AD-A041 711 AIR FORCE AVIONICS LAB WRIGHT-PATTERSON AFB OHIO F/G 17/7
CONVERSION OF COMPUTER SOFTWARE FOR THE GIMBALLED ELECTROSTATIC--ETC(U)
FEB 77 W MIKULSKI, W E SHEPHARD
UNCLASSIFIED AFAL-TR-77-8-VOL-1 NL

1 OF 2
AD
A041711

AFAL-TR-77-8
Volume I

Vol 2 - A041 677

CONVERSION OF COMPUTER SOFTWARE FOR THE GIMBALLED ELECTROSTATIC GYRO NAVIGATION SYSTEM

Volume I

REFERENCE SYSTEMS BRANCH
RECONNAISSANCE AND WEAPON DELIVERY DIVISION

FEBRUARY 1977

TECHNICAL REPORT AFAL-TR-77-8, Volume I
FINAL REPORT FOR PERIOD MAY 1973 - DECEMBER 1975

AIR FORCE AVIONICS LABORATORY
AIR FORCE WRIGHT AERONAUTICAL LABORATORIES
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433

NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

This report has been reviewed by the Information Office (IO) and is releasable to the National Technical Information Service (NTIS). AT NTIS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.

William Mikulski
WILLIAM MIKULSKI
Project Engineer

William E. Shephard
WILLIAM E. SHEPHARD
Project Engineer

FOR THE COMMANDER

Charles L Hudson
CHARLES L. HUDSON, Colonel, USAF
Chief
Reconnaissance and Weapon Delivery Division

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

AIR FORCE - 8 JUN 77 - 65

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER
(14) AFAL-TR-77-8, Vol 1

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
(6) CONVERSION OF COMPUTER SOFTWARE FOR THE GIMBALLED ELECTROSTATIC GYRO NAVIGATION SYSTEM, Volume I.

5. TYPE OF REPORT & PERIOD COVERED
(9) Final Report, May 1973 – December 1975

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)
(10) William/Mikulski
William E./Shephard

8. CONTRACT OR GRANT NUMBER(s)

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Air Force Avionics Laboratory (AFAL/RWA-3)
Air Force Wright Aeronautical Laboratories
AFSC, Wright-Patterson AFB, Ohio 45433

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
(16) Project 19270202
(17) 02

11. CONTROLLING OFFICE NAME AND ADDRESS
Same as block 9 above

12. REPORT DATE
(11) February 1977

13. NUMBER OF PAGES
(12) 127 p.

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)

15. SECURITY CLASS. (of this report)
UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)
Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)
Aircraft navigation
Inertial navigation system
Computer program
Computer softwear conversion

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The Gimballed Electrostatic Gyro Navigation System (GEANS) conversion effort consisted of the conversion of an assembly language program for the Honeywell HDC-601 computer to another assembly language program for the Singer/Kearfott SKC-2000 computer. The HDC-601 and SKC-2000 were run in real time simultaneously. The SKC-2000 real time executive automatically ynchronized with the HDC-601 so both programs ran in parallel, using the same input data from the Inertial Measurement Unit (IUM). Alignment and Navigation output

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

021670

20. ABSTRACT (Cont'd)

of both programs could then be compared and the SKC-2000 output verified. The conversion was completed successfully, the HDC-601 and SKC-2000 outputs agreeing to about 0.015 nautical miles per hour.

AFAL-TR-77-8
Volume I

FOREWORD

This report was prepared by William Mikulski and William E. Shephard of the Reference Systems Branch, Reconnaissance and Weapon Delivery Division, Air Force Avionics Laboratory, Wright-Patterson Air Force Base, Ohio.

The work was initiated under Project Work Unit Number 19270202. The report covers effort during the period May 1973 to December 1975. AFAL-TR-77-8, Volume II contains the SKC-2000 computer listing.

PRECEDING PAGE BLANK - NOT FILMED

AFAL-TR-77-8
Volume I

CONTENTS

ILLUSTRATIONS

TABLE

SECTION I

INTRODUCTION

This report describes the conversion of the computer software for a precision inertial navigation from its first developed computer code to a program for a more sophisticated and powerful machine. The inertial navigation system was the Gimballed Electrostatic Gyro Navigation System (GEANS) developed under AFAL contract by Honeywell Inc. The GEANS equipment consists of an Inertial Measurement Unit (IMU) and an electronics unit. In addition a digital computer and associated control and display unit are required.

The GEANS is unusual both because of its high accuracy (0.1 nm/hour) and because of its space stabilized mode of operation. These two features resulted in the development of unique computer software to meet the accuracy requirements and to provide for the special problems of the alignment of a space stabilized gyro and accelerometer assembly.

The purpose of the GEANS Software Conversion Project was to evaluate the machine dependency and flexibility of application of the GEANS software. This was accomplished by implementing the alignment and navigation algorithms developed originally for the Honeywell HDC-601 digital computer on the Singer-Kearfott SKC-2000. Since the SKC-2000 computer is logically similar to the SKC-2070 which is used on the B-1 aircraft, the project results will be useful in the event that GEANS is used as part of the B-1 avionics equipment.

The software which was converted was that used for Optimized GEANS and is not identical with the more recently developed SPN/GEANS software, although differences are minor.

Since it was desired to compare the navigation outputs of the original and converted programs for identical inputs, the SKC-2000 computer was arranged to operate in parallel with the HDC-601 computer. It was therefore not necessary to convert the sequencing and control portions of the HDC-601 program since the normal operation under HDC-601 control was maintained.

The project was accomplished at the Air Force Avionics Laboratory (AFAL/RWA-3) by AFAL personnel using the facilities of the Reconnaissance and Weapon Delivery Division Software Evaluation Laboratory.

SECTION II

GEANS COMPUTATIONAL REQUIREMENTS

The computational requirements of the Gimballed Electrostatic Gyro Aircraft Navigation System (GEANS) differ significantly from those of a conventional local vertical Inertial Navigator. These differences result from both the very high accuracy required and the unique method of operation as a space stabilized system.

The GEANS operates in the space stabilized mode both physically and computationally. Physical space stabilized operation means that, after initial rough platform erection, the sensor assembly containing two two-degree-of-freedom gyros and three single axis accelerometers is not rotated with respect to inertial space and thus the accelerometers are not continuously maintained in a fixed orientation with respect to the earth. Computation is also in a space stabilized coordinate frame. This means that acceleration, velocity, and position are computed in a coordinate frame which does not rotate with the earth and which is irrotational in inertial space. Subsequent, subsidiary computations based on inertial space position and elapsed time since alignment provide position and velocity in conventional earth referenced coordinates, i.e., latitude, longitude, altitude, north velocity, east velocity, and vertical velocity.

The space stabilized computational frame is fixed in inertial space at the time of transition from the alignment mode to the navigation mode. The origin of this orthogonal frame is on the

Earth's Polar Axis (EPA) with the x axis through the latitude and longitude of alignment, the z axis parallel with the EPA and y perpendicular to x and z.

The space orientation of the inertial components is fixed, except for gyro drift, at the time of initial platform erection. The approximate orientation of the accelerometers, at erection time, is z parallel with the EPA, y perpendicular to the local vertical and x perpendicular to both.

The basic computational requirements of GEANS, in addition to platform sequencing, initial erection, control, and Built-In-Test, are those required for alignment, navigation, and attitude. All functions are computed using a 32 Hertz program interrupt as the time synchronization signal. The basic computation cycle rate is 8 Hertz, allowing the computation to be distributed over 4 subcycles per major cycle.

Alignment, since the operation is space stabilized, consists of estimating the coordinate transformation matrix which relates the inertial component directions to the desired coordinate frame at time of transition from the alignment mode to the navigation mode. To account for earth's motion an earth model is used to provide ideal accelerometer outputs. These outputs are differenced with the actual, smoothed, accelerometer outputs to produce residual values.

These residual values are then used to compute an improved estimate of the transformation matrix by a minimum least squares technique.

Successive applications of this process are required to achieve an alignment accuracy commensurate with the quality of the inertial components and the specified performance of the GEANS. To insure that algorithm errors do not degrade the performance of which the GEANS equipment is capable, the algorithm accuracy requirement was established during GEANS development to be 1 second of arc. This limit was then used during analysis to establish the sophistication and complexity of the various algorithms required. It is not possible, however, to prove that this accuracy is actually attained.

Navigation computations are straight-forward although unusual because of the space stabilized coordinate frame. The special computational requirements stem primarily from the necessity of providing a very accurate gravity model and from the necessity of maintaining position preciseness of 200 ft in a total position magnitude approximating the radius of the earth (2×10^7 ft).

The basic numerical integration time interval used in the original HDC-601 computer program and maintained in the SKC-2000 conversion is 1/8 second. Rectangular integration is used. The computational load is in four 1/32-second segments using a 32-Hertz interrupt for timing synchronization. Central processor loading in the HDC-601 is approximately 70%.

SECTION III

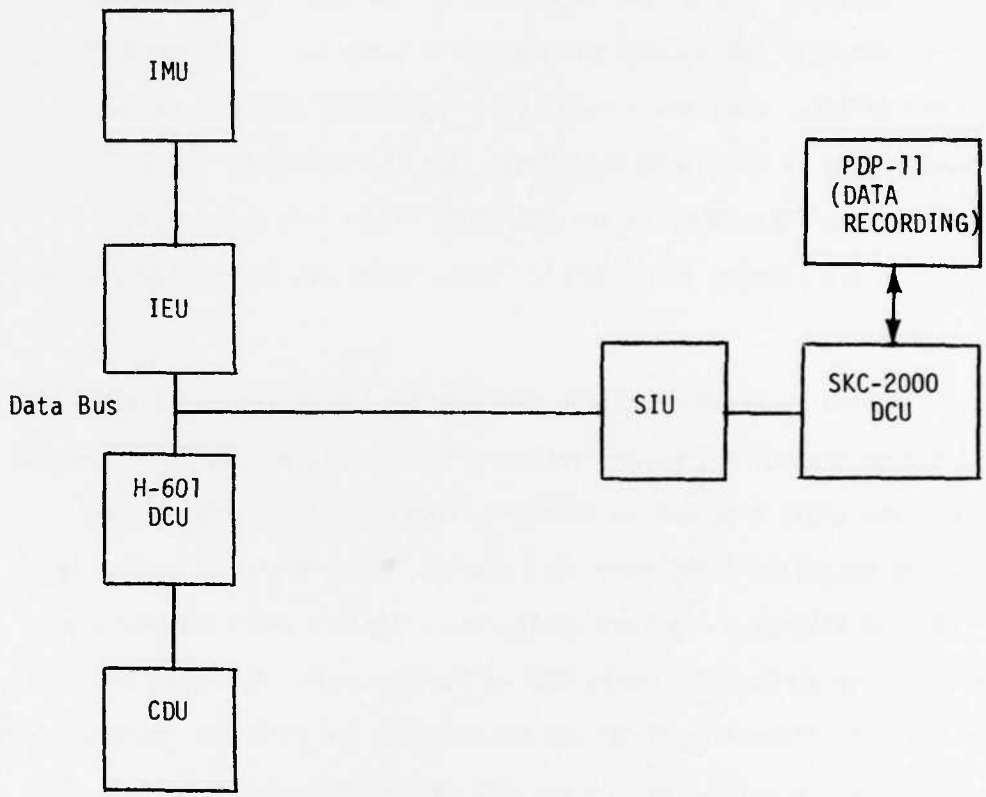
SYSTEM OPERATION AND TEST APPROACH

The basic approach to the software conversion and test was to provide for parallel operation of the SKC-2000 containing the converted program while allowing the GEANS to operate normally under the HDC-601 computer control. This approach was adopted because it allowed direct comparison of outputs based on inputs known to be identical and avoided risk of GEANS hardware damage by using the well-tested HDC-601 sequencing and Built-In-Test program functions.

Once synchronized, time recording within each computer allowed complete correlation of output data. During alignment this data allowed comparison of the results of each successive alignment matrix computation. During navigation position and velocity errors could be compared (typically at 6-minute-real-time intervals).

The arrangement shown by Figure 1 then allowed the GEANS equipment to operate normally through start-up and platform erection and into the ground alignment and navigation modes. The start-up and erection modes were not programmed on the SKC-2000. Instead, arrangements were made to synchronize the two programs (HDC-601 and SKC-2000) at the start of alignment utilizing a common 32 Hertz interrupt.

The SKC-2000 runs in synchronization with the HDC-601 to a tolerance of one 32HZ interrupt time interval. This is accomplished through a SKC-2000 routine called CDU. Both computers are turned on and the



IMU Inertial Measurement Unit
IEU Interface Electronics Unit
DCU Digital Computer Unit
CDU Computer Display Unit
SIU Special Interface Unit

Figure 1. Equipment Interconnection

HDC-601 controls spin up and sequencing of the GEANS IMU. The CDU routine monitors the HDC-601 program as it progresses through ALIGNMENT and NAVIGATION. When the HDC-601 goes into NAVIGATION, the operator manually puts it back into ALIGNMENT. This is sensed by CDU which then initiates ALIGNMENT in the SKC-2000. From this point both computers are running in parallel. These steps are shown graphically by Figure 2.

The data recording function provided by the PDP-11 is flexible and during debugging provided records of many internal variables versus time. The usual information recorded, however, is the same as the standard optimized GEANS hard copy output. This output is primarily successive alignment matrices during the alignment mode and position and velocity components during the navigation mode. These outputs are rich in information about the behavior of the software, particularly when the two separate computer outputs can be compared when it is known that they are derived from identical input velocity increment streams. As one example of data analysis, a separate CDC 6600 program was used to deduce and compare the gyro drift rate compensation being applied by the two programs.

An additional important capability was provided by the equipment shown by Figure 3. This capability is that of operating the SKC-2000 program from magnetic tapes containing recorded GEANS velocity increment data and HDC-601 output results. The velocity increment data is in

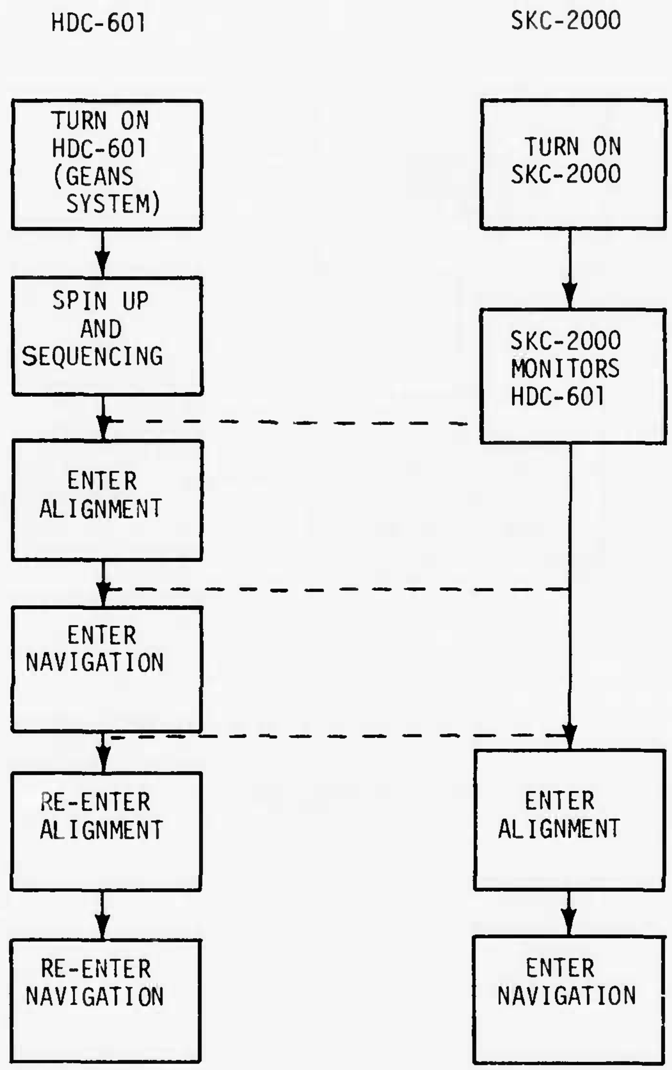


Figure 2. Steps to Run SKC-2000 in Parallel with HDC-601

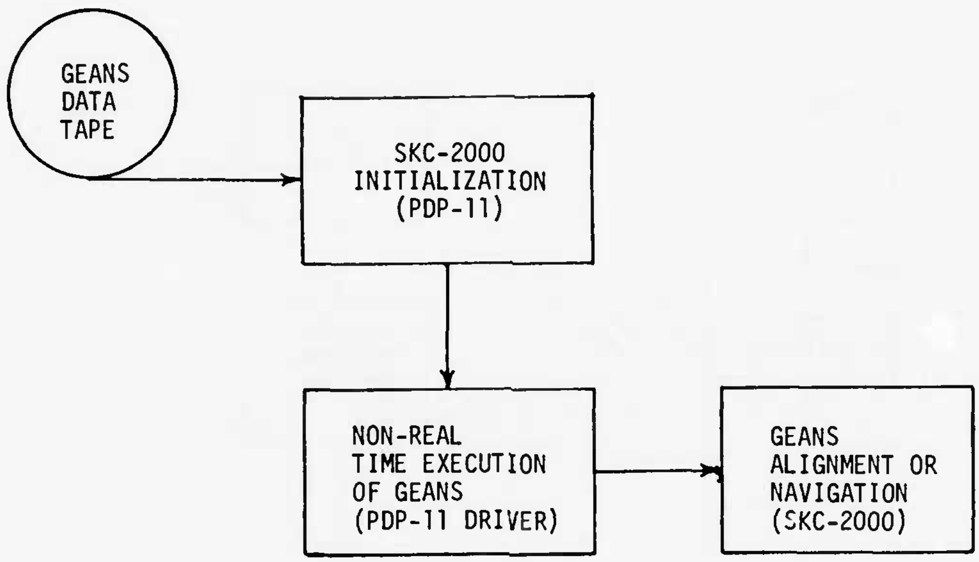


Figure 3. Steps to Run SKC-2000 from Data Tape

the form of 1/8 second sums. Operation is somewhat faster than real time. The capability was important for two reasons. Runs could be repeated, and the tape recordings could be used as input for a CDC-6600 FORTRAN program which duplicated the SKC-2000 program. The ability to run the same program with identical inputs was of great value during debugging.

SECTION IV

PROGRAM CONVERSION

GENERAL

There are several starting points and approaches which may be used to convert a reasonably complex computer program for use on another computer. Because limited resources and documentation were available for the HDC-601 SKC-2000 conversion many potential problems were avoided by adopting the following two ground rules:

1. All basic cycles, cycle times, and sampling rates were retained unchanged.

2. All algorithms except mathematical subroutines were adopted unchanged at a level expressable as FORTRAN statements.

The major deviation from the GEANS HDC-601 software implementation was the use of floating point arithmetic in the SKC-2000 software. The decision to use floating point arithmetic in the GEANS SKC-2000 software implementation was based on the following considerations: (1) the use of floating point arithmetic does not require the scaling of constants and variables in the coded program, (2) the use of floating point arithmetic significantly reduced the time required to debug the coded software, (3) the execution time of the SKC-2000 floating point software does not exceed the execution time of the HDC-601 fixed point software, (4) the GEANS SKC-2000 software conversion and implementation process could be completed and verified with less manpower and in a shorter time span if floating point arithmetic was used.

Conversion Approach

Conversion of the GEANS software was accomplished by the following steps:

1. Navigation and Alignment flow charts were developed from the HDC-601 GEANS assembly code listings. These flow charts provided an understanding of the GEANS/HDC-601 software implementation. They also were used to determine the division of functions used to satisfy timing requirements, and they proved to be a useful form of documentation of the HDC-601/GEANS software.

2. These Navigation and Alignment flow charts were studied for an understanding of GEANS and they were modified to include corrections for such things as HDC-601 code being interpreted improperly or changes made by Honeywell during their optimization program.

3. The Navigation and Alignment programs were then coded in FOCAP assembly language for the SKC-2000 computer, using the flow charts as reference. Each routine in these programs was coded separately, then run on the SKC-2000 to eliminate bugs caused by coding errors. This was the preliminary debugging phase. After this debugging phase a flow chart of the routine was drawn to reflect its implementation on the SKC-2000.

