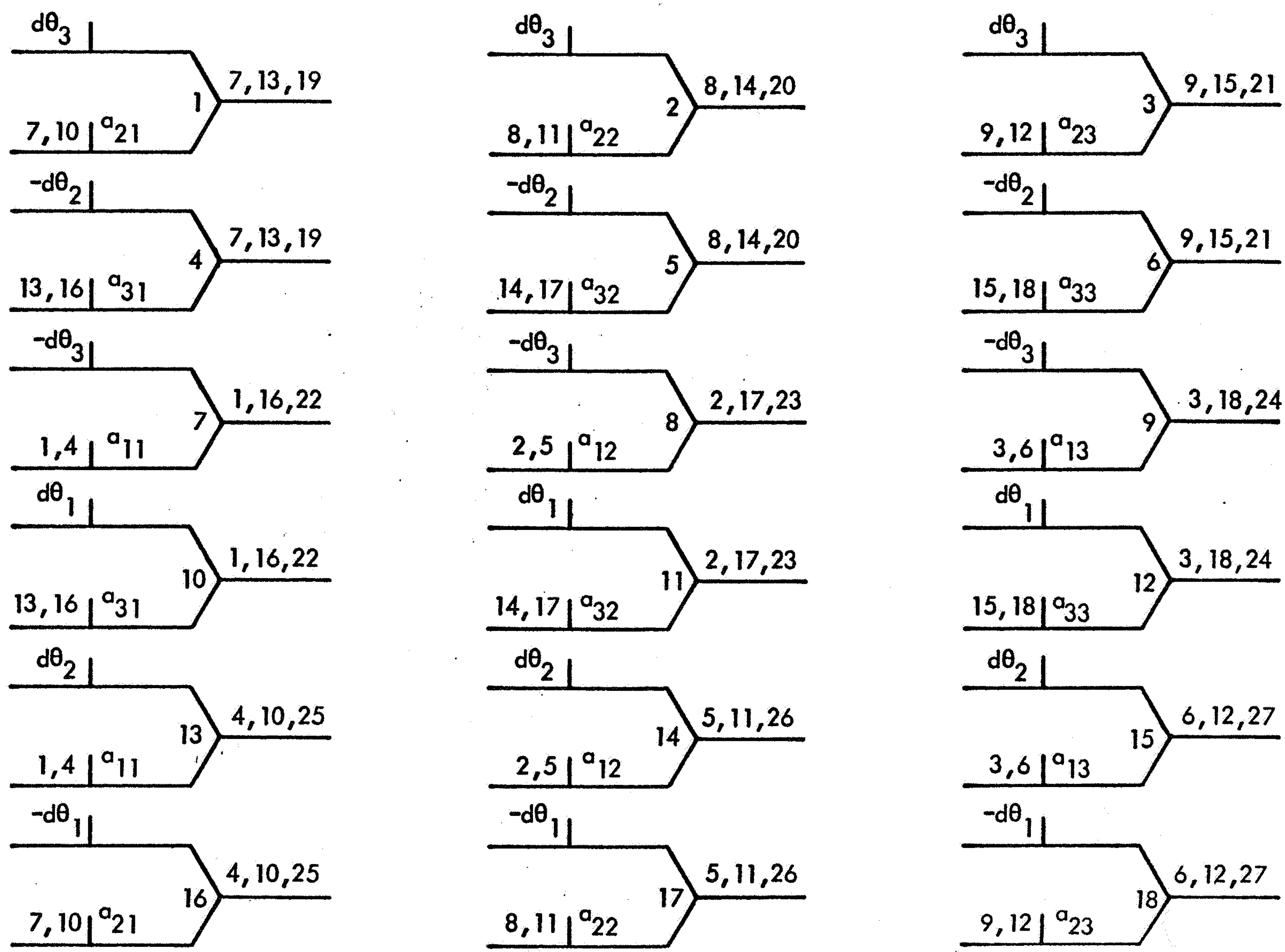


Figure 5-18. DDA Direction Cosine Mechanization for a First-Order Taylor's Series Algorithm

where

$[a_{ij}](K)$	are the direction cosines present after the Kth direction cosine update cycle
$\Delta\alpha_1, \Delta\alpha_2$ $\Delta\alpha_3$	are the positive right hand rotation increments about the X, Y, Z body axes, respectively, in the time interval between direction cosine computations

$$
\begin{aligned}
a_{11}(n+1) &= a_{11}(n) + \Delta\alpha_3 \, a_{21}(n) - \Delta\alpha_2 \, a_{31}(n) \\
a_{12}(n+1) &= a_{12}(n) + \Delta\alpha_3 \, a_{22}(n) - \Delta\alpha_2 \, a_{32}(n) \\
a_{13}(n+1) &= a_{13}(n) + \Delta\alpha_3 \, a_{23}(n) - \Delta\alpha_2 \, a_{33}(n) \\
a_{21}(n+1) &= a_{21}(n) - \Delta\alpha_3 \, a_{11}(n) + \Delta\alpha_1 \, a_{31}(n) \\
a_{22}(n+1) &= a_{22}(n) - \Delta\alpha_3 \, a_{12}(n) + \Delta\alpha_1 \, a_{32}(n) \\
a_{23}(n+1) &= a_{23}(n) - \Delta\alpha_3 \, a_{13}(n) + \Delta\alpha_1 \, a_{33}(n) \\
a_{31}(n+1) &= a_{31}(n) + \Delta\alpha_2 \, a_{11}(n) - \Delta\alpha_1 \, a_{21}(n) \\
a_{32}(n+1) &= a_{32}(n) + \Delta\alpha_2 \, a_{12}(n) - \Delta\alpha_1 \, a_{22}(n) \\
a_{33}(n+1) &= a_{33}(n) + \Delta\alpha_2 \, a_{13}(n) - \Delta\alpha_1 \, a_{23}(n)
\end{aligned}
\qquad (5\text{-}76)
$$

The integrators simulated in the computer program operate as follows:

(a) The values of the ΔX, ΔY, and ΔZ pulses are scaled the same and are of a value equal to the quantization of the Y and R registers least significant bit (K_1). This also requires the gyro quantization to be equal to this LSB quantization value.

(b) The gyro output pulses are quantized to the LSB quantization, and may be transferred to the computer in either of two ways, a ternary or binary sequence of pulses.

(c) The integrators are simulated by performing the following steps for each integrator in sequence, 1 through 18:

(1) The first operation is to update the Y registers. This is accomplished by adding the ΔY pulse to the value presently in the Y register.

$$Y_n = Y_{n-1} + \Delta Y$$

(2) Then the ΔX pulse is checked for magnitude. If ΔX is a +1 pulse,

$$R_n = R_{n-1} + Y$$

If ΔX is a -1 pulse,

$$R_n = R_{n-1} - Y$$

If ΔX is a zero pulse, $R_n = R_{n-1}$

(3) The third and last step is to check the magnitude of R_n. If $R_n \geq 1.0$, then

$$\Delta Z = +1 \text{ pulse}$$

$$R_n = R_n - 1.0$$

If $R_n \leq -1.0$,

$$\Delta Z = -1 \text{ pulse}$$

$$R_n = R_n + 1.0$$

If $|R_n| < 1.0$,

$$\Delta Z = 0.0$$

$$R_n = R_n$$

(4) Then the same procedure is followed on the next integrator.

5.4.4.8 DDA Velocity Transformation Equations

The body-mounted accelerometers are simulated along the x_b, y_b and z_b axes of the vehicle. Thus, the velocity increments measured by the accelerometers are

$$\overline{\Delta V}_b = \Delta V_{xb}\, \hat{x}_b + \Delta V_{yb}\, \hat{y}_b + \Delta V_{zb}\, \hat{z}_b$$

Since the body axes are related to the inertial axes by the a_{ij} orthonormal matrix,

$$\begin{bmatrix} x_b \\ y_b \\ z_b \end{bmatrix} = \left[a_{ij}\right] \begin{bmatrix} x_I \\ y_I \\ z_I \end{bmatrix} \tag{5-77}$$

the inertial velocity increments are obtained by the following equations (as mechanized in the simulation)

$$\begin{aligned} \Delta V_{xI} &= \Delta V_{xb}\, a_{11} + \Delta V_{yb} a_{21} + \Delta V_{zb} a_{31} \\ \Delta V_{yI} &= \Delta V_{xb}\, a_{12} + \Delta V_{yb} a_{22} + \Delta V_{zb} a_{32} \\ \Delta V_{zI} &= \Delta V_{xb}\, a_{13} + \Delta V_{yb} a_{23} + \Delta V_{zb} a_{33} \end{aligned} \tag{5-78}$$

In DDA notation, these equations are shown in Figure 5-19. These DDA integrators are operated in sequence, 19 through 30, after the 18th integrator operations are completed.

5.4.4.9 Attitude Reference Errors

The attitude reference errors of the DDA were computed by comparing the reference direction cosines with the direction cosines

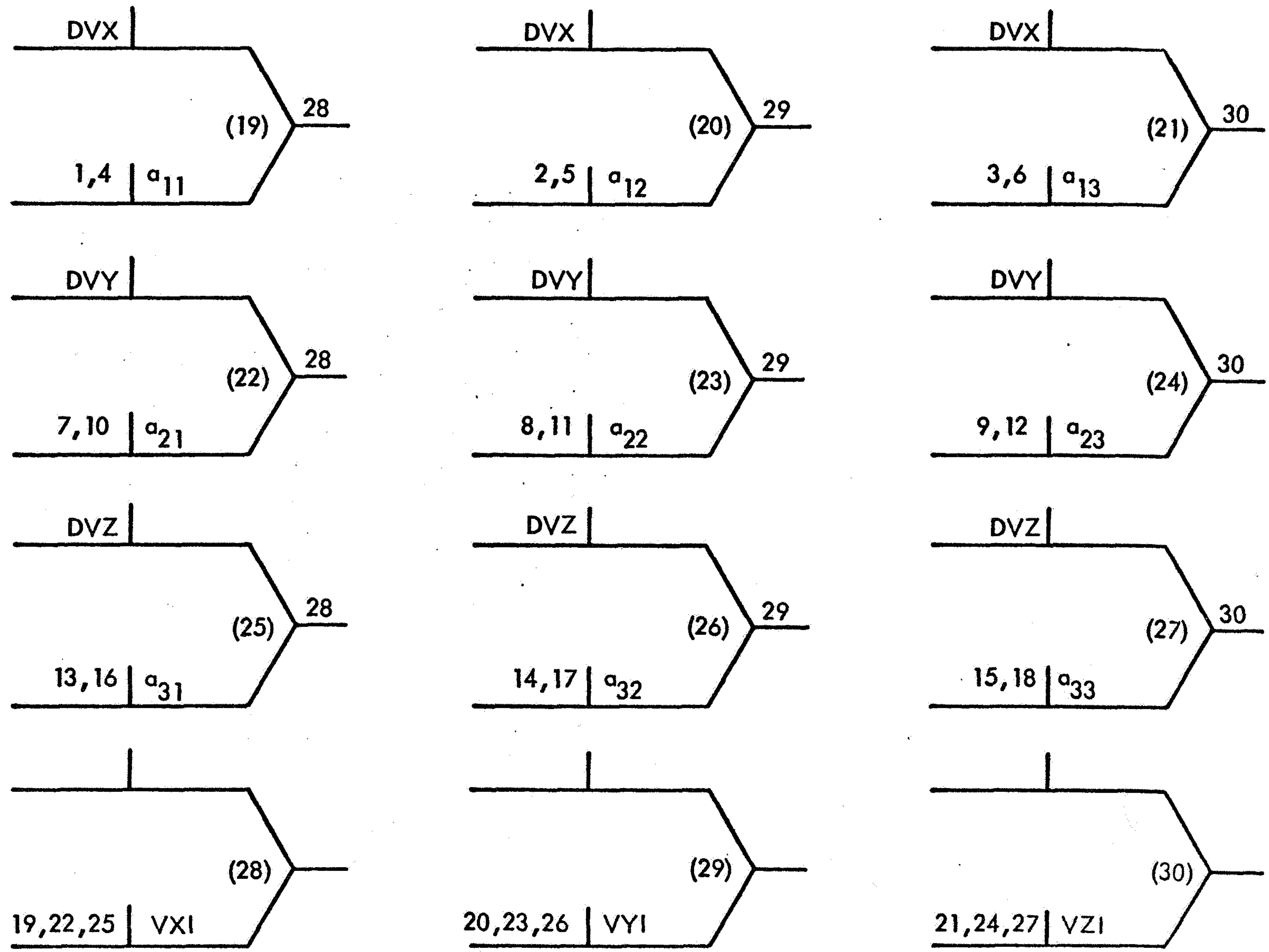


Figure 5-19. DDA Velocity Transformation

computed by the DDA. These attitude errors are defined as the positive rotation's about the vehicle body axes to align the two matrices. That is,

$$
\begin{aligned}
E_{xb} &= 0.5 \left[Z_b(DDA) \cdot Y_b(ref) - Y_b(DDA) \cdot Z_b(ref)\right] \\
E_{yb} &= 0.5 \left[Z_b(DDA) \cdot X_b(ref) - X_b(DDA) \cdot Z_b(ref)\right] \\
E_{zb} &= 0.5 \left[X_b(DDA) \cdot Y_b(ref) - Y_b(DDA) \cdot X_b(ref)\right]
\end{aligned}
\qquad (5\text{-}79)
$$

These attitude errors as described above in Equation 5-79 are computed in rads. These attitude errors are converted to arc seconds by dividing by $4.84813681 \times 10^{-6}$, as shown in Equation 5-80. The errors are printed out both in radians and arc seconds. These attitude errors are the errors of the DDA relative to the second-order Taylor's series attitude reference algorithm as used in a general purpose computer. Thus, the true attitude errors of the DDA must be computed from a true analytically computed direction cosine matrix as the second-order Taylor's series is not completely accurate.

$$
\begin{aligned}
E_{xbs} &= 10^6 \left[E_{xb}/4.848...\right] \\
E_{ybs} &= 10^6 \left[E_{yb}/4.848...\right] \\
E_{zbs} &= 10^6 \left[E_{zb}/4.848...\right]
\end{aligned}
\qquad (5\text{-}80)
$$

5.4.4.10 Input Variables

$\left.\begin{matrix} T_1 \\ T_2 \\ T_3 \end{matrix}\right\}$ the periods of the square wave limit cycles applied to the X_b, Y_b, Z_b axes

FK (K_1 in the previous discussion) - DDA least significant bit quantization in radians. FK is also the quantization of the gyro output pulses in radians.

IFA - vehicle attitude driver flag (fixed point)

IPRINT - output printout cycle interval

Y_i - initial DDA integrator Y register values

PHI1, PHI2, PHI3 — phase shift of the sinusoidal rotations experienced about the X_b, Y_b, Z_b body axes, respectively

A_1, A_2, A_3 — amplitudes in radians of the sinusoidal rotations experienced about the X_b, Y_b, Z_b axes respectively

W_1, W_2, W_3 — frequencies in radians/second of the sinusoidal rotations experienced about the X_b, Y_b, Z_b axes respectively

AA01, AA02, AA03 — constant angular accelerations in radians/second^2 experienced about the X_b, Y_b, Z_b axes

A01, A02, A03 — constant angular velocities in radians/second experienced about the X_b, Y_b, Z_b axes

W_{s1}, W_{s2}, W_{s3} — amplitudes in radians per second of the square wave limit cycles which may be applied to the X_b, Y_b, Z_b axes

TF - duration time in seconds of the computer simulation

DELT - time in seconds per DDA cycle

DELT1 - time in seconds per DDA cycle

JFA - DDA ternary or binary flag (fixed point). Ternary if JFA = 1, binary if JFA = 2

AR_{ij} - reference direction cosines

ART_{ij} - initial direction cosines for a new test case.

JFB - DDA ternary or binary flag for the velocity integrators. (ternary if JFB = 1, binary if JFB = 1)

CK2 - quantization level of the accelerometer output pulses in feet per second

A1X A1Y A1Z	constant accelerations in feet/second^2 along the Xb, Yb, and Zb body axes
A2X A2Y A2Z	ramp accelerations in feet/second^3 along the Xb, Yb, and Zb body axes
A3X A3Y A3Z	parabolic accelerations in feet/second^4 along the Xb, Yb, Zb body axes

5.4.4.11 Output Variables

Time - time in seconds

EXB, EYB, EZB - attitude errors (radians) relative to a second-order Taylor's series attitude reference algorithm

EXBS, EYBS, EZBS - the above attitude errors in seconds of arc

AR - the reference direction cosine matrix

Y(1) through Y (18) - the first 18 DDA integrator outputs. These outputs are listed in the following order, so that the direction cosines matrix as computed by the DDA may be easily compared to the reference set of direction cosines.

Y(7)	Y(8)	Y(9)
Y(1)	Y(2)	Y(3)
Y(4)	Y(5)	Y(6)
Y(13)	Y(14)	Y(15)
Y(16)	Y(17)	Y(18)
Y(10)	Y(11)	Y(12)

VX, VY, VZ - accumulated velocities along the vehicle body axes (confidence check of the simulation's operation only)

DVXI, DVYI, DVZI - inertial velocity increments (for checkout purposes only

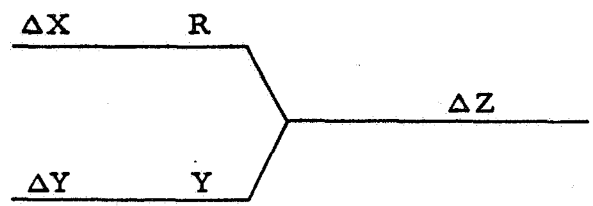
Y(19) through Y(30) - last twelve DDA outputs. The first nine are the direction cosines and the last three the inertial velocities accumulated by the DDA

Y(19)	Y(20)	Y(21)
Y(22)	Y(23)	Y(24)
Y(25)	Y(26)	Y(27)
Y(28)	Y(29)	Y(30)

VXI, VYI, VZI - reference set of inertial velocities in feet/second.

5.4.5 DDA Quantization Errors

As mentioned previously, the DDA integrators operate as follows: First the Y register is updated with the incoming ΔY pulses; if a ΔX pulse is present, the Y register is added to the R register, and if the R register has a value greater than one, a Z pulse is generated.



Consider a DDA design for the direction cosine maintenance which the ΔZ, ΔY, and ΔX pulses all have the same value of 2^{-14} rad, and the Y and R registers both have the same word length with the least significant bit worth 2^{-14} rad. Since the Y register contains a direction cosine, the number in the Y register is $\leq$ one.

The output of the DDA integrator described is $\Delta Z = Y\Delta X$. However, if the R register does not attain a value ≥ 1, the ΔZ output will be zero. Since the R register accumulates $\Sigma\, Y\Delta X$ until this sum is $\geq$ to the least significant bit of the Y register, this is equivalent to having a 28 bit GP computer. That is, the quantization error of the DDA is apparently $\pm 2^{-29}$ radians. If it may be assumed that the probability density function of this quantization error (the difference between the real variable and its discretized representation in the computer) is uniform between -2^{-29} and $+2^{-29}$ or between $-q/2$ and $+q/2$. The variance of such a distribution is given by

$$\sigma^2 = \frac{q^2}{12} \qquad (5\text{-}81)$$

the errors are additive, and independent, the resulting variance at the end of n cycles is

$$\sigma^2(n) = n\, q^2/12$$

$$\sigma\,(n) = \sqrt{n/3}\; q/2 \qquad (5\text{-}82)$$

$$\sigma\,(n) = \frac{n}{3}\, 2^{-29}$$

After a flight of 600 sec ($n = 2.16 \times 10^6$),

$$\sigma\,(n) = 1.58 \times 10^{-6} \text{ rad} \qquad (5\text{-}83)$$

The scientific simulation results of the DDA mechanization are used as a means of verifying this computation of the variance of the DDA quantization errors. The simulation results will be a combination of the first-order Taylor's series algorithm errors and the DDA mechanization errors. Since the algorithm errors are known from analytic derivations, the DDA errors can be easily determined from the simulation results.

5.4.6 DDA Simulation Results

The DDA designs presented in Table 5-13 were tested on the DDA computer simulation to evaluate their respective accuracies, and also to compare these results with the methods which may be employed on a general purpose computer. The tests which were run on the DDA simulation are tabulated in Tables 5-16 through 5-20. The results of these tests are discussed in two parts, the attitude reference results in Section 5.4.6.1 and the velocity transformation results in Paragraph 5.4.6.2. The velocity transformation results were obtained by comparing the test results with results obtained from computer simulations. . The attitude reference errors were obtained by comparison of the DDA direction cosines with hand computed direction cosines or direction cosines obtained from a 7094 computer simulation, and are presented in the next section using the following definitions.

1) Attitude Errors

The attitude errors are obtained by comparing the DDA direction cosines (a_{ij}) with the true direction cosines (b_{ij}). The attitude errors are defined to be the right hand positive rotations about the j body axis (j = x, y, z) which would be required to reduce the attitude error about this axis to zero.

$$E_x = 0.5 \, [a_{31} b_{21} + a_{32} b_{22} + a_{33} b_{23} - a_{21} b_{31} - a_{22} b_{32} - a_{23} b_{33}]$$

$$E_y = 0.5 \, [a_{11} b_{31} + a_{12} b_{32} + a_{13} b_{33} - a_{31} b_{11} - a_{32} b_{12} - a_{33} b_{13}]$$

$$E_z = 0.5 \, [a_{21} b_{11} + a_{22} b_{12} + a_{23} b_{13} - a_{11} b_{21} - a_{12} b_{22} - a_{13} b_{23}]$$

(5-84)

2) Normality

The sum of the squares of a row or column of direction cosines will equal 1.0 if the direction cosine matrix is normal. If the direction cosine matrix is not normal, the velocity increments will be transformed incorrectly to the coordinate system. The following set of normality terms are calculated in the next sections.

$$N_1^2 = a_{11}^2 + a_{12}^2 + a_{13}^2$$

$$N_2^2 = a_{21}^2 + a_{22}^2 + a_{23}^2$$

$$N_3^2 = a_{31}^2 + a_{32}^2 + a_{33}^2$$

(5-85)

3) Orthogonality

The dot product of two unit vectors in an orthonormal coordinate system is zero. The a_{ij} matrix obtained from the DDA simulation tests will be tested for its orthogonality. by computing the following quantities.

$$O_{12} = a_{11}\, a_{21} + a_{12}\, a_{22} + a_{13}\, a_{23}$$

$$O_{13} = a_{11}\, a_{31} + a_{12}\, a_{32} + a_{13}\, a_{33}$$

$$O_{23} = a_{21}\, a_{31} + a_{22}\, a_{32} + a_{23}\, a_{33}$$

(5-86)

5.4.6.1 DDA Attitude Errors

In all of the tests performed on the DDA simulation, constant slewing rates were applied equally to each of the vehicle body axes. This test was chosen as it exercised the direction cosine computations to the full extent, and provided at the same time an easily computed reference direction cosine matrix Equation(5-87).

$$\begin{bmatrix} \frac{1}{3}(1 + 2\cos \omega t) & [\frac{1}{3}(1 - \cos \omega t) + \frac{3}{3}\sin \omega t] & [\frac{1}{3}(1 - \cos \omega t) - \frac{3}{3}\sin \omega t] \\ [\frac{1}{3}(1 - \cos \omega t) - \frac{3}{3}\sin \omega t] & [\frac{1}{3}(1 + 2\cos \omega t)] & [\frac{1}{3}(1 - \cos \omega t) - \frac{3}{3}\sin \omega t] \\ [\frac{1}{3}(1 - \cos \omega t) + \frac{3}{3}\sin \omega t] & [\frac{1}{3}(1 - \cos \omega t) - \frac{3}{3}\sin \omega t] & [\frac{1}{3}(1 + 2\cos \omega t] \end{bmatrix}$$

(5-87)

where

$\omega = \sqrt{3}$ times the rotation rate applied to each vehicle axis

The tests are divided into 4 areas for the velocity comparison, and into 3 areas for the attitude reference comparison. The drivers used to test the DDA accuracy for several DDA word lengths are shown in Table 5-14. Note that tests 2 and 3 will be presented together for the attitude reference results of the DDA.

Table 5-14. DDA Driver Characteristics

Test	X_b Body Axis* Acceleration (ft/sec^2)	Y_b Body Axis* Acceleration (ft/sec^2)	Z_b Body Axis* Acceleration (ft/sec^2)	Rotation Rates** of the Vehicle (deg/sec)
1	$5.0 + 0.1t$	$20 + t + 0.5t^2$	$5.0 + 0.1t$	$\frac{50}{\sqrt{3}} (\hat{i}_x + \hat{i}_y + \hat{i}_z)$
2	$5.0 + 0.1t$	$20 + t + 0.5t^2$	$5.0 + 0.1t$	$\frac{25}{\sqrt{3}} (\hat{i}_x + \hat{i}_y + \hat{i}_z)$
3	$5.0 + 0.1t$	$20 + t + 0.25t^2$	$5.0 + 0.1t$	$\frac{25}{\sqrt{3}} (\hat{i}_x + \hat{i}_y + \hat{i}_z)$
4	$5.0 + 0.1t$	$20 + t + 0.25t^2$	$5.0 + 0.1t$	$\frac{10}{\sqrt{3}} (\hat{i}_x + \hat{i}_y + \hat{i}_z)$

* t is time in sec

**$\hat{i}_x$, $\hat{i}_y$, $\hat{i}_z$ are unit vectors along the vehicle x, y, z axes.

As mentioned earlier, the attitude reference algorithm used to maintain the DDA direction cosines was the first-order Taylor's series. In Phase I of the MSFC body-mounted study, a hand analysis of the first- and second order-Taylor's series algorithms was performed to ascertain their respective errors as they would occur in a general purpose computer (without considering word length as a variable). The results of this hand analysis are shown below for the first-order Taylor's series algorithm, using the above definitions N, 0, E_x, E_y, and E_z, for a three-axis slew.

$$N_1 = N_2 = N_3 = [\,1 + 3\Delta^2\,]^{n/2}$$

$$\text{per cycle angle error} = -\,\Delta^3$$

$$E_x = E_y = E_z \sim -n\Delta^3$$

$$0_{12} = 0_{13} = 0_{23} = \frac{1 - R^{2n}}{3} \qquad (5\text{-}88)$$

where

n is the number of attitude compute cycles

Δ is the angle increment in rad about each vehicle body axis

$$R = +\sqrt{1 + 3\Delta^2}$$

However, this error analysis does not apply for the accuracy of a first-order Taylor's series algorithm when it is mechanized on a DDA. This is because the DDA direction cosines are split into two parts; one-half in the Y register and one-half in the R register. The direction cosines stored in the Y register have a maximum value of 1 and are quantized to 2^{-K} rad (least significant bit). The R register is scaled at a maximum value of 2^{-K} rad with the least significant bit scaled to 2^{-2K} rad. Thus, the DDA has an effective word length of twice the word length of the Y register; however, this advantage is lost when the velocity increments are transformed to inertial coordinates as only the Y register is used for this purpose. As a result, normality and orthogonality errors are present in the transformation of the velocity increments, the size of which depends on the quantization of the Y register 2^{-K} rad.

For example consider the following case where a general purpose computer and a DDA are operating at the same speed. The initial direction cosine matrix is the identity matrix, and the output pul... the gyros are scaled to q rad. The direction cosine matrix of the ... computer would be

$$[a_{ij}]_{(n)} = \begin{bmatrix} 1 & q & -q \\ -q & 1 & q \\ q & -q & 1 \end{bmatrix}^n \qquad (5\text{-}...)$$

where

q is the gyro quantization in rad

n is the number of compute cycles

After 2 cycles,

$$[a_{ij}]_{(2)} = \begin{bmatrix} 1 - 2q^2 & 2q + q^2 & -2q + q^2 \\ -2q + q^2 & 1 - 2q^2 & 2q + q^2 \\ 2q + q^2 & -2q + q^2 & 1 - 2q^2 \end{bmatrix} \qquad (5\text{-}90)$$

The DDA output for the same algorithm is obtained first by examining the way a DDA integrator operates, and then to apply this knowledge to Figures 5-18 and 5-19.

As explained in Paragraph 5.4.2, the first DDA integrator operation is to update the Y register. After the Y register is updated, the R register is updated with the contents of the Y register corresponding to the input $\Delta\theta$ (gyro output pulse), and a ΔZ output pulse is generated when the R register overflows. Applying this information to the DDA diagrams on Figures 5-18 and 5-19, the direction cosines which would be used for the transformation of the velocity increments to inertial coordinates would be

$$[a_{ij}]_{(1)} = \begin{bmatrix} 1 & -q & q \\ q & 1 & -q \\ -q & q & 1 \end{bmatrix}$$

$$[a_{ij}]_{(2)} = \begin{bmatrix} 1 & -2q & 2q \\ 2q & 1 & -2q \\ -2q & 2q & 1 \end{bmatrix}$$

$$[a_{ij}]_{(3)} = \begin{bmatrix} 1 & -3q & 3q \\ 3q & 1 & -3q \\ -3q & 3q & 1 \end{bmatrix} \qquad (5\text{-}91)$$

The DDA matrices keep building up in this manner until the off diagonal direction cosine R registers overflow and send pulses to the main diagonal Y registers. Thus, it is easily seen that the direction cosine matrices in the Y register of the DDA are much different from the a_{ij} matrix computed in a general purpose computer, especially the normality and orthogonality conditions of the matrices. Since it is known that the off diagonal term integrators output pulses only when their respective R registers overflow, a maximum scale factor and normality error may be computed.

For instance, a DDA computer with the Y register scaled at q rad will be used as an example. The R register of an off diagonal direction cosine would build up as follows

$$R = q + 2q + 3q + \ldots. + nq \qquad (5\text{-}92)$$

When this sum reaches the value 1.0, an output pulse would be generated, and the R register would then be decreased by 1.0 in magnitude.

Since the sum $R = q + 2q + \ldots + nq$ is equivalent to:

$$\frac{(n)\ (n+1)}{2}\ q \qquad (5\text{-}?)$$

the number of cycles required to cause the R register to overflow may be found by solving Equation (5-93) for n

$$\frac{n\ (n+1)}{2}\ q = 1 \qquad (5\text{-}?)$$

This equation was solved for several DDA quantization values and are shown below in Table 5-15.

The maximum normality error for this worst-case condition is

$$N(2^{-10}) = 1.00184631 \qquad 0.185\%$$

$$N(2^{-12}) = 1.0004827976 \qquad 0.05\ \%$$

$$N(2^{-14}) = 1.0001206994 \qquad 0.012\%$$

$$N(2^{-16}) = 1.000030342722 \qquad 0.003\% \qquad (5\text{-}95)$$

Table 5-15. Solutions for DDA Quantization Values

q	n	(n-1) q
2^{-10} = 201.43 arc sec	45	2.45193 arc deg
2^{-12} = 50.358 arc sec	91	1.2589 arc deg
2^{-14} = 12.589 arc sec	181	0.6295 arc deg
2^{-16} = 3.147 arc sec	362	0.315 arc deg

The maximum orthogonality error is

$$
\begin{aligned}
0(2^{-10}) &= 0.00184631 \text{ R} = 381 \text{ arc sec} = 6.35 \text{ arc min} \\
0(2^{-12}) &= 0.000427976 \text{ R} = 88.4 \text{ arc sec} = 1.47 \text{ arc min} \\
0(2^{-14}) &= 0.0001206994 \text{ R} = 24.9 \text{ arc sec} = 0.41 \text{ arc min} \\
0(2^{-16}) &= 0.0000303427 \text{ R} = 6.25 \text{ arc sec} = 0.104 \text{ arc min}
\end{aligned} \tag{5-96}
$$

Of the four quantization values, it appears as though the only methods which might achieve the desired accuracy are 2^{-14} and 2^{-16} rad quantization.

The attitude reference results of the DDA simulation tests which were run on the 7094 IBM computer are tabulated below in Table 5-16. These tests verify the above results for the several quantization values for the DDA. In some cases, the attitude errors exceed the quantization of the gyro pulses. This is because each direction cosine may be in error by 3 times the Y register's least significant bit scaling due to the storage in the R registers. However, the problems which result from the non-orthogonality and nonnormality of the direction cosines stored in the Y registers do not become apparent until the results of the velocity transformation are examined.

5.4.6.2 <u>DDA Velocity Transformation Errors</u>

The velocity transformation errors are a result of the orthogonality, normality, and attitude errors of the direction cosines as discussed in Paragraph 5.4.6.1, the velocity transformation algorithm errors, and the errors due to the computer quantization of the velocity input pulses. The velocity errors which resulted from these error sources were obtained from the DDA simulation for several different computer quantization values. These results are tabulated in Tables 5-17 through 5-20, and include the effects of the direction cosine errors, the velocity quantization errors, the velocity transformation algorithm errors, the gyro quantization errors, and the DDA computer errors.

Table 5-16. DDA Attitude Errors

Test No.	Rotation Rate Applied To Each Axis (deg/sec)	Gyro Pulse Quantization (arc/sec)	DDA Cycle Speed (msec)	Time (sec)	E_x (arc sec)	E_y (arc sec)	E[illegible] (arc sec)
1	$10/\sqrt{3}$	201.4	5.0	5.0	177.4	- 88.2	-168.7
2	$10/\sqrt{3}$	50.4	1.25	5.0	42.0	- 38.8	- 25.6
3	$10/\sqrt{3}$	12.6	0.25	5.0	10.3	- 9.0	- 5.6
4*	$25/\sqrt{3}$	201.4	2.0	5.0	150.5	-150.4	- 49.[illegible]
5	$25/\sqrt{3}$	50.4	0.1	5.0	37.1	- 39.2	- 64.2
6	$25/\sqrt{3}$	50.4	0.5	5.0	37.3	- 39.7	- 63.8
7	$25/\sqrt{3}$	12.6	0.1	5.0	10.5	- 10.6	- 17.8
8	$50/\sqrt{3}$	201.4	1.0	5.0	112.8	- 85.7	43.8
9	$50/\sqrt{3}$	50.4	0.25	5.0	- 3.1	5.8	33.0
1	$10/\sqrt{3}$	201.4	5.0	10.0	23.9	-124.5	- 23.9
2	$10/\sqrt{3}$	50.4	1.25	10.0	75.5	- 40.1	- 69.0
3	$10/\sqrt{3}$	12.6	0.25	10.0	4.6	- 10.1	- 4.8
4*	$25/\sqrt{3}$	201.4	2.0	10.0	-112.8	- 85.6	43.8
5	$25/\sqrt{3}$	50.4	0.1	10.0	- 6.9	8.9	35.9
6	$25/\sqrt{3}$	50.4	0.5	10.0	- 6.1	8.8	36.0
7	$25/\sqrt{3}$	12.6	0.1	10.0	14.5	- 22.1	- 21.2
8	$50/\sqrt{3}$	201.4	1.0	10.0	149.8	- 92.9	185.6
9	$50/\sqrt{3}$	50.4	0.25	10.0	26.1	- 26.0	- 26.3
1	$10/\sqrt{3}$	201.4	5.0	15.0	24.5	-149.8	-116.3
2	$10/\sqrt{3}$	50.4	1.25	15.0	35.3	- 26.9	- 47.8
3	$10/\sqrt{3}$	12.6	0.25	15.0	*	*	*
4*	$25/\sqrt{3}$	201.4	2.0	15.0	7.92	- 24.1	-122.5
5	$25/\sqrt{3}$	50.4	0.1	15.0	5.9	19.3	19.5
6	$25/\sqrt{3}$	50.4	0.5	15.0	10.0	- 6.0	18.7
7	$25/\sqrt{3}$	12.6	0.1	15.0	16.5	- 11.3	- 18.2
8	$50/\sqrt{3}$	201.4	1.0	15.0	92.0	66.9	24.3
9	$50/\sqrt{3}$	50.4	0.25	15.0	- 29.1	12.4	60.5
1	$10/\sqrt{3}$	201.4	5.0	20.0	- 6.7	-153.1	- 83.8
2	$10/\sqrt{3}$	50.4	1.25	20.0	10.0	- 38.7	- 28.9
3	$10/\sqrt{3}$	12.6	0.25	20.0	*	*	*
4*	$25/\sqrt{3}$	201.4	2.0	20.0	149.7	- 92.9	-185.5
5	$25/\sqrt{3}$	50.4	0.1	20.0	- 5.3	- 4.1	- 24.0
6	$25/\sqrt{3}$	50.4	0.5	20.0	24.5	- 24.4	- 24.7
7	$25/\sqrt{3}$	12.6	0.1	20.0	23.1	- 25.1	- 31.0
8	$50/\sqrt{3}$	201.4	1.0	20.0	-305.5	206.7	210.3
9	$50/\sqrt{3}$	50.4	0.25	20.0	- 39.9	26.3	48.6

*Run 4 was timed for a maximum of 12 sec.

Table 5-17. DDA Velocity Errors for Test 1 (Table 5-16)

DDA Cycle Time (msec)	Gyro Pulse Quantization (arc sec)	Accelerometer Pulse Quantization (ft/sec)	Time (sec)	V_x (ft/sec)	V_y (ft/sec)	V_z (ft/sec)
0.25	50.4	0.0805	5.0	0.05	0.09	-0.03
1.0	201.4	0.322		-0.03	0.33	-0.59
0.25	50.4	0.0805	10.0	0.26	0.02	-0.07
1.0	201.4	0.322		0.06	-0.06	-0.23
0.25	50.4	0.0805	15.0	-0.004	-0.40	0.55
1.0	201.4	0.322		-0.65	-1.53	2.15
0.25	50.4	0.0805	20.0	-0.89	0.81	0.79
1.0	201.4	0.322		-3.59	3.37	-0.31

Table 5-18. DDA Velocity Errors for Test 2 (Table 5-16)

DDA Cycle Time (msec)	Gyro Pulse Quantization (arc sec)	Accelerometer Pulse Quantization (ft/sec)	Time (sec)	V_x (ft/sec)	V_y (ft/sec)	V_z (ft/sec)
10^{-4}	12.6	0.164	5.0	0.09	-0.13	-0.01
10^{-4}	12.6	0.164	10.0	0.22	0.11	0.04
10^{-4}	12.6	0.164	15.0	0.20	0.32	0.03
10^{-4}	12.6	0.164	20.0	0.40	0.54	0.76

Table 5-19. DDA Velocity Errors for Test 3 (Table 5-16)

DDA Cycle Time (msec)	Gyro Pulse Quantization (arc sec)	Accelerometer Pulse Quantization (ft/sec)	Time (sec)	V_x (ft/sec)	V_y (ft/sec)	V_z (ft/sec)
2.0	201.4	0.322	5.0	0.18	-0.17	0.23
0.5	50.4	0.164		0.22	-0.16	-0.02
0.1	50.4	0.164		0.22	0.00	0.15
2.0	201.4	0.322	10.0	0.10	0.46	-0.34
0.5	50.4	0.164		-0.07	0.11	0.01
0.1	50.4	0.164		0.09	0.11	0.17
2.0	201.4	0.322	15.0	-0.46	-0.70	1.41
0.5	50.4	0.164		-0.21	-0.20	0.21
0.1	50.4	0.164		-0.21	-0.04	0.38
2.0	201.4	0.322	20.0	2.51	-1.52	-1.32
0.5	50.4	0.164		0.63	-0.38	-0.17
0.1	50.4	0.164		0.79	-0.28	-0.16

Table 5-20. DDA Velocity Errors for Test 4 (Table 5-16)

DDA Cycle Time (msec)	Gyro Pulse Quantization (arc sec)	Accelerometer Pulse Quantization (ft/sec)	Time (sec)	V_x (ft/sec)	V_y (ft/sec)	V_z (ft/sec)
5.0	201.4	0.322	5.0	0.11	-0.07	0.20
1.25	50.4	0.164		0.006	0.07	-0.03
0.25	12.6	0.164		-0.02	0.05	0.14
5.0	201.4	0.322	10.0	0.13	-0.08	-0.15
1.25	50.4	0.164		0.12	0.03	0.06
0.25	12.6	0.164		-0.16	0.07	0.30
5.0*	201.4*	0.322*	15.0	9.73*	-7.3*	42.9*
1.25	50.4	0.164		0.26	0.01	-0.29
0.25	12.6	0.164		-0.05	0.18	0.11
5.0*	201.4*	0.322*	20.0	172.2*	-85.5*	209.5*
1.25**	50.4**	0.164**		2.85**	-0.72**	1.09**
0.25	12.6	0.164		-0.20	0.07	0.07

*This run exceeded the maximum design acceleration at t = 11.5 sec.
**This run exceeded the maximum design acceleration at t = 19.2 sec.

In some cases, the computer roundoff and integration errors in the attitude reference direction cosines and velocities are quite sizeable, — especially in runs with a cycle time of 0.1 msec. The drivers used for the tests are presented in Table 5-14. In Table 5-20, the velocity errors show the effects of exceeding the maximum design acceleration of the DDA.

The DDA diagram of the velocity transformation equations is shown in Figure 5-19. In this figure, the Y registers of the first nine integrators contain the nine direction cosines, the R registers of the first nine integrators contain the portion of the respective velocity increments transformed to inertial coordinates, and the Y registers of the last three integrators contain the accumulated velocity in inertial coordinates. Since the R registers contain as much as one velocity pulse, the velocity component may be in error by as much as three times the velocity quantization. For instance, if the accelerometer output pulses are quantized to 0.05 msec, each inertial velocity component may be in error by 0.15 msec without any of the errors previously mentioned considered. This would amount to a total velocity error of 0.26 m/sec.

The velocity transformation algorithm error analysis is presented in section 5.3 of this report. This error analysis was performed to determine which velocity transformation method was the best method. In that section, a different transformation method was recommended over the method used in the DDA navigation system. Since the DDA cycles at such a fast rate, this velocity transformation algorithm will suffice <u>if</u> the quantization of the input velocity pulses are small enough. The velocity pulses must be small or the increase in accuracy due to the DDA's fast cycle rate will be lost as it requires a finite amount of time for the velocity pulses to build up. For example, a quantization of 0.3 ft/sec pulse while under the influence of a 5g acceleration would cause velocity pulses to be output to the DDA computer every 1.8 msec.

If the DDA were operating at a cycle rate of 10,000 cps, the effective velocity transformation rate would only be 556 cps, and this would cause the velocity transformation algorithm errors to increase. Thus, to take advantage of the DDA's computation speed, the velocity pulses should be quantized to as small a value as possible. For this reason, and the results of the DDA simulation tests, the velocity pulses are recommended to be quantized at a value smaller than the preliminary design value of 0.05 msec, 0.0125 msec for example. By doing this, the velocity transformation algorithm errors would apparently be held to within one tenth of one msec for the 11,000 msec velocity flight. This would also decrease the velocity error due to the velocity quantization from 0.26 msec to 0.06 msec.

The direction cosine errors were presented in Paragraph 5.4.6.1, and were primarily composed of normality and orthogonality errors. The primary effect of these errors is to transform more velocity to inertial coordinates than the vehicle is actually experiencing. For instance, a 0.05 percent normality error would result in a 12.5 ft/sec velocity error for the vehicle to attain a velocity of 25,000 ft/sec. However, the normality error varies from zero to a maximum error (0.003 percent for a 2^{-16} rad direction cosine quantization). If one can assume that the average normality error for a 2^{-16} rad quantization is 0.001 percent, the velocity error due to the direction cosines would be 0.11 msec for a velocity of 11,000 msec.

The total velocity errors for the 11,000 msec flight, using velocity quantization of 0.0125 msec and a direction cosine quantization of 2^{-16} rad, should be ≤ 0.49 msec (3σ).

5.4.7 Conclusions

The recommended design for the DDA is for an angular quantization of 2^{-16} rad, a velocity quantization of 0.0125 msec (0.04101041 ft/sec) and a DDA cycle rate of 10,000 cps or faster. The 10,000 cps speed will allow the DDA to maintain an accurate navigation system while under influence of accelerations up to 410 ft/sec^2 and rotation rates up to 8.75 arc deg/sec. This design was arrived at from the results of the DDA simulation and hand analyses, the important results of which are summarized below.

1. The accuracy of a DDA cannot be improved by an increase in speed alone; the increase in speed must be accompanied by a corresponding decrease in quantization of the input quantities computed in the DDA. If the speed alone is increased and the quantization is not decreased, the DDA will spend these extra cycles calculating on zero input pulses and will achieve exactly the same result as a slower DDA.

2. The mechanization of a first-order Taylor's series algorithm for the maintenance of the attitude reference matrix will result in a nonnormal and nonorthogonal matrix for transformation of the velocity increments to inertial coordinates. Since the normality and orthogonality of the matrix depend on the quantization of the direction cosines, this value should be decreased until the matrix errors come within the desired accuracy limitations.

3. The accuracy of the velocity transformation algorithm is a function of the velocity quantization. Just an increase in DDA speed will not improve this accuracy as explained in (1) above.

4. The velocity errors of the DDA for the simple navigation system are caused by

   a. Attitude Errors

      1) Attitude-reference algorithm

      2) Gyro-pulse quantization

      3) DDA operation (time lags, information stored in R registers is not available).

   b. Velocity Errors

      1) Attitude errors

      2) Velocity-transformation algorithm

      3) Velocity-pulse quantization

      4) DDA operation (time lags, information in R registers is not available).

6. ALIGNMENT STUDIES

6.1 INTRODUCTION AND SUMMARY

Four methods of prelaunch alignment for a strapdown system were analyzed for the MSFC body mounted study.

- Constant gain
- White noise Kalman filter
- Variable gain filter
- Narrow band Kalman filter.

The first three methods were analyzed with a scientific simulation programmed for the IBM 7094 computer. Disappointing results were obtained from the white-noise Kalman filter. Therefore a simplified analysis of the white-noise Kalman filter and a narrow-band Kalman filter was performed to study the filter problem in detail. The poor performance of the white-noise Kalman filter was confirmed and the narrow-band Kalman filter showed promise. However, time did not permit evaluation in the scientific simulation.

All alignment methods are based on the use of a theodolite for azimuth reference information, and the accelerometer outputs for leveling information. The porro prism which reflects the theodolite signal was defined to be in the X_b, Z_b plane (Y_b vertical) and is aligned with (parallel to) the Z_b axis. The matrix to be obtained from the alignment is shown in Equation (6-1), and relates the vehicle body axes with a local level north, east, vertical coordinate system.

$$\begin{bmatrix} x_b \\ y_b \\ z_b \end{bmatrix} = \begin{bmatrix} e_{11} & e_{12} & e_{13} \\ e_{21} & e_{22} & e_{23} \\ e_{31} & e_{32} & e_{33} \end{bmatrix} \begin{bmatrix} 1_N \\ 1_V \\ 1_E \end{bmatrix} \tag{6-1}$$

where

x_b, y_b, z_b = vehicle yaw, roll, and pitch axes, respectively.

$1_N, 1_V, 1_E$ = unit vectors along the north, vertical and east directions

$\left[e_{ij}\right]$ = orthogonal matrix relating the two systems.

The final matrix relating the vehicle body axes to the desired inertial navigation axes is found by a simple rotation about the 1_V axis as the Saturn inertial guidance and navigation system is also local level at launch.

$$\begin{bmatrix} x_b \\ y_b \\ z_b \end{bmatrix} = [e_{ij}] \begin{bmatrix} \cos A_z & 0 & -\sin A_z \\ 0 & 1 & 0 \\ \sin A_z & -0 & \cos A_z \end{bmatrix} \begin{bmatrix} x_s \\ y_s \\ z_s \end{bmatrix} \qquad (6\text{-}2)$$

where

x_s, y_s, z_s = inertial reference axes

A_z = launch azimuth of the vehicle clockwise from true north.

That is, at launch y_s is vertical, x_s at an azimuth A_z clockwise from north, and $\bar{z}_s = \bar{x}_s \times \bar{y}_s$. The vehicle's trajectory is in the x_s, y_s plane.

The only errors which were considered in the analysis of the strapdown alignment methods were accelerometer bias errors, gyro constant drift errors, and the errors due to vehicle sway, which were not completely filtered out by the alignment methods. The vehicle sway used in the computer simulation was derived completely from the vehicle sway data obtained from MSFC. These vehicle sway data showed the Saturn wind induced motion on a launcher, including lateral oscillations and accelerations versus station (3250 in. for MSFC). They indicated that the vehicle is deflected away from the wind an angle γ_o, and oscillates perpendicular to the wind a maximum angle α_o with a corresponding maximum distance d_o. These lateral oscillations cause the accelerometers to pick up accelerations other than gravity which must be filtered out by the alignment method used. Typical vehicle sway data are shown below for a 99.9 percent wind. In all cases, only the above fueled frequencies and amplitudes were used in the scientific simulation tests.

	γ_o (rad)	α_o (rad)	d_o (in.)	f (cps)
Fueled	0.01025	0.00432	8.0	0.33
Unfueled	0.0134	0.00424	7.7	0.8

Using these values for vehicle sway (and no gyro or accelerometer errors), the constant gain filter required 26 minutes to attain the desired accuracy of 20 arc sec. The Kalman and variable gain filters showed no improvement over the constant gain filter as the high gains of these two filters at the start of the alignment caused a very large initial misalignment error due to the accelerometer measurement errors. For this reason a constant gain filter appears to be the best alignment method of the three tested in the scientific simulation. The simplified analysis showed that the narrow-band Kalman filter was capable of reducing the level alignment error below 20 arc sec in 16 min or less depending upon the correlation time of the swaying motion. If this performance could be verified in a full scientific simulation, then it would be the best alignment method.

The scientific simulation is described completely in Section 6.2, the theodolite correction method is described in Section 6.2.5.2, the alignment equations are described in Section 6.2.5.3, and the alignment results are presented in Section 6.3. The simplified analysis is presented in Section 6.4.

6.2 SCIENTIFIC SIMULATION DESCRIPTION

A functional block diagram of the scientific simulation developed for the analysis of prelaunch alignment methods is shown in Figure 6-1. The seven blocks in this diagram will be discussed in this section, and the equations used for their mechanization explained.

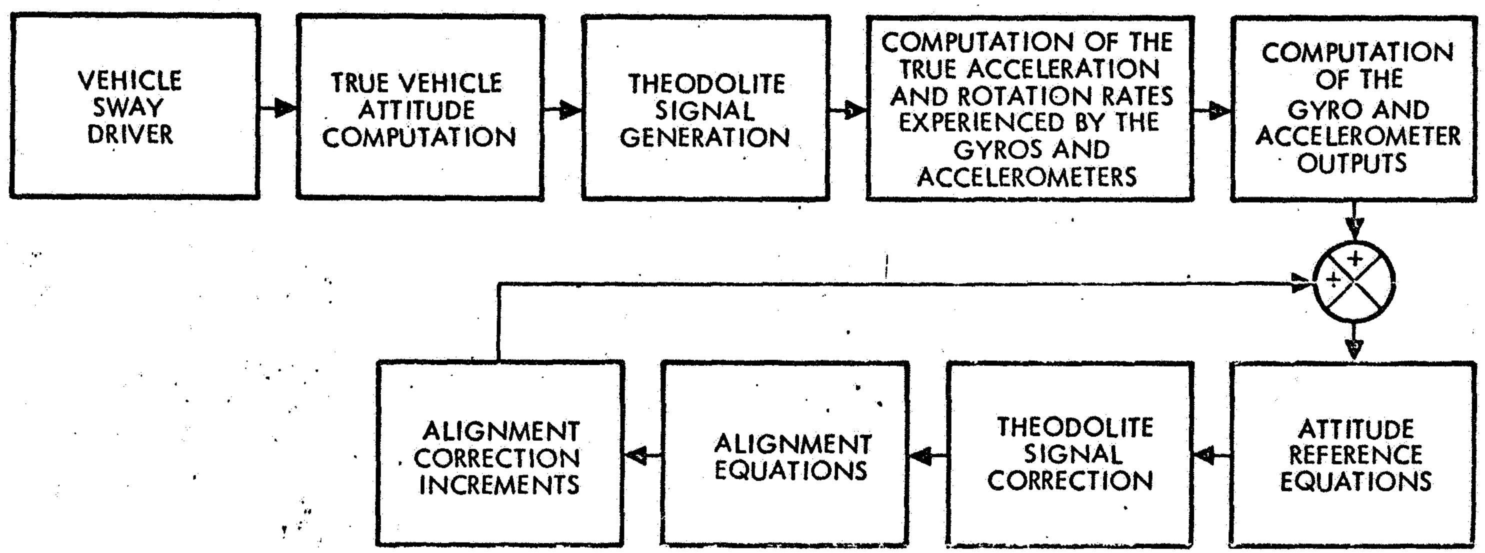


Figure 6-1. Functional Block Diagram of the Alignment Scientific Simulation

6.2.1 Vehicle Sway Driver

As mentioned briefly in the summary, the vehicle sway data were obtained from MSFC. These data showed the lateral oscillations and accelerations which would be experienced by a Saturn vehicle at any station desired. They were presented for many different wind probabilities, 99.9, 99, and 95 percent, and for both a fueled and an unfueled vehicle. The primary tests performed on the simulation were for 99.9 percent wind probability and a fueled vehicle.

When a Saturn vehicle is under the influence of a constant wind, the vehicle deflects or leans away from the wind and, if the wind is of sufficient strength, oscillates in a direction perpendicular to the wind. The vehicle sway data supplied by MSFC are graphs of the lateral deflection and lateral oscillations in inches versus the vehicle station in inches. Thus, the angle the vehicle is leaning at any station may be obtained by finding the slope of these graphs, and taking the inverse tangent of the slopes. Since the wind is constant, the deflection angle is assumed to be constant.

The lateral oscillations graphically presented are evidently the maximum lateral oscillation angles, and are also of the form $\alpha = \alpha_o \sin \omega_o t$ as the frequency is also quoted. Vehicle sway motion may be summarized as follows:

(1) The vehicle is deflected away from the wind an angle γ_o

(2) The vehicle oscillates an angle α. The oscillation frequency is 0.8 cps unfueled and 0.33 cps fueled.

The vehicle sway data, for example, at a station of 3250 in. and a 99.9 percent wind probability are:

	α_o (rad)	γ_o (rad)	d_o (in.)	f (cps)
Fueled	0.01025	0.00432	8.0	0.33
Unfueled	0.0134	0.00424	7.7	0.8

6.2.2 True Vehicle Attitude Computation

In the scientific simulation, the true vehicle attitude is computed by the multiplication of matrices. Thus the true vehicle attitude does not contain any algorithm errors as would be the case if the true attitude were maintained by a Runge-Kutta integration algorithm or equivalent.

Before any wind is applied to the vehicle, the relationship between the x_b, y_b, z_b axes and the 1_N, 1_V, 1_E axes is shown in Equation (6-3). This provides a capability for initial misalignment of the inertial instrument package to be input to the simulation.

$$\begin{bmatrix} x_b \\ y_b \\ z_b \end{bmatrix} = [b_{ij}] \begin{bmatrix} 1_N \\ 1_V \\ 1_E \end{bmatrix} \qquad (6\text{-}3)$$

The constant wind which is to be simulated may come from any azimuth Λ_o clockwise from north. Since rotations are not commutative, a new local level coordinate system, D, E, F will be defined with the E axis collinear with the wind azimuth.

$$\begin{bmatrix} 1_N \\ 1_V \\ 1_E \end{bmatrix} = \begin{bmatrix} \cos \Lambda_o & 0 & -\sin \Lambda_o \\ 0 & 1 & 0 \\ \sin \Lambda_o & 0 & \cos \Lambda_o \end{bmatrix} \begin{bmatrix} E \\ F \\ D \end{bmatrix} = [\Lambda_{ij}] \begin{bmatrix} E \\ F \\ D \end{bmatrix} \qquad (6\text{-}4)$$

The wind causes the vehicle to deflect away from the wind an angle γ_o which is a clockwise rotation γ_o about the E axis.

$$\begin{bmatrix} D \\ F \\ E \end{bmatrix} = \begin{bmatrix} \cos \gamma_o & -\sin \gamma_o & 0 \\ \sin \gamma_o & \cos \gamma_o & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} E' \\ F' \\ D' \end{bmatrix} = [\gamma_{ij}] \begin{bmatrix} E' \\ F' \\ D' \end{bmatrix} \qquad (6\text{-}5)$$

After the wind reaches sufficient strength, the vehicle oscillates back and forth in a direction perpendicular to the wind. This is equivalent to a clockwise rotation $\alpha = \alpha_o \sin \omega_o t$ about the D' axis.

$$\begin{bmatrix} D' \\ F' \\ E' \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix} \begin{bmatrix} E'' \\ F'' \\ D'' \end{bmatrix} = [\alpha_{ij}] \begin{bmatrix} E'' \\ F'' \\ D'' \end{bmatrix} \qquad (6\text{-}6)$$

From these matrix relationships, the true vehicle attitude may now be computed for the D", F", E" and x_b, y_b, z_b axes are related as shown in Equation (6-7), as both sets of axes are body fixed coordinates.

$$\begin{bmatrix} E'' \\ F'' \\ D'' \end{bmatrix} = \Lambda^T_{ij} b^T_{ij} \begin{bmatrix} x_b \\ y_b \\ z_b \end{bmatrix} \qquad (6\text{-}7)$$

Thus

$$\begin{bmatrix} l_N \\ l_V \\ l_E \end{bmatrix} = [\Lambda_{ij}]\,[\gamma_{ij}]\,[\alpha_{ij}]\,[\Lambda^T_{ij}]\,[b^T_{ij}] \begin{bmatrix} x_b \\ y_b \\ z_b \end{bmatrix} \qquad (6\text{-}8)$$

6.2.3 Vehicle Sway Acceleration and Rotation Rate Computation

The vehicle sway motion at the 3250 in station is assumed to be a sinusoidal motion of the form (based on the MSFC vehicle sway information)

$$d = d_o \sin \omega_o t \qquad (6\text{-}9)$$

and the corresponding lateral oscillation is of the form

$$\alpha = \alpha_o \sin \omega_o t \qquad (6\text{-}10)$$

The lateral distance the vehicle moves and the vehicle sway rotation rate are, in vector notation

$$
\begin{aligned}
\bar{d} &= d_o \sin \omega_o t \bar{D}'' \\
\bar{\omega}_s &= \dot{\alpha} \bar{E}'' = \alpha_o \omega_o \cos \omega_o t \bar{E}'' = \omega_s \bar{E}''
\end{aligned}
\qquad (6\text{-}11)
$$

where

d_o = maximum lateral deflection in feet the vehicle will experience

ω_o = radian frequency of the vehicle's lateral oscillation

α_o = maximum angle the vehicle rotates during the lateral oscillations

$\bar{D}''$ = unit vector along the D'' axis

$\bar{E}''$ = unit vector along the E'' axis

The velocity of the vehicle due to vehicle sway is computed from the equations for $\bar{d}$ and $\bar{\omega}$ by taking the derivative of $\bar{d}$ with respect to time.

$$
\begin{aligned}
\dot{\bar{d}} &= d_o \omega_o \cos \omega_o t \bar{D}'' + d_o \sin \omega_o t \dot{\bar{D}}'' \\
\dot{\bar{d}} &= d_o \omega_o \cos \omega_o t \bar{D}'' + d_o \sin \omega_o t \left[\alpha_o \omega_o \cos \omega_o t\right] \bar{F}''
\end{aligned}
\qquad (6\text{-}12)
$$

The acceleration experienced by the accelerometers is the time derivative of $\dot{\bar{d}}$

$$
\begin{aligned}
\ddot{\bar{d}} = &- d_o \omega_o^2 \sin \omega_o t \bar{D}'' + \alpha_o d_o \omega_o^2 \cos^2 \omega_o t \bar{F}'' - \alpha_o \omega_o^2 d_o \sin^2 \omega_o t \bar{F}'' \\
&+ d_o \omega_o \cos \omega_o t \left[\alpha_o \omega_o \cos \omega_o t\right] \bar{F}'' \qquad (6\text{-}13) \\
&- d_o \sin \omega_o t \left[\alpha_o \omega_o \cos \omega_o t\right]^2 \bar{D}''
\end{aligned}
$$

$$\ddot{\bar{d}} = \bar{D}'' \left[-d_o \omega_o^2 \sin \omega_o t - d_o \sin \omega_o t \, \alpha_o^2 \, \omega_o t \, \alpha_o^2 \, \omega_o^2 \cos^2 \omega_o t \right]$$

$$+ \bar{F}'' \left[\begin{array}{l} \alpha_o d_o \omega_o^2 \cos^2 \omega_o t - \alpha_o d_o \omega_o^2 \sin^2 \omega_o t \\ \qquad\qquad + \alpha_o d_o \omega_o^2 \cos^2 \omega_o t \end{array} \right] \tag{6-14}$$

$$\ddot{\bar{d}} = A_D \bar{D}'' + A_F \bar{F}'' \tag{6-15}$$

$$A_D = - d_o \omega_o^2 \sin \omega_o t \left[1 + \alpha_o^2 \cos^2 \omega_o t \right]$$

$$A_D = - d_o \omega_o^2 \sin \omega_o t \left[1 + \frac{\alpha_o^2}{2} \left(1 + \cos 2 \omega_o t \right) \right] \tag{6-16}$$

$$A_F = \alpha_o d_o \omega_o^2 \left[2 \cos^2 \omega_o t - \sin^2 \omega_o t \right]$$

$$A_F = \alpha_o d_o \omega_o^2 \left[\cos^2 \omega_o t + \cos 2 \omega_o t \right]$$

$$A_F = \frac{\alpha_o d_o \omega_o^2}{2} \left[1 + \cos 2 \omega_o t + 2 \cos 2 \omega_o t \right]$$

$$A_F = \alpha_o d_o \omega_o^2 \left[1 + 3 \cos 2 \omega_o t \right] \tag{6-17}$$

The acceleration experienced by the accelerometers is due to vehicle sway and gravity. From the above calculations, the total acceleration is

$$\bar{A} = g \hat{1}_V + A_D \bar{D}'' + A_F \bar{F}'' \tag{6-18}$$

Similarly, the total rotation rates experienced by the gyros is composed of earth rate and the rotational rate caused by vehicle sway. The total rotational velocity is

$$\bar{\omega}_s = \omega_e \cos \lambda \, \hat{1}_N + \omega_e \sin \lambda \, \hat{1}_V + \omega_s \bar{E}'' \tag{6-19}$$

Since the relationship between the vehicle body axes and the several co... nate axes are known, these accelerations and angular velocities are ea... converted to accelerations and angular velocities in body coordinates ... the matrix relationships described in Section 6.2.2. These matrix pr... will be redefined below so one variable may be used in this conversion process. By definition

$$\begin{bmatrix} E'' \\ F'' \\ D'' \end{bmatrix} = [S_{ij}] \begin{bmatrix} X_b \\ Y_b \\ Z_b \end{bmatrix} = [\Lambda^T_{ij}]\,[b^T_{ij}] \begin{bmatrix} X_b \\ Y_b \\ Z_b \end{bmatrix} \quad (6\text{-}2\ldots$$

$$\begin{bmatrix} l_N \\ l_V \\ l_E \end{bmatrix} = [h_{ij}] \begin{bmatrix} X_b \\ Y_b \\ Z_b \end{bmatrix} = [\Lambda_{ij}]\,[\gamma_{ij}]\,[\alpha_{ij}]\,[S_{ij}] \begin{bmatrix} X_b \\ Y_b \\ Z_b \end{bmatrix} \quad (6\text{-}2\ldots$$

The rotation rates and accelerations experienced by the inertial instrume... using these definitions are

$$\begin{aligned} \omega_{xg} &= \omega_e \cos\lambda\, h_{11} + \omega_e \sin\lambda\, h_{21} + \omega_s C_{11} \\ \omega_{yg} &= \omega_e \cos\lambda\, h_{12} + \omega_e \sin\lambda\, h_{22} + \omega_s C_{k2} \\ \omega_{zg} &= \omega_e \cos\lambda\, h_{13} + \omega_e \sin\lambda\, h_{23} + \omega_s C_{13} \end{aligned} \quad (6\text{-}22)$$

$$\begin{aligned} A_{xa} &= gh_{21} + A_F S_{21} + A_D S_{31} \\ A_{ya} &= g\, h_{22} + A_F S_{22} + A_D S_{32} \\ A_{za} &= gh_{23} + A_F S_{23} + A_D S_{33} \end{aligned} \quad (6\text{-}23)$$

where

$\omega_{xg}, \omega_{yg}, \omega_{zg}$ = rotational rates experienced by the X_b, Y_b, Z_b strapdown gyros, respectively

A_{xa}, A_{ya}, A_{za} = accelerations experienced by the X_b, Y_b, Z_b strapdown accelerometers, respectively

The accelerometer outputs already compensated for angular rates and gyro outputs as mechanized in the scientific computer simulation are:

$$\Delta V_{xb} = A_{xa}\,\Delta t_1 + \Delta V_{xb} + b_{xa}\,\Delta t_1$$
$$\Delta V_{yb} = A_{ya}\,\Delta t_1 + \Delta V_{yb} + b_{ya}\,\Delta t_1 \qquad (6\text{-}24)$$
$$\Delta V_{zb} = A_{za}\,\Delta t_1 + \Delta V_{yb} + b_{za}\,\Delta t_1$$

$$\Delta\alpha_1 = \omega_{xg}\,\Delta t_1 + \Delta\alpha_1 + \Delta\alpha_{xa} + b_1\,\Delta t_1$$
$$\Delta\alpha_2 = \omega_{yg}\,\Delta t_1 + \Delta\alpha_2 + \Delta\alpha_{ya} + b_2\,\Delta t_1 \qquad (6\text{-}25)$$
$$\Delta\alpha_3 = \omega_{zg}\,\Delta t_1 + \Delta\alpha_3 + \Delta\alpha_{za} + b_3\,\Delta t_1$$

where

$\Delta V_{xb}, \Delta V_{yb}, \Delta V_{zb}$ = velocity increments accumulated by the X_b, Y_b, Z_b accelerometers (ft/sec)

b_{xa}, b_{ya}, b_{za} = accelerometer bias errors (ft/sec^2)

$\Delta\alpha_1, \Delta\alpha_2, \Delta\alpha_3$ = attitude increments accumulated by the gyros about the X_b, Y_b, Z_b axes (rad)

b_1, b_2, b_3 = constant drift errors of the body mounted gross (rad/sec)

Δt = time interval over which the incremental integration is computed.