4. When the complete Alignment and Navigation programs had been coded an extensive number of timing checks were done on each routine to determine execution time. These checks were done under worst case

conditions; that is, the maximum number of instructions were being executed within each routine. After this data had been collected a Navigation sub-executive program and an Alignment sub-executive program were written. These sub-executive programs provide the proper sequencing of the four separate branches of Alignment and Navigation. Each branch is executed at 1/8-second intervals. Thus, a knowledge of the execution time of the routines that comprise these programs was necessary to insure that each branch would execute within the given 1/8-second time interval.

5. After Alignment and Navigation had been decoded, the GEANS real time routines were coded, debugged, and flow charted in the same way.

6. While the GEANS algorithms were being converted to run on the SKC-2000 a parallel effort was under way to develop a double precision math subroutine library. All variables and constants are defined within the library - there are no external references. Thus, the subroutine library can be used in other programs as well as GEANS. It has been supplied to the COLLINS RADIO CORP. for use in the GPS program and the double precision multiply and divide routines have been supplied to SOFTECH for inclusion in version 2 of their J3B Compiler.

7. To aid in the accompanying analytical studies and to provide a debugging tool, a CDC-6600 program was generated which duplicates the functions of the SKC-2000 program at FORTRAN statement level. This program proved very useful and is available for GEANS error modeling and simulation.

Computer Differences

There are a number of differences between the HDC-601 computer and the SKC-2000 computer which were significant in performing a software conversion: (1) the word length of the HDC-601 computer is 16 bits single precision, 31 bits double precision; the word length of the SKC-2000 is 32 bits single precision and 64 bits double precision, (2) the HDC-601 has one index register; the SKC-2000 has 64 index registers, (3) the SKC-2000 has hardware floating point capability; the HDC-601 does not, and (4) the HDC-601 has approximately 80 instructions; the SKC-2000 also has approximately 80 instructions; however, these instructions are in general not compatible on a one-for-one basis. The primary characteristics of each computer are shown by Table 1.

TABLE I

COMPARISON OF SKC-2000 AND HDC-601 COMPUTERS

CHARACTERISTIC	SKC-2000	HDC-601
Word Length	32	16
Registers	64	1
Floating Point	Yes	No
No. of Instructions	Approximately 80	Approximately 80
Double Precision	31 Bits Fixed Point	Up to 56 Bits Floating Point
Multiplication Time	10μs for Single Precision 100μs for Double Precision	6μs for Single Precision; 75μs for Double Precision

Accuracy Considerations

To attain the computational accuracy of the GEANS software, approximately 90% of the navigation and alignment algorithm was coded in double precision on the HDC-601. This significantly affected the execution time of a multiplication or a division operation. For example, a fixed point single precision multiplication on the HDC-601 requires approximately 10 microseconds while a fixed point double precision multiplication requires approximately 75 microseconds. As multiplication is the single most often performed operation, this differential is a significant factor in program execution time.

The word length of the SKC-2000 is 32 bits which is double the HDC-601 bit word length. The use of floating point arithmetic on the SKC-2000 requires 9 bits to define the sign and exponent. Thus, only 23 bits are available for the data word. In contrast, the use of fixed point arithmetic on the HDC-601 requires 1 bit to define the sign and thus effectively 31 bits are available for the data word. Therefore, to retain the same bit accuracy on the SKC-2000 using floating point arithmetic as is achieved using double precision fixed point arithmetic on the HDC-601 required the use of double precision. Because of the 64 bit double precision word length of the SKC-2000, effectively 56 bits are available for the data word and an additional 24 bits of accuracy can be achieved.

As the additional 24 bits of accuracy is not required to achieve the computational accuracy of the GEANS/HDC-601 Navigation and Alignment software, a double precision multiplication and a double precision divide routine were written. The multiplication routine accomplishes its task through multiplication by parts and produces at least 46 bits of accuracy. The divide routine uses recursive division and produces at least 42 bits of accuracy. Both routines require approximately 100 microseconds to execute. Thus, both the accuracy and timing requirements of GEANS were met.

SECTION V

INTERFACE HARDWARE (SPECIAL ELECTRONICS ASSEMBLY)

The acquisition of IMU output data from the GEANS data bus required provision of a Direct Memory Access (DMA) Serial Interface for the SKC-2000 computer compatible with the GEANS System Duplex Serial Data Bus.

After consultation with Singer-Kearfott Division of Singer, Inc., manufacturers of the SKC-2000, the selected approach was to install a Parallel DMA Input/Output Channel (PIOC) (similar to that used on the B1 type (SKC-2070) into the SKC-2000 Input/Output Unit at the Singer-Kearfott Plant. The Parallel DMA Channel then became the driver for a Serial Conversion Board (SCB) designed, constructed, and installed into the SKC-2000 I/O Unit by AFAL/RWA. The SCB does the data format conversions between the GEANS Serial Format, a format acceptable by the ports of the PIOC which does the DMA transfers with SKC-2000 core memory.

INTERFACE OPERATION

The Parallel Output Channel (POC), when issued a software command, accesses a control word specifying an external device address, word count, and a data block start address. The first data word is then accessed and transferred to the SCB in two 16-bit halves. The SCB reassembles the word and transmit it in serial format. During the serial transmission the POC is accessing the next data word for transfer to the SCB. This continues until the data block word-count is exhausted and the POC generates an End-of-Transfer Status Bit and/or program interrupt.

The Parallel Input Channel (PIC), when issued a software command, accepts external input requests from the SCB. After the SCB has received a serial data word and converted it into parallel format, it generates an external input request to the PIC. The PIC then accesses its control word and accepts the input data in two 16-bit halves. The SCB can then receive the next serial input while the PIC is storing its word in core memory. After the PIC control word word-count is exhausted, the PIC generates an End-of-Transfer status Bit and/or program interrupt.

An additional feature of the SCB is that one of the output external device address discretes allows the POC to generate a program controlled reset to the SCB. After either a system master reset or program controlled SCB reset, the SCB input channel ignores all external activity until after the SCB output channel has serially transmitted one word which is simultaneously routed into the SCB input channel to act as a frame marker and interface self-test word. The SCB also buffers the GEANS 32-Hertz Clock into an SKC-2000 program interrupt input to act as software and input/output synchronizer for the SKC-2000.

The particular construction selected for the GEANS Interface allowed the input/output software driver to be modular and entirely interrupt driven which reduced the complexity of the SKC-2000 GEANS Executive Software. The actual interrupt driver routines totaled approximately 160 words of core program instructions.

Further details of this interface are supplied in design documentation available from the Air Force Avionics Laboratory, AFAL/RWA-4, Wright-Patterson AFB, Ohio.

SECTION VI

MATHEMATICAL MODEL

The software conversion was not a direct instruction by instruction translation because of fundamental differences between the machines. In addition, there was no provision for continuous Honeywell analytical support. For these reasons, it was necessary to develop a thorough understanding of the theory of operation of the GEANS alignment and navigation programs, and to provide a basic mathematical model for use at the AFAL.

This was accomplished by a review of all available technical information and the rederivation of some basic equations, in particular the classic least squares fine alignment approach. An extract of the basic mathematical background information was prepared and can be found in Appendix A.

SECTION VII

CDC 6600 SIMULATION

An aid which was decided upon very early in the task effort and which was found to be extremely useful during all phases was a CDC 6600 program written in FORTRAN-IV. This program does not simulate the decoding of the time division multiplex data from the GEANS Inertial Measuring Unit, but instead operates from 1/8 second sums of velocity increments from a magnetic tape. All of the alignment and navigation calculations are duplicated at statement level.

No attempt was made to change word length or other machine characteristics to resemble the SKC 2000 or HDC-601. The full word length of the CDC 6600 was used,as were the CDC 6600 system mathematical functions. The CDC 6600 number processing was therefore generally more precise than the SKC-2000. This difference was itself a useful attribute because it allowed immediate evaluation of whether or not an error appearing in the SKC-2000 program was due to accumulated computational errors.

The computational flow charts of the CDC 6600 and SKC 2000 are identical for those program segments which are duplicated by the CDC-6600 program.

Since the CDC 6600 programming task was much less complex than that for the SKC-2000, the FORTRAN program preceded the SKC-2000 program during all of the software development. The availability of this program made possible the discovery of many errors early in the programming task and reduced the final SKC-2000 debugging task to primarily that of isolating and removing errors in the input decoding process.

SECTION VIII

TEST RESULTS

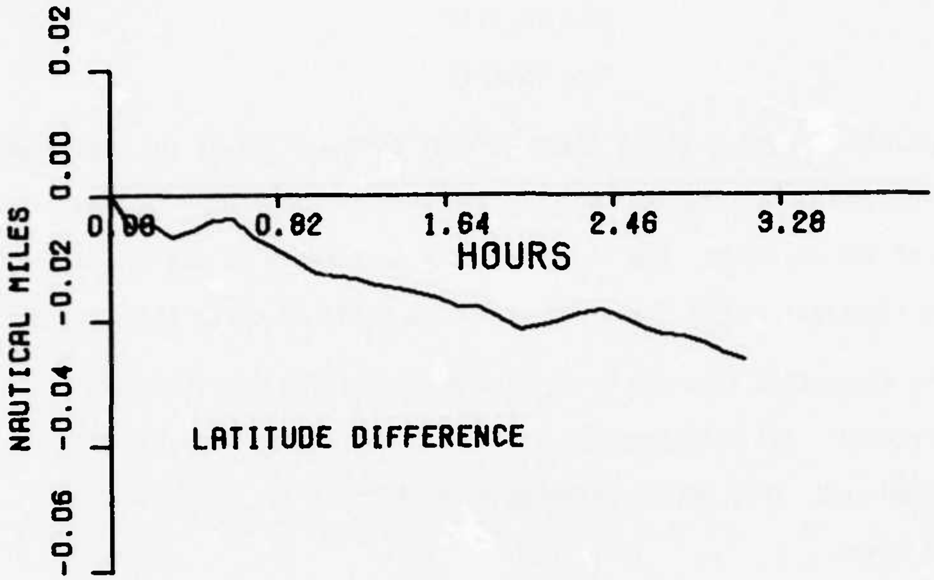
Comparative testing of the GEANS HDC-601 software versus the converted SKC-2000 software was conducted by synchronizing the two programs after erection of the platform. The input velocity increments to the two programs were thus identical except for a possible time shift of up to 1/32 second.

After transition into the Navigation Mode position output versus time was recorded for each computer and the difference between the two outputs computed. This output difference is shown by the graphs on the following pages.

As can be seen from Figures 4 through 8, the difference between the two outputs was generally 0.015 nautical miles per hour or about one tenth the system accuracy specification. This performance met the conversion project objectives.

The form of the plotted curves suggests that the difference between the two outputs is caused by an effective gyro drift rate difference. Since the acceleration inputs are from the same platform, such a difference would have to be due to a difference in the gyro drift rate compensation computations.

The computations are accomplished by subroutine DC. A rough analysis showed that the two drift compensation calculations can differ enough to account for the apparent drift rate difference but the analysis was not pursued further since the comparative performance met project objectives.



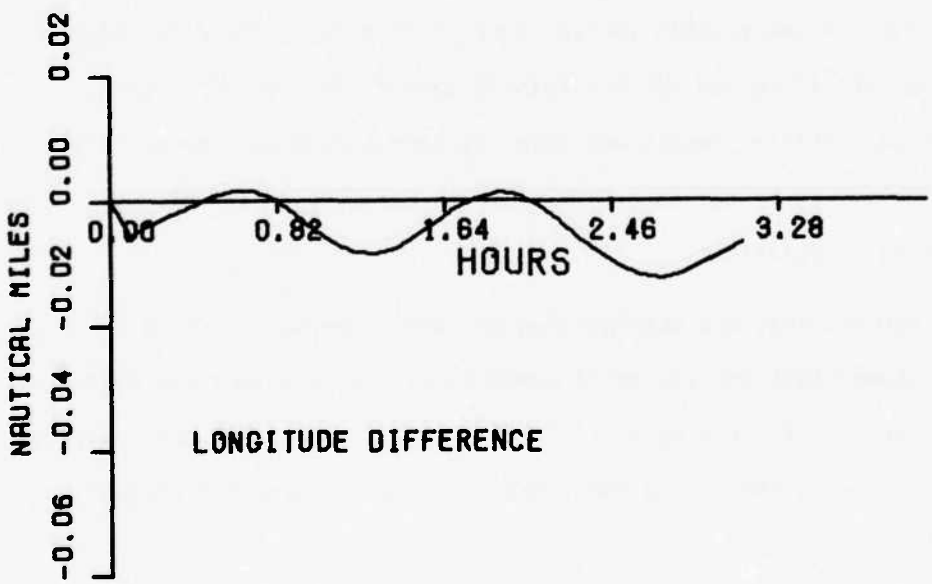
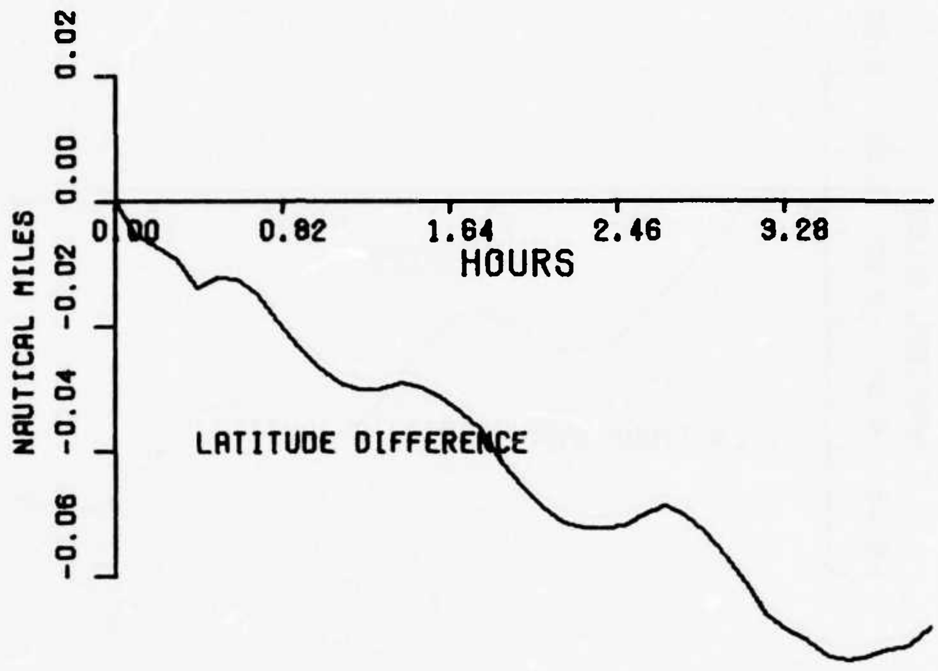


Figure 4. Run Number 09171



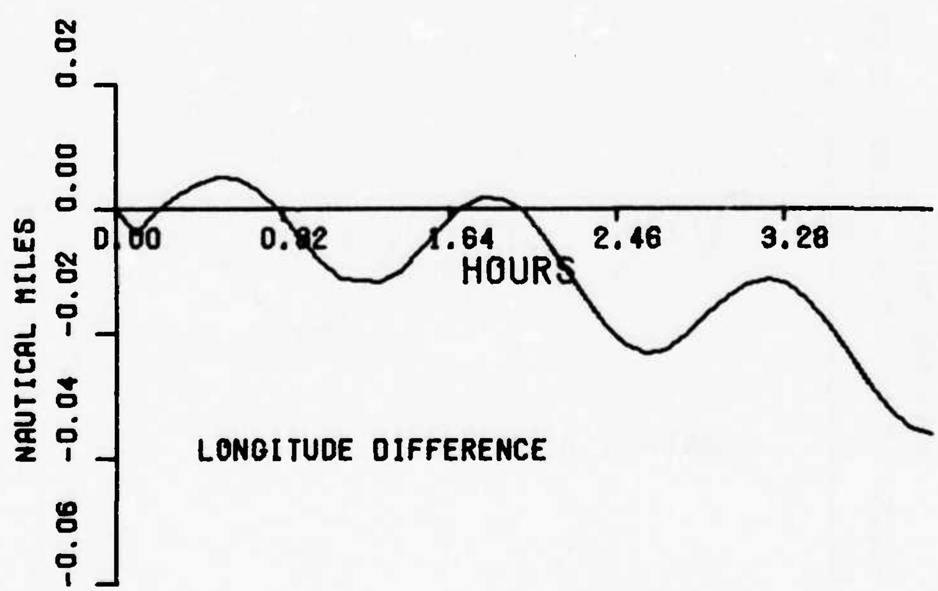
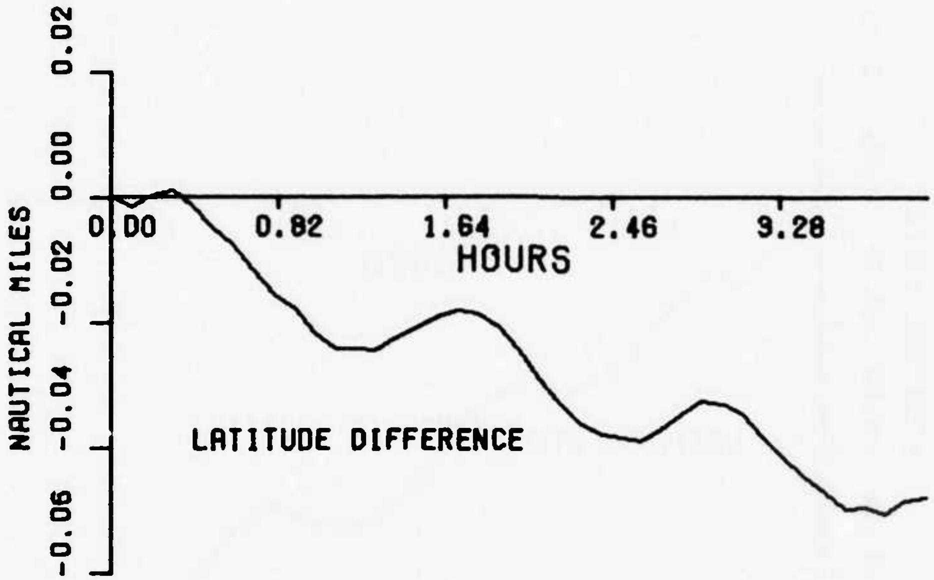


Figure 5. Run Number 09181



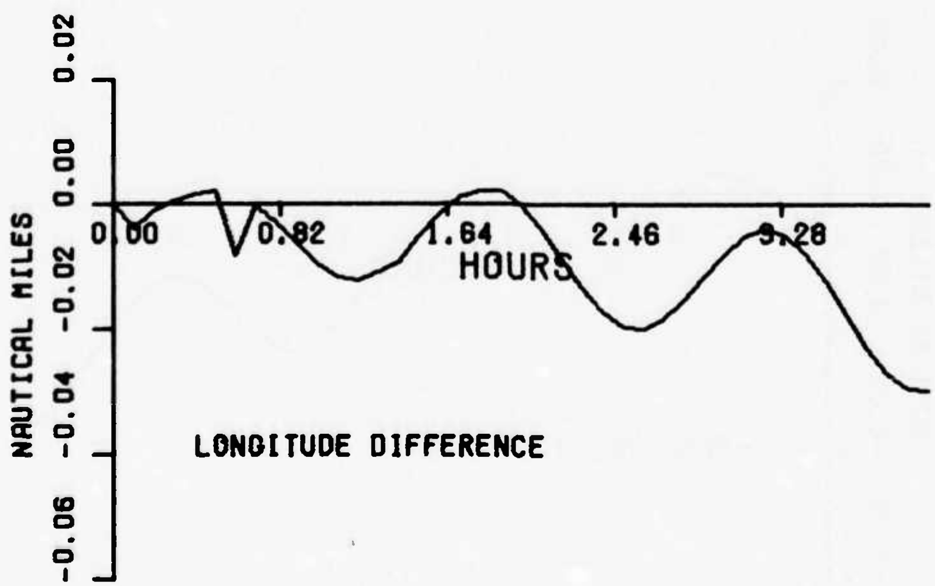


Figure 6. Run Number 09191

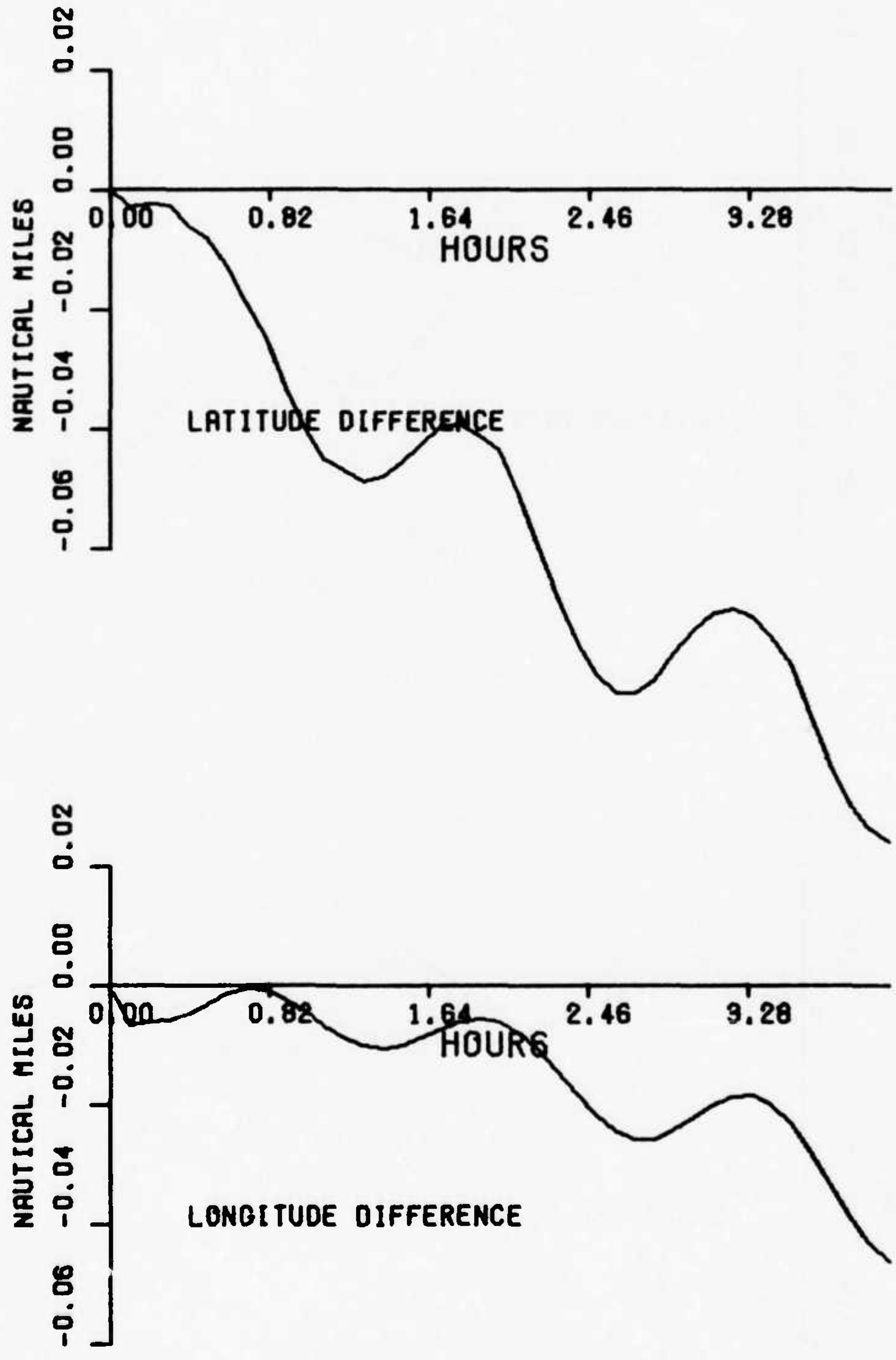
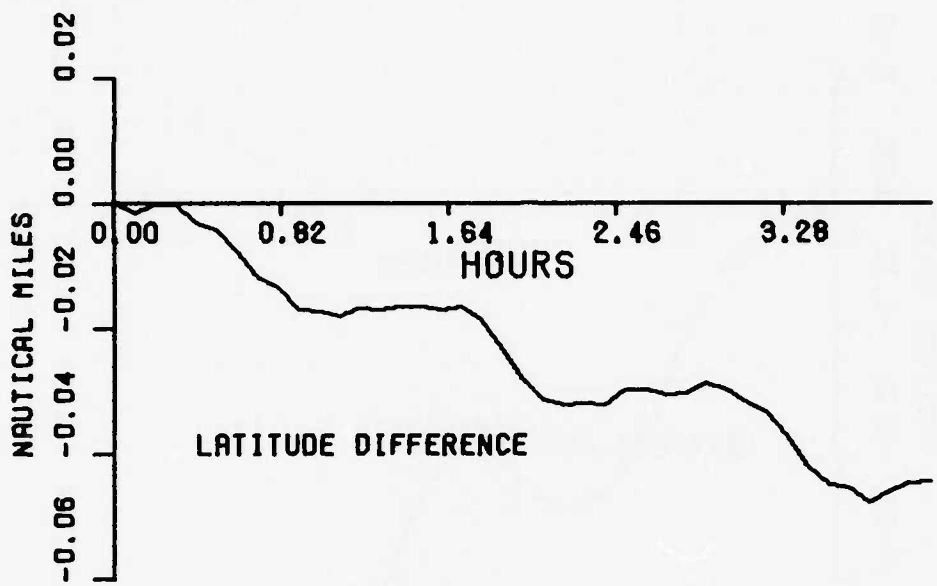


Figure 7. Run Number 09241



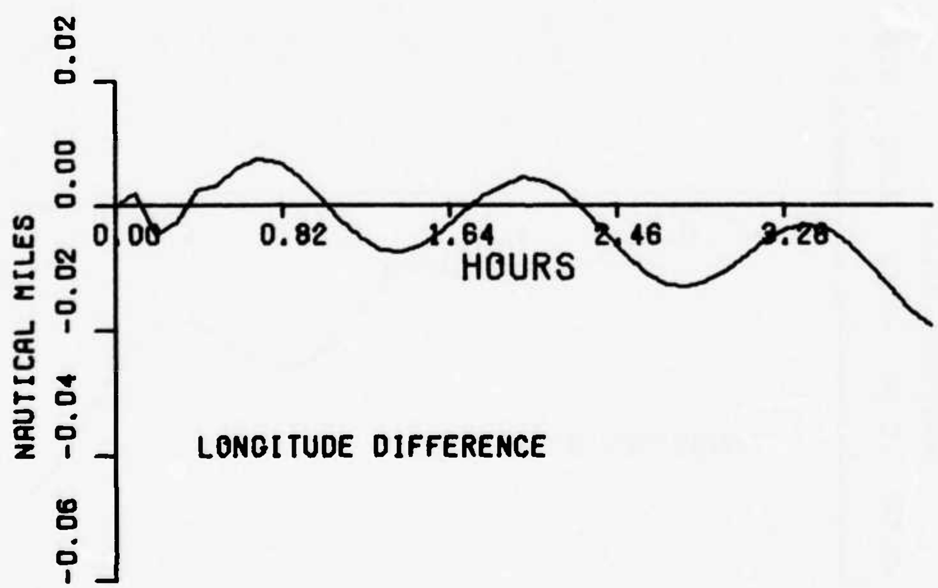


Figure 8. Run Number 11111

SECTION IX

DISCUSSION

The GEANS software conversion effort consisted of the conversion of an assembly language program for the Honeywell HDC-601 computer to another assembly language program for the Singer/Kearfott SKC-2000 computer. Since the architecture of the two machines is quite different in many respects, a one for one instruction translation was not possible. The more powerful features of the SKC-2000 such as floating point arithmetic and multiple index registers were taken advantage of during the conversion process.

The GEANS program was also written in FORTRAN for the AFAL CDC-6600 General Purpose computer to obtain a thorough understanding of the Alignment and Navigation algorithms involved and to provide comparative data for checkout. Real Inertial Measurement Unit data supplied by Honeywell on magnetic tape was used as input for both the CDC-6600 version and SKC-2000 version of the alignment and navigation programs. This provided a high degree of confidence in the CDC-6600 program because of the repeatability of the input data. It also provided a proven tool to debug the SKC-2000 program, since the output of the SKC-2000 program could be compared to the output of the CDC-6600 program. Thus a great deal of debugging was done in non-real time.

The HDC-601 and SKC-2000 were run in real time simultaneously. The SKC-2000 real time executive automatically synchronized with the HDC-601 so both programs ran in parallel, using the same input data from the IMU. Alignment and Navigation output of both programs could then be compared, and the SKC-2000 output verified.

The conversion was completed successfully, the HDC-601 and SKC-2000 outputs agreeing to about 0.015 nautical miles per hour.

The program was shown to have a small degree of machine and algorithm dependency but that this dependency causes an apparent gyro drift rate change of less than one-tenth of that allowable by specification and has negligible effect on system accuracy.

The program conversion reported herein was from assembly language to assembly language. Although the use of a Higher Order Language (HOL) would not affect accuracy performance significantly, time loading and memory requirements might change. It is therefore recommended that portions of the SKC-2000 program be converted to the HOL J3B and the time and memory requirements of the resulting assembly language program be compared with those of the corresponding program segments coded directly in assembly language.

APPENDIX A

GEANS MATHEMATICAL BACKGROUND

The material in this appendix is a short summary of the mathematical derivations contained in the program documentation generated by Honeywell Incorporated under USAF contract and has been included in this report for convenience. For a more complete treatment the Honeywell report* should be consulted.

COORDINATE FRAMES

The coordinate frames shown in Figure 9 were selected for GEANS.

The origins of all coordinate frames are located at the center of the earth and are portrayed as shown for convenience only.

(X_I, Y_I, Z_I) is the inertial frame in which all computations are performed. The Z_I axis is parallel to the earths rotation axis and Y_I points east at the beginning of navigation.

(X_N, Y_N, Z_N) is a local level coordinate frame associated with the position of the aircraft. Y_N points east and Z_N points north at all times.

(X_E, Y_E, Z_E) is an earth fixed coordinate frame. X_E lies along the Greenwich Meridian and Z_E lies along the earths rotation axis.

(X_P, Y_P, Z_P) which is not shown is the platform coordinate axis and is aligned with the accelerometer triad.

(X_G, Y_G, Z_P) which also is not shown is the gyro coordinate axis. The relationship between the gyro coordinate axis and the platform axis is defined later.

*Russ Sittloh et al., "AN/ASN-101 Optimization Program," Interim Report AFAL-TR-2043-IT-2, Honeywell Inc., St. Petersboro, Florida, January 1973.

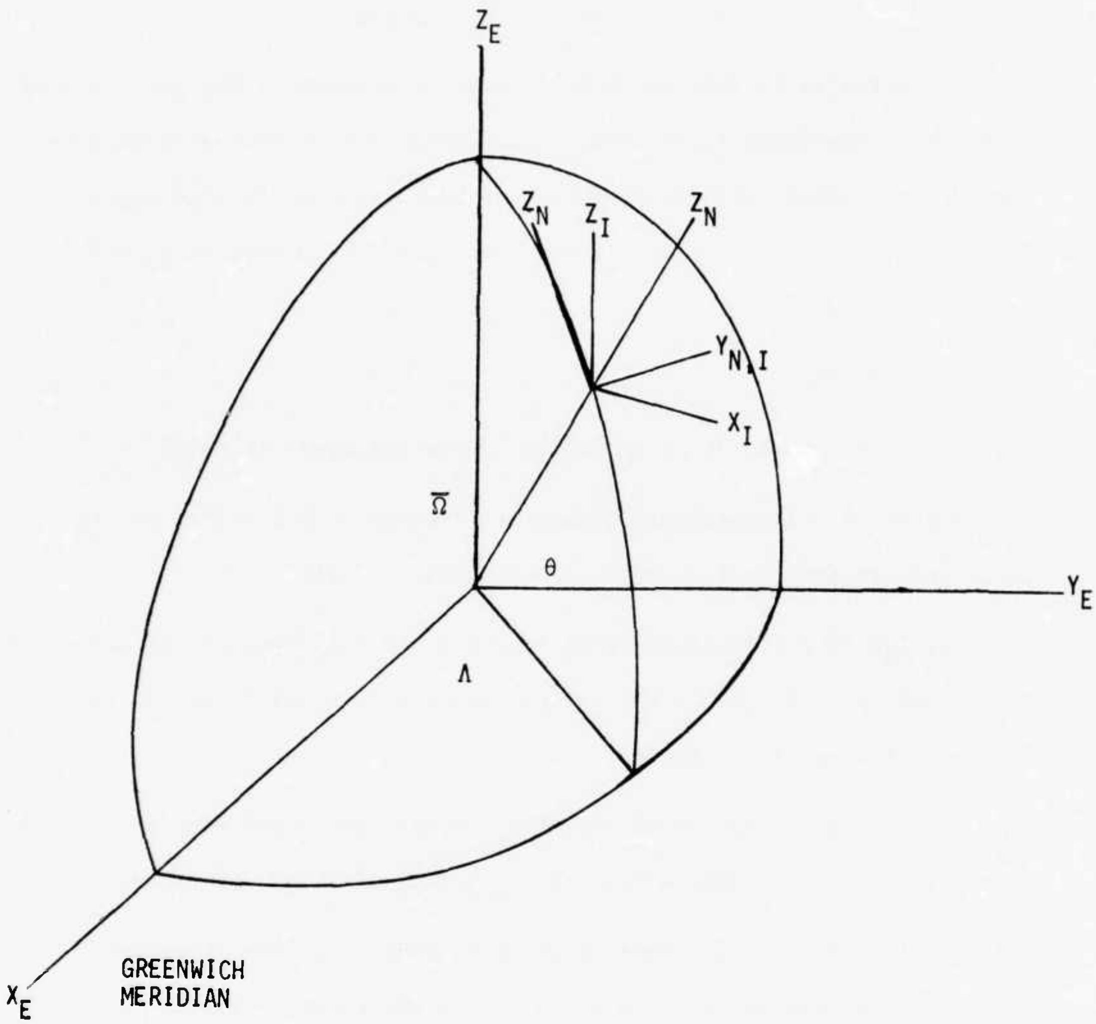


Figure 9. Coordinate Frame

COORDINATE TRANSFORMATIONS

The following coordinate transformations are readily derived from Figure 9.

$$C_E^I = \begin{bmatrix} \cos(\Lambda_0 + \Omega t) & \sin(\Lambda_0 + \Omega t) & 0 \\ -\sin(\Lambda_0 + \Omega t) & \cos(\Lambda_0 + \Omega t) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (1)$$

Λ_0 - initial longitude

Ω - earths rotation rate

$$C_N^E = \begin{bmatrix} \cos\theta \ \cos\Lambda & \cos\theta \ \sin\Lambda & \sin\theta \\ -\sin\Lambda & \cos\Lambda & 0 \\ -\sin\theta \ \cos\Lambda & -\sin\theta \ \sin\Lambda & \cos\theta \end{bmatrix} \quad (2)$$

θ - latitude

Λ - longitude

$$C_G^P = \begin{bmatrix} 0 & 0 & 1 \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 \\ -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 \end{bmatrix} \quad (3)$$

BASIC NAVIGATION EQUATIONS

The specific force equation for an accelerometer in inertial space is

$$\bar{f}_I = \bar{a}_I + \bar{G}_I \tag{4}$$

where

a_I - acceleration of the point

G_I - gravitation at the point

This equation can be written in terms of inertial velocity as:

$$\dot{V}_{XI} = a_{XI} + G_{XI} \tag{5}$$

$$\dot{V}_{YI} = a_{YI} + G_{YI} \tag{6}$$

$$\dot{V}_{ZI} = a_{ZI} + G_{ZI} \tag{7}$$

The calculation of inertial position from inertial velocity can be determined as:

$$\dot{r}_{XI} = V_{XI} \tag{8}$$

$$\dot{r}_{YI} = V_{YI} \tag{9}$$

$$\dot{r}_{ZI} = V_{ZI} \tag{10}$$

As written above, these mechanization equations are unstable. They can be stabilized by use of an altimeter as follows:

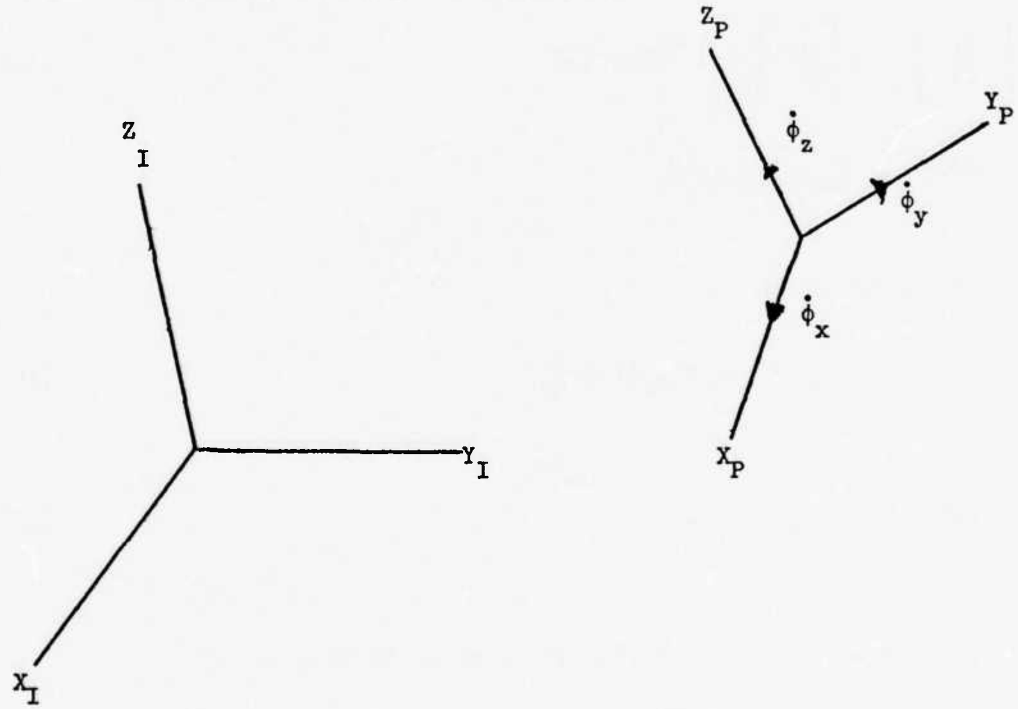
$$\dot{r}_{XI} = V_{XI} - K(R_{ec} + h - R_I)r_{XI} \tag{11}$$

$$\dot{r}_{YI} = V_{YI} - K(R_{ec} + h - R_I)r_{YI} \tag{12}$$

$$\dot{r}_{ZI} = V_{ZI} - K(R_{ec} + h - R_I)r_{ZI} \tag{13}$$

To obtain stability the altimeter must also be used in the computation of the gravity components in Equations 5-7.

PLATFORM TO COMPUTATION FRAME TRANSFORMATION



By definition, the computation frame is non-rotating with respect to inertial space. It is assumed in the sequel that the rotation of the platform coordinate can be described by small angular rotations.

An arbitrary vector $\bar{r}$ can be expressed in the two frames as:

$$\bar{r}_I = C_I^P \bar{r}_P \tag{14}$$

and the derivative as

$$\dot{\bar{r}}_I = \dot{C}_I^P \bar{r}_P + C_I^P \dot{\bar{r}}_P \tag{15}$$

Utilizing different notation the derivative can also be expressed as:

$$\left[\frac{d\bar{r}}{dt}\right]_I = \left[\frac{d\bar{r}}{dt}\right]_P - \bar{\Omega}(P,I) x \bar{r} \tag{16}$$

$$C_P^I \dot{\bar{r}}_I = \dot{\bar{r}}_P + \underline{\Omega}_P(P,I)\bar{r}_P \tag{17}$$

Equating equations 15 and 17

$$C_P^I \dot{C}_I^P \bar{r}_P + \dot{\bar{r}}_P = \dot{\bar{r}}_P + \underline{\Omega}_P(P,I)\bar{r}_P \tag{18}$$

or

$$\dot{C}_I^P \bar{r}_P = C_I^P \underline{\Omega}_P(P,I)\bar{r}_P \tag{19}$$

Since $\bar{r}$ is arbitrary, then a solution to Equation 19 is:

$$\dot{C}_I^P = C_I^P \underline{\Omega}_P(P,I) \tag{20}$$

where

$$\underline{\Omega}_P(P,I) = \begin{bmatrix} 0 & \dot{\phi}_z & -\dot{\phi}_y \\ -\dot{\phi}_z & 0 & \dot{\phi}_x \\ \dot{\phi}_y & -\dot{\phi}_x & 0 \end{bmatrix} \tag{21}$$

ACCELEROMETER ERROR MODEL

The accelerometer error model used in GEANS is:

$$\begin{bmatrix} a_{xp} \\ a_{yp} \\ a_{zp} \end{bmatrix} = \begin{bmatrix} k_1 & \beta_1 & -\alpha_1 \\ -\beta_2 & k_2 & \alpha_2 \\ \beta_3 & -\alpha_3 & k_3 \end{bmatrix} \begin{bmatrix} a_{xa} - b_x \\ a_{ya} - b_y \\ a_{za} - b_y \end{bmatrix} \tag{22}$$

where a_{xa}, a_{ya}, a_{za} are the measured accelerations.