6.2.4 Theodolite Reference Signal Generation

The azimuth reference signal is provided by a theodolite ... located along the astronomical north axis for simulation purposes, ... an angle of inclination of 26 degrees with the horizontal. Due to th... clination angle, the theodolite measures the azimuth with an error ... tional to the tangent of the inclination angle and the deflection angle ... vehicle.

The porro prism is aligned with the Z body axis in the X_b - ... The reflected signal is essentially a dot product of the Z_b axis with a ... collinear with the line of sight from the theodolite to the porro prism.

The unit vector defining the theodolite's line of sight to the porro prism is

$$\hat{i}_{th} = \cos i\ \hat{i}_N - \sin i\ \hat{i}_V$$

where

- i = theodolite's angle of inclination with the horizontal

From Section 6.2.3, the Z_b axis is defined to be

$$\hat{Z}_b = h_{13}\ \hat{i}_N + h_{23}\ \hat{i}_V + h_{33}\ \hat{i}_E$$

Thus, the theodolite signal U_{zm} is:

$$\hat{Z}_b \cdot \hat{i}_{th} = h_{13} \cos i - \sin i h_{23}$$

$$U_{zm}' = \frac{\hat{Z}_b \cdot \hat{i}_{th}}{\cos i} = h_{13} - \tan i h_{23}$$

The error in the theodolite signal is $(\tan i)\ (h_{23})$ which is approximately one-half the vehicle deflection about the X_b axis for a 26-degree inclination angle.

6.2.5 Flight Computer Equations

After the true vehicle attitude has been computed, the true accelerations and rotation rates and the resulting inertial instrument outputs are obtained, the flight computer equations which are used during alignment are then processed. These equations which include the attitude reference equations, theodolite signal correction equations, and the alignment equations are then processed in the order they would be used in the flight computer.

6.2.5.1 Attitude Reference Equations

The primary purpose of the attitude reference equations in a strapdown system is to update and maintain the direction cosine matrix that relates the vehicle axes to the desired navigation or guidance coordinate system. For the Saturn application, the attitude reference equations update the matrix that relates the X_b, Y_b, Z_b body axes to the inertial navigation coordinate axes X_s, Y_s, Z_s.

$$\begin{bmatrix} X_b \\ Y_b \\ Z_b \end{bmatrix} = \begin{bmatrix} a_{ij} \end{bmatrix} \begin{bmatrix} X_s \\ Y_s \\ Z_s \end{bmatrix} \qquad (6\text{-}29)$$

where

X_b = vehicle yaw axis

Y_b = vehicle roll axis

Z_b = vehicle pitch axis

X_s, Y_s, Z_s = inertial navigation axes, with Y_s vertical at launch, X_s at an azimuth Z_z clockwise from north,

and

Z_s = $(\overline{X}_s) \times (\overline{Y}_s)$. The vehicle's trajectory is in the X_s, Y_s plane

The alignment of the body mounted system is accomplished by obtaining the matrix which relates the vehicle body axes to a local level system with one axis north, one east, and the third vertical.

$$\begin{bmatrix} X_b \\ Y_b \\ Z_b \end{bmatrix} = \begin{bmatrix} e_{ij} \end{bmatrix} \begin{bmatrix} 1_n \\ 1_v \\ 1_E \end{bmatrix} \tag{6-30}$$

where

X_b, Y_b, Z_b = vehicle yaw, roll, and pitch axes, respectively

1_n, 1_v, 1_E = unit vectors defining the local level system: 1_n north, 1_E east, and 1_v vertical

e_{ij} = an orthogonal direction cosine matrix which is obtained during alignment and relates the vehicle body axes to the local level system.

When the attitude reference equations go inertial, the $[a_{ij}]$ matrix which relates the vehicle body axes to the inertial reference coordinate system by multiplying the $[e_{ij}]$ matrix by another matrix as shown below

$$[a_{ij}] = [e_{ij}] \begin{bmatrix} \cos A_z & 0 & -\sin A_z \\ 0 & 1 & 0 \\ \sin A_z & 0 & \cos A_z \end{bmatrix} \tag{6-31}$$

where

$[a_{ij}]$ = orthogonal matrix relating the vehicle body axes and the X_s, Y_s, Z_s coordinate axes.

$[e_{ij}]$ = orthogonal matrix relating the vehicle body axes and the local level north system.

A_z = vehicle launch azimuth clockwise from north.

The attitude reference equations update the direction cosine matrix with a second-order Taylor's series algorithm. This algorithm is shown in matrix notation below exactly as mechanized in the alignment simulation.

$$[a_{ij}]_n = \begin{bmatrix} 1 - \frac{1}{2}(\Delta\alpha_2^2 + \Delta\alpha_3^2) & \Delta\alpha_3 + 0.5\,\Delta\alpha_1\,\Delta\alpha_2 & -\Delta\alpha_2 + 0.5\,\Delta\alpha_1\,\Delta\alpha_3 \\ -\Delta\alpha_3 + 0.5\,\Delta\alpha_1\,\Delta\alpha_2 & 1 - \frac{1}{2}(\Delta\alpha_1^2 + \Delta\alpha_3^2) & \Delta\alpha_1 + 0.5\,\Delta\alpha_2\,\Delta\alpha_3 \\ \Delta\alpha_2 + 0.5\,\Delta\alpha_1\,\Delta\alpha_3 & -\Delta\alpha_1 + 0.5\,\Delta\alpha_2\,\Delta\alpha_3 & 1 - 0.5\,(\Delta\alpha_2^2 + \Delta\alpha_3^2) \end{bmatrix} [a_{ij}]_{(n-1)} \qquad (6\text{-}32)$$

6.2.5.2 Theodolite Signal Correction Equations

As described in Section 6.2.4, the theodolite signal is in error by an amount proportional to the tangent of the theodolite's inclination angle times the 1_v, Z_b direction cosine. Two correction methods are mechanized in the computer simulation as shown below

$$\Delta VAC = \Delta VAC + \Delta V_z$$

$$\Delta\eta = \frac{C_1 \Delta V_z}{G T_a} + C_2 \frac{\Delta VAC}{Gt} \qquad (6\text{-}33)$$

$$e_{31D} = U'_{zm} + K_1 e_{32} + K_2 \Delta\eta$$

The first method of correcting the theodolite signal is to add the tangent of i times the e_{32} direction cosine to the theodolite signal U'_{zm}. This is accomplished by setting K_2 to zero and setting K_1 equal to the tangent of i.

The second method is to use the velocity increments from the z accelerometer

$$a_{31D} = U_{zm}' + \tan i \frac{\Delta V_z}{GT_a} \qquad (6-34)$$

This method is mechanized by setting K_2 equal to the tangent of i and C_1 equal to 1.

The third method is to accumulate the z accelerometer outputs and then compensate by adding this term to the theodolite signal.

$$a_{31D} = U_{zm}' + \tan i \frac{\left[\sum_{t=0}^{T} \Delta V_z\right]}{GT} \qquad (6-35)$$

This method is mechanized by setting C_1 to zero, C_2 to 1, K_2 equal to the tangent of i. Of the three methods, this method reduces the azimuth error due to the theodolite the best.

6.2.5.3 Alignment Equations

Four different alignment methods were tested on the computer simulation, and are shown in Figures 6-2, 6-3, and 6-4. These methods include two constant gain methods, one Kalman filter method, and one variable gain method.

The equations shown in the next few pages do not include the initial direction cosine matrix computation. The initial direction cosine matrix is computed as shown in Figure 6-2 except using the velocities accumulated over 240 sec.

6.3 ALIGNMENT SIMULATION RESULTS

Three alignment methods were tested on the scientific computer simulation. The primary difference between these methods was the alignment time required, as they were all essentially low gain filters. In the next few paragraphs, the results for the constant, white noise Kalman, and variable gain filters are presented. In all cases, the theodolite signal is corrected using the method selected in Phase I [Equation (6-36)] .

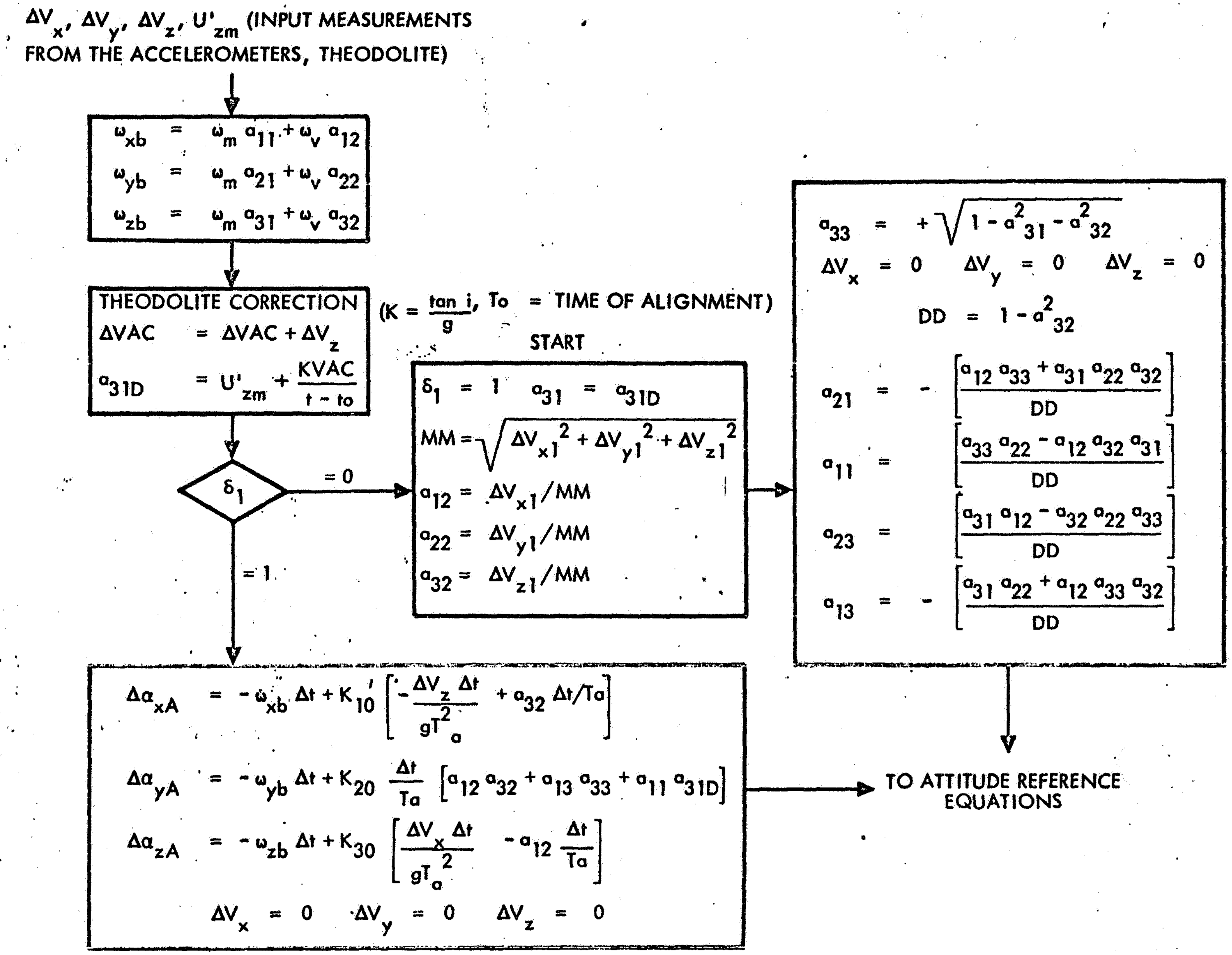


Figure 6-2. Constant Gain Alignment Method

$\Delta V_x, \Delta V_y, \Delta V_z, U'_{zm}$

$$\omega_{xb} = \omega_n a_{11} + \omega_v a_{12}$$
$$\omega_{yb} = \omega_n a_{21} + \omega_v a_{22}$$
$$\omega_{zb} = \omega_n a_{31} + \omega_v a_{32}$$

THEODOLITE CORRECTION

$$\Delta VAC = \Delta VAC + \Delta V_z$$
$$a_{31D} = U'_{zm} + \frac{K \Delta VAC}{t - to}$$

$$MM = +\sqrt{\Delta V_x^2 + \Delta V_y^2 + \Delta V_z^2}$$
$$a_{12D} = \Delta V_x / MM$$
$$a_{22D} = \Delta V_y / MM$$
$$a_{32D} = \Delta V_z / MM$$
$$a_{33D} = +\sqrt{1 - a^2_{31D} - a^2_{32D}}$$
$$\Delta V_x = \Delta V_y = \Delta V_z = 0$$
$$DD = 1 - a^2_{32D}$$

$$a_{21D} = \frac{1}{DD}\left[-a_{12D}\, a_{33D} - a_{31D}\, a_{22D}\, a_{32D}\right]$$
$$a_{11D} = \frac{1}{DD}\left[a_{33D}\, a_{22D} - a_{12D}\, a_{32D}\, a_{31D}\right]$$
$$a_{23D} = \frac{1}{DD}\left[a_{31D}\, a_{12D} - a_{32D}\, a_{22D}\, a_{33D}\right]$$
$$a_{13D} = -\frac{1}{DD}\left[a_{31D}\, a_{22D} \cdot a_{12D}\, a_{33D}\, a_{32D}\right]$$

δ_1

$\delta_1 = 0$

$$\delta_1 = 1$$
$$[a_{ij}] = [a_{ijD}]$$
$$K_{17}' = K_{17}$$

$\delta_1 = 1$

$$E_x = a_{21D}\, a_{31} + a_{22D}\, a_{32} + a_{23D}\, a_{33}$$
$$E_y = a_{11D}\, a_{31} + a_{12D}\, a_{32} + a_{13D}\, a_{33}$$
$$E_z = a_{11D}\, a_{21} + a_{12D}\, a_{22} + a_{13D}\, a_{23}$$

$$K_{12}' = K_{12} + K_{24} K_{27}$$
$$K_{17}' = K_{17} + K_{24} (K_{12} + K_{12}')$$
$$\Delta = K_{13} + K_{17}'$$
$$Q_1 = K_{17}'/\Delta \quad Q_2 = K_{12}'/\Delta$$

$$K_{17} = K_{17}' [1 - Q_1]$$
$$K_{27} = K_{27} - K'_{12} Q_2$$
$$K_{12} = K_{12}' [1 - Q_1]$$

$$\Delta\alpha_{xA} = -\omega_{xb} \Delta t + Q_1 E_x \Delta t / Ta$$
$$\Delta\alpha_{yA} = -\omega_{yb} \Delta t + Q_1 E_y \Delta t / Ta$$
$$\Delta\alpha_{zA} = -\omega_{zb} \Delta t + Q_1 E_z \Delta t / Ta$$

TO ATTITUDE REFERENCE EQUATIONS

(INPUT MEASUREMENTS

Figure 6-3. Kalman Filter Alignment Method

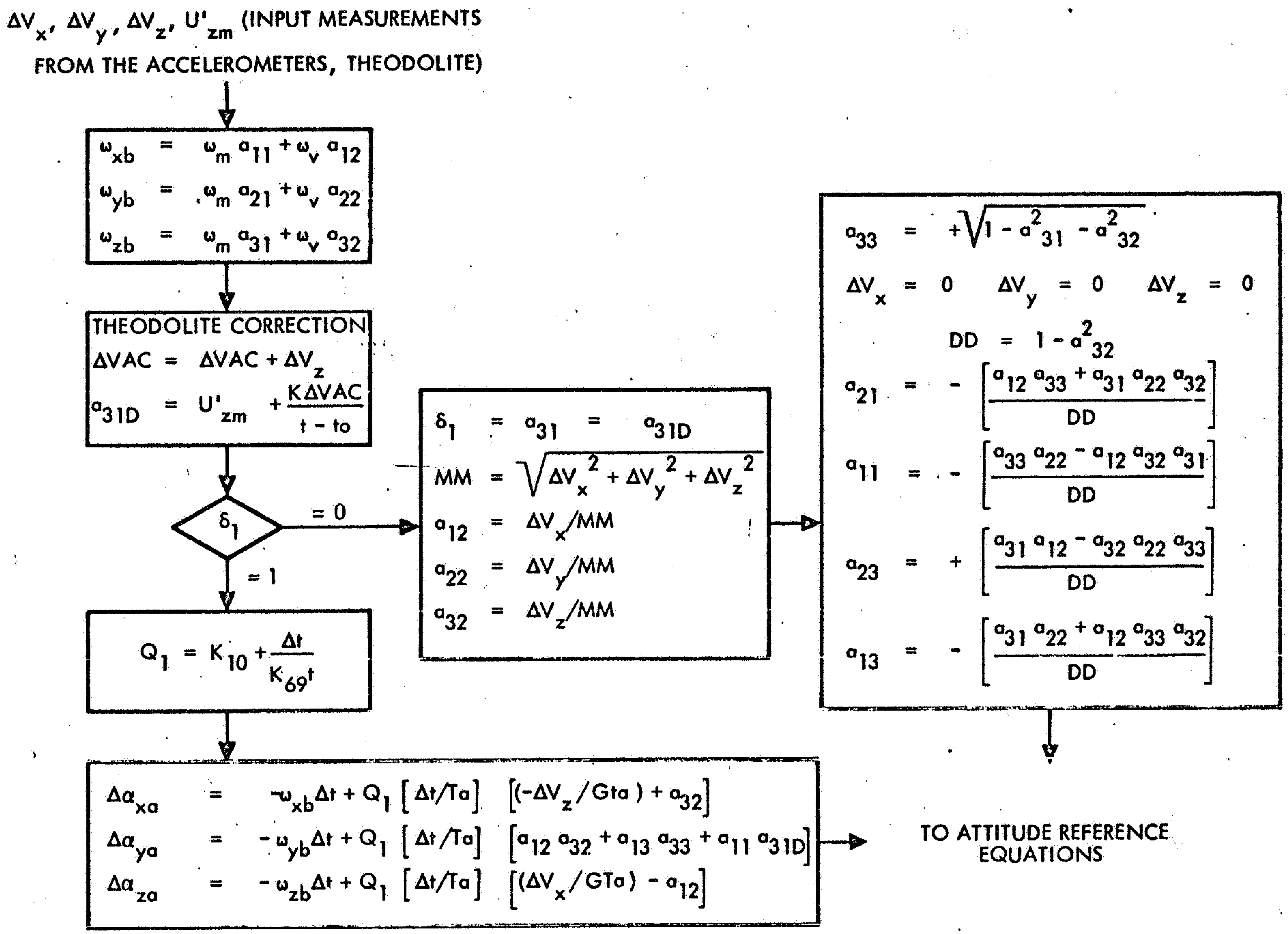


Figure 6-4. Variable Gain Alignment Method

$$A_{31D} = U'_{zm} + \frac{\tan i}{g(t - T_o)} \left[\sum_{T_o}^{t - T_o} \Delta V_z \right]$$

where

U'_{zm} = theodolite signal

i = angle of inclination of the theodolite

T_o = time the alignment is started (sec)

g = gravity in ft/sec^2

ΔV_z = output of the z accelerometer in ft/sec

t = time in sec

In Phase I of the MSFC body mounted study, the constant gain filter aligned to the desired accuracy of 20 arc sec in 1 hour. This alignment was achieved using an initial direction cosine matrix equal to the identity matrix and with alignment gains of 2.5 x 10^{-5}.

Since it was desirable to reduce this alignment time, an initial direction cosine matrix was computed in lieu of using the identity matrix, as this decreased the initial attitude error from 3600 arc sec to less than 900 arc sec. The computation of the initial direction cosine matrix is accomplished using the equations shown on the following page. This initial direction cosine matrix is essentially the average direction cosine matrix as the velocity increments used were accumulated over a relatively long time, 3 to 4 minutes. If the vehicle deflects to a constant position, the alignment may be concluded after the matrix is computed. However, since the vehicle is swaying in the wind, this matrix is only an average direction cosine matrix and may be in error by α_o degrees — the maximum sway angle. During the 99.9 percent winds which were used to test the alignment, α_o is 0.00434 rad or 895 arc sec. Thus, after the average direction cosine matrix has been computed, the maximum attitude error is 900 arc sec and this error must be reduced to less than 20 arc sec with a filtering process.

The constant gain filter, using gains of 5×10^{-5}, reduced this error to less than 20 arc sec in 25 minutes — a considerable improvement over the results of Phase I.

The Kalman filter, although expected to finish the alignment much faster and with better accuracy than the constant gain filter, did not show any improvement over the constant gain filter. Due to the high gains of the Kalman filter at alignment start (0.5), and the accelerometer measurement errors, the alignment was very unstable at first as the initial alignment error increased from 900 arc sec to 16,000 arc sec. After approximately 2 minutes, the Kalman filter started to converge to the desired alignment in an oscillatory manner indicating that the white-noise Kalman filter used was definitely not the best filter to solve the alignment problem. This result occurs because of the fundamental mismatch between the sinusoidal errors and the white-noise Kalman error model.

In an attempt to achieve a faster alignment, a least-squares fit method was also tested on the computer simulation. This method computed a variable alignment gain which decreased to a desired steady state alignment gain inversely proportional to time. Since the initial alignment

$$K_{10} = K_{10f} + \frac{1}{K(t - T_a)} \qquad (6\text{-}37)$$

where

K_{10} = alignment gain

K_{10f} = desired steady state alignment gain

t = time in seconds

T_a = time the alignment starts

K = constant which controls the rate of change of the alignment gain

gain did not start as high as the Kalman filter gain, this method did not diverge like the Kalman filter. However, this filter caused the initial attitude error to diverge from 900 to 1400 arc sec before starting to reduce the alignment error. Thus, this method also did not compare favorably with the constant gain approach.

6.4 NARROW-BAND KALMAN FILTER FOR VERTICAL ALIGNMENT

The performance of the white-noise Kalman filter in the scientific simulation was disappointing. It was hypothesized that the reason for the poor performance was an inadequate representation of the errors caused by the swaying acceleration of the vehicle by the white noise model, since these errors are sinusoidal or nearly so. Therefore, a Kalman filter in which the sway error is modeled as a narrow-band gaussian process was also studied. This type of filter is considerably more complicated than the white noise filter, and is therefore more demanding in its computer capacity requirements.

6.4.1 General Approach

It would have been possible to modify the scientific simulation program to include the NBKF (narrow-band Kalman filter) as an option. However, the NBKF might have offered no advantage over the present techniques, so it appeared wiser to examine it thoroughly in a special IBM 7094 computer program before such action.

This special program is a linearized, statistical, single axis error analysis. Thus quantization and roundoff errors are neglected. There is no simulation using deterministic or Monte Carlo inputs; rather the second-order statistics of the inputs and the model of the system are used to calculate the second-order statistics of the alignment error. Only one axis is considered. Since the program is an error analysis, only the errors are calculated. That is, there is no direct representation of a platform being torqued or a direction cosine matrix being slewed, and the results apply equally well to either. No error source other than the swaying motion of the vehicle is considered.

6.4.2 Narrow Band Noise

A narrow-band gaussian process may be represented by

$$x = x_c \cos \omega t + x_s \sin \omega t \tag{6-38}$$

where x_c and x_s are independent gaussian random processes having the same spectrum which is band limited about zero frequency. For convenience in this study, x_c and x_s are taken as first order Gauss-Markov processes. Their spectra are thus not really band limited, but perhaps the actual sway error spectra aren't either. Figure 6-5 shows the resulting model of the measurements. In the figure:

θ = actual misalignment

θ_m = measured misalignment

U_c, U_s = independent Gaussian white noises of equal intensity

n_c, n_s = independent Gauss-Markov processes with equal spectra

n = narrow band noise

ω = center frequency of the narrow band noise

τ = time constant of the Gauss-Markov process

s = Laplace operator

t = time

The standard deviations of the noises n, n_c, and n_s, are all equal

$$\sigma_n = \sigma_{n_c} = \sigma_{n_s} \tag{6-39}$$

The purpose of the NBKF is to estimate θ, given the measurements θ_m, n_c and n_s are also estimated as by-products.

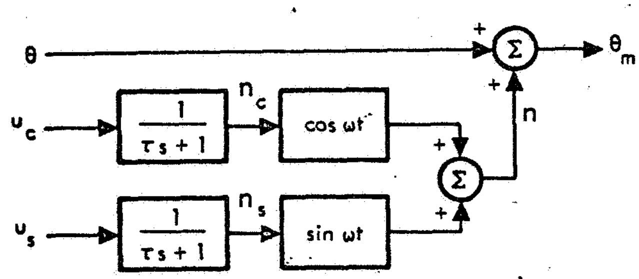


Figure 6-5. Measurement Model

6.4.3 Kalman Filter

The following nomenclature is used:

x = state vector including system and error states

$\hat{x}$ = estimate of x

$\tilde{x}$ = error in $\hat{x}$ ($\tilde{x} = \hat{x} - x$)

y = perfect measurement of a linear combination of states

$\hat{y}$ = estimate of y calculated from $\hat{x}$

M = measurement matrix $\left(M = \frac{\partial y}{\partial x}\right)$

K = filter matrix

Φ = state transition matrix

P = covariance matrix $\left(P = \langle \tilde{x}\tilde{x}^T \rangle\right)$

Q = noise matrix

t^+ = after measurement at time t

t^- = before measurement at time t

T = sampling period

When a measurement is made, $\hat{x}$ and P are updated.

$$\hat{x}(t^+) = \hat{x}(t^-) + K(y - \hat{y}) \tag{6-40}$$

where

$$y = Mx; \quad \hat{y} = M\hat{x}(t^-) \tag{6-41}$$

and

$$P(t^+) = (I - KM)\, P(t^-) \tag{6-42}$$

In between measurements $\hat{x}$ and $\hat{P}$ must be propagated through time

$$\hat{x}(t + T) = \Phi(t + T;\, t)\, \hat{x}\,(t) \tag{6-43}$$

and

$$P(\hat{t} + T) = \Phi(t + T;\, t)\;\; P(t)\Phi^T(t + T;\, t) + Q(t + T;\, t) \tag{6-44}$$

The Kalman filter matrix is

$$K = PM^T \left(MPM^T\right)^{-1} \tag{6-45}$$

The states are identified as follows

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} \theta \\ n_c \\ n_s \end{bmatrix} \tag{6-46}$$

The M matrix is

$$M = [\,1 \;\cos \omega t \;\sin \omega t\,] \tag{6-47}$$

The Φ matrix is:

$$\Phi = \begin{bmatrix} 1 & 0 & 0 \\ 0 & e^{-\frac{T}{\tau}} & 0 \\ 0 & & e^{-\frac{T}{\tau}} \end{bmatrix} \tag{6-48}$$

The Q matrix is:

$$Q = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \sigma_n^{\,2}(1-e^{-\frac{2T}{\tau}}) & 0 \\ 0 & 0 & \sigma_n^{\,2}(1-e^{-\frac{2T}{\tau}}) \end{bmatrix} \tag{6-49}$$

The initial P matrix is:

$$P(0) = \begin{bmatrix} \sigma_\theta^{\,2} & 0 & 0 \\ 0 & \sigma_n^{\,2} & 0 \\ 0 & 0 & \sigma_n^{\,2} \end{bmatrix} \tag{6-50}$$

where σ_θ is the standard deviation of the original alignment error. The value of $P_{11}^{\;1/2}$ at any time represents the standard deviation σ of the remaining alignment error at that time.