GYRO ERROR MODEL

The gyro error model used in GEANS is:

$$\begin{bmatrix} \dot{\phi}_x \\ \dot{\phi}_y \\ \dot{\phi}_z \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{2}}{2}\frac{W_o}{HW_1} & \frac{\sqrt{2}}{2}\frac{W_o}{HW_2} & 0 \\ \frac{\sqrt{2}}{2}\frac{W_o}{HW_1} & -\frac{\sqrt{2}}{2}\frac{W_o}{HW_2} & 0 \\ 0 & 0 & -\frac{W_o}{HW_1} \end{bmatrix} \left\{ \begin{bmatrix} \gamma_{10} - \Delta\gamma_{10} \\ \gamma'_{10} - \Delta\gamma'_{10} + \gamma'_{18}\dot{R}^{+} - \gamma'_{19}\dot{R}^{-} \\ \gamma_{20} - \Delta\gamma_{20} \end{bmatrix} \right.$$

$$
+ \begin{bmatrix} \gamma_{11} & \gamma_{12} & \gamma_{13} \\ -\gamma'_{11} & -\gamma'_{13} & -\gamma'_{12} \\ \gamma_{21} & \gamma_{22} & \gamma_{23} \end{bmatrix} \begin{bmatrix} a_{xg} \\ a_{yg} \\ a_{zg} \end{bmatrix} + \begin{bmatrix} f_1 \alpha_{10} \\ f_2 \alpha'_{10} + (\alpha'_{18} R^+ - \alpha'_{19} R^-) f_2 \\ f_1 \alpha_{20} \end{bmatrix}
$$

$$
\left. + \begin{bmatrix} f_1 \alpha_{11} & f_1 \alpha_{12} & f_1 \alpha_{13} \\ -f_2 \alpha'_{11} & -f_2 \alpha'_{13} & -f_2 \alpha'_{12} \\ f_1 \alpha_{21} & f_1 \alpha_{22} & f_1 \alpha_{23} \end{bmatrix} \cdot \begin{bmatrix} a_{xg} \\ a_{yg} \\ a_{zg} \end{bmatrix} \right\} \tag{23}
$$

where

$$
\begin{bmatrix} a_{xg} \\ a_{yg} \\ a_{zg} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 \\ -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 \end{bmatrix} \begin{bmatrix} a_{xp} \\ a_{yp} \\ a_{zp} \end{bmatrix} \tag{24}
$$

GRAVITY MODEL

The gravity model used in the GEANS Navigation Software is:

$$
G_{XI} = \left\{ A_2 {*} [A_5 - 6 \sin^2\theta + 9 \sin^4\theta] + A_0 P^3 - A_1 P^5 (A_3 - \sin^2\theta) \right\} r_{XI} \tag{25}
$$

$$
G_{YI} = \left\{ A_2 {*} [A_5 - 6 \sin^2\theta + 9 \sin^4\theta] + A_0 P^3 - A_1 P^5 (A_3 - \sin^2\theta) \right\} r_{YI} \tag{26}
$$

$$
G_{ZI} = \left\{ A_2 {*} [A_6 + A_5 - 10 \sin^2\theta + 9 \sin^4\theta] + A_0 P^3 - A_1 P^5 (A_3 + A_4 - \sin^2\theta) \right\} r_{ZI} \tag{27}
$$

where

$$P = 1 - \frac{\Delta R}{R} + \left(\frac{\Delta R}{R}\right)^2 \tag{28}$$

$$\Delta R = h - [21385 \sin^2\theta\ (1. + 0.00503 \cos^2\theta\ )] \tag{29}$$

$$A_0 = -1.536227137E-6 \tag{30}$$

$$A_1 = 1.247433376E-8 \tag{31}$$

$$A_2 = -1.593661308E-11 \tag{32}$$

$$A_3 = 0.2 \tag{33}$$

$$A_4 = 0.4 \tag{34}$$

$$A_5 = 0.4285714286 \tag{35}$$

$$A_6 = 1.714285714 \tag{36}$$

LATITUDE, LONGITUDE CALCULATION

From Figure 9, the computation of latitude and longitude can be readily derived as:

$$\theta = \theta_0 + \tan^{-1}\left[\frac{Z_I}{(X_I^2 + Y_I^2)^{1/2}}\right] \tag{37}$$

$$\Lambda = \Lambda_0 - \Omega t + \tan^{-1}\left[\frac{Y_I}{X_I}\right] \tag{38}$$

LOCAL-LEVEL VELOCITY COMPUTATION

The relationship of the radius vector in the computational and earth fixed coordinates frames can be expressed as:

$$\bar{r}_I = C_I^E \bar{r}_E \tag{39}$$

and the time derivations as:

$$\dot{\bar{r}}_I = \dot{C}_I^E \bar{r}_E + C_I^E \dot{\bar{r}}_E \tag{40}$$

$$\dot{\bar{r}}_E = C_E^I \dot{\bar{r}}_I - C_E^I \dot{C}_I^E \bar{r}_E \tag{41}$$

$$C_N^E \dot{\bar{r}}_E = C_N^I \left[\dot{\bar{r}}_I - \dot{C}_I^E C_E^I \bar{r}_I\right] \tag{42}$$

The expression $C_N^E \dot{\bar{r}}_E$ is the velocity with respect to the earth expressed in local-level coordinates and can be written as:

$$C_N^E \dot{\bar{r}}_E = \begin{bmatrix} V_V \\ V_E \\ V_N \end{bmatrix} \tag{43}$$

INTEGRATION

Euler integration is used in the GEANS Mechnaization. This can be expressed as:

$$\bar{X}(t + \Delta t) = \bar{X}(t) + f(\bar{x}(t), t)\ \Delta t \tag{44}$$

for the differential equation

$$\dot{\bar{X}}(t) = f(\bar{X}(t), t) \tag{45}$$

The accelerometers used in the GEANS platform are really velocity meters and their output is a change in velocity $\Delta\overline{V}_I$ over a specified time interval Δt. Thus the average acceleration over the time period Δt is

$$\bar{a}_I = \frac{\Delta\overline{V}_I}{\Delta t} \tag{46}$$

Thus, the mechanization equations 5-7, and 11-13 can be written as:

$$r_{XI}(t+\Delta t) = r_{XI}(t)+V_{XI}(t)\Delta t + K(R_{EC}+h(t) - R_I)\, r_{XI}(t) \tag{47}$$

$$r_{YI}(t+\Delta t) = r_{YI}(t)+V_{YI}(t)\Delta t + K(R_{EC}+h(t) - R_I)\, r_{YI}(t) \tag{48}$$

$$r_{ZI}(t+\Delta t) = r_{ZI}(t)+V_{ZI}(t)\Delta t + K(R_{EC}+h(t) - R_I)\, r_{ZI}(t) \tag{49}$$

$$V_{XI}(t+\Delta t) = V_{XI}(t) + \Delta V_{XI} + G_{XI}\Delta t \tag{50}$$

$$V_{YI}(t+\Delta t) = V_{YI}(t) + \Delta V_{YI} + G_{YI}\Delta t \tag{51}$$

$$V_{ZI}(t+\Delta t) = V_{ZI}(t) + \Delta V_{ZI} + G_{ZI}\Delta t \tag{52}$$

The accelerometer error model as:

$$\begin{bmatrix} \Delta V_{xp} \\ \Delta V_{yp} \\ \Delta V_{zp} \end{bmatrix} = \begin{bmatrix} K_1 & \beta_1 & -\alpha_1 \\ -\beta_2 & K_2 & \alpha_2 \\ \beta_3 & -\alpha_3 & K_3 \end{bmatrix} \begin{bmatrix} \Delta V_{xa}-b_x\Delta t \\ \Delta V_{ya}-b_y\Delta t \\ \Delta V_{za}-b_z\Delta t \end{bmatrix} \tag{53}$$

The gyro error model as:

$$\begin{bmatrix} \phi_x(t+\Delta t) \\ \phi_y(t+\Delta t) \\ \phi_z(t+\Delta t) \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{2}}{2}\frac{W_o}{HW_1} & \frac{\sqrt{2}}{2}\frac{W_o}{HW_2} & 0 \\ \frac{\sqrt{2}}{2}\frac{W_o}{HW_1} & -\frac{\sqrt{2}}{2}\frac{W_o}{HW_2} & 0 \\ 0 & 0 & -\frac{W_o}{HW_1} \end{bmatrix} \begin{bmatrix} \gamma_{10}-\Delta\gamma_{10} \\ \gamma_{10}'-\Delta\gamma_{10}'+\gamma_{18}\dot{R}^{+}-\gamma_{19}\dot{R}^{-} \\ \gamma_{20}-\Delta\gamma_{20} \end{bmatrix} \Delta t .$$

$$+\begin{bmatrix} \gamma_{11} & \gamma_{12} & \gamma_{13} \\ -\gamma'_{11} & -\gamma'_{13} & -\gamma'_{12} \\ \gamma_{21} & \gamma_{22} & \gamma_{23} \end{bmatrix} \begin{bmatrix} \Delta V_{xg} \\ \Delta V_{yg} \\ \Delta V_{zg} \end{bmatrix} + \begin{bmatrix} f_1\alpha_{10} \\ f_2\alpha'_{10} + (\alpha'_{18}R^+ - \alpha'_{19}R^-) f_R \\ f_1\alpha_{20} \end{bmatrix} \Delta t$$

$$+\begin{bmatrix} f_1\alpha_{11} & f_1\alpha_{12} & f_1\alpha_{13} \\ -f_2\alpha'_{11} & -f_2\alpha'_{13} & -f_2\alpha'_{12} \\ f_1\alpha_{21} & f_1\alpha_{22} & f_1\alpha_{23} \end{bmatrix} \begin{bmatrix} \Delta V_{xg} \\ \Delta V_{yg} \\ \Delta V_{zg} \end{bmatrix} \tag{54}$$

where

$$\begin{bmatrix} \Delta V_{xg} \\ \Delta V_{yg} \\ \Delta V_{zg} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 \\ -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 \end{bmatrix} \tag{55}$$

Note that since C_I^P is continually updated $\phi_x(t)$, $\phi_y(t)$, $\phi_z(t) = 0$.

The platform to computation frame transformation is computed as:

$$C_I^P (t+\Delta t) = C_I^P (t) + C_I^P (t)\, \underline{\Omega}_p (P, I)\, \Delta t \tag{56}$$

$$C_I^P (t+\Delta t) = C_I^P (t) + C_I^P (t)\underline{A}(t) \tag{57}$$

where

$$\underline{A} = \begin{bmatrix} 0 & \phi_z & -\phi_y \\ \phi_z & 0 & \phi_x \\ \phi_y & -\phi_x & 0 \end{bmatrix} \tag{58}$$

APPENDIX B

PROGRAM ORGANIZATION AND FLOW CHARTS

SKC-2000 PROGRAM ORGANIZATION

The organization of the converted program for use by the SKC-2000 computer, which follows closely that of the original HDC-601 computer program, is described below and is shown graphically by the flow charts (Pages 62 through 120) of this appendix. The software is divided into 5 decks. These decks are:

NAV - navigation, which includes all navigation subroutines plus drift computation and the navigation data base.

SUBLIB - the math subroutine package

INIT - navigation and alignment initialization routines and their data base.

ALIGN - alignment subroutines plus data base

RTEXEC - real time executive and real time routines plus data base

The data base in each deck is organized into common data blocks, a local variable area, and a local constants area. The common blocks include one universal variables common data area. This includes all variables common to all decks. There is one universal constant common area. In addition each deck has common areas for data it shares with other decks but may not share with all decks. A brief description follows:

1. WLDCOM - GEANS world variable data. Contains all variables used throughout GEANS.

2. NIACOM - NAV, INIT, and ALIGN variables. Contains those variable used in navigation, initialization and alignment.
3. NICOM - NAV and INIT variable data. Contains variables used in navigation and initialization.
4. MATCOM - world matrix and miscellaneous data. Most matrices, vectors, and some miscellaneous data resides here.
5. CONCOM - GEANS world constants. All universial constants and system calibration data are in this COMMON.
6. IACOM - Init and ALIGN variables data area. Variables common to initialization and alignment.

NAVIGATION

The SKC-2000 GEANS navigation program consists of four branches, each branch being executed at 1/8-second intervals. The navigation scheduler, NSCH controls the selection of branches. This program starts out with subroutine IA, the sub-executive for navigation. This routine does the actual scheduling of branching and subroutine calls. A description of subroutines, branch by branch, follows.

Branch 1:

IC - computes accelerometer bias and scale factor. Also compensates for non-orthogonality between the accelerometer coordinate frame and the actual physical mounts.

ID - rotates acceleration vector from platform reference frame to navigation reference frame.

IE - computation of the gravity vector for the present position.

IF - computes earth raduis and delta radius. Computes vertical damping vector

IG - Double integration for velocity and distance.

Branch 2:

IH - Latitude and longitude computation

IJ - computation of velocity in local vertical co-ordinates, ground speed, and earth relative velocity. Also computes the position update matrix, V1.

Branch 3:

NAVO - navigation output routine. Outputs navigation variables to a buffer when they are accessed by the PDP-11. Output occurs at 0.1 hour intervals.

Branch 4:

IL - drift componsation computation. This routine is executed once per second. It constructs an update matrix, DCAR. Execution period is controlled by variable DCON.

IM - this routine updates the AJ matrix using update matrix DCAR. It compensates the AJ matrix for gyro drift. This occurs at one second intervals. Saves present AJ in matrix SA.

RTAL - this is the return to alignment decision routine. If the software MODE switch is $\geq$ 4 subroutine FENT is called and the system returns to Alignment. If MODE is < 4 the system remains in navigation.

ALIGNMENT

The SKC-2000 GEANS alignment program, like the navigation program, contains four branches. It is controlled by the alignment scheduler, ASCH, and starts with the alignment sub-executive IIA which does the same type of scheduling that the navigation sub-executive does. A description of the branches and subroutines follows:

Branch 1:

IIC - this is subroutine IC in navigation - computes accelerometer bias and scale factor, and accelerometer non-orthogonality compensation.

IID - this routine calculates the filtered values of the delta V's which are used in the alignment solution. Both coarse and fine filter values are calculated until NMO >7. After NMO is >7 the coarse filter values are no longer calculated.

IIE - this is subroutine ID in navigation (see above).

Branch 2:

IIF - drift compensation computation (subroutine IL in navigation). It is executed once per second as in navigation.

Branch 3:

IIG - subroutine IM in navigation. Also executed once per second, as in navigation.

IIH - This routine calculates reference delta V's RDVX, RDVY and RDVZ. It then does residual summing.

IIK - computes earth polar axis (EPA) solution

IIM - solves for local level alignment about north without attempting to correct for azimuth error.

IIO - least squares solution. The intermediate step in solving for the final alignment correction angles.

IIP - Computes the AJ matrix with corrections about X, Y, and Z as computed in each of the alignment modes, and with a rotation about Z for earth movement since the previous update.

ALNO - Navigation output routine.

Branch 4:

IIR - Go to nav decision. This routine is executed once every 1/8 second. The criteria for entering navigation are as follows:

1. Navigation mode is selected
2. Fine alignment must be complete (NMO .GE.8).

INITIALIZATION

The deck called INIT contains the following initialization routines.

FENT - First entry. This routine initial system variables for alignment and drift computation. It loads a call to the alignment sub-executive in the vector table of the executive

NAVI - Navigation initialization. Also replaces VECT align call with call to Nav.

RSET - Selects solution time and made for alignment. Alignment solution cycle counter vs mode number (NMO).

RTEXEC - Contains the following routines:

EXEC - this is a initialization routine used to clear the real time status of the SKC 2000 and reset the DMA input and output channels. Initialization is then followed by the real time executive. It is originated at 7800_{16} and is entered automatically when the SKC-2000 is reset and set to RUN.

CDUI - initializes routine CDU

BDSI - initializes routine DECODE (DECD)

CDU - synchronizes the SKC-2000 alignment program with the HDC-601.

INTIO - interrupt 10 (32 HZ interrupt) processing routine.

INT05 - interrupt 5 (DMA input complete) processing routine.

INT04 - interrupt 4 (DMA output complete) processing routine

DECD - unpacks SIDL words and stores them for use by other programs. Repacks BITE bits for use by the BITE routine. Diagnostic checks are used for verification of rotor speed and delta V's.

OUTPUT DATA FOR ALIGNMENT AND NAVIGATION

Standard

The standard output is designed for use when routine, record runs are made and provides sufficient information for evaluation of performance and some, minimum, debugging. The output is similar to that of the Honeywell 601 during alignment and navigation.

Standard - Alignment

The output during alignment is only at the end of each alignment calculation, that is each AJ update as the result of Least Squares Solutions and the proceding AJ revisions. The outputs are recorded on tape for subsequent generation of hard copy and are displayed on the CRT until the next AJ update. Drift correction of AJ is not regarded as an AJ update. The first two updates are 1 second apart, the next three are 8 seconds apart and the remainder are 150 seconds or more apart. The quantities displayed are Time, NMO, SX, SY, SZ, and the sine elements of AJ row major. The format is 4E15.8

Units are not a problem for this output form since all quantities involved are unitless.

Standard - Navigation

Upon and after beginning of navigation the quantities listed below are acquired every ΔT where ΔT is a nominal value of 360 seconds.

The data are recorded on tape for possible subsequent generation of hard copy or punched cards and is displayed on the CRT until the next observation time.

The items listed are not exactly those of the Honeywell 601.

The quantities are also not all directly available from SKC 2000 storage in the form desired. Necessary processing is indicated below, and is done by the PDP-11/40 data acquisition program RGATTY. Table 1 lists data displayed and the process by which it is derived.

MATH SUBROUTINE LIBRARY

The GEANS math subroutine library is written to handle double precision arguments. It is completely self contained. That is, all constants and variables are contained within the same deck - there are no external references. Temporary storage is handled with a push-down stack, with the address of the stack in index register 6. The stack is located in LSI memory for maximum speed of execution. The library contains the following math subroutines: SINCOS (sine and cosine), DECSQ (square root), DECATN (arc tangent), MULFD (double precision multiply), DVFD (double precision divide), and EXP (exponential). It also contains a 3 x 1 vector add a 3 x 1 vector subtract routine (VECADD & VECSUB), a 3 x 3 matrix add routine (MATDAD) a 3 x 3 single precision matrix multiply routine (MULS33), a 3 x 3 double precision matrix multiply routine (MUL33), and a (3 x 3) (3 x 1) vector multiply routine (MULD31).

All vector operations are limited to 3 x 1 vectors and all matrix operations are limited to 3 x 3 operations. Since GEANS Alignment and Navigation are limited to operations on vectors of these dimensions it was decided not to write generalized matrix handling routines.

TABLE 1

STANDARD NAVIGATION OUTPUT

ITEM	SKC 2000 FORM	OUTPUT FORM	PROCESS
TIME	SECONDS SINCE ORIGIN	MINUTES IN NAV	(TIME-TIME IN NAV)/60.0
DELTA LAT	LATITUDE IN π RADIANS	DELTA LAT IN MIN = N.M.	(LATITUDE-LOADED LAT)*π *57.295*60.0
DELTA LONG	LONGITUDE IN π RADIANS	DELTA LONG IN MIN = N.M.	(LONGITUDE-(LOADED LONG*COS (LOADED LONG))) *π*57.295*60.0
RSS	N/A	N.M.	$[(\Delta LAT)^2 + (\Delta LONG)^2]^{\frac{1}{2}}$
VV	M/S	M/S	-
VE	M/S	M/S	-
VN	M/S	M/S	-
VX	M/S	M/S	-
VY	M/S	M/S	-
VZ	M/S	M/S	-

The math subroutine library has been optimized and contains a short instruction density of 66%. This can be improved somewhat by further optimization of the code. This library can be included in any program that requires use of double precision arithmetic.

Each routine has a temporary storage area that contains as many words as the routine needs, including one word for the return address. The return address is at the top of the stack. Upon entry the contents of XR6 is decremented the number of full words necessary for temporary storage. Before the return XR6 is incremented the same number of words. All routines are re-entrant.

SINCOS - This routine computes both the sine and cosine of the argument passed to it. It also assumes the argument it receives is in π radians.

The sine function F(x) - Sin x is evaluated by the modified Taylor expansion:

$$\text{Sin}\,\frac{\pi}{2}X = \sum_{K=0}^{5} \alpha_{2K+1} X^{2K+1}, \text{ where } |x| \leq 1$$

This approximation gives a relative error less than $.5 \times 10^{-9}$, the maximum occurring near X=1. So the error in Sin $\frac{\pi}{2}$ X does not exceed $.5 \times 10^{-9}$.

For the sine function $f(y) = \text{Sin } y$ to operate it is necessary to reduce y to the interval $-\left[\frac{\pi}{2}, \frac{\pi}{2}\right]$.

Then: $\text{Sin } y \equiv \text{Sin } (\pi - y)$
$\text{Sin } y \equiv \text{Sin } (y - 2\pi)$

To do this form two parts: $n = \left[\frac{2}{\pi} y\right]$ (integer part) and $f = \left\{\frac{2}{\pi} y\right\}$ (fractional part). The two low order bits of $n, b_1\ b_o$ indicate what quadrant y is in. Let $m - b_1\ b_o$. The table below gives the quadrant as a function of m.

m (binary)	Quadrant	Y^1 = Y modulo 2π
00	1	$\left[\varepsilon \quad 0_0, \frac{\pi}{2}\right)$
01	2	$\left[\varepsilon \quad \frac{\pi}{2}, \pi\right]$
10	3	$\left[\varepsilon \quad \pi, \frac{3\pi}{2}\right]$
11	4	$\left[\varepsilon \quad \frac{3\pi}{2}, 2\pi\right]$

With the quadrant known approximate identities can be used to express Sin y as a function of a modified argument $x = g(f)$ where $x \ \varepsilon \ [0,1]$

if m is	then set x	using the identity
00	f	$\text{Sin } y' \equiv \text{Sin } y'$
01	1-f	$\text{Sin } y' \equiv \text{Sin } (\pi - y')$
10	-f	$\text{Sin } y' \equiv \text{Sin } (-(y' - \pi))$
11	f-1	$\text{Sin } y' \equiv \text{Sin } (-(2\pi - y'))$

For any y, $y^1 = y \text{ modulo } 2\pi = m\frac{\pi}{2} + \frac{\pi}{2}$ the constants α_{2k+1} are as follows:

α_1	1.57079673
α_2	- .64596486
α_3	.079692215
α_4	- .0046761082
α_5	.00015209895

DECSQ - returns the square root of the argument passed to it.

A floating point number can always be written in the form

$$X = \begin{cases} m.2^{2k} & \text{if the exponent is even} \\ m.2^{2k}.2 & \text{if the exponent is odd} \end{cases}$$

where $.5 \leq |m| \leq 1$ and $k = 0, \pm 1, \pm 2, \ldots$

Thus the square root is

$$1) \quad \sqrt{X} = \begin{cases} 2^k \cdot \sqrt{m} & \text{if the exponent is even} \\ 2^k \cdot \sqrt{m} \cdot \sqrt{2} & \text{if the exponent is odd} \end{cases}$$

To calculate the square root Heron's iterative process is used. An initial approximation, y_o, to $\sqrt{m}$ is made. Then the recursive values are computed.

$$Y_{i+1} = \frac{1}{2}\left\{Y_i + \frac{m}{y_i}\right\}, \; (i = 0, 1, 2, \ldots) \text{ until}$$

Y_n is such that the discrepency is invisible to the machine:

$$| Y_n - \sqrt{m} | \leq \varepsilon \leq 2^{-31}$$

The initial approximation for $\sqrt{m}$ is:

.5826924 M + .41730760 on the interval [.5, 1]

Once y_o is obtained three iterations are required to obtain $\sqrt{m}$. then $\sqrt{x}$ is calculated using equation 1.

DECATN - returns TAN^{-1} (Y/X) in π radians.

Let U = Y/X. The algorith used to evaluate $\tan^{-1} X$ (arctan X) is only valid for X ε (0, 1), so it is necessary to reduce X to that interval. This is done by using the identities:

$$\tan^{-1}(-X) = -\tan^{-1}(X) \qquad \text{when } X < 0$$

$$\tan^{-1}(0) = 0 \qquad \text{when } X = 0$$

$$\tan^{-1}(X) = \tan^{-1}(X) \qquad \text{when } 0 \leq X \leq 1$$

$$\tan^{-1}(X) = \frac{\pi}{2} - \tan^{-1}\left(\frac{1}{X}\right) \qquad \text{when } X > 1$$

Thus $\tan^{-1} X$ can be expressed in terms of a $\tan^{-1} Y$ where

$$Y = g(X) = X \text{ or } \frac{1}{X}$$

Now if $0 < Y \leq 1$ the following algorithm is used:

$$\tan^{-1} Y = \tan^{-1} Z + C$$

where:

$$\begin{cases} \text{if } X < 2 - \sqrt{3} \text{ then } Z = X \text{ and } C = 0 \\ \text{if } X \geq 2 - \sqrt{3} \text{ then } Z = \dfrac{X\sqrt{3} - 1}{X + \sqrt{3}} \text{ and } C = \dfrac{\pi}{6} \end{cases}$$

$$\text{and} \quad \tan^{-1} = \sum_{k=0}^{4} \alpha_{2k+1} Z^{2k+1}$$

The constants α_{2k+1} are as follows:

α_1 = .9999999984

α_2 -.3333328936

α_3 .1999653478

α_4 -.1417346061

α_5 .0949195495

MULFD - returns a double precision ($\geq$ 46 bits of accuracy) product of two input arguments.

Algorithm is multiplication by parts:

$$P = X * Y$$

where

$$X = X_A + X_B$$

$$Y = Y_A + Y_B$$

$$P_{AB} = X_A * Y_B$$

$$P_{BA} = X_B * Y_A$$

$$P_{AA} = X_A * Y_A$$

$$P' = P_{AB} + P_{BA} + P_{AA} \quad \text{where } P' = P + \varepsilon$$

$$\text{and} \quad \varepsilon \leq P * 2^{-46} \quad \text{approximately}$$

DVFD - returns a double precision ($\geq$42 bits of accuracy) quotient of two input arguments.

Algorithm is recursive division

$$Q = \frac{d}{D}$$

$$Q_1 = D_A \quad d$$

$$\varepsilon_1 = d - Q_1 D$$

$$Q_2 = D_A \quad \varepsilon_1$$

$$Q^1 = Q_1 + Q_2 \quad \text{where } Q^1 = Q + \varepsilon$$

$$\text{and} \quad \varepsilon \leq Q * 2^{-42}$$

EXP - this routine is part of this library although it is not utilized in the GEANS program.

To calculate the exponential function

$$f(x) = e^x$$

Cody and Ralston's algorithm is used. This is as follows:

Set $X \varepsilon (- \infty, \infty)$

Now

$$e^x = 2^{\log_2 e^x} = 2^{x \quad \log_2 e}$$

Set $n = [X \log_2 e]$ (integer part) and

$r = \{ X \log_2 e\}$ (fractional part). Then

$$e^x = 2^n\, 2^r, \quad |r| < 1$$

Now

$$2^r = e^{\ln 2^r} = e^{r \cdot \ln 2} = \left(e^{1/2 r \cdot \ln 2} \right)^2$$

Define $Y = r\frac{\ln 2}{2}$. Then $Y \in \left[-\frac{\ln 2}{2}, \frac{\ln 2}{2} \right]$

On this interval the approximate to e^y

$$\left(e^y\right) = 1 + \cfrac{2Y}{A_o - y - \cfrac{A_1}{B_1 + Y^2}}$$

is nearly the best (in Chebyschev's sense) which has a relative error of less than 10^{-9}.

where:

A_o = 12.015016753875

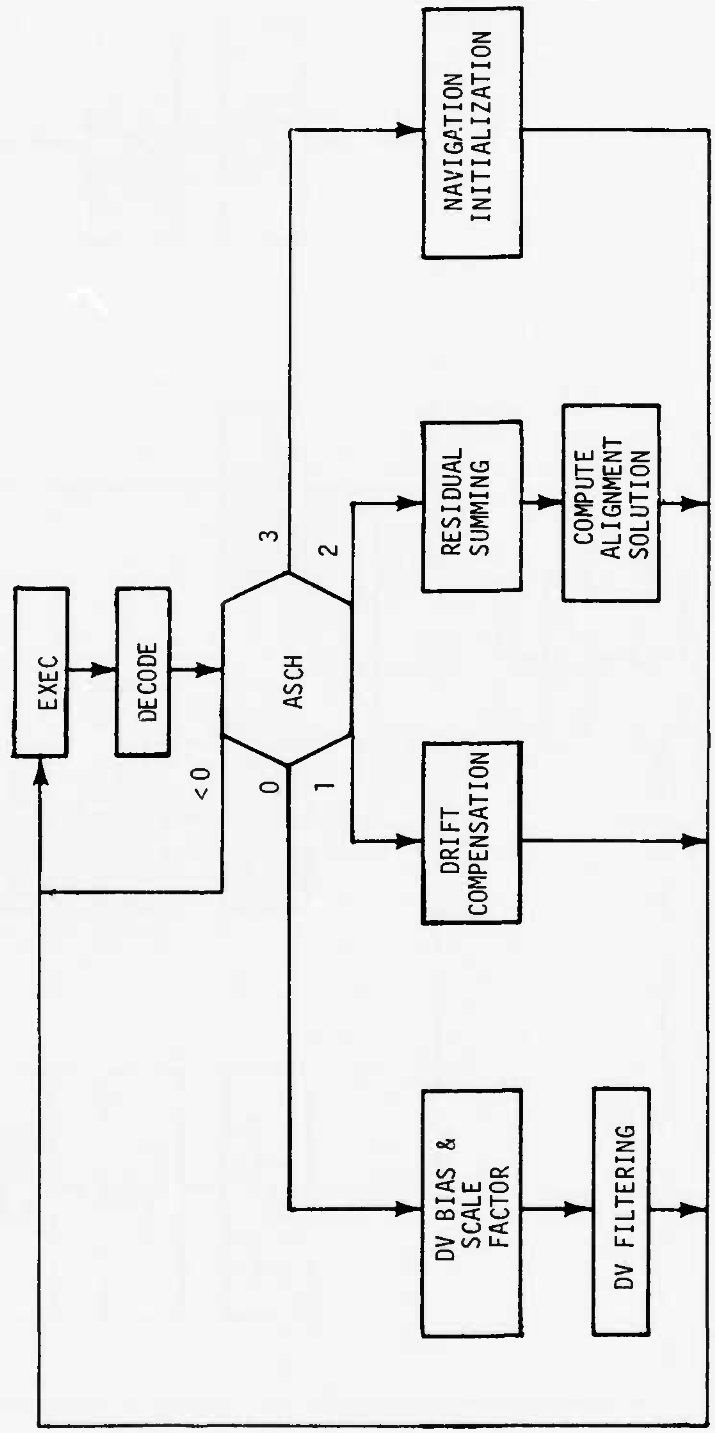
A_1 = 601.8042666979565

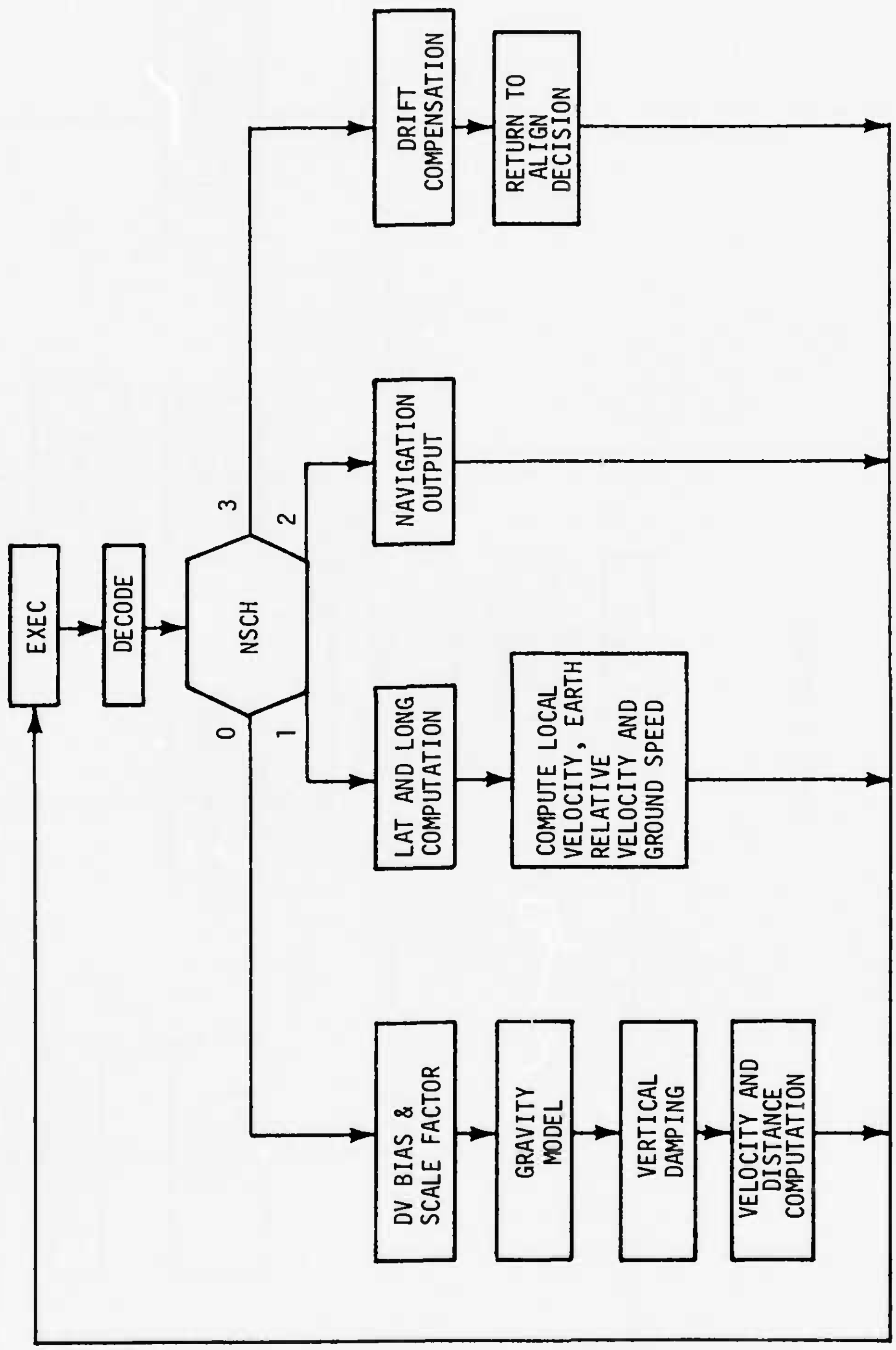
B_1 = 60.090190731926

FLOW CHARTS

The charts on the following pages reflect the converted SKC-2000 program in its final form.