6.4.4 The Real World

In order for the above Kalman filter to be optimal, the real world must be exactly represented by the Kalman model. Obviously, this will not be the case in a practical system. For the purposes of this study it is assumed that the real world has the same form as the Kalman model, but that the parameters (σ_θ, σ_n, τ, ω) may have different values. The resulting errors are calculated statistically. If these errors are small enough to be encouraging, then the other error sources such as non-linearity, quantization, roundoff, cross-coupling, and a different form of the real world error model may be examined in the simulation program.

Let all quantities with a superscript star (x^*) represent real world values, while the unstarred quantities continue to represent the values used in the Kalman filter. In order to keep track of the errors, it is necessary to use a 6 by 6 covariance matrix.

$$P^* = \left\langle \begin{bmatrix} \hat{x}^* \\ x^* \end{bmatrix} [\hat{x}^{*T} \ x^{*T}] \right\rangle \tag{6-51}$$

When a measurement is made, x^* is updated by

$$\hat{x}^*(t^+) = \hat{x}^*(t^-) + K(y^* - \hat{y}^*) \tag{6-52}$$

where K is calculated from (45)

$$y^* = M^* x^* \tag{6-53}$$

and

$$\hat{y}^* = M\hat{x}^* \tag{6-54}$$

Combining (6-51), (6-52), and (6-53) gives

$$\hat{x}^*(t^+) = [I - KM]\hat{x}^* + KM^* x^* \tag{6-55}$$

since x^* is unchanged by a measurement, we have

$$\begin{bmatrix} \hat{x}^*(t^+) \\ x^*(t^+) \end{bmatrix} = \begin{bmatrix} I - KM & KM^* \\ 0 & I \end{bmatrix} \begin{bmatrix} \hat{x}^*(t^-) \\ x^*(t^-) \end{bmatrix} \tag{6-56}$$

and

$$P^*(t^+) = \begin{bmatrix} I - KM & KM^* \\ 0 & I \end{bmatrix} P^*(t^-) \begin{bmatrix} I - KM & KM^* \\ 0 & I \end{bmatrix}^T \tag{6-57}$$

In between measurements x^* and x^* are propagated through time by

$$\hat{x}^*(t + T) = \Phi(t + T; t)\hat{x}^*(t) \tag{6-58}$$

$$x^*(t + T) = \Phi^*(t + T; t)x^*(t) + U^*(t + T; t) \tag{6-59}$$

where $U^*(t + T; t)$ is the effect of the white noises acting through T. Therefore P^* is propagated by

$$P^*(t+T; t) = \begin{bmatrix} \Phi(t+T; t) & 0 \\ 0 & \Phi^*(t+T; t) \end{bmatrix} P^*(t) \begin{bmatrix} \Phi(t+T; t) & 0 \\ 0 & \Phi^*(t+T; t) \end{bmatrix}^T + \begin{bmatrix} 0 & 0 \\ 0 & Q^* \end{bmatrix} \tag{6-60}$$

where

$$Q^* = \langle U^* U^{*T} \rangle . \tag{6-61}$$

The error is given by

$$\tilde{x}^* = \hat{x}^* - x^* \tag{6-62}$$

or

$$\tilde{x}^* = [I \quad -I] \begin{bmatrix} \hat{x}^* \\ x^* \end{bmatrix} . \tag{6-63}$$

Therefore

$$\langle \tilde{x}^* \tilde{x}^{*T} \rangle = [I \quad -I] P^* \begin{bmatrix} I \\ -I \end{bmatrix} \tag{6-64}$$

and the alignment error is square root of the 1, 1 term of the resultant matrix.

$$\sigma^* = (P^*_{11} - 2P^*_{14} + P^*_{44})^{\frac{1}{2}} \tag{6-65}$$

Both σ and σ^* are calculated. σ represents what the Kalman filter thinks the error is and σ^* represents what it actually is. The initial value of the P^* matrix is:

$$P^{*}(0) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_\theta^{*2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_n^{*2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_n^{*2} \end{bmatrix}$$ (6...

since the initial estimates of the states are zero. Figure 6-6 shows a flow chart of the computer program.

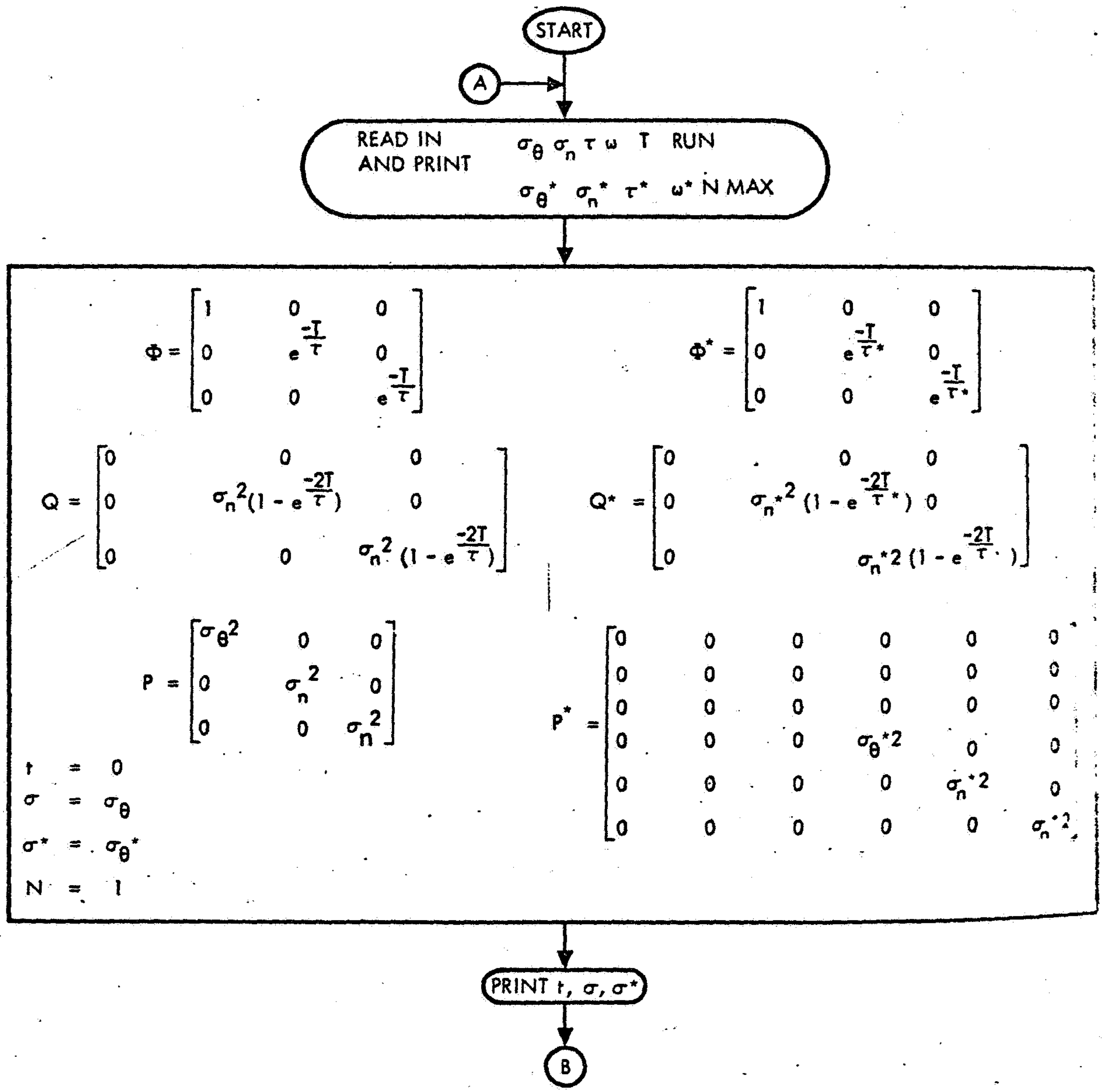


Figure 6-6. Flow Chart

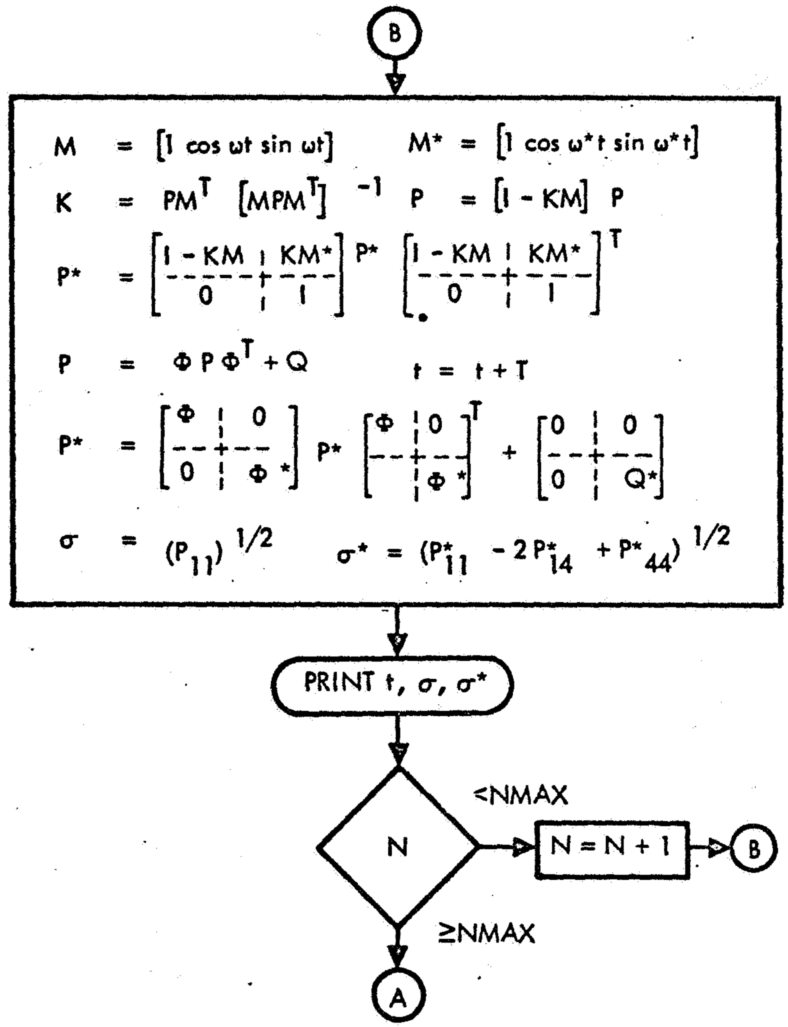


Figure 6-6. Flow Chart (Continued)

6.4.5 Figure of Merit

Since the alignment goal is 20 arc sec, the time required for the alignment error to become and remain less than 20 arc sec would be the ideal figure of merit. However many of the runs would require too much computer time to actually reach this value. Therefore a figure of merit is chosen. If the standard deviation of the error does become and remain less than 20 arc sec during the run, then the time required to do so is the figure of merit. If not, it is assumed that the variance is proportional to the reciprocal of time

$$\sigma^2 = \frac{k}{t - t_0}$$

The values of σ at the end of the run and at 80 percent of the end of the run are used to solve for k_2 and t_0 in (6-67) which is then used to calculate the time required for σ^2 to be reduced to 20 arc sec. This estimate is taken as the figure of merit, and is referred to as the "alignment time."

6.4.6 Runs Made

Runs 1-26 had erroneous inputs because of a key-punching error. Since the answers were correct for the inputs used, even though the inputs were not those intended, useful information was obtained from them and they were not discarded. When rerun with the proper inputs (modified in some cases on the basis of information from runs 1-26) the originally intended runs became numbers 27-52. Only the latter are presented here because they cover all of the results of interest. Table 6-1 shows the values of the input quantities for each run.

Three major cases were studied in which the Kalman filter assumed the swaying accelerations to be

(1) Pure sinusoids, $\tau = 1.0 \times 10^{10}$ sec (runs 27-31)
(2) Narrow-band noise, $\tau = 200$ sec (runs 32-42)
(3) White noise, $\tau = 1.0 \times 10^{-10}$ (runs 43-52)

For each case, the parameters of the real world model were varied to explore the filter sensitivity. Table 6-2 shows the figures of merit for σ and σ^* for each run.

The runs are plotted in Figures 6-7 through 6-32, where σ is indicated by Δ and σ^* by ∇. For runs 27, 28, and 29, σ^* is in error after 1.5 seconds, because of loss of significance.

6.4.7 Sinusoidal Kalman Filter

The sinusoidal Kalman filter works very well when the real world and the filter model are identical, converging to zero error in three samples or 1.5 seconds. However, it is sensitive to changes in ω^* and τ^*. A 10 percent variation in ω^* prevents it from ever converging. It assumes it has finished after three samples and makes no further corrections even if the error is still large in actuality. The filter is not sensitive to changes in σ_θ^* or σ_n^*, reaching zero error in three samples if ω^* and τ^* are correct. Because of its sensitivity the sinusoidal Kalman filter has no practical value.

6.4.8 Narrow-Band Kalman Filter

When the actual swaying accelerations are sinusoidal, and ω^* is correct, the alignment time is increased from 1.5 to 10 sec for $\tau = 200$ sec. However, a 10 percent variation in ω^* further increases the alignment time to only about 1.2 min, rather then infinity.

When the actual swaying accelerations are narrow band noise with $\tau^* = 200$ sec the alignment time is about 16 min. The filter does not seem to be overly sensitive to variations in σ_θ^*, σ_n^*, ω^*, or τ^*.

6.4.9 White-Noise Kalman Filter

The white-noise Kalman filter is very insensitive to variations in ω^* and τ^* and only moderately sensitive to variations in σ_θ^* and σ_n^*. However, it takes much too long to converge. It is possible to cause it to converge more rapidly by reducing σ_n to a fraction of σ_n^*, but this procedure leads to the initial error growth experienced in the scientific simulation.

6.4.10 Conclusions

The NBKF shows promise in reducing the alignment time when the IMU is subjected to sinusoidal or narrow band swaying accelerations. The actual alignment time depends upon the correlation time of the swaying motion, τ^*, which is not presently known. The longer τ^* is, the better are the results which can be obtained from the NBKF. The sinusoidal Kalman filter is not practical.

Table 6-1. Runs Made

Run	σ_θ	σ_n	τ	ω	σ_θ^*	σ_n^*	τ^*	ω^*	T	[illegible]
	$\widehat{sec}$	$\widehat{sec}$	sec	rad/sec	$\widehat{sec}$	$\widehat{sec}$	sec	rad/sec	sec	
27	2160	17500	$1x10^{10}$	π	2160	17500	$1x10^{10}$	1.1π	.5	
28								$.9\pi$		
29							200	π		
30					4320		$1x10^{10}$			
31					1080					
32			200		2160			1.1π		18[illegible]
33								π		
34								$.9\pi$		
35							200	1.1π		
36								$.9\pi$		
37							400	π		
38							100			
39					4320		200			
40					1080					
41					2160				.25	
42									.125	
43			$1x10^{-10}$	0			$1x10^{10}$	1.1π	.5	
44								π		
45								$.9\pi$		
46							200	1.1π		
47								π		
48								$.9\pi$		
49							400	π		
50							100			
51					4320		200			
52					1080					

Note: When an entry is blank, take the next non-blank entry above.

Table 6-2. Figure of Merit

Run	Time for σ	Time for σ*
27	1.5 sec	∞ sec
28	1.5	∞
29	1.5	∞
30	1.5	1.5
31	1.5	1.5
32	15.99 min	1.189 min
33	15.99	0.1667
34	15.99	1.195
35	15.99	12.48
36	15.99	16.38
37	15.99	7.95
38	15.99	31.90
39	15.99	15.88
40	15.99	15.98
41	13.63	13.63
42	12.79	12.79
43	106.2 hrs	15.85 hrs
44	106.2	14.80
45	106.2	17.08
46	106.2	16.00
47	106.2	15.25
48	106.2	16.70
49	106.2	15.02
50	106.2	15.60
51	106.2	61.10
52	106.2	3.74