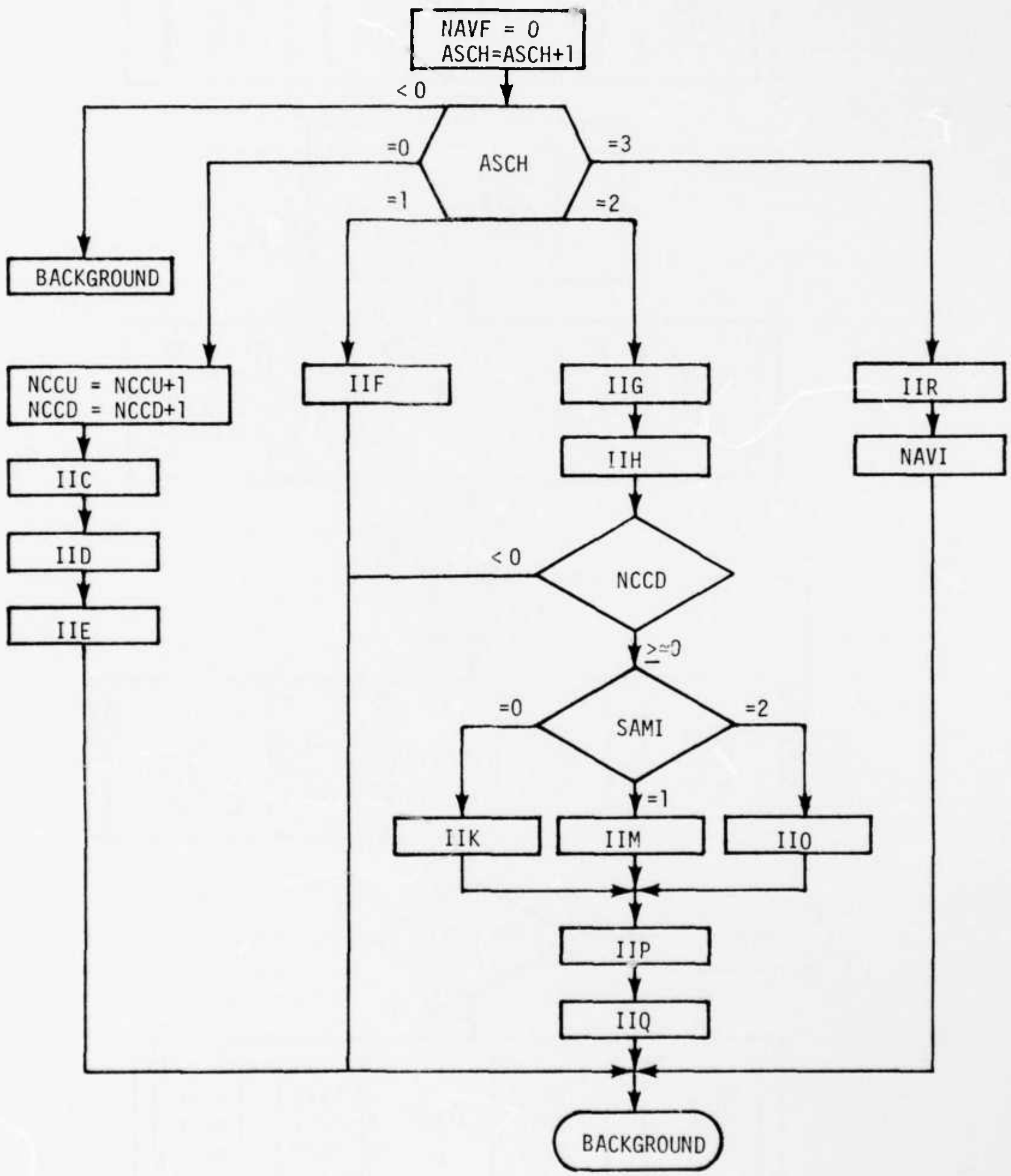
GEANS ALIGNMENT



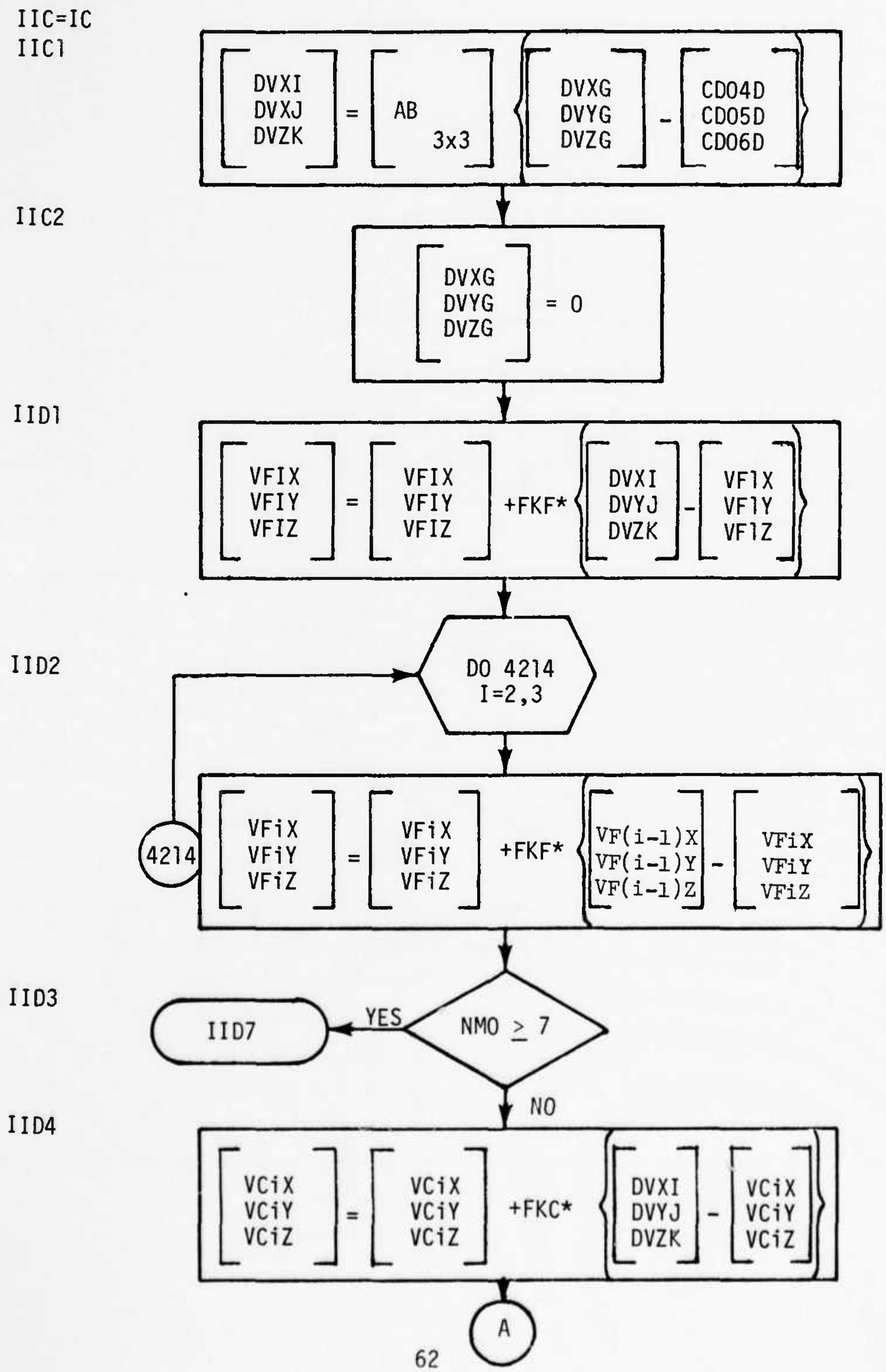


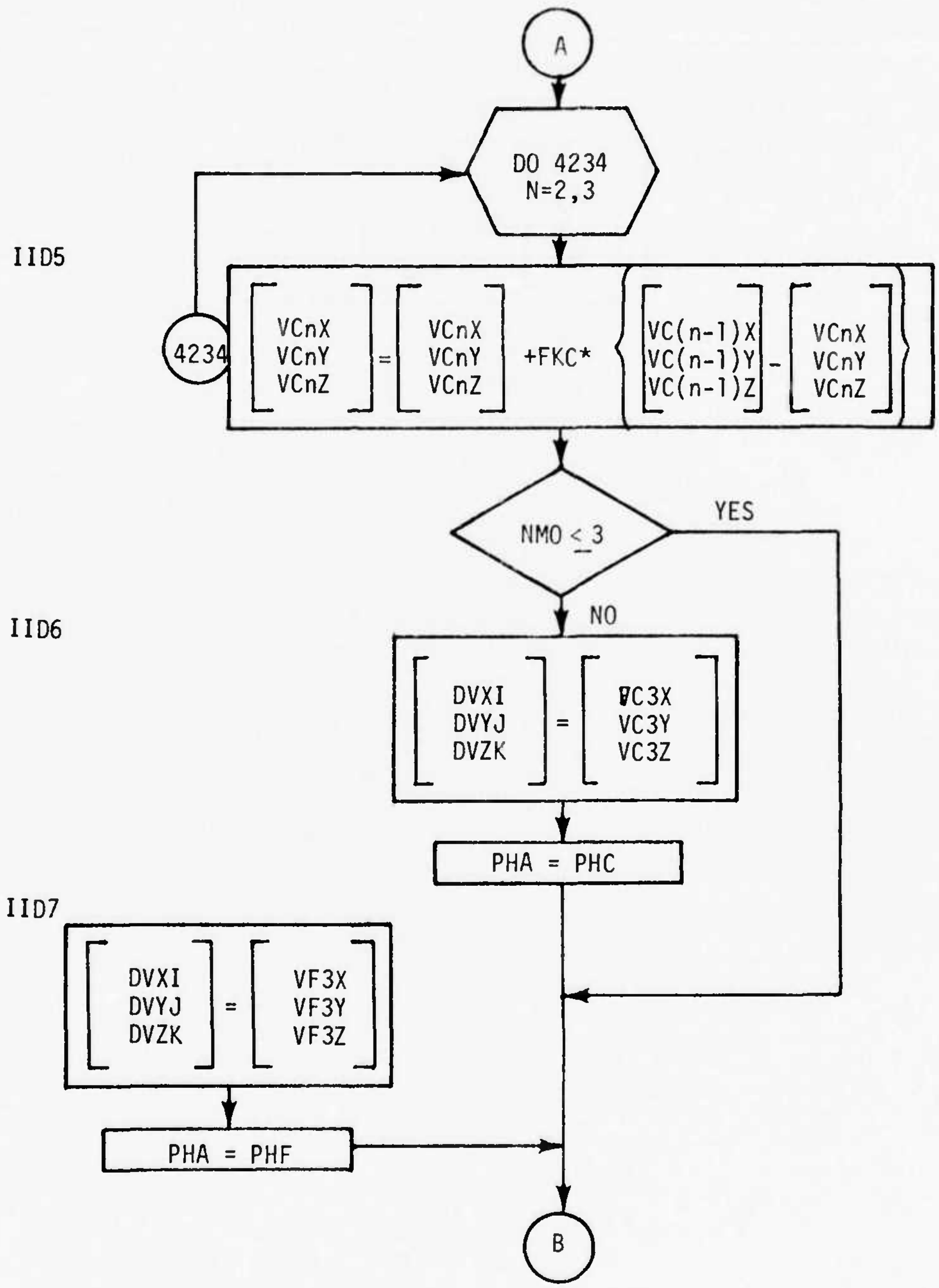
GEANS NAVIGATION

ALIGNMENT SUB-EXECUTIVE

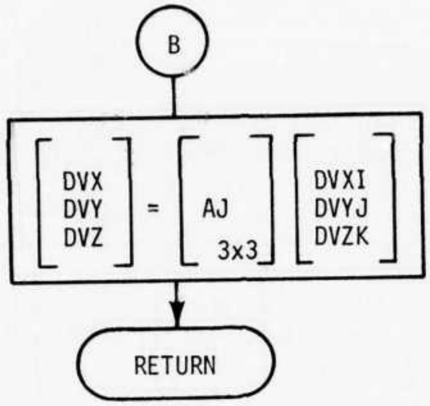


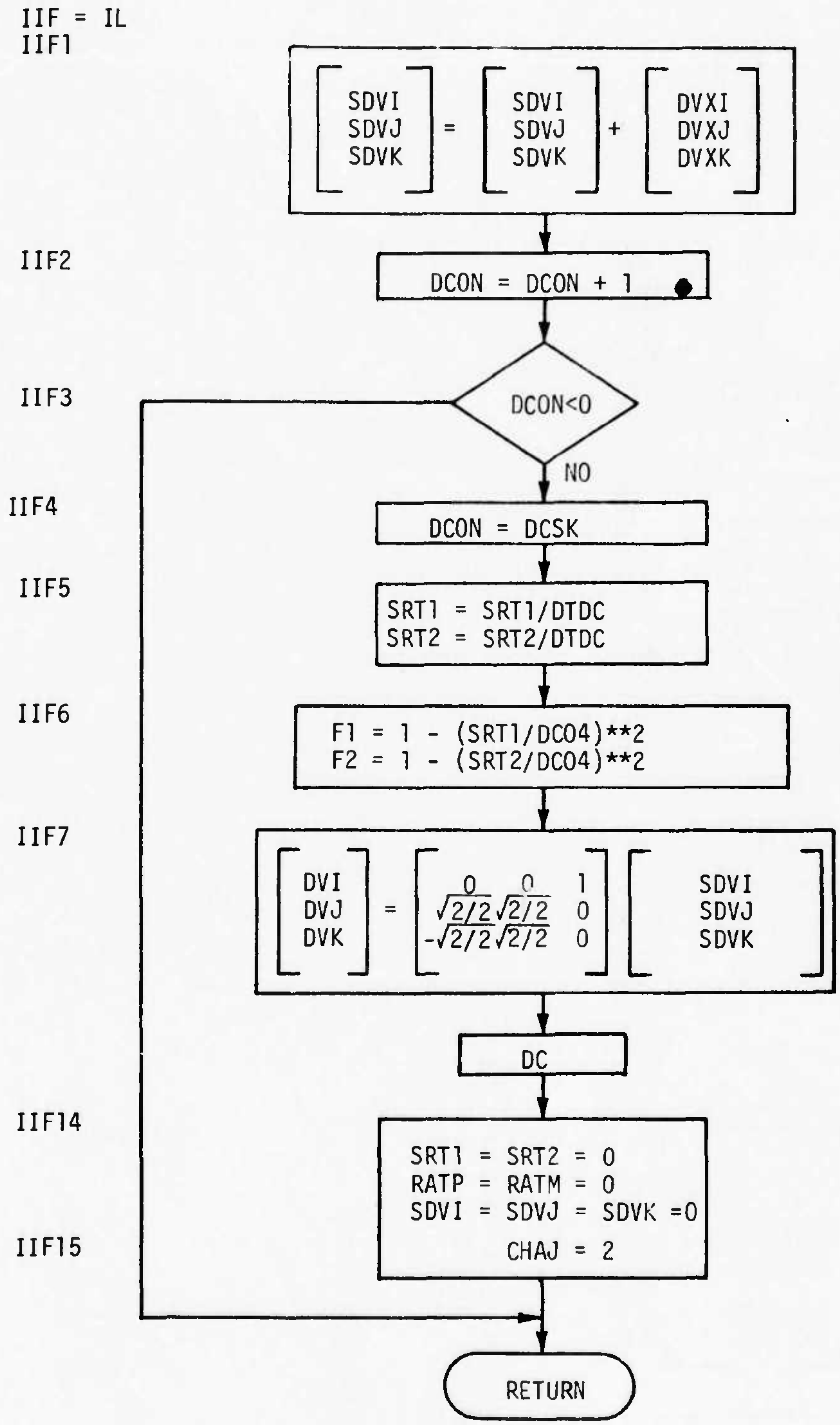
LOW PASS FILTER




A
DO 4234
N=2,3
IID5
4234
VCnX
VCnY
VCnZ
=
VCnX
VCnY
VCnZ
+FKC*
VC(n-1)X
VC(n-1)Y
VC(n-1)Z
-
VCnX
VCnY
VCnZ
NMO ≤ 3
YES
NO
IID6
DVXI
DVYJ
DVZK
=
VC3X
VC3Y
VC3Z
PHA = PHC
IID7
DVXI
DVYJ
DVZK
=
VF3X
VF3Y
VF3Z
PHA = PHF
B

IIE=ID
IIEI




IIF = IL
IIF1
SDVI
SDVJ
SDVK
=
SDVI
SDVJ
SDVK
+
DVXI
DVXJ
DVXK
IIF2
DCON = DCON + 1
IIF3
DCON<0
NO
IIF4
DCON = DCSK
IIF5
SRT1 = SRT1/DTDC
SRT2 = SRT2/DTDC
IIF6
F1 = 1 - (SRT1/DC04)**2
F2 = 1 - (SRT2/DC04)**2
IIF7
DVI
DVJ
DVK
=
0 0 1
√2/2 √2/2 0
-√2/2 √2/2 0
SDVI
SDVJ
SDVK
DC
IIF14
SRT1 = SRT2 = 0
RATP = RATM = 0
SDVI = SDVJ = SDVK =0
IIF15
CHAJ = 2
RETURN

IIF12 IIF8 IIF9 IIF10

$$\begin{bmatrix} \emptyset_x \\ \emptyset_y \\ \emptyset_z \end{bmatrix} = \underbrace{\begin{bmatrix} \frac{DC42}{SRT1} & \frac{DC42}{SRT2} & 0 \\ \frac{DC42}{SRT1}- & \frac{DC42}{SRT2} & 0 \\ 0 & 0 & \frac{DC43}{SRT2} \end{bmatrix}}_{\text{TRANSFORM TO PLATFORM}} \Bigg\{ \underbrace{DTDC * \begin{bmatrix} CD16 \\ CD18 \\ CD17 \end{bmatrix} + \begin{bmatrix} 0 \\ CD28*RATP \\ -CD29*RATM \\ 0 \end{bmatrix}}_{\text{G INDEPENDENT, SPEED INDEPENDENT}} + \underbrace{\begin{bmatrix} GM \\ 3x3 \end{bmatrix} \begin{bmatrix} DVI \\ DVJ \\ DVK \end{bmatrix}}_{\text{G DEPENDENT, SPEED INDEPENDENT}} + \underbrace{\begin{bmatrix} F1*CD30 \\ F2*CD32 \\ SRT2 \\ +(1-DC04)(CD42*RATP-CD43*RATM) \\ F1*CD31 \end{bmatrix}}_{\text{G INDEPENDENT, SPEED DEPENDENT}} +$$

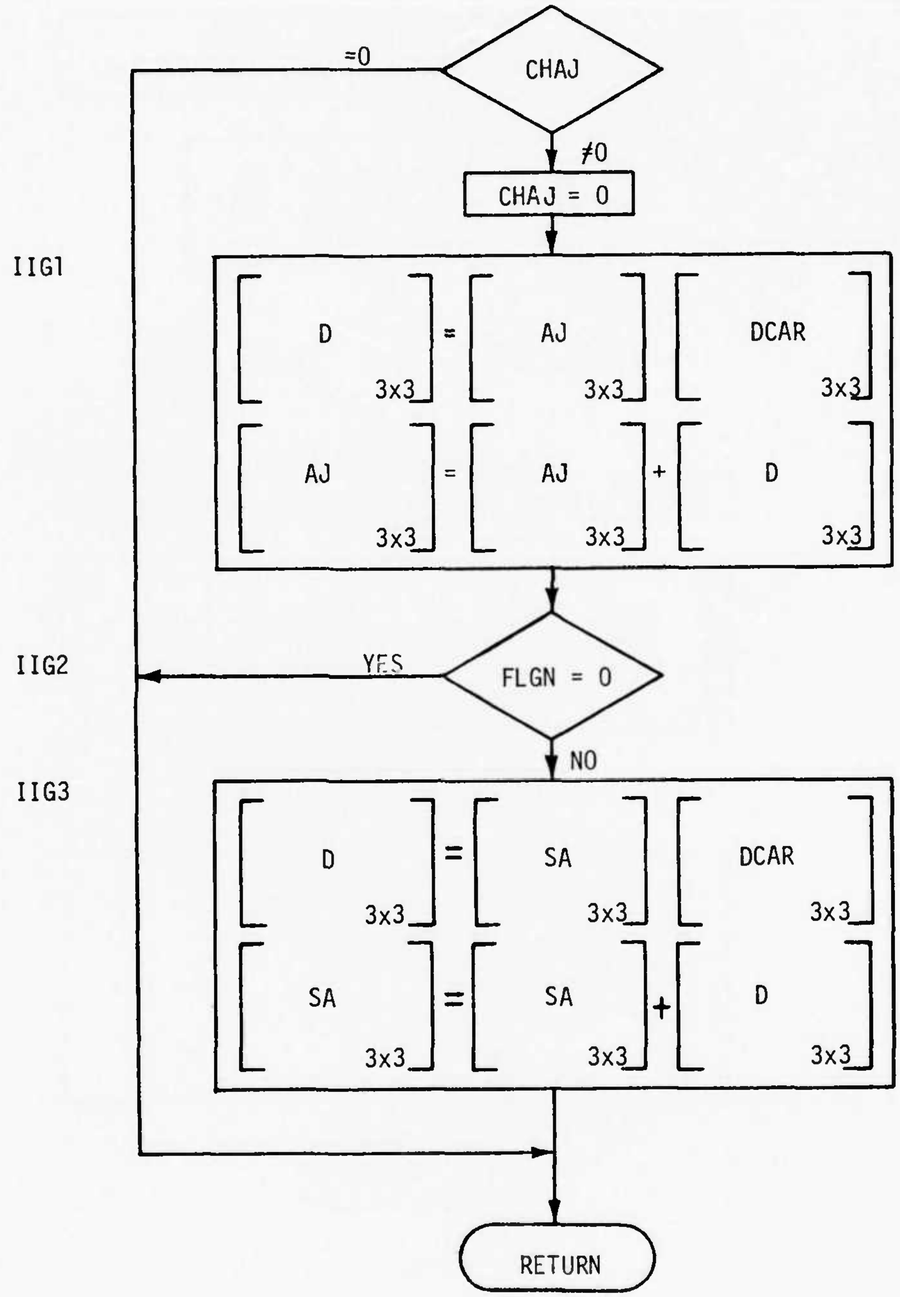
IIF11

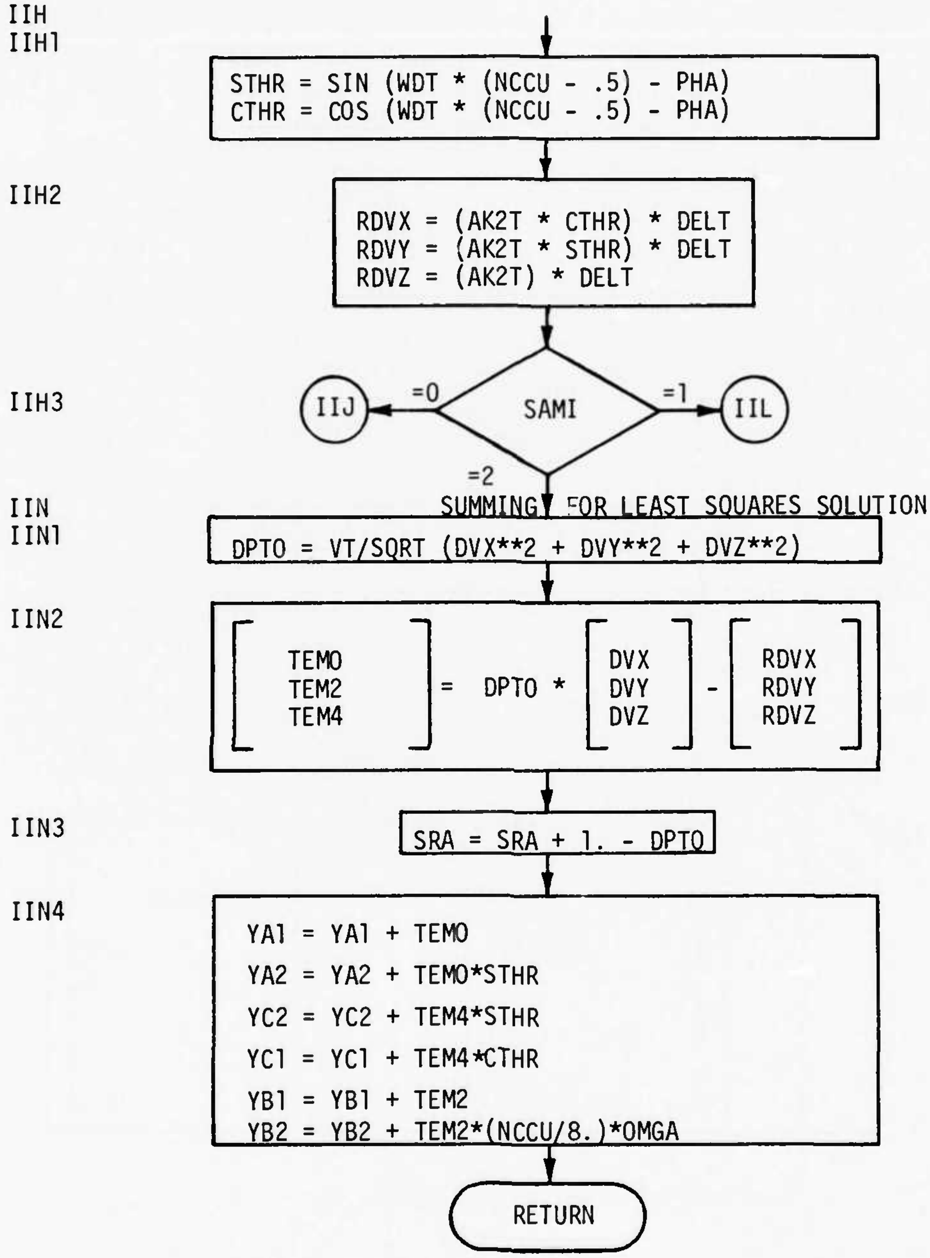
$$\underbrace{\begin{bmatrix} F1*CD33 & F1*CD34 & F1*CD35 \\ -F2*CD39 & -F2*CD40 & -F2*CD41 \\ F1*CD36 & F1*CD37 & F1*CD38 \end{bmatrix} \begin{bmatrix} DVI \\ DVJ \\ DVK \end{bmatrix}}_{\text{G \& SPEED DEPENDENT}} \Bigg\}$$