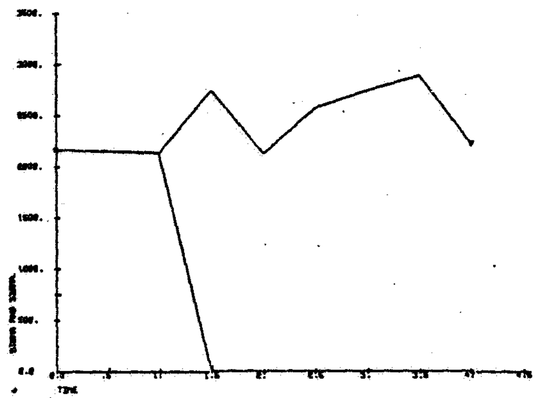


Figure 6-7. Narrow-band Kalman Filter, Run No. 27

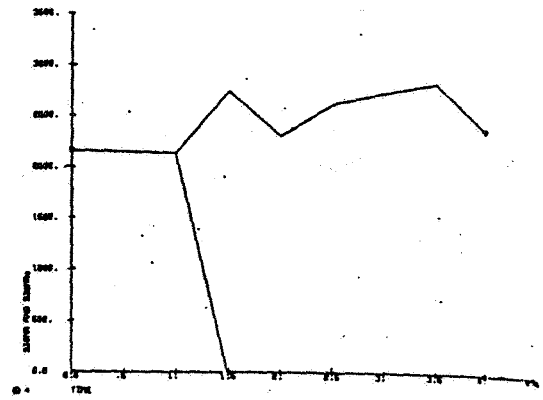


Figure 6-8. Narrow-band Kalman Filter, Run No. 28

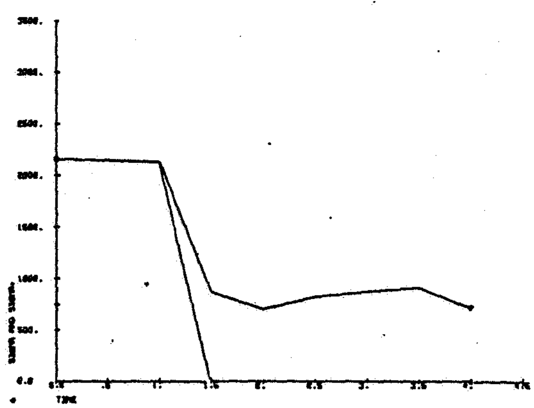


Figure 6-9. Narrow-band Kalman Filter, Run No. 29

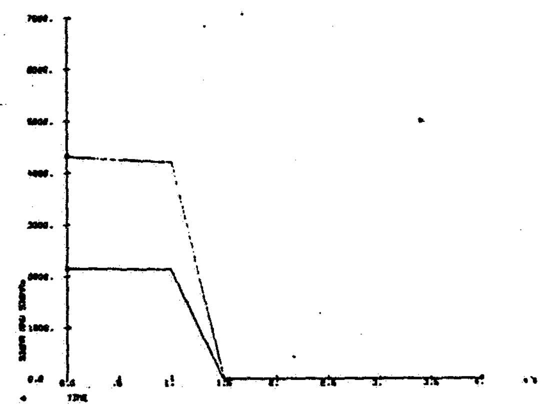


Figure 6-10. Narrow-band Kalman Filter, Run No. 30

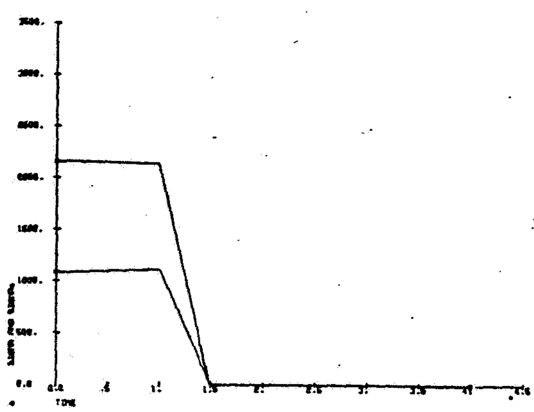


Figure 6-11. Narrow-band Kalman Filter, Run No. 31

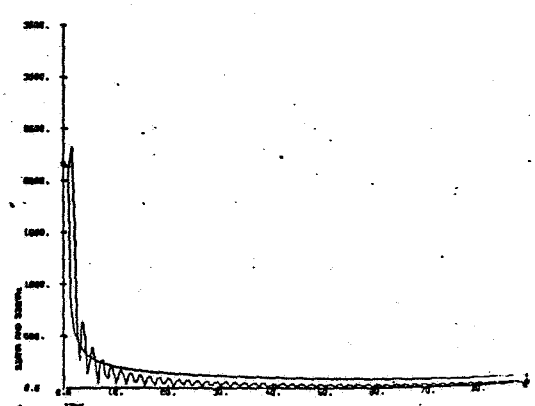


Figure 6-12. Narrow-band Kalman Filter, Run No. 32

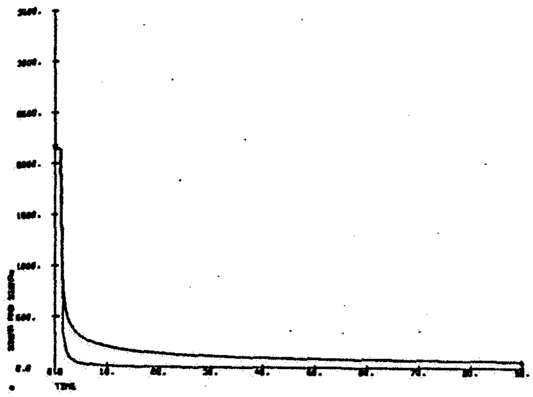


Figure 6-13. Narrow-band Kalman Filter, Run No. 33

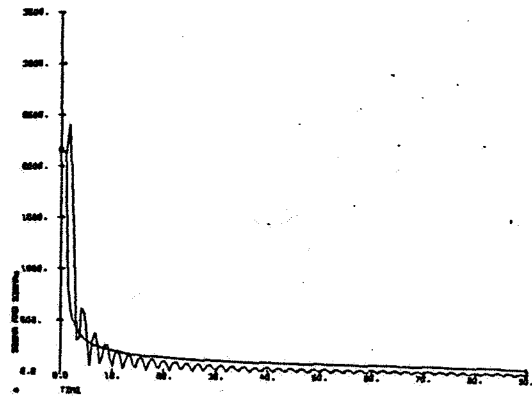


Figure 6-14. Narrow-band Kalman Filter, Run No. 34

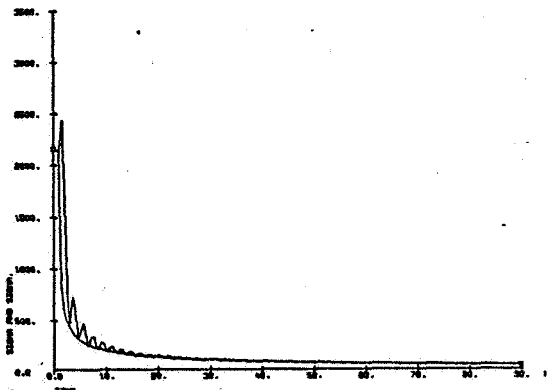


Figure 6-15. Narrow-band Kalman Filter, Run No. 35

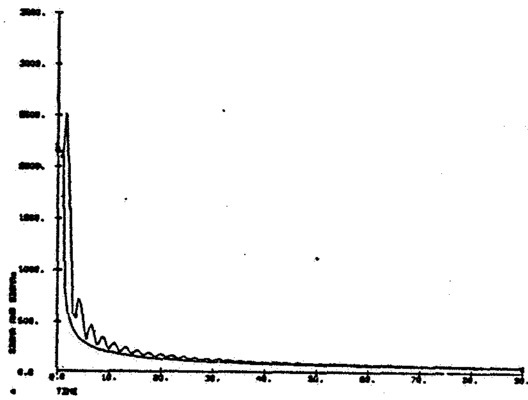


Figure 6-16. Narrow-band Kalman Filter, Run No. 36

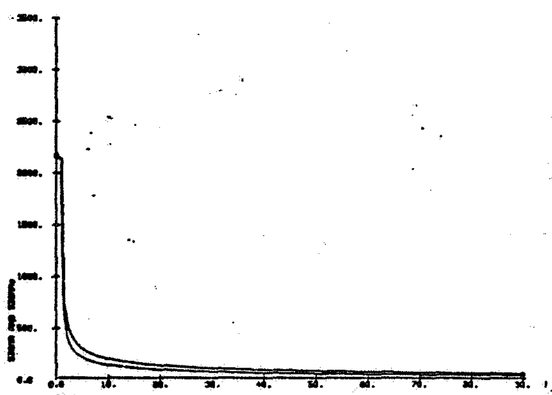


Figure 6-17. Narrow-band Kalman Filter, Run No. 37

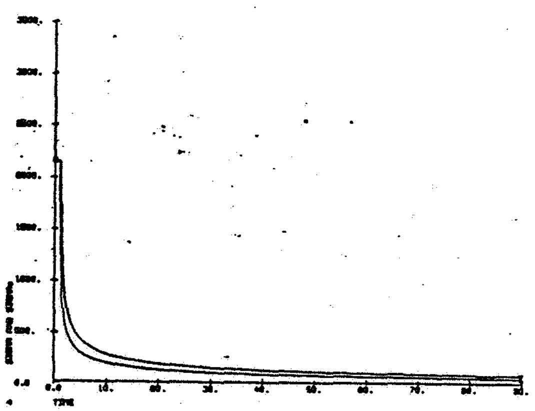


Figure 6-18. Narrow-band Kalman Filter, Run No. 38

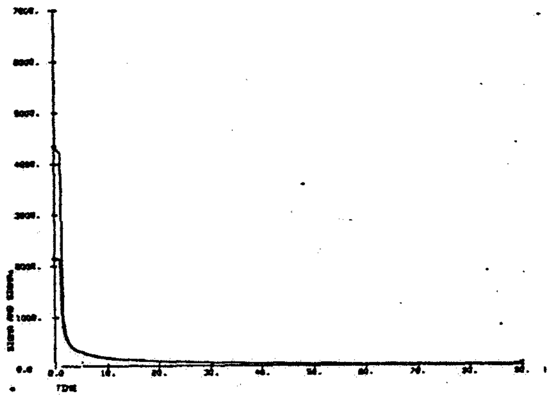


Figure 6-19. Narrow-band Kalman Filter, Run No. 39

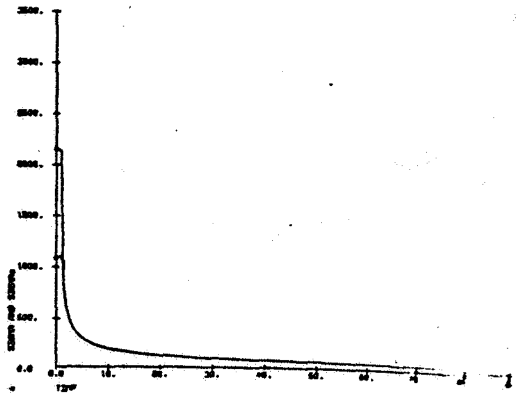


Figure 6-20. Narrow-band Kalman Filter, Run No. 40

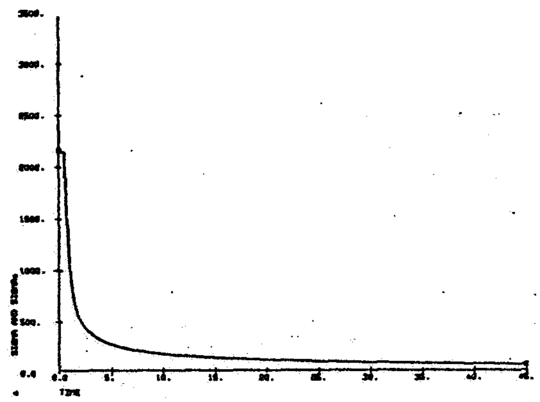


Figure 6-21. Narrow-band Kalman Filter, Run No. 41

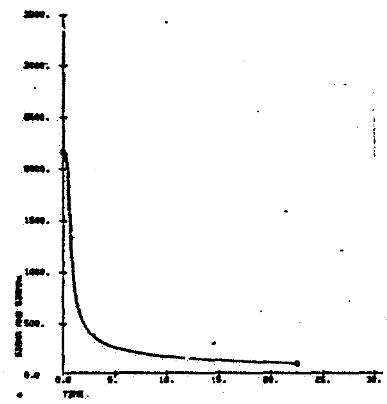


Figure 6-22. Narrow-band Kalman Filter, Run No. 42

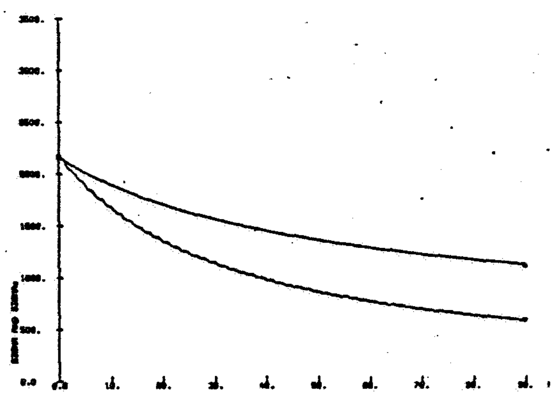


Figure 6-23. Narrow-band Kalman Filter, Run No. 43

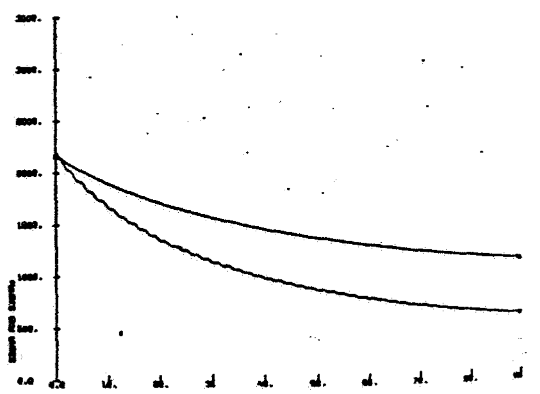


Figure 6-24. Narrow-band Kalman Filter, Run No. 44

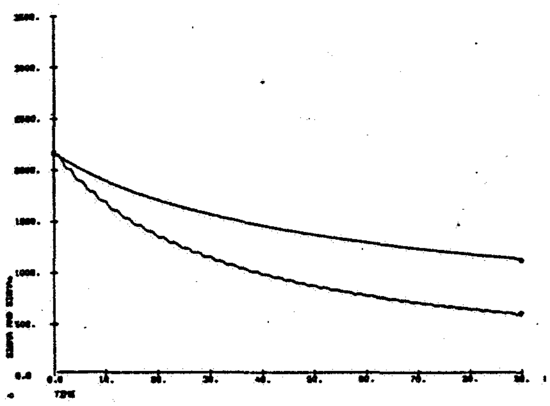


Figure 6-25. Narrow-band Kalman Filter, Run No. 45

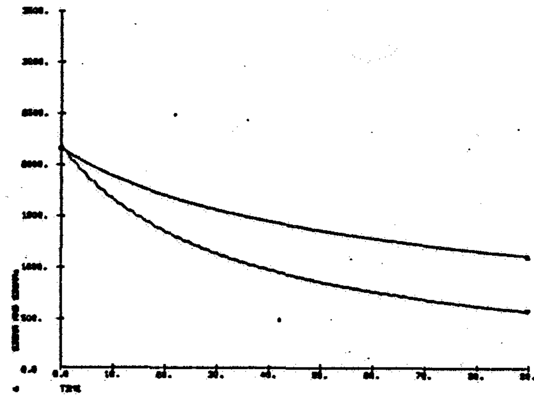


Figure 6-26. Narrow-band Kalman Filter, Run No. 46

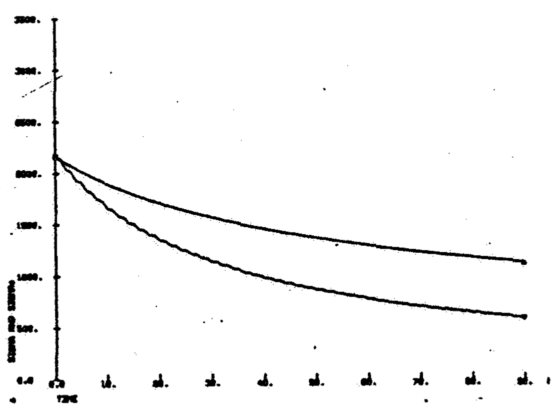


Figure 6-27. Narrow-band Kalman Filter, Run No. 47

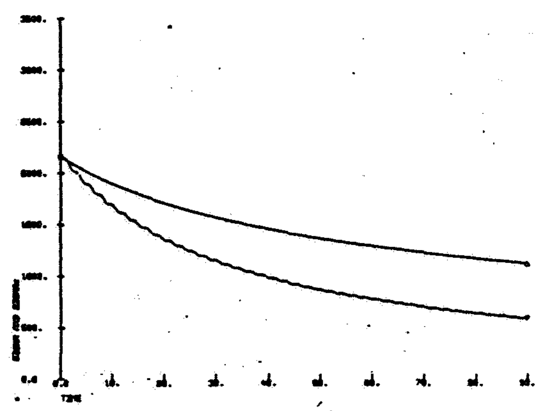


Figure 6-28. Narrow-band Kalman Filter, Run No. 48

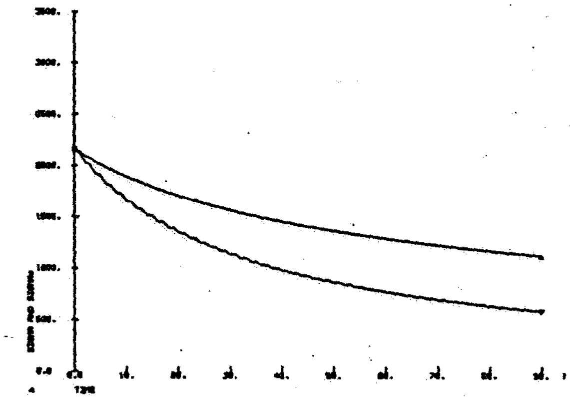


Figure 6-29. Narrow-band Kalman Filter, Run No. 49

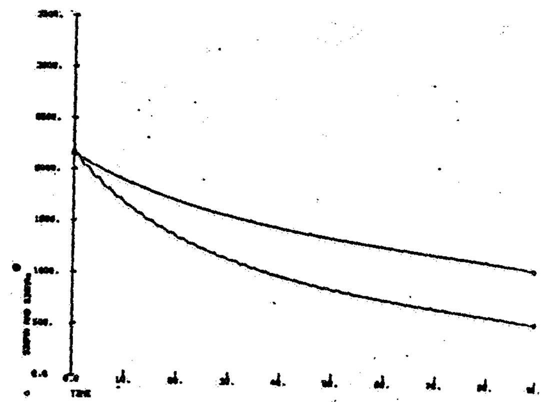


Figure 6-30. Narrow-band Kalman Filter, Run No. 50

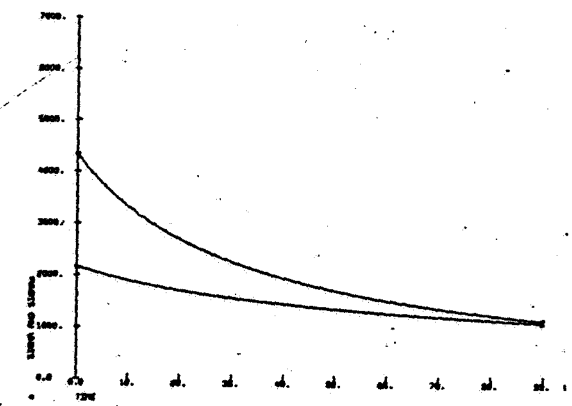


Figure 6-31. Narrow-band Kalman Filter, Run No. 51

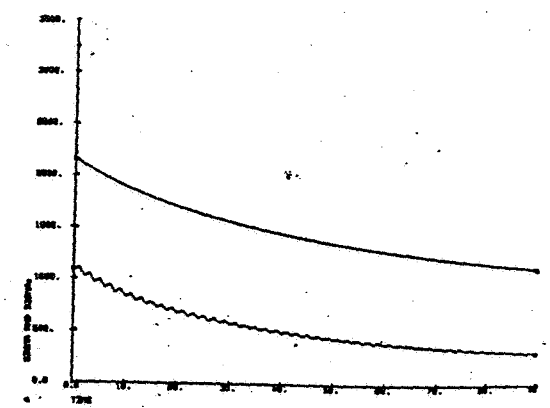


Figure 6-32. Narrow-band Kalman Filter, Run No. 52

7. AIRBORNE COMPUTER IMPLEMENTATION STUDIES

7.1 INTRODUCTION

This section will investigate some of the techniques available for mechanizing the SIRA equations in an airborne computer. A general purpose (GP) computing device will be considered. Since instrument outputs are in incremental form, digital differential analyzer methods will also be considered. Generally, DDA's have been used to solve equations of rather limited number and complexity because of the rather large investment in hardware required. Therefore, in most instances use will depend upon whether or not hardware can be time-shared between computations in order to bring the total hardware count within reasonable bounds.

Kalman
o. 50

The number of equations to be implemented, approximately 50, illustrates the computational load which the device must carry. The complexity of the equations varies from a simple multiplication of a variable by a constant to the comparison of two 3 x 3 matrices. The performance rate at which these must be performed varies with the equation, and for most DDA implementations varying speeds are very awkward to accommodate.

There is also the necessity to perform one set of computations during the alignment mode and a slightly different set after liftoff. This constraint again is cumbersome for a DDA to handle and implies a logical restructuring and reinitialization of certain functions.

In the paragraphs to follow, six configurations of hardware are discussed using IC's, MOS's, delay lines, and cores for a DDA implementation, and IC's for both a general-purpose computer and a hybrid.

During Phase I the number of DDA computational elements needed to implement the equations was specified. This number is substantially the same and the information is repeated in Table 7-1 with additional elements added to accommodate the new alignment procedure.

alman
52

Table 7-1. DDA Hardware Requirements

	Integrators	Adders	Samplers	Total
Flight Equations*	87	33	14	134
Flight Equations**	99	51	14	164
Flight Equations***	133	51	11	195
Alignment Equations	31	11	1	43
New Alignment Equations	44	3	0	47
Maximum Total****	208	65	12	285

*Assumes no extrapolation of guidance commands.

**Assumes direction cosines are received by unit which performs linear extrapolation on these

***Assumes commanded angles are received by unit. Unit extrapolates; forms sines/cosines, and commanded A_{ij}'s.

****Implies use of all alignment equations and flight equations.

7.2 SUMMARY

The problems associated with the alignment computation in a DDA were not satisfactorily resolved. Further study is recommended. The concept of alignment studied is quite easily adaptable to either the GP or hybrid configuration. It is possible that other alignment schemes would be more adaptable to the DDA structure.

Table 7-2 delineates some of the differences between the various computer implementation approaches. The set of equations used for the comparison are the simplest set of flight equations. Additional circuitry must also be added to the flat-pack count to accommodate delay line read and write amplifiers and core memory circuitry where appropriate. The power totals reflect this additional circuitry but do not include a power supply itself.

The large number of flat-packs associated with an all IC approach simply emphasizes the size of the computational problem and the amount of communication necessary between basic computational units. The circuits chosen for this application were integrated, 2 gates, 1 flip-flop, or one 8-bit register per flat-pack. Flip-flop toggle rates of 2 MHz and pair delays of 80 nsec were assumed.

The MOS approach indicates the potential advantages of multifunctions per flat-pack. Clock rates of 500 kHz were considered as maximum at this time.

During Phase I, the core memory DDA was considered the optimum DDA design for the system. The increased data rates and new alignment techniques force this structure to extremely high internal rates. The power totals reflect the requirements of these fast devices (namely, 100 mw per flip-flop and 50 mw per dual gate). The speeds required were toggle rates of 40 mHz and pair delays of 20 nsec.

The delay line approach suffers considerably from the requirement to communicate much data between delay lines. These devices also require a weight, volume, and power expenditure much above all others considered. The data rates within the delay line were 10 MHz. The rates external to the delay line were essentially 5 MHz. The system, as described, should be considered an approach and not necessarily an optimum configuration.

Table 7-2. Airborne Computer Comparisons

Characteristic	Iteration Rate	Number of Flat Packs	Power (w)	Flexibility	Remarks
System Type					
IC's	100 kHz	2600	122	Poor	Includes 129-8 bit register packs.
MOS	33 kHz	510	25	Poor	Multi-function flat-packs.
Core DDA	10 kHz	675	52	Poor	Requires core stack and electronics.
Delay Line	62.5 kHz	940	116	Poor	Requires 27 read & 27 write amplifiers plus delay lines.
GP Computer (24-bit)	132 Hz	1060	37	Good	Requires core stack and electronics.
Hybrid (18-bit)	250Hz 100kHz	850	36	Good	Requires core stack & electronics. Includes 36-8 bit register packs

The GP approach is essentially the one described in the Phase I report. Input/ output equipment has been modified to reflect system changes. Word length has also been increased to 24 bits. The hybrid configuration takes advantage of the desirable characterisitics of both a DDA and a GP device. Alignment and fast data rates were accommodated easily. Much of the input hardware in the GP configuration was traded for computational hardware in the hybrid. The ability to custom-fit a small section of the computations enabled a reasonably economical solution.

Table 7-2 shows DDA's presently structured to accommodate the flight equations. If all the other equations were to be implemented in a similar fashion, the DDA systems become prohibitively complex. The maximum number of computational elements required is 285. This includes all alignment and commanded angle extrapolations. This would essentially double the hardware totals shown above unless some kind of logical re-structuring and subsequent re-initialization is done. This restructuring itself would entail additional hardware and complexity.

Table 7-2 does indicate the competitiveness of the approaches if the simplest set of equations are to be implemented. Additional functions lend favor to either the hybrid or conventional GP configurations. Additionally, short of pure redundancy, there does not appear to be any means whereby operational readiness can be determined within the DDA's. An operational check (during countdown, for example) can be done by the GP or hybrid with relative ease.

The basic computational elements of which the DDA is comprised permit conventional development and checkout techniques to be used. The system as a whole is quite another matter, however, primarily be-cause of the inability to break out and exercise small functional units. Development of these devices would further be handicapped by the intimate relationship between the equations and the logical structure. Parallel design of the logical structure and program could not be accomplished. Further system changes would be very difficult to fit into an existing system.

Gyro and accelerometer inputs are incremental . All other data (commanded angles and outputs to Saturn LVDA) are most efficiently handled by transferring whole binary numbers. Therefore, although the

DDA can accept incremental data, a whole number interface is desirable as its output mechanism. Similarly, the GP device can output whole binary numbers readily, but the input logic must accept incremental data. The hybrid configuration of hardware seems to offer a reasonable solution to these opposing constraints. The gyro data (to 46 kHz) can be handled very effectively by the DDA portion of the hybrid. This reduces the computational load on the GP to the point where a 250 Hz major iteration cycle is possible. The conversion from incremental to whole number form is accomplished by the communications scheme between the DDA and GP sections of the hybrid. In addition, the alignment schemes considered lend favor to a hybrid configuration in that only the expenditure of time is necessary to permit their solution.

7.3 SYSTEM CHANGES DURING PHASE II OF STUDY

During Phase II of this study it was recognized that the alignment scheme was too time-consuming. It was also observed that the nominal vehicle turning rates considered in Phase I were unrealistic. That is, the 1.4 deg/sec turning rate has been changed to 13 deg/sec. Accelerometer data rates have remained constant in the GP mechanization. Improved alignment time necessitated the inclusion of a set of equations which generate an approximation to the actual attitude reference matrix, instead of using the identity matrix as the starting point for alignment. This set of equations and the new nominal vehicular turning rates have a profound impact on the structure of the computing element. These new criteria have changed the direction of the study and will be reflected in the following paragraphs. Additionally, the scheme used to output data from the computing instrument has been modified to permit the transmission of whole numbers as compared to the incremental scheme described in the Phase I report.

7.3.1 Alignment Mechanization

Several approaches were considered for the satisfactory solution of alignment mechanization. Some of the solutions depend upon the configuration of hardware used to mechanize the equations.

The equations derived in order to improve the alignment time are given below. They are essentially the only equations which have changed since the Phase I study was completed. A flow diagram of the computations is shown in Figure 7-1.

$$A_{31} = A_{31\,D}$$

$$MM = +\sqrt{\Delta V_{x1}^2 + \Delta V_{y1}^2 + \Delta V_{z1}^2}$$

$$A_{12} = \Delta V_{x1}/MM$$

$$A_{22} = \Delta V_{y1}/MM$$

$$A_{32} = \Delta V_{z1}/MM$$

$$A_{33} = +\sqrt{1 - A_{31}^2 - A_{32}^2}$$

$$\text{Set } \Delta V_{x1} = \Delta V_{y1} = \Delta V_{z1} = 0$$

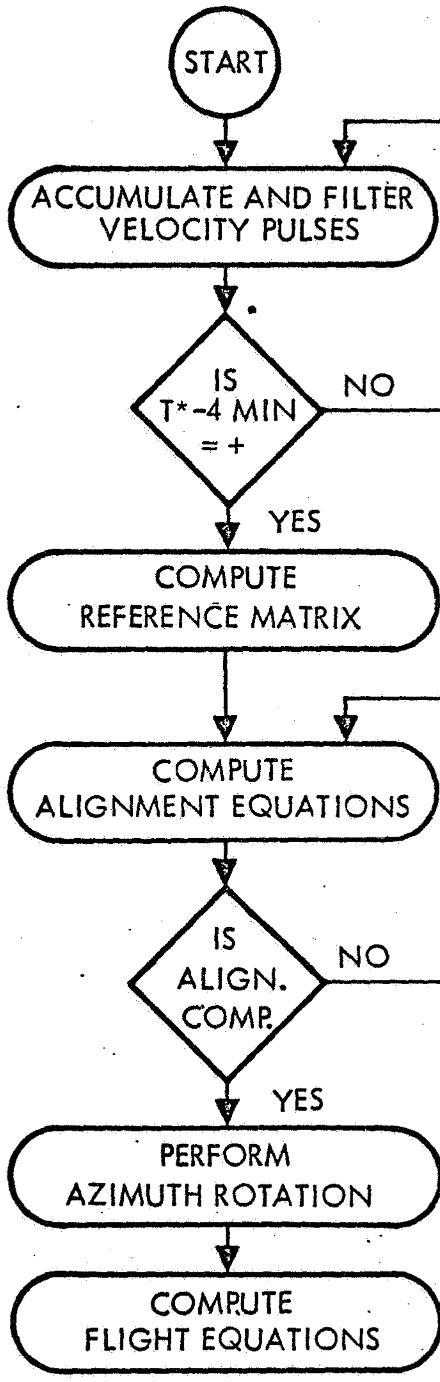
$$DD = 1 - A_{32}^2$$

$$A_{21} = -\left[\frac{A_{12}\,A_{33} + A_{31}A_{22}A_{32}}{DD}\right]$$

$$A_{11} = \frac{A_{33}A_{22} - A_{12}A_{32}A_{31}}{DD}$$

$$A_{23} = \frac{A_{31}A_{12} - A_{32}A_{22}A_{33}}{DD}$$

$$A_{13} = -\left[\frac{A_{12}A_{33}A_{32} + A_{31}A_{22}}{DD}\right]$$



*T = TIME FROM START

Figure 7-1. Alignment Flow Diagram

Note from Figure 7-1 that the initial reference matrix calculation is transferred just once. The velocity pulses are accumulated and an initial approximation made of the attitude reference matrix. The alignment increment computation is then used for the final alignment.

It is possible that the initial attitude reference matrix could be evaluated by ground equipment or the primary guidance computer and then transferred to the airborne computer. This would reduce the complexity of the computing instrument by approximately 45 integrators and 6 adders, or their equivalent in a general-purpose device.

Another possibility is an increased iteration rate, and thus, faster alignment. This assumes an initial reference matrix such as the identity matrix as was used during Phase I studies. Verification of this approach by a simulated alignment would be necessary.

These calculations are trivial, of course, if a general-purpose computer is the computing instrument (80 cells of program storage).

7.3.2 Iteration Rate

The ramifications of the increased data rates change markedly the requirements of a satisfactory computing element.

The increment size on the gyros has been tentatively selected as 4.7 arc sec. This compares with 10 arc sec during Phase I of this study. Moreover, the nominal vehicle turning rate has been changed from 1.4 to 13 deg/sec. This implies a data rate of 10 kHz or equivalently 100 μsec for a DDA to process one bit of incremental data. The maximum turning rate of 60 deg/sec corresponds to a data rate of 46.2 kHz or 21.6 μsec/bit of data.

The data rates of the accelerometer inputs, on the other hand, have remained at 805 bits/sec, or 1.25 msec per increment. Thus, there is one computation, the attitude reference equations which must be executed at a significantly higher rate than all the other computations.

7.4 DDA IMPLEMENTATIONS

Since several techniques for alignment exist, the following tradeoff studies assume the use of the simplest set; i.e., a first-order Taylor series approximation for the attitude reference computations, no extrapolation of guidance commands, and a simple identity matrix for the initial matrix upon which alignment is performed. The implications of including additional functions over and above this basic set are discussed where appropriate.

7.4.1 Conventional Flat-Pack Integrated Circuit Hardware

It is certainly possible to construct a computing element exclusively of active elements (i.e., flip-flops, gates). Such an implementation is inefficient in terms of the number of computations performed per flat-pack; nevertheless the implementation is straightforward. Due to the large number of calculations, it is desirable to construct a small set of functional units which can be used to solve all the equations. Certainly with all computations being done in parallel, serial operation within the small functional units should be considered. The two functional units required to solve the equations are an integrator and an adder. Figure 7-2 is a block diagram for the integrator. An adder or time buffer element is similar in construction, but requires no R register, Δx input, or $R \pm Y$ adder. The number of flat-packs assuming two gates, one flip-flop, or one 8-bit shift register per flat-pack is also listed. Another element, an integrator without its own Y register, is also listed.

Type	Number of Flat-Packs	Power* per Element (w)
Full Integrator	30	1.45
Half Integrator	14	0.70
Adder Element	16	0.75

*Registers at 200 mw/FP
Gates, etc. at 25 mw/FP

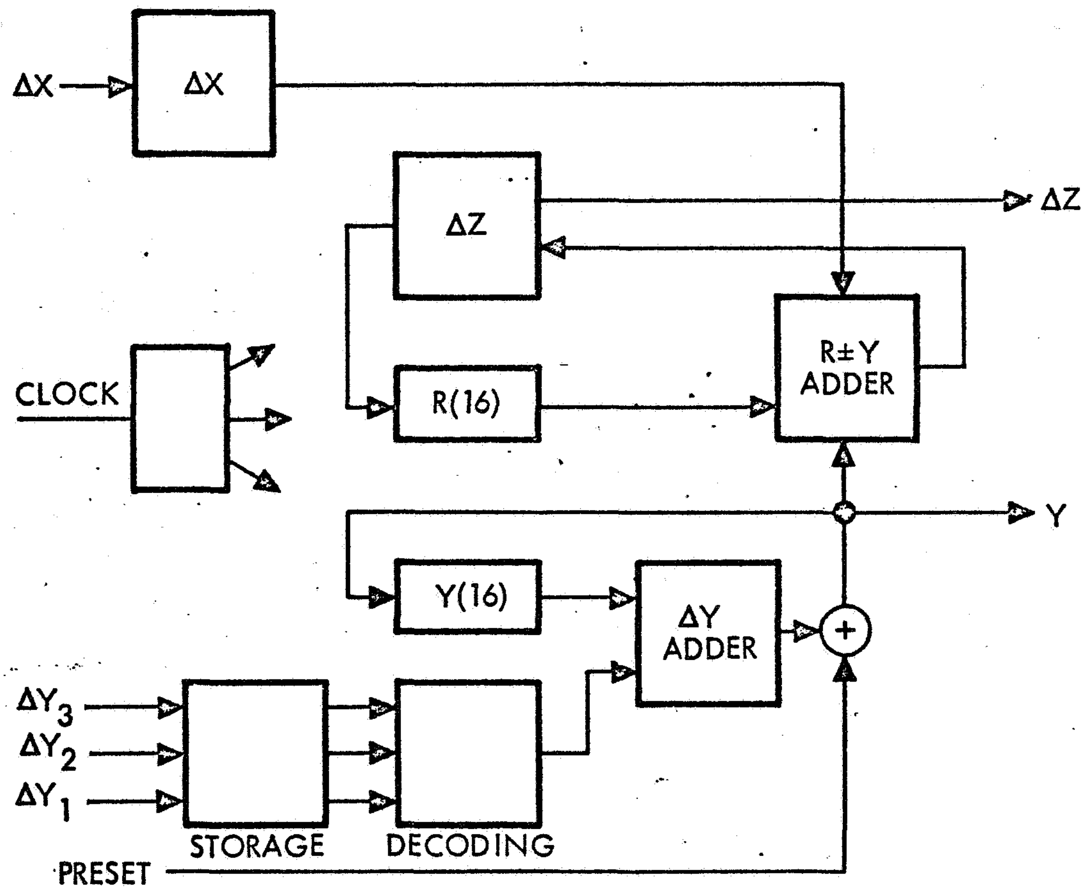


Figure 7-2. Integrated Circuit Mechanization of DDA Integrator

With a completely parallel operation, at least one simplification technique exists for reducing the total amount of necessary hardware. The direction cosines (used in many computations) need only be stored in one integrator. These data can then be routed to the appropriate integrators where needed.

The computations which can share a common Y register are itemized below. Obviously, the savings include not only the Y register itself, but also the adder and control circuitry.

Computation	Savings
Gyro to Accelerometer Coordinator	9 Y's
Attitude Reference Equation	9 Y's
Velocity Res to Inertial Space	9 Y's
Attitude Error Calculation	9 Y's
Earth Rate to Body Axes	9 Y's
Alignment Increment Equations	6 Y's
Total	51 Y's

The minimum number of elements necessary to mechanize the strapdown equations is 162 elements.

		Flat-Packs	Power (w)
Full Integrators	67	2010	97
Half Integrators	51	714	36
Adders	44	704	31
Total	162	3428	164 w

To these totals must be added the necessary circuitry to enable a) communication with outside world, b) initialization of DDA, c) sequencing and/or restructuring the device to facilitate the proper sequencing of the calculations. All of the above considerations are related in that some form of common hardware could be the solution.

Since all adders and R registers would be set initially to zero, this leaves 67 full integrators whose Y registers will have to be preset. This implies the ability to address uniquely 67 registers which involves an addressing scheme of 6 binary bits. A simple binary counter scheme involving approximately 100 flat-packs could be used. If the counter is addressable by an external source, the Y registers could be accessed randomly. The characteristics of the external equipment determine desirability.

Obviously, there is a need for an input/output register used in conjunction with the 6-bit counter described above. It could be implemented with 8 bits per flat-pack shift registers. This communication scheme, coupled with several discrete signals, could be the vehicle by which data are also outputted from the DDA. A block diagram of the system is shown in Figure 7-3 below.

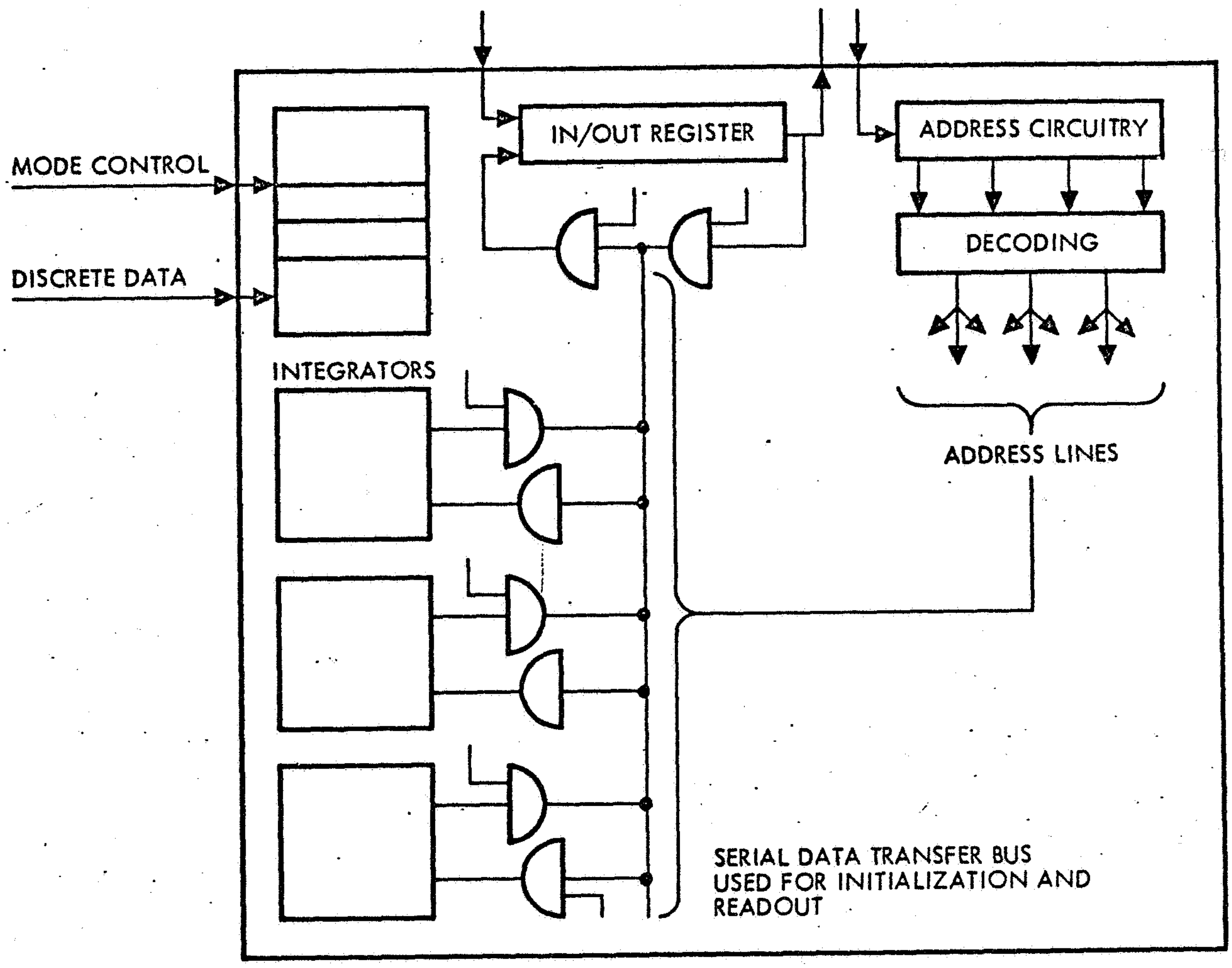


Figure 7-3. Communication Technique

The total number of flat-packs is:

	Flat-Packs	Power (w)
Arithmetic Units	3428	164.0
Addressing	100	2.5
In/Out Register	5	0.5
Discrete Register	20	0.5
Serial Bus	70	1.75
Total	3623	169.25 w

Presently available circuits permit a maximum iteration rate of approximately 100 kHz. This would obviously negate the requirement for input buffers even at maximum vehicular turning rates (60 deg/sec).

If alignment must be done according to Figure 7-5 and the equations described earlier, the implementation shown in Figure 7-4 must be accommodated.

These equations require 45 integrators and 6 adders for their mechanization. As can be seen from Figure 7-5, approximation is made once at the beginning of alignment and as such is not well suited to DDA implementation. It would, of course, be possible to restructure the DDA via flip-flops and gating to accommodate the equations; however, there is still the necessity of initializing this newly structured DDA.

7.4.2 MOS Devices

It is appropriate to mention recent developments in the field of MOS devices. There are presently available DDA elements fabricated in a single 72 x 86 mil area flat-pack. This device coupled with two suitable shift registers forms a complete DDA integrator. Although the maximum clock rate is relatively slow, 500 kHz, the packing density is significantly greater than that obtainable with presently integrated circuit devices.

A system comprised of these devices would appear very similar to the one shown in Figure 7-3 in block diagram form. There still exists the necessity to time buffer ΔZ's which are added to form ΔX's in other integrators.

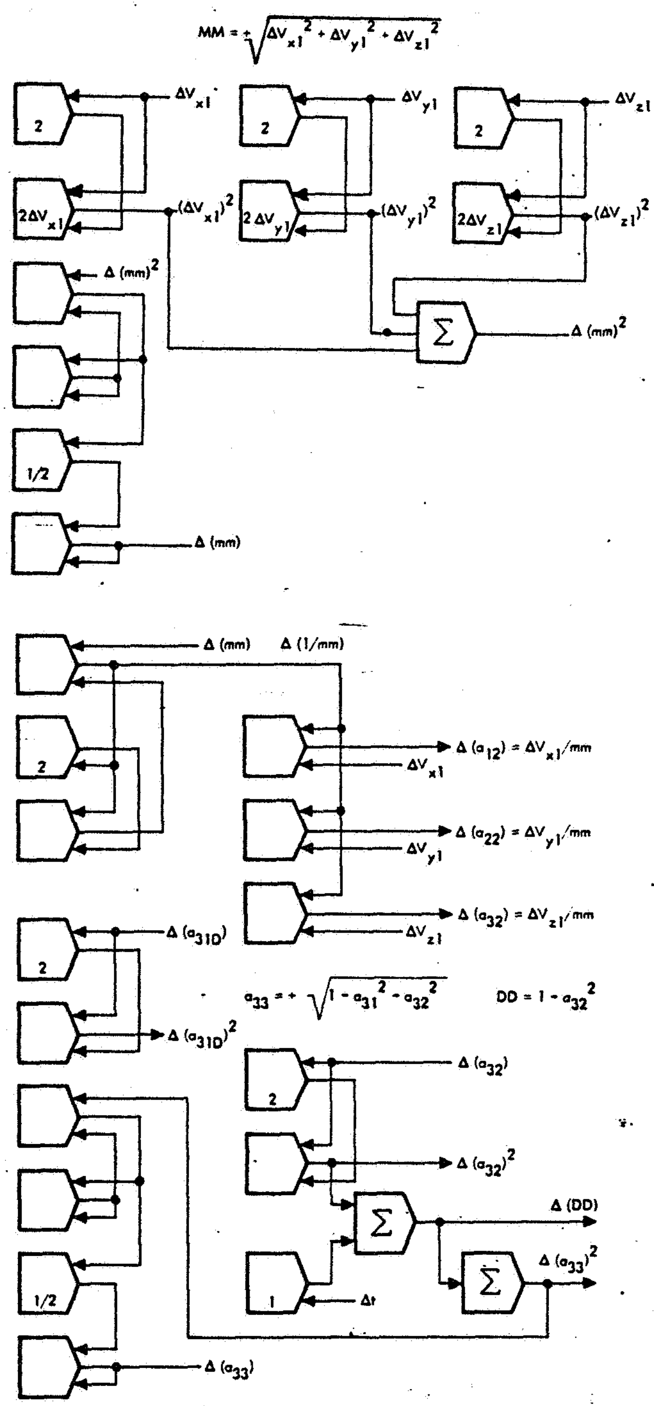


Figure 7-4. DDA Alignment Implementation

$$a_{21} = -\frac{(a_{12}\,a_{33} + a_{31}\,a_{22}\,a_{32})}{DD}$$

$$a_{11} = \frac{(a_{33}\,a_{22} - a_{12}\,a_{32}\,a_{31})}{DD}$$

$$a_{23} = \frac{(a_{31}\,a_{12} - a_{32}\,a_{22}\,a_{33})}{DD}$$

$$a_{13} = -\frac{(a_{31}\,a_{22} - a_{12}\,a_{33}\,a_{32})}{DD}$$

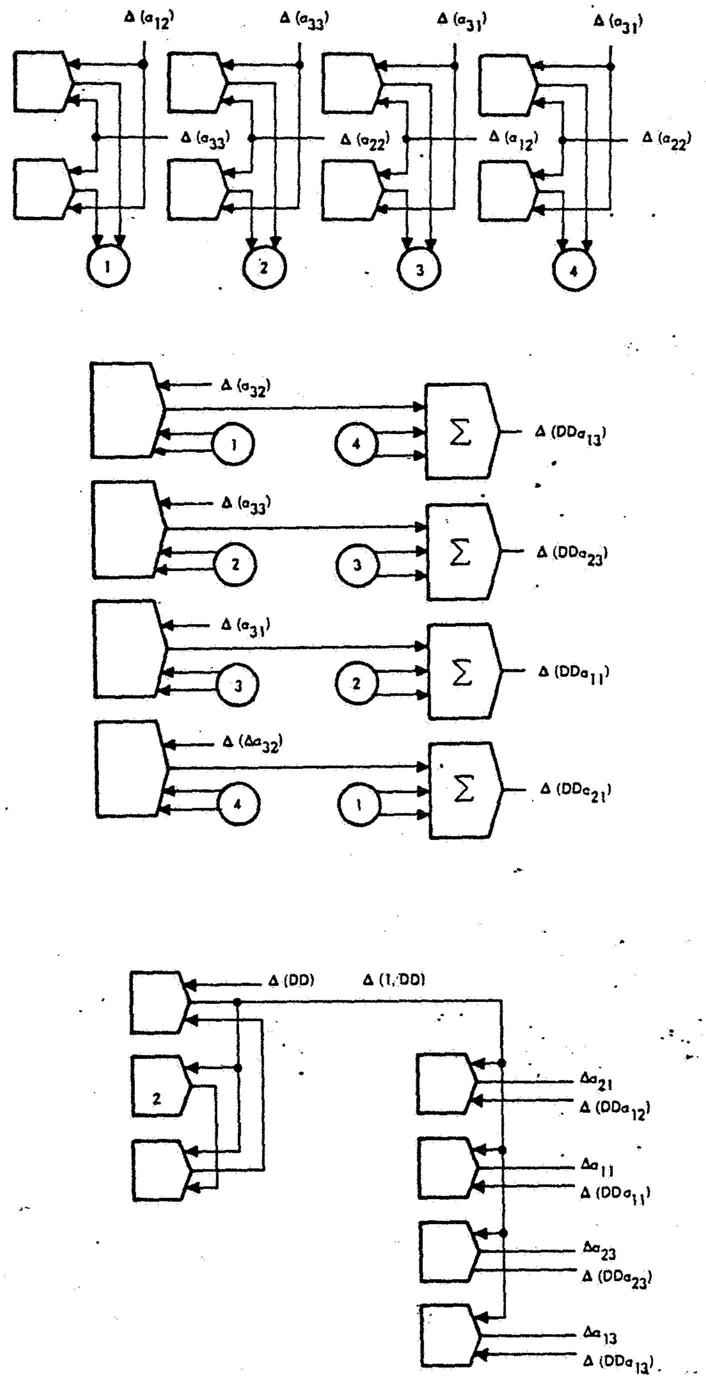
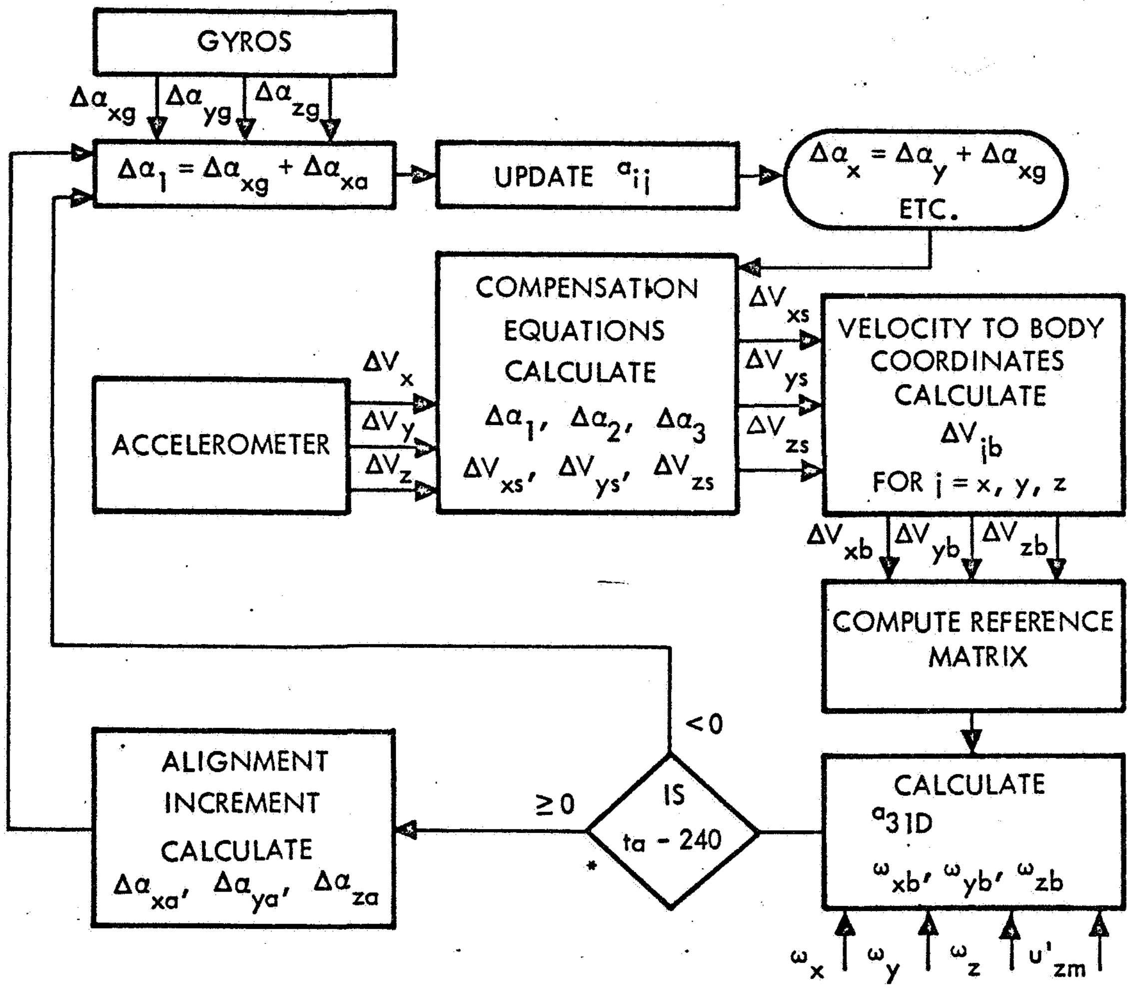


Figure 7-4. DDA Alignment Implementation (Continued)



ta = TIME FROM START OF ALIGNMENT OF ATTITUDE REFERENCE MATRIX

Figure 7-5. DDA Alignment Flow Diagram

The other considerations, such as initialization and communication channels, are also required of this system. Hardware totals are shown below. The significant advantages are reduced size and power.

	Flat-Packs	Power (w)
Arithmetic Units	378	30.000*
Addressing	100	0.800
In/Out Register	5	0.400
Discrete Register	20	0.160
Serial Bus	70	0.560
Total	573	31.900

*DDA element requires 80 mw of power
Dual 16-bit shift registers require 150 mw of power
Adder element (full adder/flat-pack) requires 25 mw of power

The obvious disadvantages of these devices are the lack of operational data, lack of experience using such devices, their nonstandard interface with other solid-state circuits, and reduced maximum clock rates. They will be worthy of serious thought if and when the above factors are satisfactorily resolved.

7.4.3 Core Memory Technique

This paragraph will describe the characteristics of a DDA similar in nature to the one specified in the Phase I final report. Its organization was similar in many respects to a general-purpose digital computer. Specifically, data associated with a particular "integrator" are retrieved from a word-oriented core stack. The appropriate arithmetic, $y + \Delta y$'s, $R + \Delta x(y)$ is performed on the data and the result restored in the same cell location in the stack. The Δz's are routed to many "integrators" (cells) as Δy's or Δx's by a unique wiring arrangement. The transmittal of $+\Delta z$ to some integrators and the same $-\Delta z$ to others is also accomplished by a wiring technique already described in the Phase I report.

The primary advantage of using this technique is that the same arithmetic unit is used to perform all calculations. Thus the rather large expenditure of hardware to implement the 16-bit parallel binary adder is amortized over all of the computations. The number of computational units required to execute the flight equations was 134 (integrators, samplers, adders). The time allowed during the Phase I study effort was 1 ms. This has since been reduced by a factor of 10. It would be possible to execute the attitude reference equations more frequently than the other computations. The attitude reference set required 32 computational elements. This leaves 102 for all other calculations. If the 32 devoted to the reference matrix are to be processed every 100 μsec and the rest every 1.24 ms (1/805 bits/sec), the required unit cycle time (UCT) would be

$$1.24 \text{ ms} = [(102)(UCT) + (12.4)(32)(UCT)]$$

or

$$1.24 \text{ ms} = 498 \text{ UCT}$$

$$\therefore \text{ UCT} = 2.5 \ \mu\text{sec}$$

The above estimate is without doubt optimistic since there will be two different computation rates within the DDA.

Another alternative would be to process all 134 elements within the 100 μsec interval. This demands a unit cycle time of 0.805 μsec. Sylvania has recently disclosed integrated circuits capable of permitting these fast iteration rates. Flip-flops which can be clocked at 40 MHz and gates with pair delays of 20 nsec even after proper derating are seen to be available in the future. A core memory capable of 0.8 μsec is also feasible. Therefore a design predicated on using state-of-the-art hardware could be done; however, the development would entail all the problems associated with using present state-of-the-art devices.

7.4.4 Large Serial Memory Devices

Several configurations of delay line storage elements were considered as candidates. Since there is a minimum of 134 integrators and adders, the task of communicating information between them is a formidable one. A single delay line as the common storage element has the obvious advantage of an efficient storage element; it also provides a convenient technique for initializing the system. During actual computations, however, communicating with other integrators implies either multiple passes through the single delay line or multiple delay lines or a random access storage element for the 134 Δz's. This also involves the existence of a relatively complex addressing scheme.

In order to reduce the lag in solution time within the DDA, it is desirable to accommodate more than one Δy per iteration. The optimum number for the strapdown equations is about three or four. This means that some of the integrators should be able to randomly address three or four Δy's and a Δx, which in turn requires that some integrators need four or five addresses of 8 bits per address. This does yield a very flexible system, in that reprogramming could be accommodated with relative ease. A completely flexible system, as described, would require that many integrators carry with them much excess addressing capacity which would never be used simply because they require a single Δy and a Δx.

A single delay line for storage is completely incompatible with the necessary speed at which this unit must operate. There would be no great advantage involved in replacing each of the 16-bit registers specified in the all-parallel DDA described earlier with delay line storage elements. The package would be larger and the electronics necessary to drive and sample the delay line would tend to offset any power advantage. Hence, the delay line storage element will be useful if it can serve as a common storage element for multiple integrators.

Therefore, it is necessary to group the calculations into subsets using the reduction of communication between them as one of the criteria for selecting the subset itself. Since it is desirable to operate on all data at the same rate (so as to negate the need for pulse buffering), the task of forming optimum subsets of equations is relatively complex.

The flow of incremental data through the DDA is serial from equation to equation. Since time did not permit the consideration of all possible configurations, it was decided to evaluate one which would minimize the lag in solution time. Assume that an "adder" element consumes computational time equivalent to that of an integrator and call this a unit cycle time (UCT). The flow diagram in Figure 7-6 illustrates the minimum number of UCT's for each block of computations provided that parallelism is applied within each block.

The equations to be solved within each block are given below for one axis.

$$\Delta \dot{a}_x b = 50\,(\Delta a_{xg} - \dot{a}_{xb}\Delta t) \qquad (7\text{-}1)$$

$$\Delta a_{ij} = \Delta \alpha_{(i+2)} a_{(i+1)}{}^{j} - \Delta \alpha_{(i+1)} a_{(i+2)}{}^{j} \quad \text{(9 total)} \qquad (7\text{-}2)$$

$$E_x = -\Big[a_{21}\Delta a_{31R} + a_{31R}\Delta a_{21} + a_{22}\Delta a_{32R} + a_{32R}\Delta a_{22} + a_{23}\Delta a_{33R} + a_{33R}\Delta a_{23}\Big] \qquad (7\text{-}3)$$

$$\Delta \alpha_1' = b_{11}\Delta \alpha_x + b_{21}\Delta \alpha_y + b_{31}\Delta \alpha_z \qquad (7\text{-}4)$$

$$\Delta V_{xs} = C_{1x}\Delta V_x - C_{2x}\Delta \alpha_1' - C_{3x}\Delta t \qquad (7\text{-}5)$$

$$\Delta V_{xb} = b_{11}\Delta V_{xs} + b_{12}\Delta V_{ys} + b_{13}\Delta V_{zs} \qquad (7\text{-}6)$$

$$\Delta V_{xI} = a_{11}\Delta V_{xb} + a_{21}\Delta V_{yb} + a_{31}\Delta V_{zb} \qquad (7\text{-}7)$$

An assignment of delay line elements to specific functions is shown in Figure 7-7. These will execute the flight equations for the strapdown system. Notice that the delay lines are grouped in triplets for solving the equations from blocks 3, 4, 5, 6, and 7; pairs for block 2, and single units for block 1. Each delay line is broken up (timewise) into 4 word times. Within each word time an integration and a servo-addition can be accomplished.

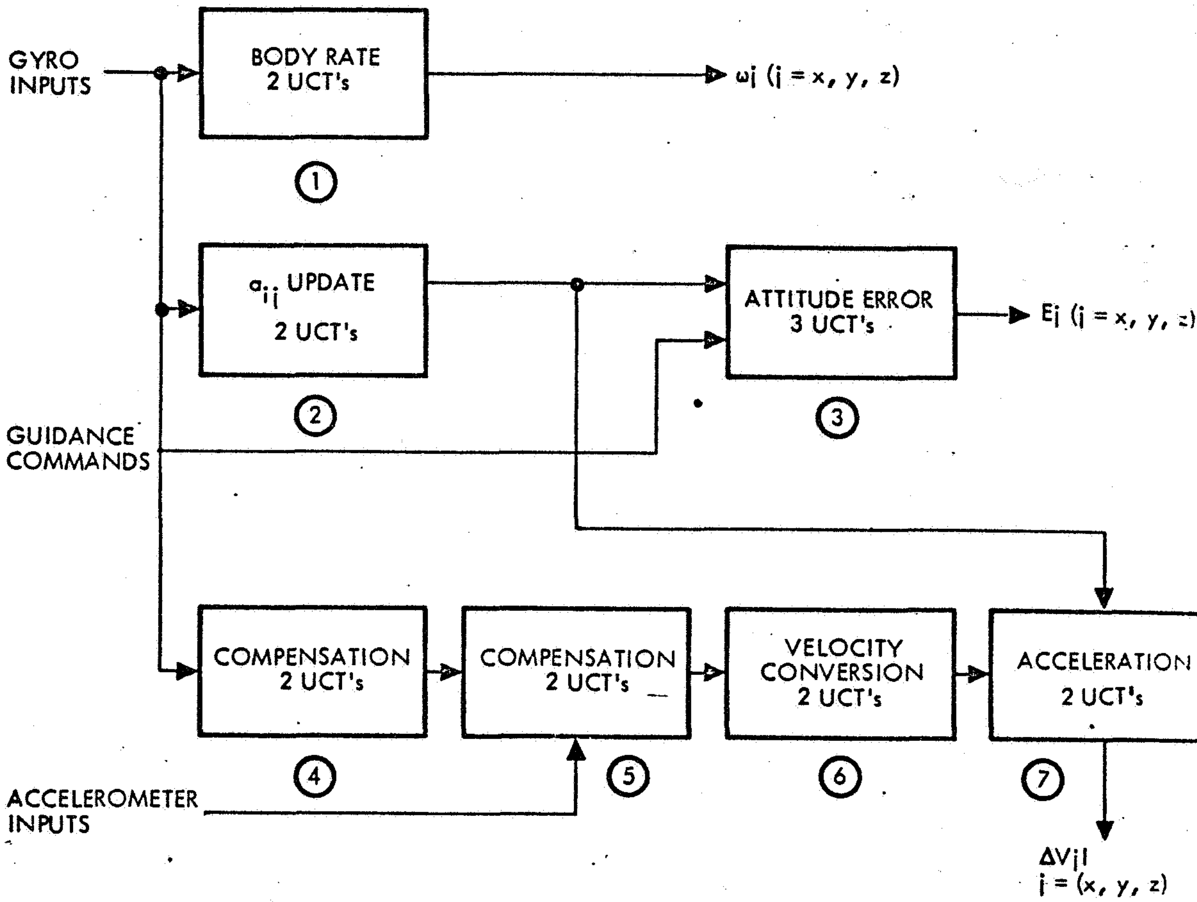


Figure 7-6. Flight Equations Flow Diagram

WORD TIME	4	3	2	1	FUNCTION
EQUATION BLOCK	7	6	5	4	
DELAY LINE					
1	(a_{11})	(b_{11})	(c_{1x})	(b_{11})	
2	(a_{21})	(b_{12})	(c_{2x})	(b_{21})	ΔV_{X1}
3	Σ (a_{31})	Σ (b_{13})	Σ (c_{3x})	Σ (b_{31})	
4	(a_{21})	(b_{21})	(c_{1y})	(b_{12})	
5	(a_{22})	(b_{22})	(c_{2y})	(b_{22})	ΔV_{Y1}
6	Σ (a_{32})	Σ (b_{33})	Σ (c_{3y})	Σ (b_{32})	
7	(a_{13})	(b_{31})	(c_{1z})	(b_{13})	
8	(a_{23})	(b_{32})	(c_{2z})	(b_{23})	ΔV_{Z1}
9	Σ (a_{33})	Σ (b_{33})	Σ (c_{3z})	Σ (b_{33})	

Figure 7-7. Delay Line Assignments

WORD TIME	4	3	2	1	FUNCTION
EQUATION BLOCK	1	3 AND 1	3 AND 1	2	
DELAY LINE					
10		a_{31R}	a_{21}	a_{21}	Δa_{11}
11				$\Sigma\ a_{31}$	
12		a_{32R}	a_{22}	a_{31}	Δa_{21}
13				$\Sigma\ a_{11}$	E_x
14	50	50	$\dot{a}_{xb}$	a_{11}	Δa_{31}
15		$\Sigma\ a_{33R}$	$\Sigma\ a_{23}$	$\Sigma\ a_{21}$	$\dot{a}_{xb}$
16		a_{11R}	a_{31}	a_{22}	Δa_{12}
17				$\Sigma\ a_{32}$	
18		a_{12R}	a_{32}	a_{32}	Δa_{22}
19				$\Sigma\ a_{12}$	E_y
20	50	50	$\dot{a}_{yb}$	a_{12}	Δa_{32}
21		$\Sigma\ a_{13R}$	$\Sigma\ a_{33}$	$\Sigma\ a_{22}$	$\dot{a}_{yb}$
22		a_{11R}	a_{21}	a_{23}	Δa_{13}
23				$\Sigma\ a_{33}$	
24		a_{12R}	a_{22}	a_{33}	Δa_{23}
25				$\Sigma\ a_{13}$	E_z
26	50	50	$\dot{a}_{zb}$	a_{13}	Δa_{33}
27		$\Sigma\ a_{13R}$	$\Sigma\ a_{23}$	$\Sigma\ a_{23}$	$\dot{a}_{zb}$

Figure 7-7. Delay Line Assignments (Continued)

The data in the delay line is envisaged as shown below, where R_n is the least significant bit of the remainder and Y_N the least significant bit of the Y register. The right-most bits precede the left-most bits in leaving the delay line.

$$R_0 \; Y_0 \; R_1 \; Y_1 \; - - - - - - - - - \; R_{n-1} \; Y_{n-1} \; R_n \; Y_n$$

Thus, the operands for the integration are interleaved in a common delay line. During the second half of word time 1, the increments which must be summed before they are used as Δx's in the next integrator are summed. One delay line within a triplet or pair is assigned the task of summing Δ's. The other delay lines simply wait until the summation is complete.

For example, during word time 1 (first half), delay lines 1, 2, and 3 and their respective arithmetic units are executing the equation

$$\Delta a_1^1 = b_{11}\Delta a_x + b_{21}\Delta a_y + b_{31}\Delta a_z. \tag{7-8}$$

During the second half of word time 1, delay line 3 is summing the resulting Δz's from delay lines 1, 2, and 3. The output of this summer is then used as a Δx input to integrators 2, 5, and 8 during word time 2.

The equations from blocks 1, 2, and 3 are all performed by delay lines 10 through 27. The data shown on the assignment chart are, of course, the contents of the Y register for that particular integrator. Delay lines 3, 6, 9, 11, 13, 15, 17, 19, 21, 23, 25, and 27 have an arithmetic unit associated with them which will permit summation of Δ's to be performed. All other delay lines have a more conventional arithmetic unit. A block diagram for those with summation logic is shown in Figure 7-8.

A block diagram of the necessary control functions to permit initialization, distribution, and readout is shown in Figure 7-9.

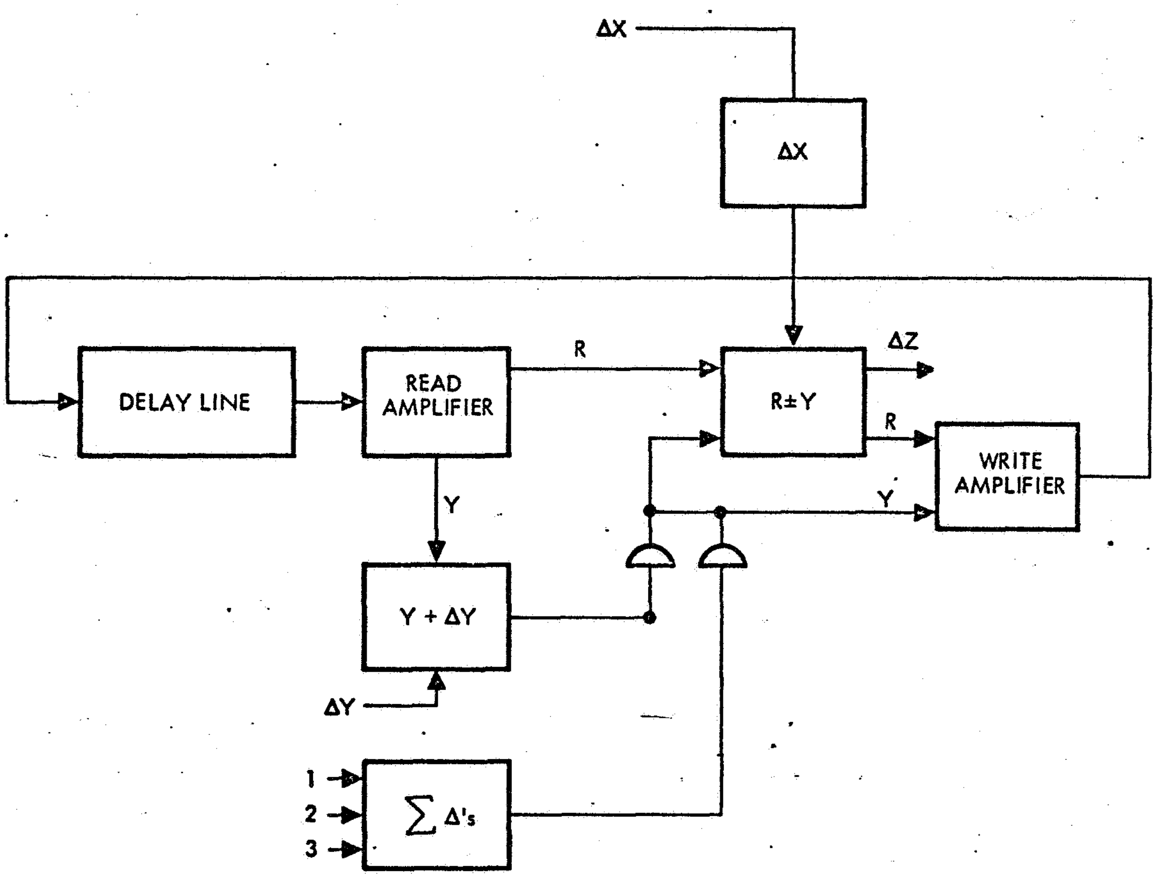


Figure 7-8. Individual Delay Line and Arithmetic Unit

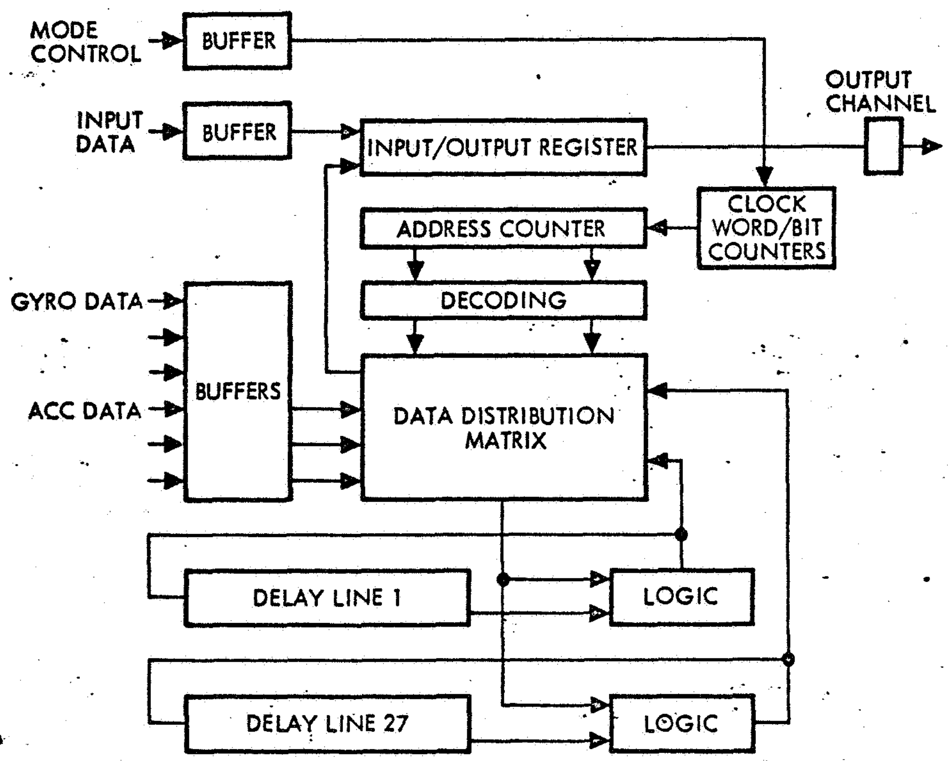


Figure 7-9. DDA Delay Line Storage Elements

7.5 GENERAL-PURPOSE COMPUTER

The study conducted during Phase I concerning a GP approach included alignment functions such as the ones new to this phase of the DDA study. Memory sizing and timing estimates are still valid. The new modified form of second-order Taylor series expansion is sufficiently similar to the one used in the Phase I study so as to require no new sizing effort.

During the Phase I study, it was assumed that the output of the GP computer should be in incremental form. This was a rather severe penalty to impose upon the GP configuration and further consideration has led to a communication channel whereby a single serial output register coupled with discrete information would not only meet requirements but also be more desirable.

In addition, the gyro data input repetition rate has increased. The input accumulators must now accommodate up to 7.6 ms/21.6 μsec/bit or 3519 bits. The size of the accumulators thus increases from 8 to 12 bits each. The word length has similarly been increased from 18 to 24 bits.

The influence of these changes are included in Table 7-2. The block diagram for the GP configuration is shown in Figure 7-10.

7.6 HYBRID COMPUTER

Gyro data are received at 46.2 kHz; however most of the equations need not be calculated at this rate. This preliminary study has revealed the desirability of having certain functions computed at different rates. Thus a certain amount of general purpose capability is required, especially during the alignment phase of operation.

It becomes practical to investigate the possibilities of using a hybrid configuration wherein the attitude reference equations are "solved" by the DDA and all other functions computed by the general purpose section. On the surface it would appear that many of the items troublesome to a large DDA would be avoided. A block diagram of the system is shown in Figure 7-11.