IIF13

$$\begin{bmatrix} DCAR \\ 3x3 \end{bmatrix} = \begin{bmatrix} 0 & Q_z & \emptyset_y \\ -\emptyset_z & 0 & -\emptyset_x \\ -\emptyset_y & Q_x & 0 \end{bmatrix}$$

IIG = IM


=0
CHAJ
≠0
CHAJ = 0
IIG1
D 3x3 = AJ 3x3 DCAR 3x3
AJ 3x3 = AJ 3x3 + D 3x3
IIG2
YES
FLGN = 0
NO
IIG3
D 3x3 = SA 3x3 DCAR 3x3
SA 3x3 = SA 3x3 + D 3x3
RETURN


IIH
IIH1
STHR = SIN (WDT * (NCCU - .5) - PHA)
CTHR = COS (WDT * (NCCU - .5) - PHA)
IIH2
RDVX = (AK2T * CTHR) * DELT
RDVY = (AK2T * STHR) * DELT
RDVZ = (AK2T) * DELT
IIH3
IIJ
=0
SAMI
=1
IIL
=2
IIN
SUMMING FOR LEAST SQUARES SOLUTION
IIN1
DPTO = VT/SQRT (DVX**2 + DVY**2 + DVZ**2)
IIN2
TEMO
TEM2
TEM4
= DPTO *
DVX
DVY
DVZ
-
RDVX
RDVY
RDVZ
IIN3
SRA = SRA + 1. - DPTO
IIN4
YA1 = YA1 + TEMO
YA2 = YA2 + TEMO*STHR
YC2 = YC2 + TEM4*STHR
YC1 = YC1 + TEM4*CTHR
YB1 = YB1 + TEM2
YB2 = YB2 + TEM2*(NCCU/8.)*OMGA
RETURN

SUMMING FOR E.P.A. SOLUTION

IIJ

IIJ1

$$\begin{bmatrix} VAXI \\ VAYJ \\ VAZK \end{bmatrix} = \begin{bmatrix} VAXI \\ VAYJ \\ VAZK \end{bmatrix} + \begin{bmatrix} DVXI \\ DVYJ \\ DVZK \end{bmatrix}$$

IIJ2

$$\begin{bmatrix} VAX \\ VAY \\ VAZ \end{bmatrix} = \begin{bmatrix} VAX \\ VAY \\ VAZ \end{bmatrix} + \begin{bmatrix} DVX \\ DVY \\ DVZ \end{bmatrix}$$

RETURN

SUMMING FOR LOCAL LEVEL SOLUTION

IIL

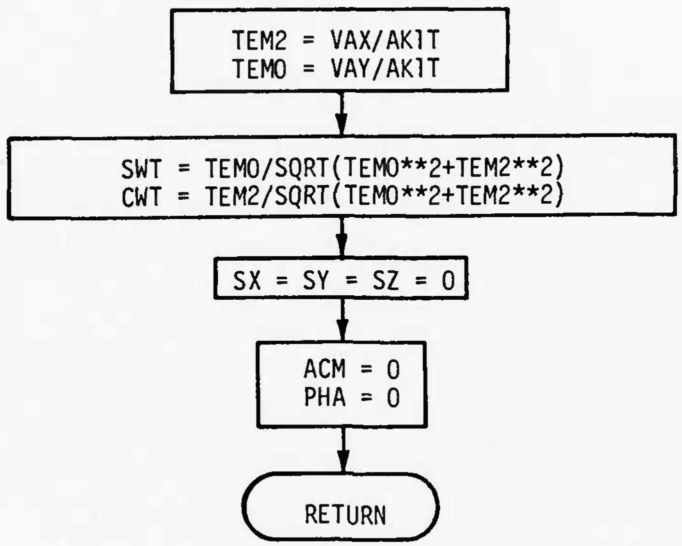
IIL1

$$\begin{bmatrix} VAX \\ VAY \\ VAZ \end{bmatrix} \begin{bmatrix} VAX \\ VAY \\ VAZ \end{bmatrix} + \begin{bmatrix} DVX \\ DVY \\ DVZ \end{bmatrix} - \begin{bmatrix} RDVX \\ RDVY \\ RDVZ \end{bmatrix}$$

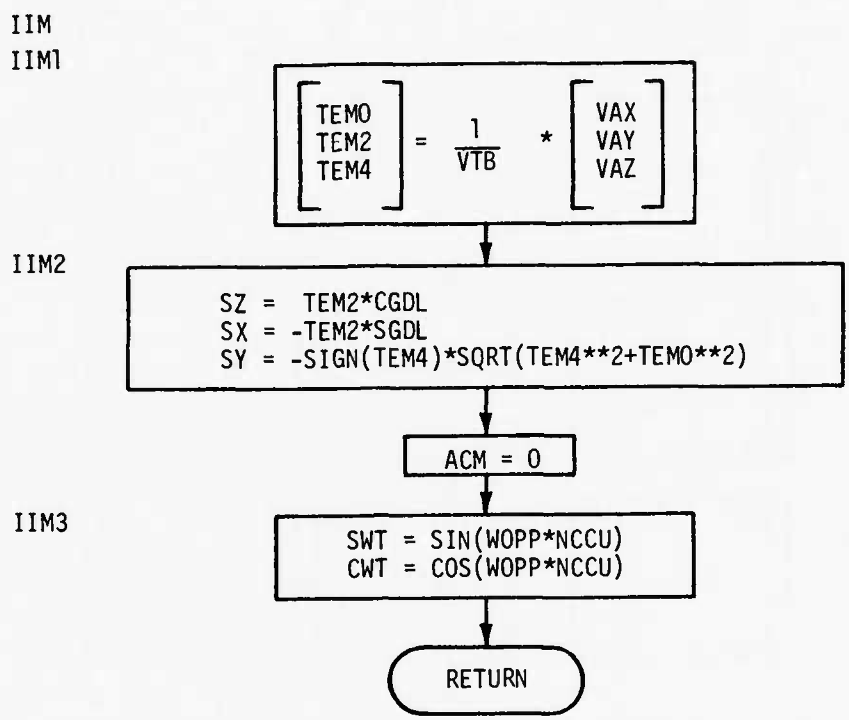
RETURN

E.P.A. SOLUTION

IIK
IIK2



LOCAL LEVEL SOLUTION



LEAST SQUARES SOLUTION

IIØ
IIØ1

$$\begin{bmatrix} XA \\ & 2x1 \end{bmatrix} = \begin{bmatrix} {}^{a}MCSI \\ & * \end{bmatrix} \begin{bmatrix} YA \\ & 2x1 \end{bmatrix}$$

↓

$$\begin{bmatrix} XB \\ & 2x1 \end{bmatrix} = \begin{bmatrix} {}^{/b}MCSI \\ & * \end{bmatrix} \begin{bmatrix} YB \\ & 2x1 \end{bmatrix}$$

↓

$$\begin{bmatrix} XC \\ & 2x1 \end{bmatrix} = \begin{bmatrix} {}^{c}MCSI \\ & * \end{bmatrix} \begin{bmatrix} YC \\ & 2x1 \end{bmatrix}$$

↓

IIØ2

```
SX = XC(2)/AK1T
SZ - XA(2)/AK1T
SY = -XC(1)/AK1T
ACM = SRA/NCCU

SX = SX+SGDL*(SZ*CGDL-XB(1)/VTC-SX*SGDL)
SZ = SZ-CGDL*(SZ*CGDL-XB(1)/VTC-SX*SGDL)
```