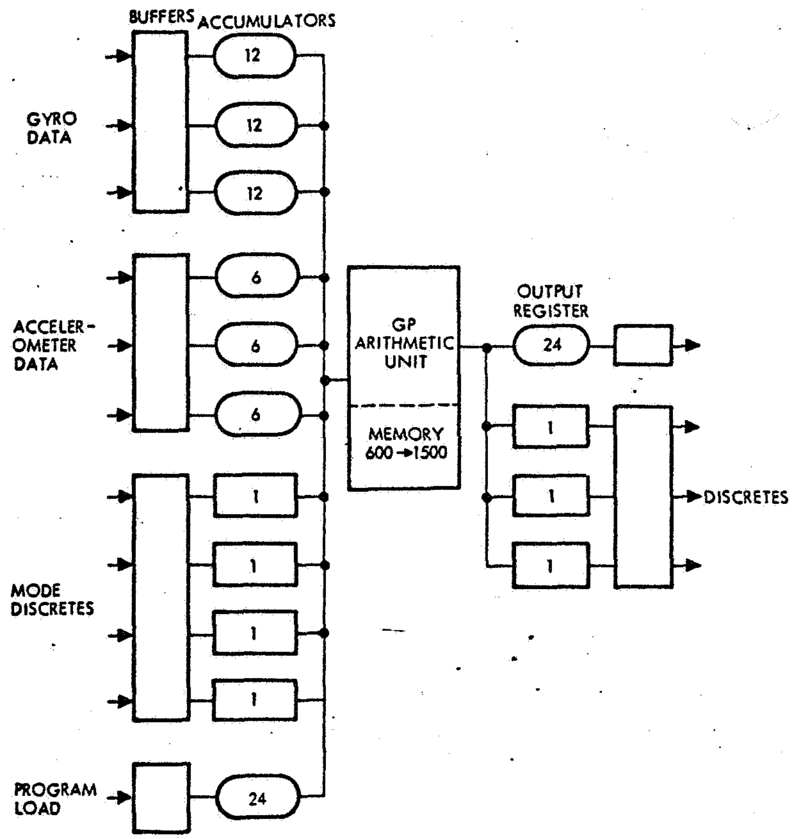


Figure 7-10. GP Mechanization

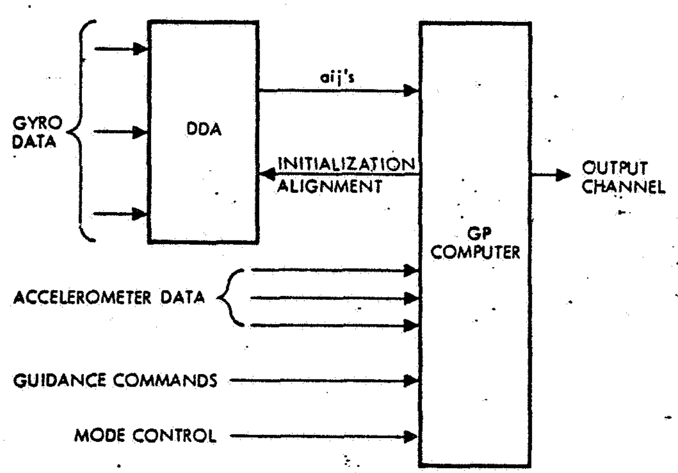


Figure 7-11. Hybrid Computer Mechanization

Since the GP device does not need to solve the attitude reference equations, the maximum iteration rate can be increased by about a factor of two. These equations, when solved by the GP computer, required the equivalent of 420 adds and 30 multiplies. In the example computer (TRW's LEM) this entailed 3.6 ms of computing time. This load is now borne by the DDA and the iteration time can be decreased from 7.6 ms to 7.6 - 3.6 = 4.0 ms. The iteration rate is then 250 Hz for all equations except the attitude reference equations which are executed at 100 kHz. (Unit cycle time for the DDA is tentatively set at 10 µsec.)

Naturally difficulties are encountered when one tries to mate these two nonsimilar devices. Time must be expended to transmit quantities to the GP device. A serial or parallel transfer system could be devised depending upon the characteristics of the GP device. The TRW LEM computer could accept this data either way by virtue of its logical construction. Since it is desirable to be able to preset (initialize) the DDA, a simple serial input/output register would be required in the DDA. A scheme for communication with the LEM computer is shown in Figure 7-12.

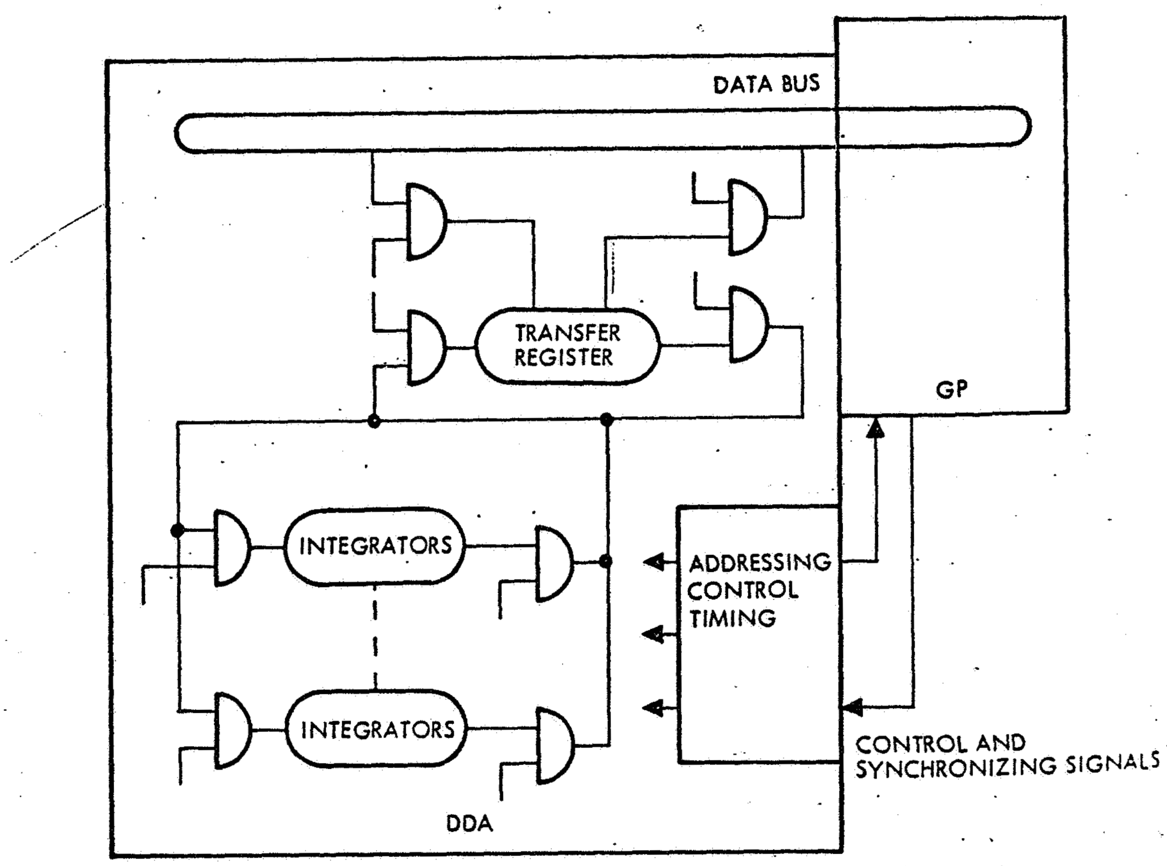


Figure 7-12. Communication Technique (Hybrid)

Two possibilities exist for sampling the A_{ij}'s. If they must be sampled all at once, sample registers (one for each A_{ij}) must be included within the DDA. It has been tentatively decided that serial sampling would not degrade system performance (this should, however, be verified). A binary counter could be included to "address" or route the A_{ij}'s serially both into and out of the DDA. It would take a maximum of 10 μsec to load the transfer register within the DDA. An additional 15 μsec would be required for the GP to retrieve the data. Thus all 9 direction cosines could be exchanged within 300 μsec.

Since the raw gyro data no longer need to be sent to the GP device, new equations must be generated which

a) Permit body rate data to be extracted from the available quantities

b) Allow the GP device to compute new A_{ij}'s during alignment instead of the angle increments

c) Enable the angle increment information to be formed for use in the accelerometer compensation equations. A cursory examination of this indicated that it is possible (by holding two successive A_{ij} matrices) to form increments in the same manner as the attitude error signals are generated. These increments can then be used in the required equations.

Because the computations within the DDA are self-contained, (i.e., partial results or incremental data such as ΔA_{ij} are not needed elsewhere) certain simplifications can be made within the unit itself. As was shown on a block diagram transmitted informally from NASA to TRW Systems, this results in a much simpler configuration. With proper consideration given to initialization and input and output, this system qualifies as a candidate for use in a hybrid configuration.

For the computations (first-order Taylor series) all A_{ij}'s must be available serially for every iteration. However since there is no need for ΔA_{ij}'s outside these units; they can be added to their respective A_{ij}'s as shown in Figure 7-13. Furthermore, the sharing of Y registers eliminates 9 registers required in a conventional approach.

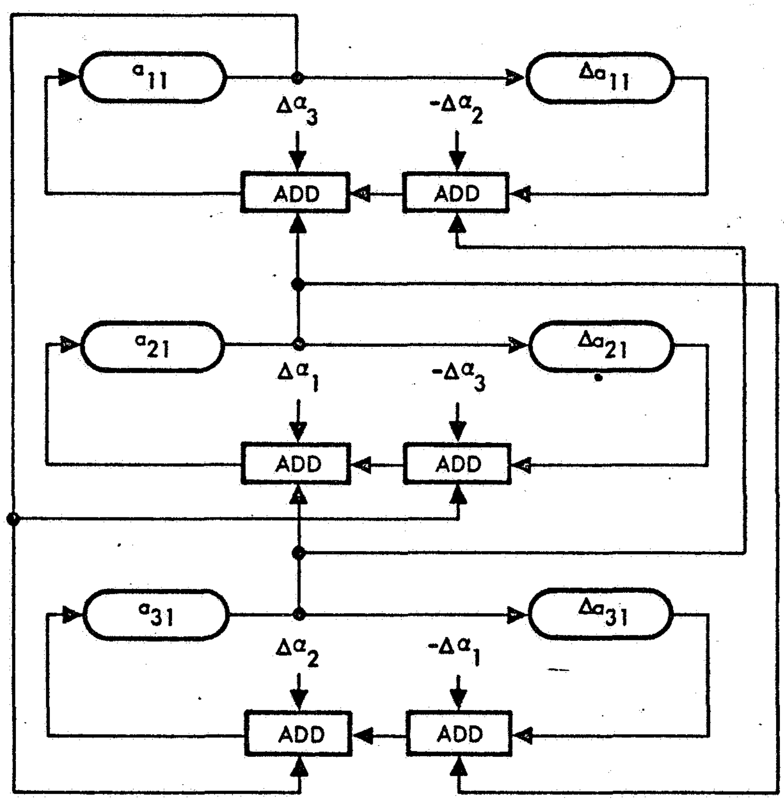


Figure 7-13. DDA Section of Hybrid (Single Axis)

APPENDIX I. ADDITIONAL ALGORITHM STUDIES

1. WHOLE NUMBER COMPUTER ALGORITHM TO REDUCE COMMUTATIVITY ERROR

1.1 Introduction

The existence of commutativity error in a strapdown attitude reference comes about not from any fundamental principle, but from the way in which most strapdown gyroscopic sensors are implemented. That is, the components of angular velocity are individually integrated, sampled, and quantized, and the resulting increments of angle are all that is available to the algorithm that determines attitude.

The information as to the variation of the direction of the angular velocity vector during the sampling period which was available from the unmodified components of the angular velocity, has been lost and must be assumed. The usual assumption is that the direction of the angular velocity vector is unchanging during the sampling period. Whenever the assumption differs from the actuality, commutativity errors occur.

At least four approaches to the reduction of commutativity error exist:

- Increase the sampling rate of the basic algorithm.
- Use a simplified algorithm at the increased sampling rate while keeping the basic algorithm at a slow rate.
- Modify the sensor to provide additional information such as angular acceleration and use an algorithm capable of making use of this information.
- Attempt to recover some of the lost information by use of differentiation and use an algorithm capable of making use of this information.

The last is the approach taken here.

1.2 Magnitude of the Problem

Consider an IMU subjected to a circular coning motion about the x axis such that the coning half-angle is β and the coning angular frequency is Ω. The angular velocity is

$$\{\omega\} = \begin{Bmatrix} \Omega(1 - \cos\beta) \\ \Omega \sin\beta \cos\Omega t \\ \Omega \sin\beta \cos(\Omega t + \frac{\pi}{2}) \end{Bmatrix} \qquad \text{(I-1)}$$

Note that a small constant angular velocity is present on the x axis. It has been introduced to cause the IMU to return to its original orientation after each coning cycle, facilitating analysis. If the angular velocity components are individually integrated and sampled, and the resultant angular increments are used in a perfect algorithm assuming constant direction of the angular velocity vector during the sampling period, then the commutativity error drift rate about the x axis, as is shown in Section I-1.6,

$$\omega_D = \Omega^3 T^2 \beta^2 / 12 \qquad \text{(I-2)}$$

where T is the sampling period. Consider the case

$$\Omega = 1 \text{ cps} = 2\pi \text{ rad/sec}$$

$$T = 1/32 \text{ sec}$$

$$\beta = 0.5 \text{ deg} = .00873 \text{ rad}$$

then

$$\omega_D = .154 \times 10^{-5} \text{ rad/sec} = .317 \text{ deg/hr}$$

It can be seen that commutativity error can have considerable effect on a high accuracy system.

1.3 Approach to the Problem

The gyros will be sampled at twice the basic attitude algorithm sampling rate. Once each sampling period, prior to each execution of the basic algorithm, the six angular increments will be preprocessed to obtain three angular increments for input to the basic algorithm. The preprocessor is the heart of the technique. It operates under the assumption that the angular acceleration of the IMU is constant during the sampling period. The three outputs are the three angular increments which an ideal IMU would produce if it were rotated from the orientation at the start of the sampling period to the orientation at the end of the sampling period at constant angular velocity. Therefore the basic algorithm can be of the usual variety which assumes constant direction of the angular velocity vector.

1.4 Derivation of Preprocessor

Quaternions will be used to facilitate the derivation. The quaternion ρ defining the orientation of the IMU satisfies the differential equation

$$\dot{\rho} = \tfrac{1}{2}\rho\omega \qquad \text{(I-3)}$$

where

$$\omega = \omega_1 i + \omega_2 j + \omega_3 k + 0 \qquad \text{(I-4)}$$

and the ω_i are the angular velocity components. Assume

$$\omega(t) = \omega(t_0) + \dot{\omega}(t_0)(t - t_0) \qquad \text{(I-5)}$$

and expand ρ in a Taylor series

$$\rho(t) = \rho(t_0) + \dot{\rho}(t_0)(t - t_0) + \ddot{\rho}(t_0)\frac{(t - t_0)^2}{2!} + \ldots \qquad \text{(I-6)}$$

Substitution of (I-5) and (I-6) in (I-3) gives

$$\begin{aligned}\dot{\rho}(t) = \tfrac{1}{2}\{&\rho(t_0)\omega(t_0) + [\rho(t_0)\dot{\omega}(t_0) + \dot{\rho}(t_0)\omega(t_0)](t - t_0) \\ &+ [2\dot{\rho}(t_0)\dot{\omega}(t_0) + \ddot{\rho}(t_0)\omega(t_0)]\frac{(t - t_0)^2}{2!} + \ldots\}\end{aligned} \qquad \text{(I-7)}$$

Equation (I-3) can be used to eliminate the derivatives of ρ from (I-7)

$$\begin{aligned}\dot{\rho}(t) = \tfrac{1}{2}\rho(t_0)\{&\omega(t_0) + [\dot{\omega}(t_0) + \frac{\omega^2(t_0)}{2}](t - t_0) \\ &+ [\omega(t_0)\dot{\omega}(t_0) + \frac{\omega^3(t_0)}{4} + \frac{\dot{\omega}(t_0)\omega(t_0)}{2}]\frac{(t - t_0)^2}{2!} + \ldots\}\end{aligned} \qquad \text{(I-8)}$$

The value of ρ at the end of the sampling period is

$$\rho(t_0 + T) = \rho(t_0) + \int_{t_0}^{t_0+T} \dot{\rho}(t)dt \qquad \text{(I-9)}$$

Therefore

$$\rho(t_0 + T) = \rho(t_0)\{1 + \tfrac{1}{2}\omega_0(t_0)T + \tfrac{1}{2}[\dot{\omega}(t_0) + \frac{\omega^2(t_0)}{2}]\frac{T^2}{2!} + \tfrac{1}{2}[\omega(t_0)\dot{\omega}(t_0) + \frac{\omega^3(t_0)}{4} + \frac{\dot{\omega}(t_0)\omega(t_0)}{2}]\frac{T^3}{3!} + \ldots\} \qquad \text{(I-10)}$$

The gyro outputs may be represented by

$$\theta_1 = \int_{t_0}^{t_0+T/2} [\omega(t_0) + \dot{\omega}(t_0)(t - t_0)]dt = \frac{\omega(t_0)T}{2} + \frac{\dot{\omega}(t_0)T^2}{8} \qquad \text{(I-11)}$$

and

$$\theta_2 = \int_{t_0+T/2}^{t_0+T} [\omega(t_0) + \dot{\omega}(t_0)(t - t_0)]dt = \frac{\omega(t_0)T}{2} + \frac{3\dot{\omega}(t_0)T^2}{8} \qquad \text{(I-12)}$$

Solving (I-11) and (I-12) simultaneously gives

$$\omega(t_0) = (3\theta_1 - \theta_2)/T \qquad \text{(I-13)}$$

$$\dot{\omega}(t_0) = 4(\theta_2 - \theta_1)/T^2 \qquad \text{(I-14)}$$

Substitution of (I-13) and (I-14) into (I-10), discard of terms higher than second order in θ_1 and θ_2 and introduction of

$$\rho(t_0 + T) = \rho(t_0)\Delta\rho \qquad \text{(I-15)}$$

gives

$$\Delta\rho = 1 + \frac{\theta_1 + \theta_2}{2} + \frac{-9\theta_1^{\,2} + 19\theta_1\theta_2 + 11\theta_2\theta_1 - 9\theta_2^{\,2}}{24} \qquad \text{(I-16)}$$

Let the quaternions θ_1 and θ_2 be expressed in terms of their components

$$\theta_i = \theta_{i1}i + \theta_{i2}j + \theta_{i3}k + 0; \qquad i = 1, 2 \qquad \text{(I-17)}$$

Then

$$\Delta\rho = \left[\frac{\theta_{11}+\theta_{21}}{2} + \frac{\theta_{12}\theta_{23}-\theta_{13}\theta_{22}}{3}\right]i + \left[\frac{\theta_{12}+\theta_{22}}{2} + \frac{\theta_{13}\theta_{21}-\theta_{11}\theta_{23}}{3}\right]j$$

$$+ \left[\frac{\theta_{13}+\theta_{23}}{2} + \frac{\theta_{11}\theta_{22}-\theta_{12}\theta_{21}}{3}\right]k + 1 \qquad \text{(I-18)}$$

$$+ \frac{9(\theta_{11}{}^2+\theta_{12}{}^2+\theta_{13}{}^2) - 30(\theta_{11}\theta_{21}+\theta_{12}\theta_{22}+\theta_{13}\theta_{23}) + 9(\theta_{21}{}^2+\theta_{22}{}^2+\theta_{23}{}^2)}{24}$$

If $\Delta\alpha_i$; $i = 1, 2, 3$ represent the outputs of the hypothetical IMU performing the same orientation change at constant angular velocity, then

$$\Delta\rho = (\Delta\alpha_1 i + \Delta\alpha_2 j + \Delta\alpha_3 k)\frac{\sin\frac{\Delta\alpha_0}{2}}{\Delta\alpha_0} + \cos\frac{\Delta\alpha_0}{2} \qquad \text{(I-19)}$$

where

$$\Delta\alpha_0 = (\Delta\alpha_1{}^2 + \Delta\alpha_2{}^2 + \Delta\alpha_3{}^2)^{\frac{1}{2}} \qquad \text{(I-20)}$$

Therefore

$$\Delta\alpha_i = \frac{2\sin^{-1}(\Delta\rho_1{}^2 + \Delta\rho_2{}^2 + \Delta\rho_3{}^2)^{\frac{1}{2}}}{(\Delta\rho_1{}^2 + \Delta\rho_2{}^2 + \Delta\rho_3{}^2)^{\frac{1}{2}}}\Delta\rho_i; \quad i = 1, 2, 3 \qquad \text{(I-21)}$$

Substitution of the components of $\Delta\rho$ from (I-18) into (I-21) and discarding terms higher than second order gives the preprocessing algorithm

$$\Delta\alpha_1 = \theta_{11} + \theta_{21} + \tfrac{2}{3}(\theta_{12}\theta_{23} - \theta_{13}\theta_{22}) \qquad \text{(I-22)}$$

$$\Delta\alpha_2 = \theta_{12} + \theta_{22} + \tfrac{2}{3}(\theta_{13}\theta_{21} - \theta_{11}\theta_{23}) \qquad \text{(I-23)}$$

$$\Delta\alpha_3 = \theta_{13} + \theta_{23} + \tfrac{2}{3}(\theta_{11}\theta_{22} - \theta_{12}\theta_{21}) \qquad \text{(I-24)}$$

1.5 Accuracy Improvement

The commutativity error drift rate about the x axis in the presence of circular coning, as is shown in Section I-1.6, is

$$\omega_D = \Omega^5 T^4 \beta^2/960 \qquad \text{(I-25)}$$

For the same case as before,

$$\omega_D = .738 \times 10^{-8} \text{ rad/sec} = .00152 \text{ deg/hr}$$

If the sampling rate had merely been doubled, without use of the pre-processor, the drift rate would have been reduced to

$$\omega_D = .385 \times 10^{-6} \text{ rad/sec} = .0793 \text{ deg/hr}$$

and the basic algorithm would have had to be computed twice as often. The ratio of improvement is about 50:1 for this case.

1.6 Commutativity Error in the Presence of Circular Coning Motion

From (I-1) the angular velocity applied to the IMU for circular coning motion is represented by the quaternion

$$\omega = \Omega(1 - \cos\beta)i + \Omega\sin\beta\cos\Omega t j + \Omega\sin\beta\cos(\Omega t + \tfrac{\pi}{2})k \qquad \text{(I-}$$

The quaternion defining the orientation of the IMU is the solution of the differential equation

$$\dot{\rho} = \tfrac{1}{2}\rho\omega; \qquad \rho(0) = 1 \qquad \text{(I-}$$

which is

$$\rho(t) = \tfrac{1}{2}[(1 - \cos\beta)\sin\Omega t i + \sin\beta\sin\Omega t j$$
$$- \sin\beta(1 - \cos\Omega t)k + (1 + \cos\beta) + (1 - \cos\beta)\cos\Omega t] \qquad \text{(I-}$$

The ideal quaternion at the $n+1^{th}$ sampling instant is related to the ideal quaternion at the n^{th} sampling instant by

$$\rho(nT + T) = \rho(nT)\Delta\rho(nT) \qquad \text{(I-}$$

Therefore the ideal $\Delta\rho$ is given by

$$\Delta\rho(nT) = \rho^*(nT)\rho(nT + T) \qquad \text{(I-}$$

and

$$
\begin{aligned}
\Delta\rho(nT) \;=\;& \tfrac{1}{2}\{(1 - \cos\beta)\sin\Omega T i \\
&+ \sin\beta[-\sin n\Omega T(1 - \cos\Omega T) + \cos n\Omega T \sin\Omega T]j \\
&- \sin\beta[\cos n\Omega T(1 - \cos\Omega T) + \sin n\Omega T \sin\Omega T]k\,\} \\
&+ 1 - \tfrac{1}{2}(1 - \cos\beta)(1 - \cos\Omega T)
\end{aligned}
\tag{I-31}
$$

Now consider an IMU where the individual components of angular velocity are integrated and sampled. The gyro outputs for the n^{th} sampling instant are

$$
\theta(nT) = \int_{nT}^{nT+T} \omega dt \tag{I-32}
$$

so that

$$
\begin{aligned}
\theta(nT) = \;& \Omega T(1 - \cos\beta)i + \sin\beta[-\sin n\Omega T(1-\cos\Omega T) + \cos n\Omega T \sin\Omega T]j \\
&- \sin\beta[\cos n\Omega T(1 - \cos\Omega T) + \sin n\Omega T \sin\Omega T]k
\end{aligned}
\tag{I-33}
$$

If the algorithm assumes that the angular velocity is constant during the sampling period, the quaternion representing the rotation is

$$
\Delta\rho' = \theta\frac{\sin\frac{1}{2}\theta_0}{\theta_0} + \cos\tfrac{1}{2}\theta_0 \tag{I-34}
$$

where

$$
\theta_0 = (\theta_1{}^2 + \theta_2{}^2 + \theta_3{}^2)^{\frac{1}{2}} \tag{I-35}
$$

Substitution of the components of θ from (I-33) into (I-35) and discard of higher than second order terms in β gives

$$
\theta_0{}^2 \cong 2\beta^2(1 - \cos\Omega T) \tag{I-36}
$$

Therefore

$$
\frac{\sin\frac{\theta_0}{2}}{\theta_0} \cong \tfrac{1}{2}[1 + \beta^2(1 - \cos\Omega T)/12] \tag{I-37}
$$

$$
\cos\frac{\theta_0}{2} \cong 1 - \beta^2(1 - \cos\Omega T)/4 \tag{I-38}
$$

Discarding terms of higher than second-order in β gives

$$\Delta\rho' = \tfrac{1}{2}\{\Omega T\ \beta^2/2\ i + \beta[-\sin n\Omega T(1-\cos\Omega T) + \cos n\Omega T \sin \Omega T]j$$
$$- \beta[\cos n\Omega T(1-\cos\Omega T) + \sin n\Omega T \sin\Omega T]k\} + 1 - \beta^2(1-\cos\Omega T)/4$$

Discarding terms of higher than second-order in β in Equation (I-31) gives an expression for the ideal $\Delta\rho$

$$\Delta\rho = \tfrac{1}{2}\{\sin\Omega T\ \beta^2/2\ i + \beta[-\sin n\Omega T(1-\cos\Omega T) + \cos n\Omega T \sin \Omega T]j$$
$$- \beta[\cos n\Omega T(1-\cos\Omega T) + \sin n\Omega T \sin\Omega T]k\} + 1 - \beta^2(1-\cos\Omega T)/4$$

Note that (I-39) and (I-40) differ only in the $\Delta\rho_1$ term. The angular error per cycle about the x axis may be approximated by

$$\epsilon = 2(\Delta\rho'_1 - \Delta\rho_1) \qquad \text{(I-...)}$$

$$\epsilon = (\Omega T - \sin\Omega T)\beta^2/2 \qquad \text{(I-...)}$$

$$\epsilon \cong \Omega^3 T^3 \beta^2/12 \qquad \text{(I-...)}$$

The drift rate is then

$$\omega_D = \epsilon/T = \Omega^3 T^2 \beta^2/12 \qquad \text{(I-...)}$$

which is the required result. Simulation of the exact equations has demonstrated that this expression is accurate for coning half angles less than about ten degrees and for sampling rates greater than about six samples per coning cycle.

Now suppose the commutativity preprocessing algorithm is used. The gyro output angles are

$$\theta_1(nT) = \int_{nT}^{nT+\frac{T}{2}} \omega dt \qquad \text{(I-...)}$$

and

$$\theta_2(nT) = \int_{nT+\frac{T}{2}}^{nT+T} \omega dt \tag{I-46}$$

If the expressions for the components of θ_1 and θ_2 from (I-45) and (I-46) are substituted into (I-22), (I-23), and (I-24) and the results treated as the θ components were above, it is found that

$$\omega_D = \Omega^5 T^4 \beta^2/960 \tag{I-47}$$

2. HYBRID-INCREMENTAL WHOLE NUMBER ALGORITHM

2.1 Introduction

Incremental, whole number, and hybrid incremental-whole number digital computers have all been proposed for use in strapdown inertial navigation systems. If the relative merits of these three hardware configurations are to be fairly compared, an appropriate algorithm must be selected for each. Much effort has been expended in the study of algorithms for incremental and whole number computers. This section proposes an algorithm for a hybrid computer and discusses some of the problems of implementation.

2.2 Description of Algorithm

The algorithm consists of two parts. The incremental part contains Euler parameter attitude calculations and body axis velocity calculations performed at a frequency of about 1 to 5 kHz. The whole number part contains the velocity transformation, gravity calculations and velocity integration performed at a frequency of about 1 to 5 Hz. The use of body axis velocity calculations in the incremental part of the algorithm permits the velocity transformation to be performed at a low sampling rate in the whole number algorithm. This technique greatly reduces the complexity of the incremental part of the computer without overburdening the whole number part.

The Euler parameter calculations are:

$$
\begin{aligned}
d\rho_1 &= \tfrac{1}{2}(\rho_4 d\theta_1 - \rho_3 d\theta_2 + \rho_2 d\theta_3) \\
d\rho_2 &= \tfrac{1}{2}(\rho_3 d\theta_1 + \rho_4 d\theta_2 - \rho_1 d\theta_3) \\
d\rho_3 &= \tfrac{1}{2}(-\rho_2 d\theta_1 + \rho_1 d\theta_2 + \rho_4 d\theta_3) \\
d\rho_4 &= \tfrac{1}{2}(-\rho_1 d\theta_1 - \rho_2 d\theta_2 - \rho_3 d\theta_3)
\end{aligned}
\qquad \text{(I-[illegible])}
$$

where the ρ_i are the Euler parameters, the $d\rho_i$ are increments of the ρ_i, and the $d\theta_i$ are the angular increments from the gyroscopes.

The body axis velocity calculations are:

$$
\begin{aligned}
dV_{B_1} &= dV_1 - d\theta_2 V_{B_3} + d\theta_3 V_{B_2} \\
dV_{B_2} &= dV_2 - d\theta_3 V_{B_1} + d\theta_1 V_{B_3} \\
dV_{B_3} &= dV_3 - d\theta_1 V_{B_2} + d\theta_2 V_{B_1}
\end{aligned}
\tag{I-49}
$$

where the V_{B_i} are the components of body axis velocity, the dV_{B_i} are increments of the V_{B_i}, and the dV_i are the velocity increments from the accelerometers.

The velocity transformation may be performed in either of two ways, depending upon which way is most efficient for the particular whole number computer instruction repertoire selected. The first method is:

$$
\Delta \underline{V}_I = B\underline{V}_B \tag{I-50}
$$

where

$$
B = \begin{bmatrix}
\rho_1^2 - \rho_2^2 - \rho_3^2 + \rho_4^2 & 2(\rho_1\rho_2 - \rho_3\rho_4) & 2(\rho_3\rho_1 + \rho_2\rho_4) \\
2(\rho_1\rho_2 + \rho_3\rho_4) & -\rho_1^2 + \rho_2^2 - \rho_3^2 + \rho_4^2 & 2(\rho_2\rho_3 - \rho_1\rho_4) \\
2(\rho_3\rho_1 - \rho_2\rho_4) & 2(\rho_2\rho_3 + \rho_1\rho_4) & -\rho_1^2 - \rho_2^2 + \rho_3^2 + \rho_4^2
\end{bmatrix} \tag{I-51}
$$

and the second method is

$$
\begin{aligned}
\mu_1 &= \rho_4 V_{B_1} - \rho_3 V_{B_2} + \rho_2 V_{B_3} \\
\mu_2 &= \rho_3 V_{B_1} + \rho_4 V_{B_2} - \rho_1 V_{B_3} \\
\mu_3 &= -\rho_2 V_{B_1} + \rho_1 V_{B_2} + \rho_4 V_{B_3} \\
\mu_4 &= \rho_1 V_{B_1} + \rho_2 V_{B_2} + \rho_3 V_{B_3}
\end{aligned}
\tag{I-52}
$$

and

$$
\begin{aligned}
\Delta V_{I_1} &= \rho_4\mu_1 - \rho_3\mu_2 + \rho_2\mu_3 + \rho_1\mu_4 \\
\Delta V_{I_2} &= \rho_3\mu_1 + \rho_4\mu_2 - \rho_1\mu_3 + \rho_2\mu_4 \\
\Delta V_{I_3} &= -\rho_2\mu_1 + \rho_1\mu_2 + \rho_4\mu_3 + \rho_3\mu_4
\end{aligned}
\qquad \text{(I-53)}
$$

where $\Delta \underline{V}_I$ is the incremental initial velocity for one whole number computation cycle, B is the direction cosine matrix, and the μ_i are temporary storage. The first method requires 19 multiplies and 21 adds (exclusive of shifts) while the second method requires 24 multiplies and 17 adds, so that the first method would probably be more efficient than the second.

Whenever $\underline{V}_B$ is transmitted to the whole number section of the computer for use in (I-50) or (I-52), it is zeroed out in the Y registers of the incremental part of the computer. However, when the ρ_i are transmitted to the whole number section for use in (I-51) or (I-52) and (I-53), they are not zeroed out.

The velocity integration may be performed by a number of techniques. The following technique, using trapezoidal integration, is quite efficient:

$$
\begin{aligned}
&\Delta \underline{V}_I \leftarrow \Delta \underline{V}_I + \frac{T}{2}\,\underline{a}_G(\underline{x}_I) \\
&\underline{x}_I \leftarrow \underline{x}_I + T\underline{V}_I \\
&\Delta \underline{V}_I \leftarrow \Delta \underline{V}_I + \frac{T}{2}\,\underline{a}_G(\underline{x}_I) \\
&\underline{x}_I \leftarrow \underline{x}_I + \frac{T}{2}\,\Delta \underline{V}_I \\
&\underline{V}_I \leftarrow \underline{V}_I + \Delta \underline{V}_I
\end{aligned}
\qquad \text{(I-54)}
$$

where T is the whole number iteration period, $\underline{a}_G$ is the gravitational acceleration, $\underline{x}_I$ is the inertial position, and $\underline{V}_I$ is the inertial velocity. The equations comprising (I-54) must be performed in the order given except that the last two may be interchanged.

2.3 Implementation of Algorithm - Euler Parameters

The implementation of the whole number part of the computer will not be discussed here. The implementation of (I-48) would require 12 DDA integrations. This would be an inefficient implementation, however. An incremental computer requires only 4 Y registers and 4 R registers to solve (I-48). Each R register would have to be processed three times per cycle and each Y register once per cycle. The R registers should all be processed three times before the Y registers are processed. This technique ("parallel") gives third order drift errors. Three other orders of processing were investigated and found to give second order drift errors:

Input Serial:

Process R registers for $d\theta_1$

Process Y registers

Repeat for $d\theta_2$ and $d\theta_3$

R Serial:

Process R_1 three times

Process Y_1

Repeat for R_2, Y_2; R_3, Y_3; R_4, Y_4

Y Serial:

Process R_2, R_3, R_4 with Y_1

Process Y_2, Y_3, Y_4

Repeat with Y_2, Y_3, Y_4

Figure I-1 shows a partial flow chart of a possible organization for an incremental computer to solve (I-48). It is a serial-parallel arrangement with four adders to perform the 16 additions required per cycle. The iteration period is broken up into four intervals T_1, T_2, T_3, and T_4. The R registers are processed during T_1, T_2, and T_3 and the Y registers during T_4. Not shown are the provisions for zeroing the dz_i at the beginning of each iteration period or the buffering for the $d\theta_i$. Ternary inputs are used, with the $d\theta_i$ appearing as a pulse on either a plus line or a minus line or neither, but not both. The R registers and Y registers all circulate and might be

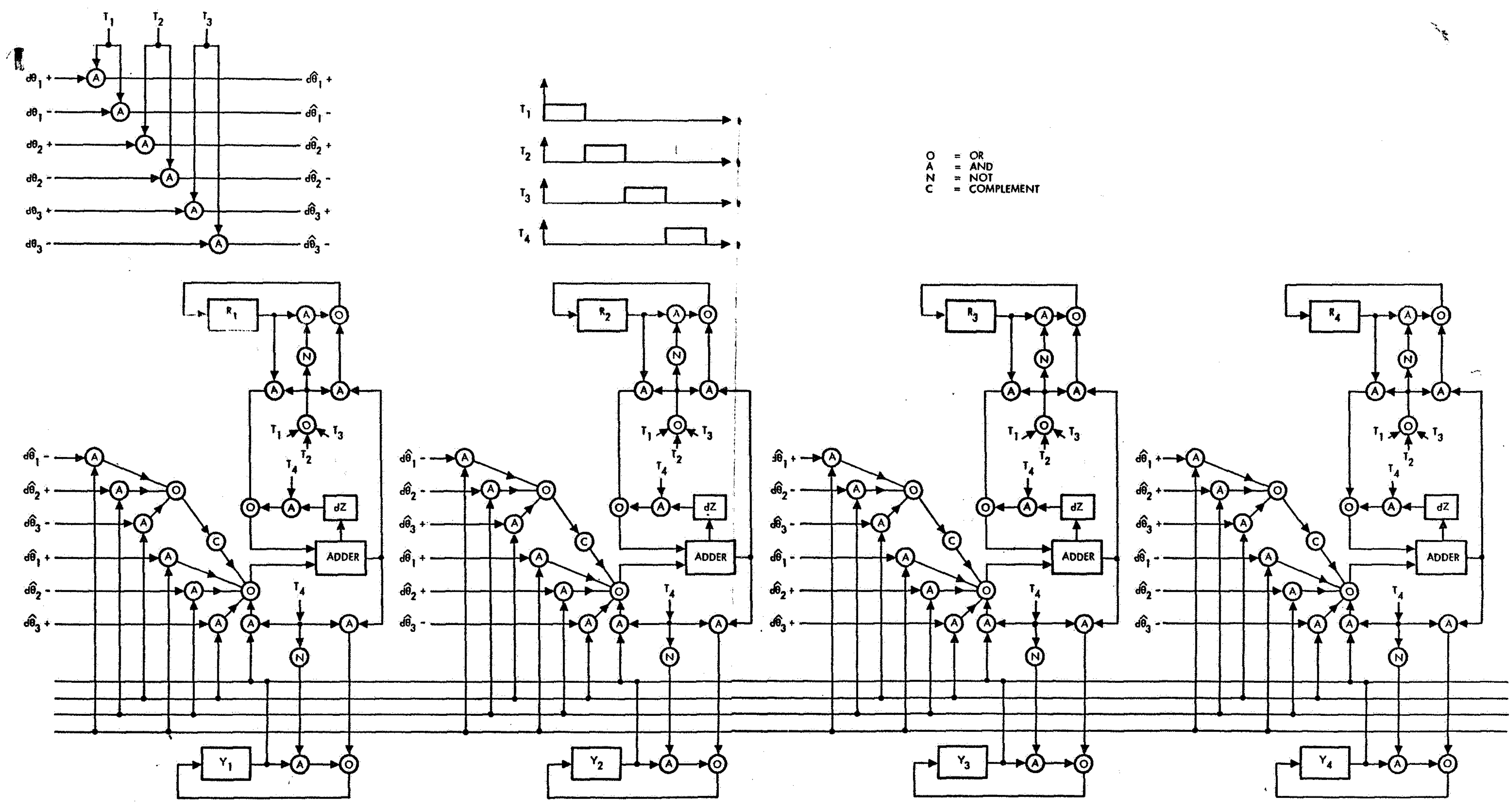


Figure I-1. Possible Incremental Computer Organization for Euler Parameters implemented by active microcircuits or delay lines. Depending upon the iteration rate and the type of circuitry selected, a completely serial organization may be feasible.

2.4 Scaling - Euler Parameters

The incremental computer organization can be kept simple if no more than one overflow of any R register is permitted per iteration. In each iteration period, three Y registers are added to each R register. The maximum value of an Euler parameter is one and the maximum value of the sum of three Euler parameters can easily be shown to be $3^{\frac{1}{2}}$. Therefore if the Euler parameters are scaled at any scaling greater than $3^{\frac{1}{2}}$, the sum of three of the scaled parameters cannot exceed unity. A convenient scaling is 2^1.

2.5 Implementation of Algorithm - Body Axis Velocity

An example organization has not been worked out for the body axis velocity equations. Six DDA integrators would be required. An incremental computer requires three Y registers and three R registers. For (I-49) the R registers must be processed twice each and the Y registers once each per iteration. Provision must be made for the accumulation of up to three dY increments. A somewhat more accurate algorithm than (I-49) can be achieved at the expense of processing each Y register a second time.

$$\begin{aligned}
&\underline{V}_B \leftarrow \underline{V}_B + \tfrac{1}{2}d\underline{V} \\
&V_{B_1} \leftarrow V_{B_1} + \tfrac{1}{2}dV_1 - d\theta_2 V_{B_3} + d\theta_3 V_{B_2} \\
&V_{B_2} \leftarrow V_{B_2} + \tfrac{1}{2}dV_2 - d\theta_3 V_{B_1} + d\theta_1 V_{B_3} \\
&V_{B_3} \leftarrow V_{B_3} + \tfrac{1}{2}dV_3 - d\theta_1 V_{B_2} + d\theta_2 V_{B_1}
\end{aligned} \qquad \text{(I-55)}$$

In this algorithm half of the velocity increments are added to the body axis velocity before the Coriolis terms are computed and half are added afterwards. The scaling depends upon the choice of iteration rates and the vehicle environment.

2.6 Initialization

The body axis velocity computations require no initialization, because the Y registers are zeroed every whole number computing cycle. The Euler parameter computations, however, must be initialized. At least two possibilities exist:

(1) Four slewing integrators accepting Y register inputs from the whole number computer and driving the Y registers containing the ρ's directly.

(2) Three slewing integrators accepting Y register inputs from the whole number computer and producing $d\theta_i$ outputs, plus provision for forcing the constraint (I-56) at computer turn on.

The methods are compared below:

	Method 1	Method 2
Number of integrators required	4	3
Special startup circuitry required	no	yes
Whole number computer can correct orthogonality	yes	no
Whole number computer can slew angular rates directly without calculating Euler parameter rates	no	yes

Unless orthogonality corrections were to be applied to the Euler parameters during flight, the above equipment would only be used during alignment. Provisions must be made that the $d\theta_i$ or $d\rho_i$ resulting from the gyro inputs are not lost when slewing inputs are present. One approach would be to suppress the appropriate slewing integrator whenever a nonzero $d\theta_i$ or $d\rho_i$ appears. The slewing integrator would take longer to complete its slew, but no information would be lost.

2.7 Orthogonality

The orthogonality errors of the direction cosine matrix B may become large enough to require correction in very high accuracy applications.

Whether or not such corrections are required in a specific case must be determined by further analysis. The requirement that B be orthogonal corresponds to a requirement that the Euler parameters satisfy the constraint.

$$\rho_o^{\,2} = \sum_{i=1}^{4} \rho_i^{\,2} = 1 \qquad \text{(I-56)}$$

The correction may be applied either to the Euler parameters themselves or to the transformed velocity components. The Euler parameter correction is:

$$\rho_i \leftarrow \frac{\rho_i}{\rho_o} \qquad i = 1, 2, 3, 4 \qquad \text{(I-57)}$$

and the velocity correction is:

$$\Delta \underline{V}_I \leftarrow \frac{\Delta V_I}{\rho_o^{\,2}} \qquad \text{(I-58)}$$

The latter is simpler, involving three components instead of four and requiring no square root operation. The division can be eliminated in either case by means of a first or higher order approximation. If the first initialization method were used, (I-57) may be implemented by calculating slewing commands in the whole number section of the computer and sending them to the incremental section. If the second initialization method were used, (I-58) would be more suitable, performed entirely in the whole number section.

APPENDIX II. ADDITIONAL CONING STUDIES

The coning results documented previously in Reference 1, Section 5 were obtained for four attitude reference algorithms while under the influence of a very limited type of vehicle coning. The vehicle coning errors were analyzed only for one case cycles applied to two of the vehicle axes. In this memo, these previous results are expanded for the second-order Taylor series algorithm to include vehicle coning which results from the application of the following limit cycles:

(1) Equal frequency limit cycles

  (a) 90-degree phased, different amplitudes ($\leq$ one degree)

  (b) phase $\neq$ 90 degrees, equal amplitudes (-180 deg < phase < $+180$ deg)

  (c) phase $\neq$ 90 degrees, different amplitudes*

(2) Different frequency limit cycles

  (a) through (c) as described above

(3) Since the previous results were obtained only for frequencies up to one-half the sampling frequency (f_s), the above tests (a) through (c) will also be performed for frequencies higher than $f_{s/2}$.

(4) The previous coning results were presented on a graph of percent error of the true vehicle coning rate versus the limit cycle frequency normalized to the computer sampling frequency (see Figure II-1). After the f/f_s ratio reached 0.01, these previous results were affected by the roundoff quantization errors of the 7094 computer. To examine the effects of computer word length, several coning runs were performed on a double precision 7094 computer simulation and an Interpretive Computer Simulation of the LEM 18 bit computer.

---

*For these coning tests, the phase was varied from -180 deg to $+180$ deg and the amplitudes were ≤ 1 deg.

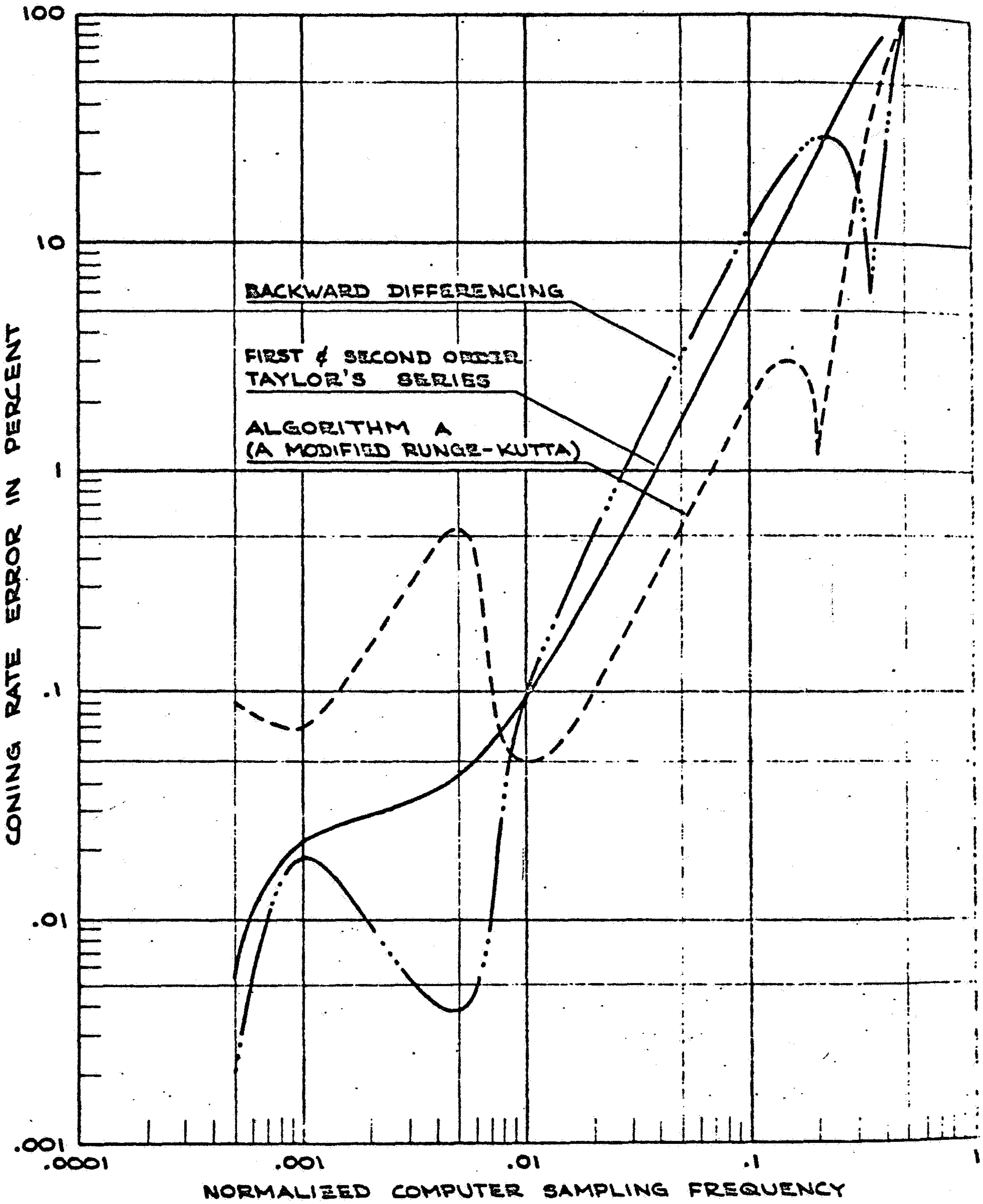


Figure II-1.

Previous Coning Study Results for Four Attitude Algorithms

The equal frequency limit cycle tests which were performed for different combinations of amplitudes and phasing showed that the true vehicle coning rate was proportional to the two amplitudes, the frequency, and the sine of the phase between the two limit cycles which were applied to the two vehicle axes. That is, the true vehicle coning rate ω_c about the third vehicle axis may be computed from Equation (II-1) or Equation (II-2).

$$\omega_c = A_1 A_2 \frac{\omega}{2} \sin \phi \text{ rad/sec} \qquad \text{(II-1)}$$

$$\omega_c = \frac{a_1 a_2}{2\omega} \sin \phi \text{ rad/sec} \qquad \text{(II-2)}$$

where

ω is the limit cycle radian frequency (rad/sec)

A_1, A_2 are the limit cycle amplitudes in radians

a_1, a_2 are the limit cycle amplitudes (rad/sec)

ϕ is the phase shift between the two limit cycles.

The percent error of the computer for any combination of phasing or amplitudes behaved exactly the same as the previous results which were obtained for equal amplitudes and 90 degree phasing. This implies that the errors due to coning are just due to sampling errors as the application of different amplitudes did not affect the percent error.*

The same type of tests were performed for frequencies higher than one-half the sampling frequency. The results of these tests are included in Figure II-2, which is also a graph of the percent error of the true vehicle coning rate versus normalized frequency. This graph also shows the effects of computer word length as the coning errors are shown for the 18 bit computer, the single precision 7094 computer, and the double precision 7094 computer. From the graph on Page II-4, the percent error due to sources other than roundoff appears to be of the form

$$\% \text{ error} = 100 \left[1 - \frac{\sin \omega T}{\omega T} \right]^{*} \qquad \text{(II-3)}$$

where

ω is the limit cycle frequency in radians per second

T is the computer sampling period in seconds ($T = \frac{1}{f_s}$)

*These conclusions and Equation II-3 appear to be very strongly justified by the simulation results, but should be analytically proved before accepted as correct.

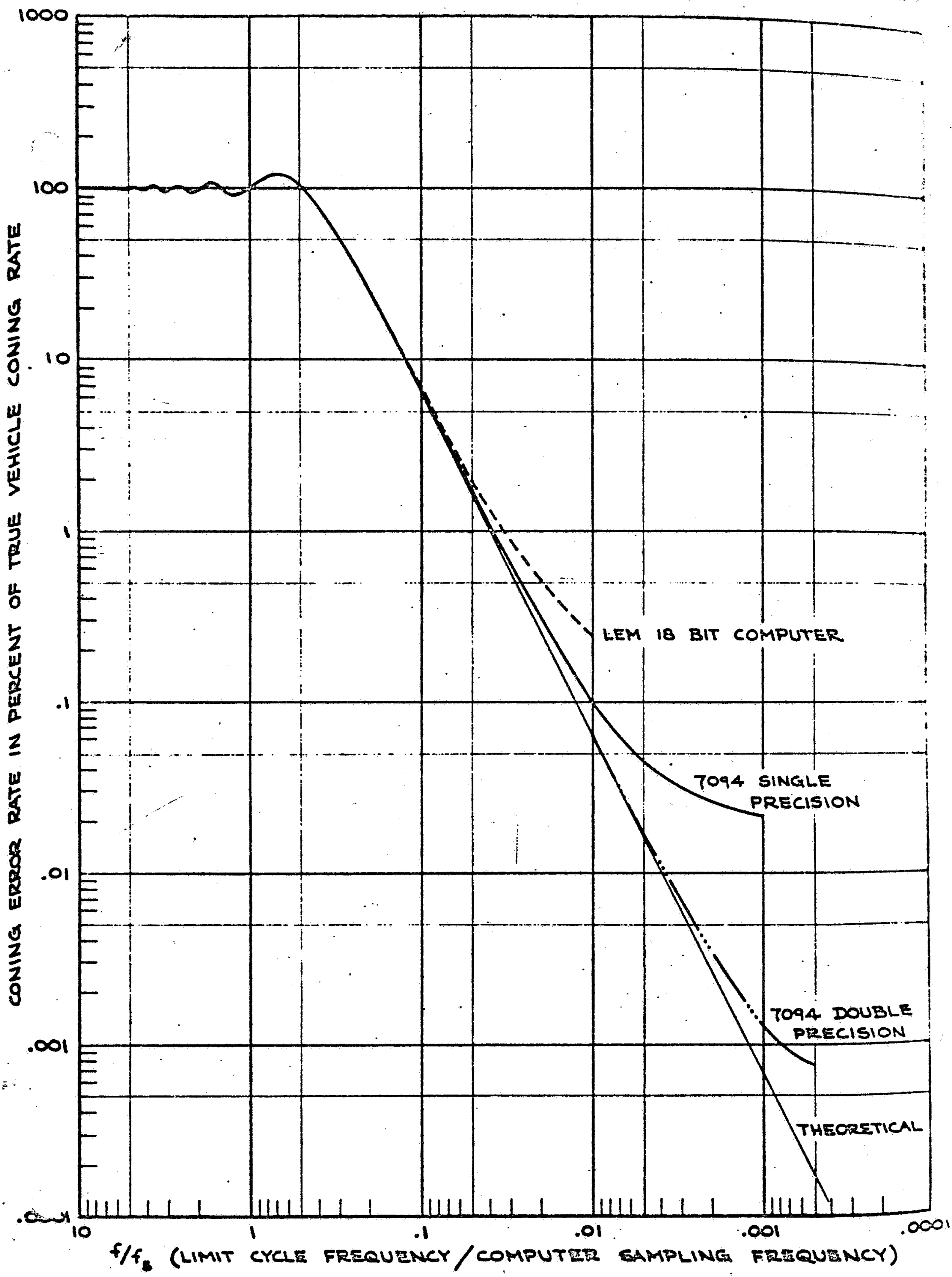


Figure II-2. Coning Study Results for the Second-order Taylor Series Algorithm

T
combi
two si
vehicl
the ph
the ve
again
differ
direct
path c
This c

T
zero a
does c
exactl
a coni

C
Figure
the cc
coning
gain
fictit

T
which
vehicl
and ca
less t
vibrat
the hi
errors

The different frequency tests were performed for several different amplitude combinations, and different starting phase shifts. These tests showed that when two sinusoidal limit cycles of different frequency are applied about two of the vehicle's axes, the vehicle will cone in one direction about the third axis until the phasing between the two sinosoids reaches zero or 180 deg. At this time the vehicle will begin to cone in the opposite direction until the sinusoids are again in phase or 180 degrees out of phase. Thus, the vehicle coning motion for different frequency limit cycles is simply a rotation about the third axis in one direction, and then in the opposite direction. One coning trajectory, i.e., the path of the tip of the vehicle about a reference vertical axis, is shown in Figure II-3 This coning motion returns to the original coning motion after $\left|\frac{1}{f_1 - f_2}\right|$ seconds.

The coning drift errors for different frequency limit cycles cancel out or are zero after an integral number of $\left|\frac{1}{f_1 - f_2}\right|$ seconds. The direction cosine matrix does drift in one direction about the vehicle's third axis, but this drift is exactly canceled when the direction is reversed. Thus, one method of eliminating a coning drift is to use a vehicle which is slightly non-symmetric.

One possible application for the results of the coning studies presented in Figure II-2, besides providing an easy calculation of coning drift errors, is to analyze the coning errors which may occur during vibrations due to fictitious coning. The coning percent error curve shown in Figure II-2 is converted for this purpose to a gain versus normalized frequency curve (Figure II-4). The gain is the ratio of the fictitious coning output error to the fictitious coning input.

This graph shows that the direction cosines are not affected by vehicle vibrations which occur at an integral number times one-half the sampling frequency. Also, the vehicle vibrations higher than one-half the sampling frequency are greatly attenuated and cause a smaller effect than the vehicle vibrations which occur at frequencies less than one-half the sampling frequency. Due to this attenuation, the vehicle vibrations need only be analyzed at low frequencies (0 to 2 f_s, for example) as the high frequency vibrations should have a negligible effect on vehicle attitude errors.

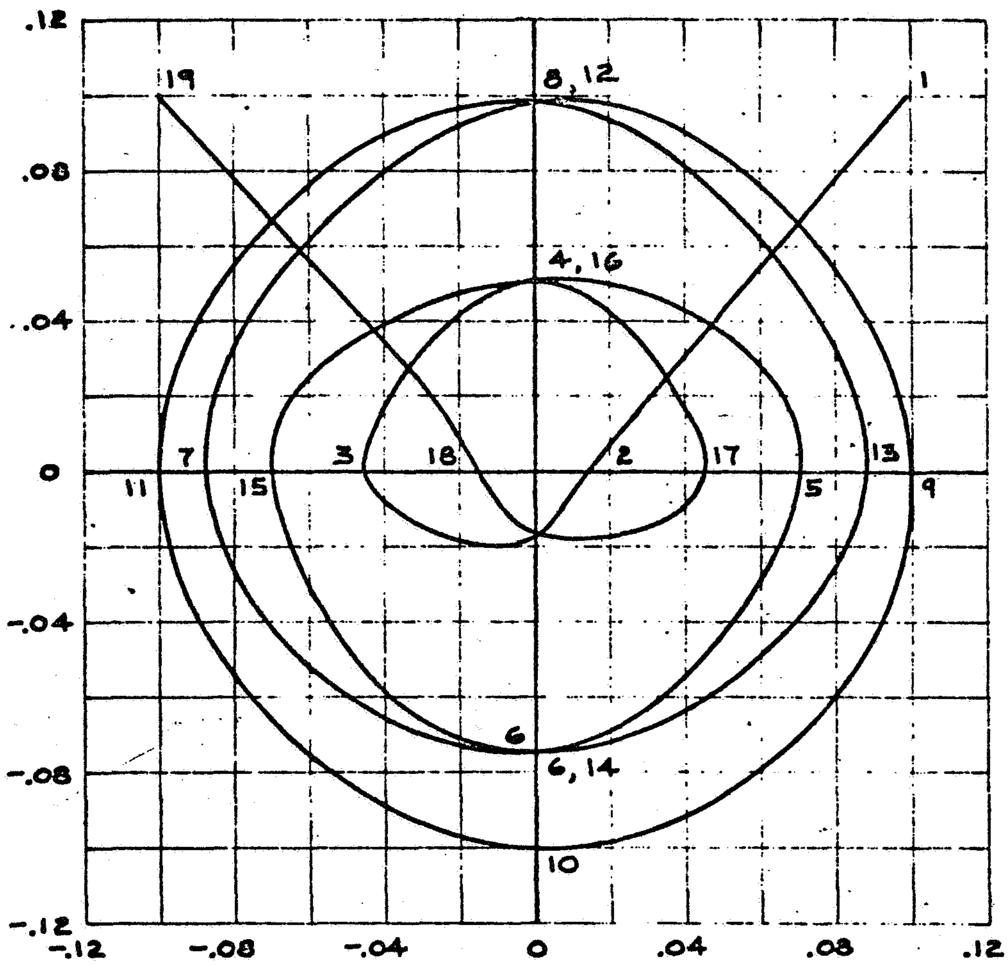


Figure II-3.

Coning Motion for 1 CPS and 0.9 CPS Limit Cycles (Vehicle Follows the Trajectory Shown from Point 1 to Point 19, and from 19 back to 1) (etc.)

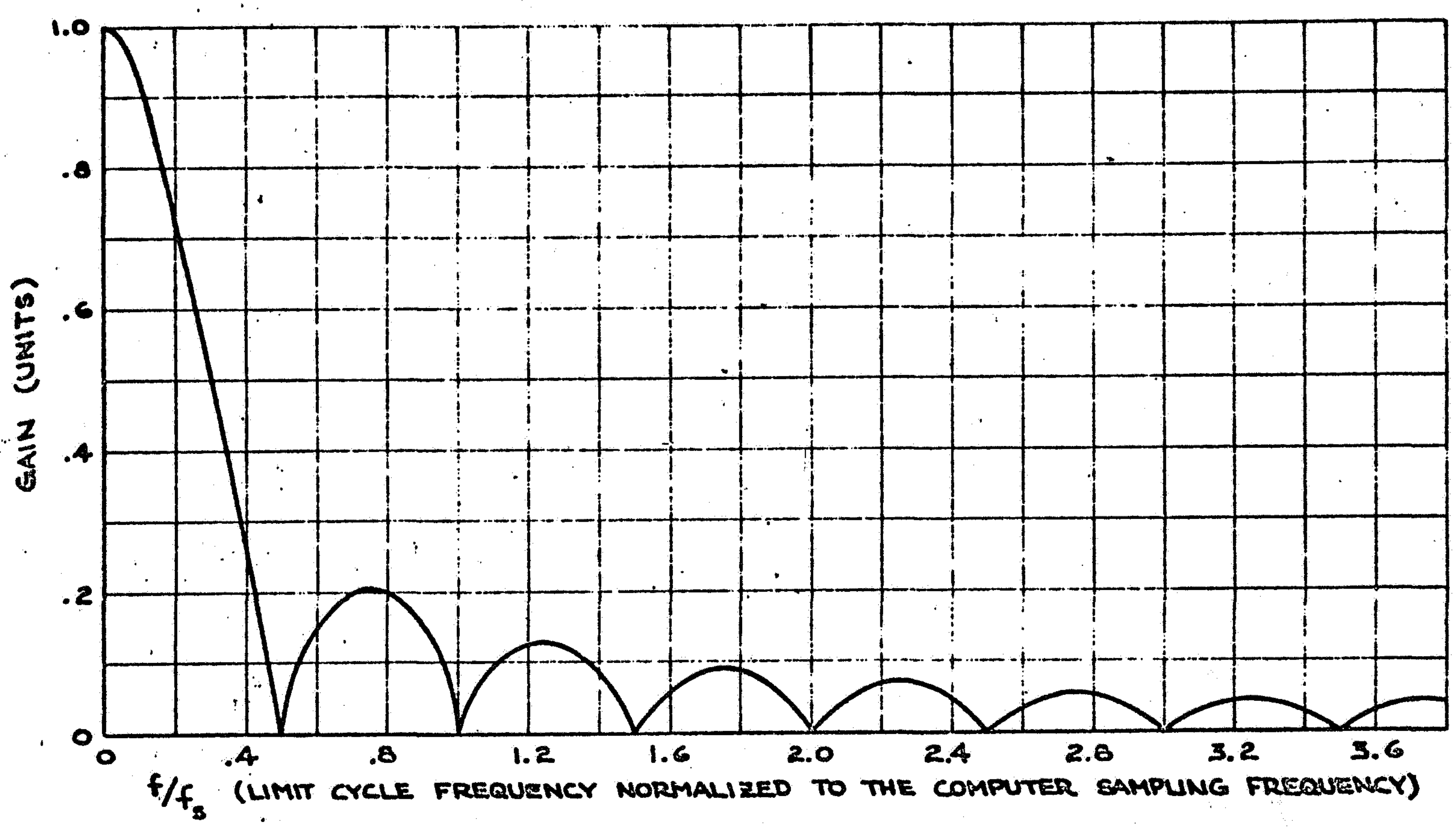


Figure II-4.

Second-Order Taylor Series Gain Versus
Normalized Frequency