↓

II04 = IIM3

```
SWT = SIN(WOPP*NCCU)
CWT = COS(WOPP*NCCU)
```

↓

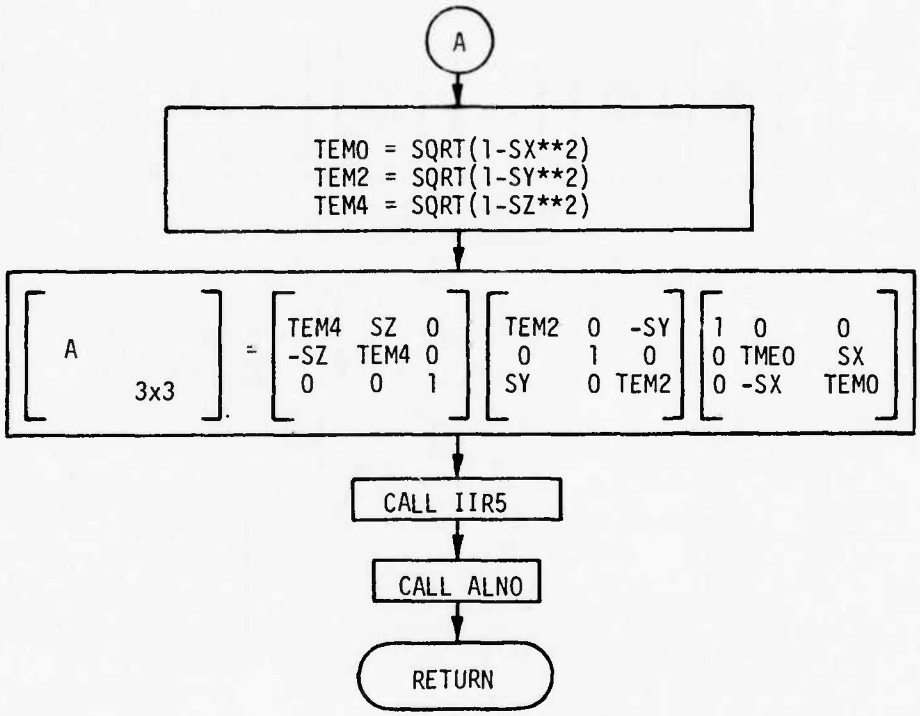
A

COMPUTE ΔA MATRIX AND ΔAJ MATRIX

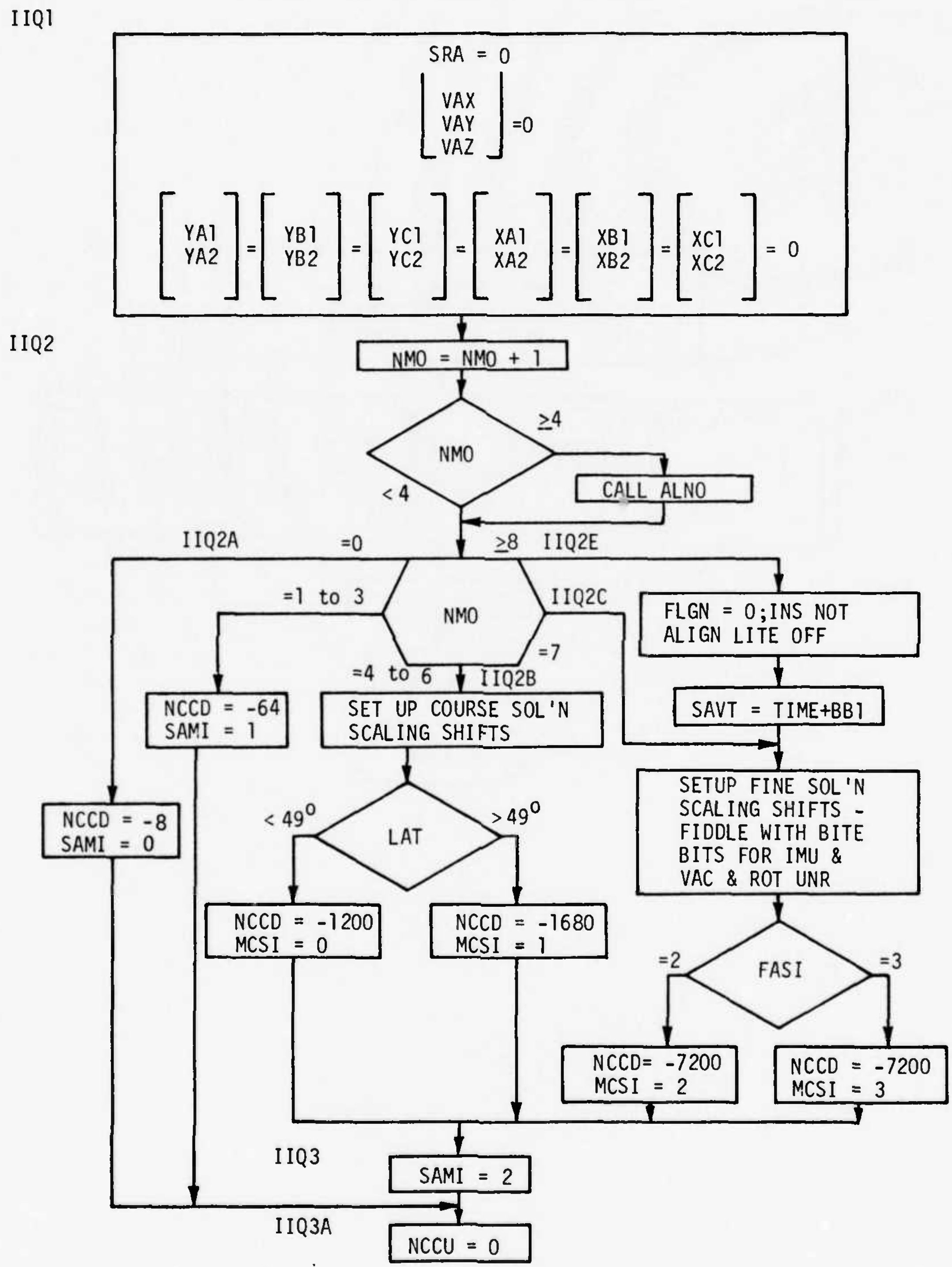
IIP
IIP1

IIP2

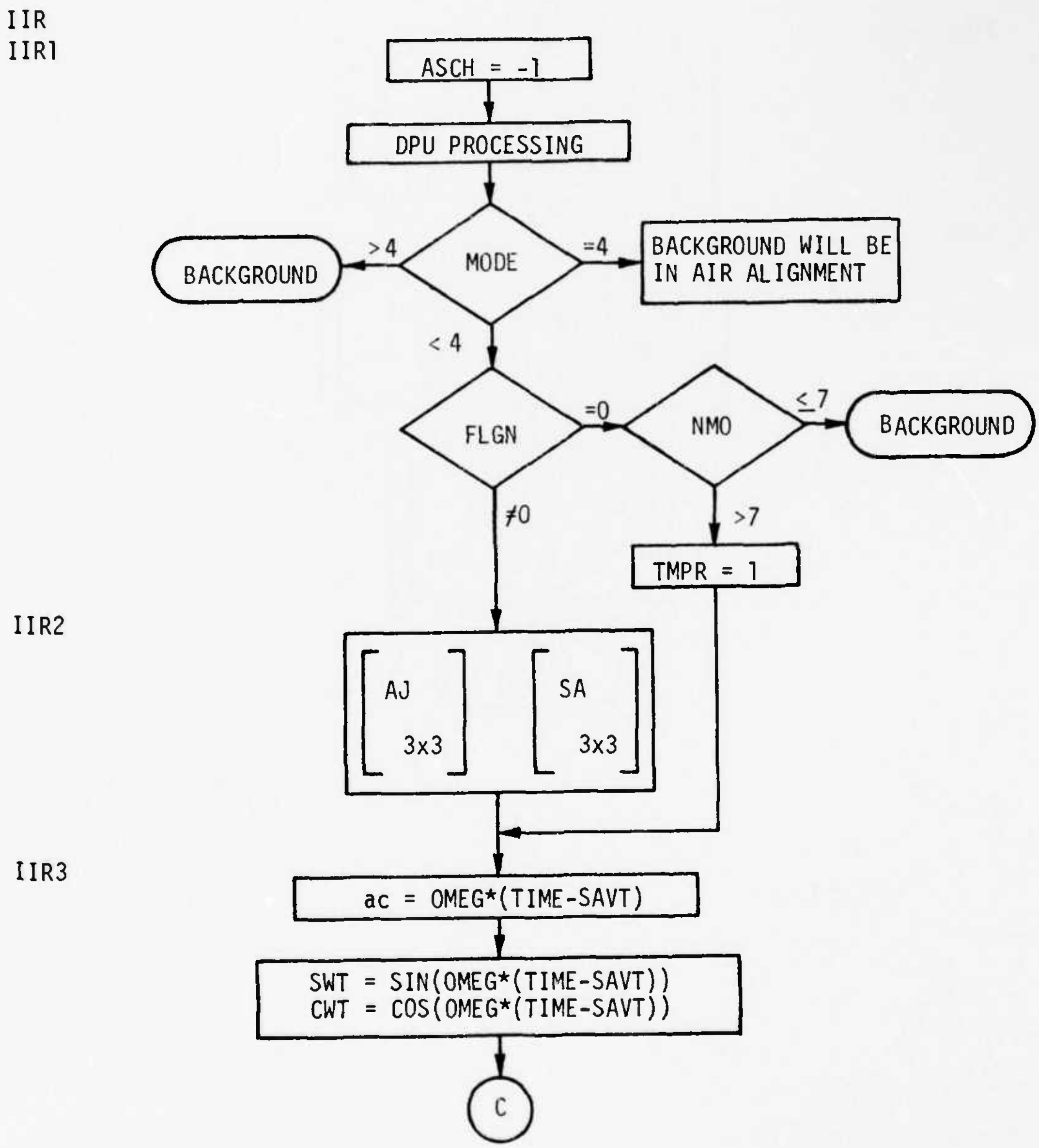
IIP3



SUBROUTINE RESET

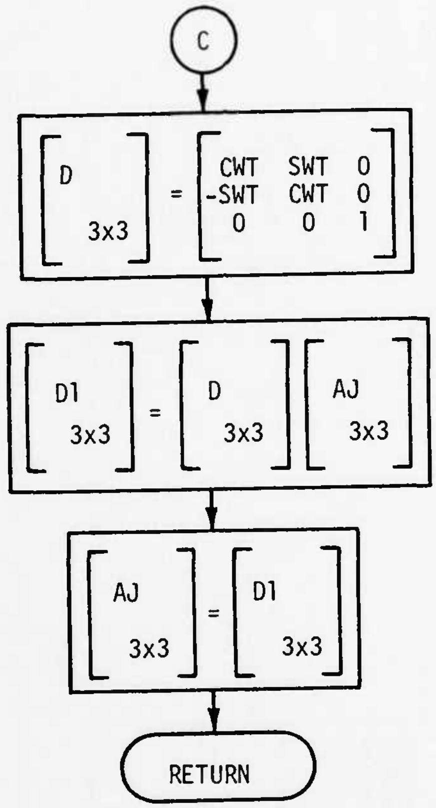


GO TO NAV DECISION



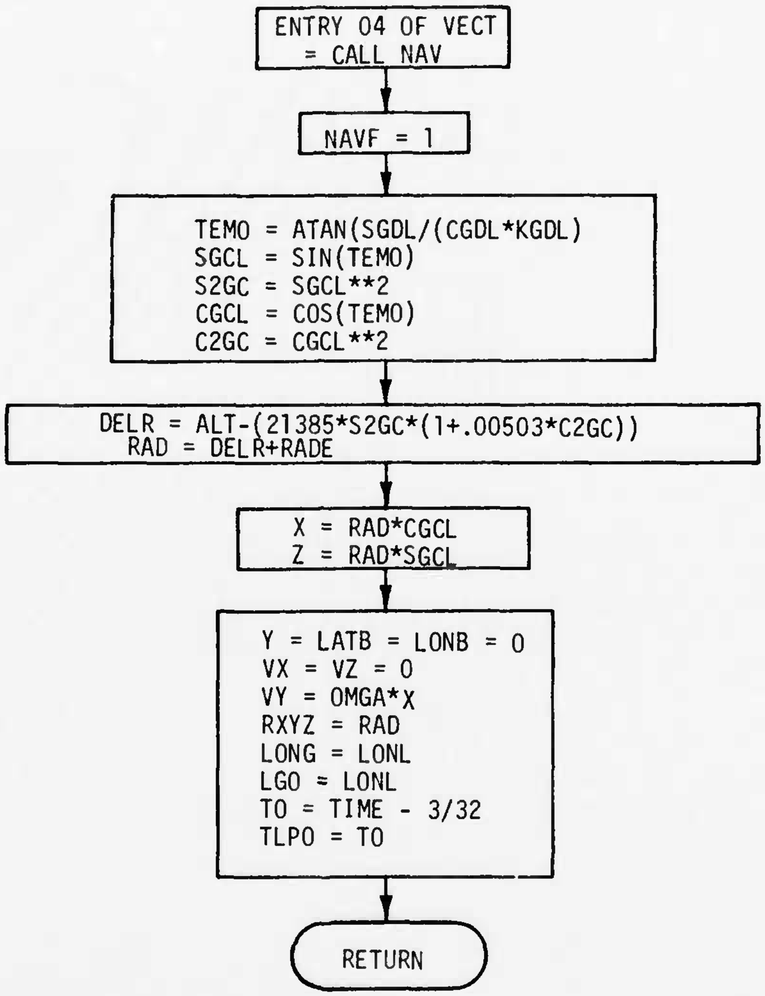
IIR5

IIR6


C
D
3x3
=
CWT SWT 0
-SWT CWT 0
0 0 1
D1
3x3
=
D
3x3
AJ
3x3
AJ
3x3
=
D1
3x3
RETURN

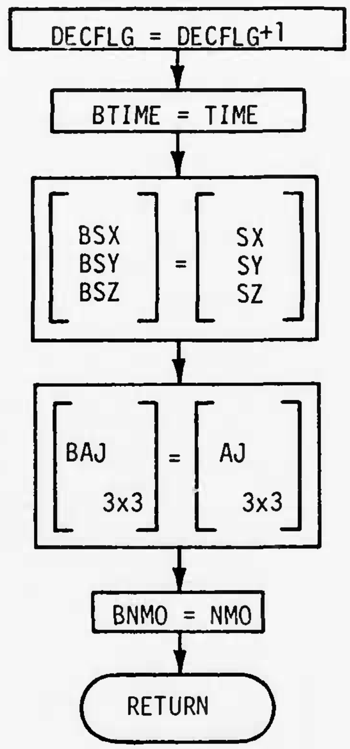
NAVIGATION INITIALIZATION

NAVI

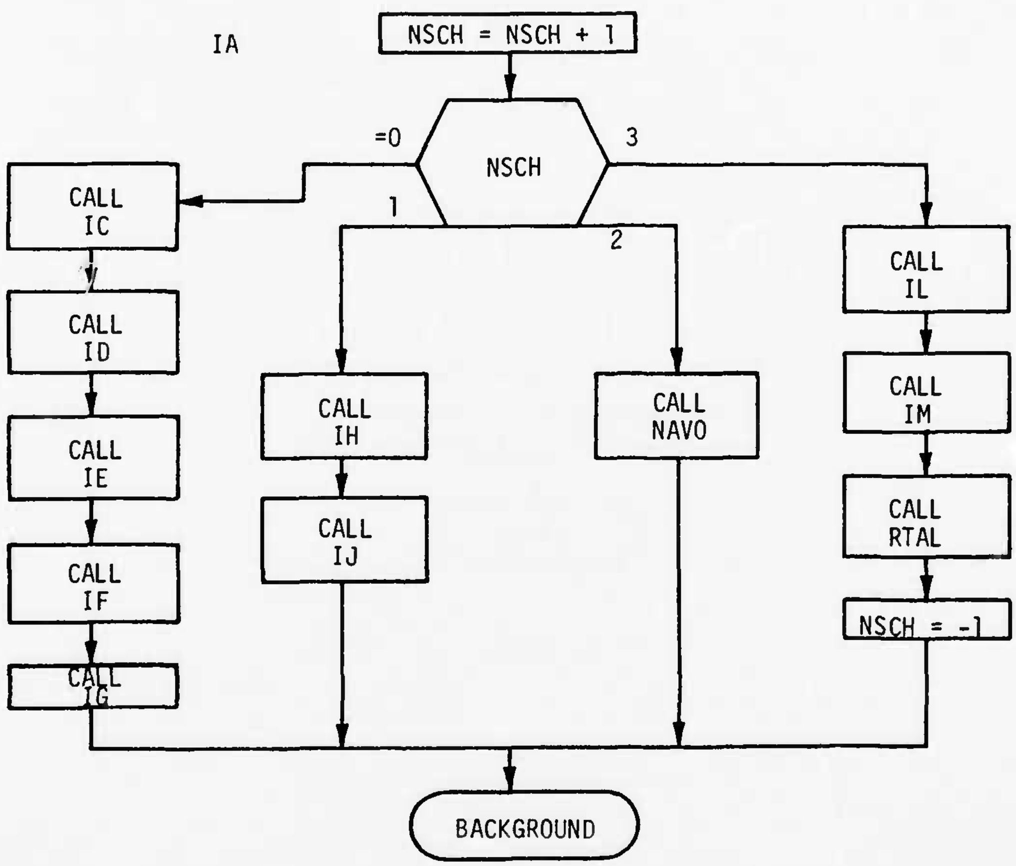


ALIGNMENT OUTPUT ROUTINE

ALNO



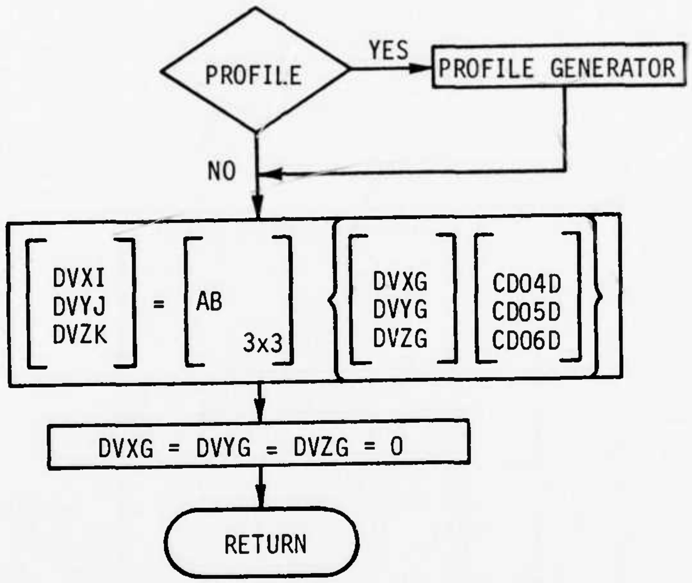
NAVIGATION SUB-EXECUTIVE



ACCELEROMETER BIAS & SCALE FACTOR COMPUTATION
AND NON-ORTHOGONALITY COMPENSATION

IC

IC1



ROTATION FROM PLATFORM FRAME TO NAVIGATION FRAME

ID
ID1

$$\begin{bmatrix} DVX \\ DVY \\ DVZ \end{bmatrix} = \begin{bmatrix} AJ & \\ & 3x3 \end{bmatrix} \begin{bmatrix} DVXI \\ DVYJ \\ DVZK \end{bmatrix}$$

RETURN

GRAVITY MODEL

IE
IE1

P = 1 - (DELR/RADE) + (DELR/RADE)**2

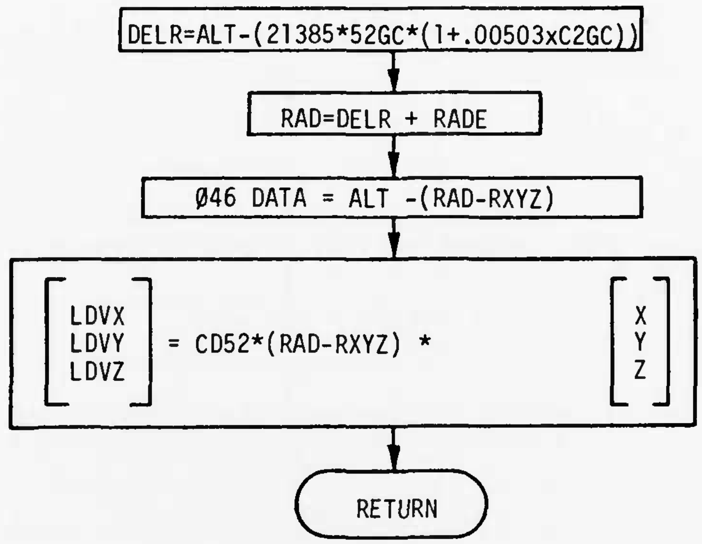
IE2

```
[GXDT]   [GCA2*(GCA5-6*S2GC+9*S2GC**2)+GCA0*P**3-GCA1((GCA3-S2GC)*P**5)*X              ]
[GYDT] = [GCA2*(GCA5-6*S2GC+9*S2GC**2)+GCA0*P**3-GCA1((GCA3-S2GC)*P**5)*Y              ]
[GZDT]   [GCA2*(GCA6+GCA5-10*S2GC+9*S2GC**2)+GCA0*P**3-GCAI((GCA4+GCA3-S2GC)*P**5)*Z   ]
```

RETURN

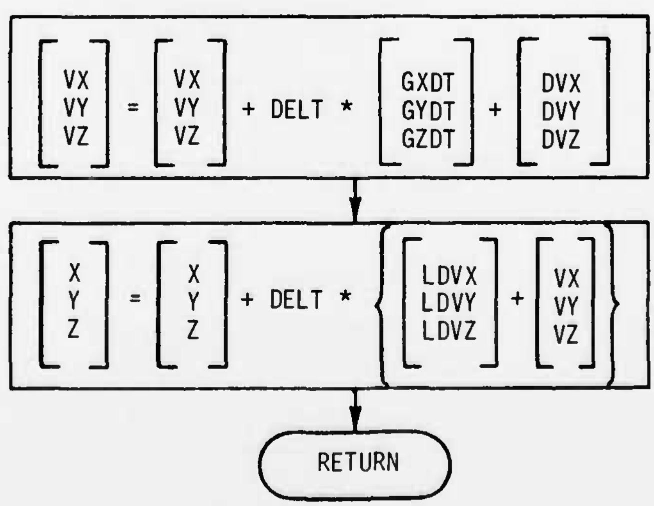
VERTICAL DAMPING COMPUTATION

IF
IF2
IF3
IF1



DOUBLE INTEGRATION FOR VELOCITY AND DISTANCE

IG
IG1



LATITUDE AND LONGITUDE COMPUTATION

IH

IH1 LAT = LATB+ATAN((Z*KGDL)/SQRT(X**2+Y**2))

IH2 Ø42DATA = MSH OF LAT
043DATA = LSH OF LAT

IH3 LONG = LONB+ATAN (Y/X)-((TIME-TO)*OMEG-LGO)

IH4 Ø44DATA = MSH OF LONG
045DATA = LSH OF LONG

IH5 SCLG = SIN((LONB+ATAN (Y/X))

IH6 CCLG = COS((LONB+ATAN (Y/X))

IH7 SWT = SIN((TIME-TO)*OMEG-LGO)

IH8 CWT = COS((TIME-TO)*OMEG-LGO)

IH9 RXYZ = SQRT(X**2+Y**2+Z**2)

IH10 SGCL = Z/RXYZ

IH11 S2GC = SGCL**2

IH12 CGCL = SQRT(X**2+Y**2)/RXYZ

IH13 C2GC = CGCL**2

IH14 SGDL = SIN(LAT)

IH15 CGDL = COS(LAT)

LOCAL VERTICAL CO-ORDINATES AND GROUND SPEED

IJ

IJ1

$$\begin{bmatrix} VV \\ VE \\ VN \end{bmatrix} = \begin{bmatrix} CGDL*CCLG & CGDL*SCLG & SGDL \\ -SCLG & CCLG & 0 \\ -SGDL*CCLG & -SGDL*SCLG & CGDL \end{bmatrix} \begin{bmatrix} VX+Y*OMGA \\ VY-X*OMGA \\ VZ \end{bmatrix}$$

IJ2

$$\begin{bmatrix} VXE \\ VYE \\ VZE \end{bmatrix} = \begin{bmatrix} CWT & SWT & 0 \\ -SWT & CWT & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} VX+Y*OMGA \\ VY-X*OMGA \\ VZ \end{bmatrix}$$

IJ3

$$\begin{bmatrix} Ø48 \\ Ø4A \\ Ø4C \end{bmatrix} = \begin{bmatrix} LSH\ OF\ VV \\ LSH\ OF\ VE \\ LSH\ OF\ VN \end{bmatrix}$$

$$\begin{bmatrix} Ø47 \\ Ø49 \\ Ø4B \end{bmatrix} = \begin{bmatrix} MSH\ OF\ VV \\ MSH\ OF\ VE \\ MSH\ OF\ VN \end{bmatrix}$$

IJ4

$$VEL2 = \begin{bmatrix} VV \\ VE \\ VN \end{bmatrix} \cdot \begin{bmatrix} VV \\ VE \\ VN \end{bmatrix}$$

IJ5

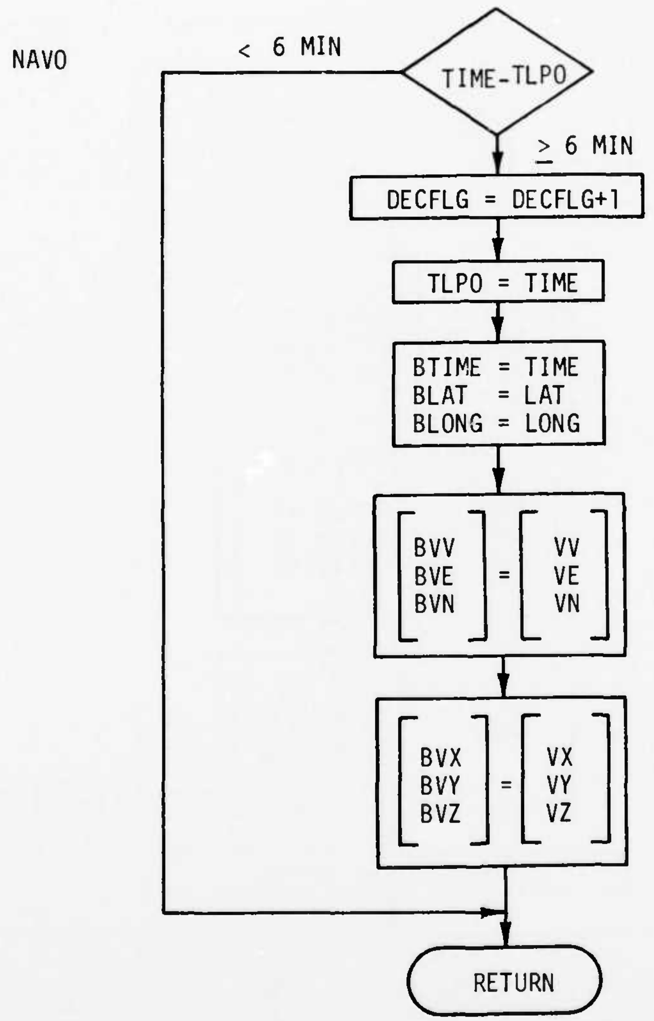
GS = SQRT (VEL2-VV**2)

IJ6

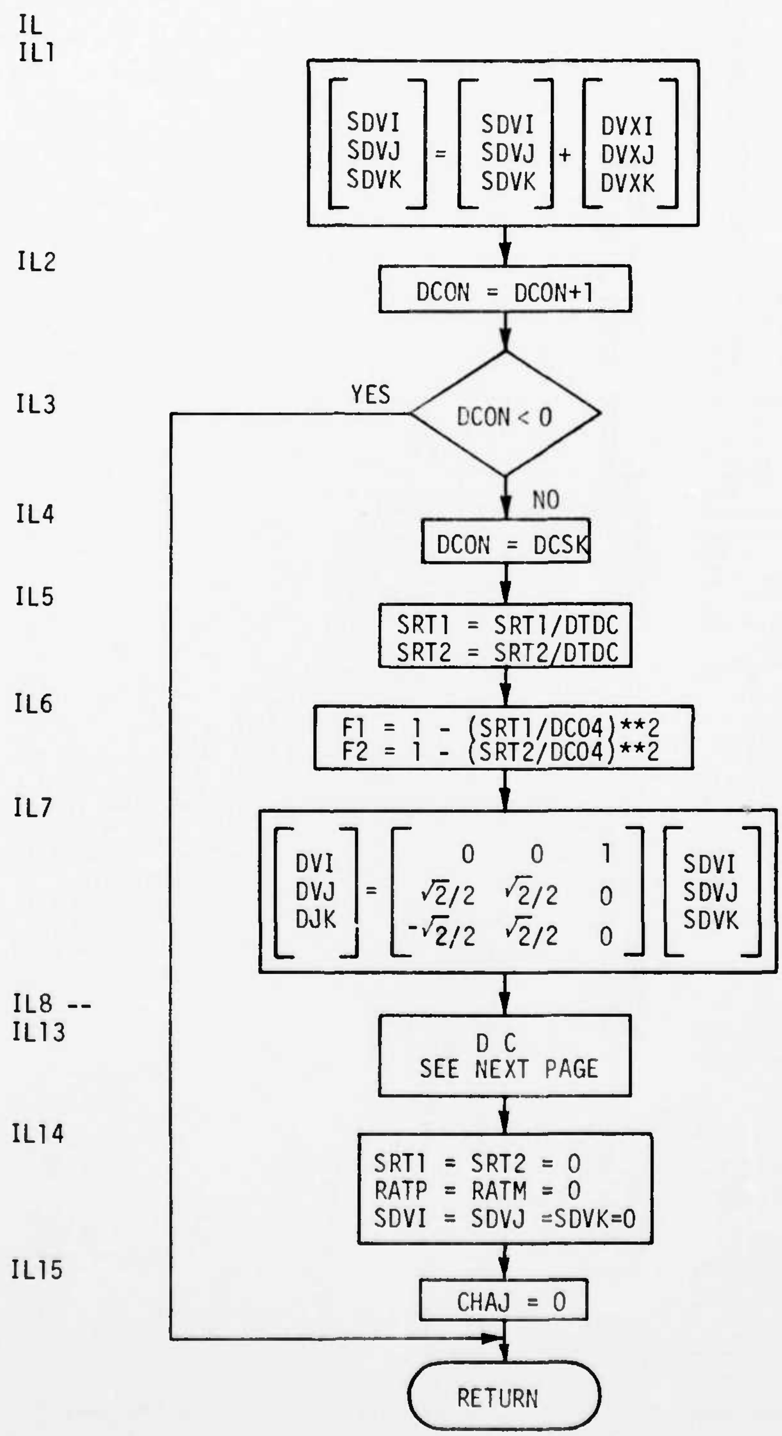
$$\begin{bmatrix} V1 \\ 3x3 \end{bmatrix} = \begin{bmatrix} CEDL*CCLG & CGDL*SCLG & SGDL \\ -SCLG & CCLG & 0 \\ -SGDL*CCLG & -SGDL*SCLG & CGDL \end{bmatrix} \begin{bmatrix} AJ \\ 3x3 \end{bmatrix}$$

RETURN

NAVIGATION OUTPUT ROUTINE



DRIFT COMPENSATION



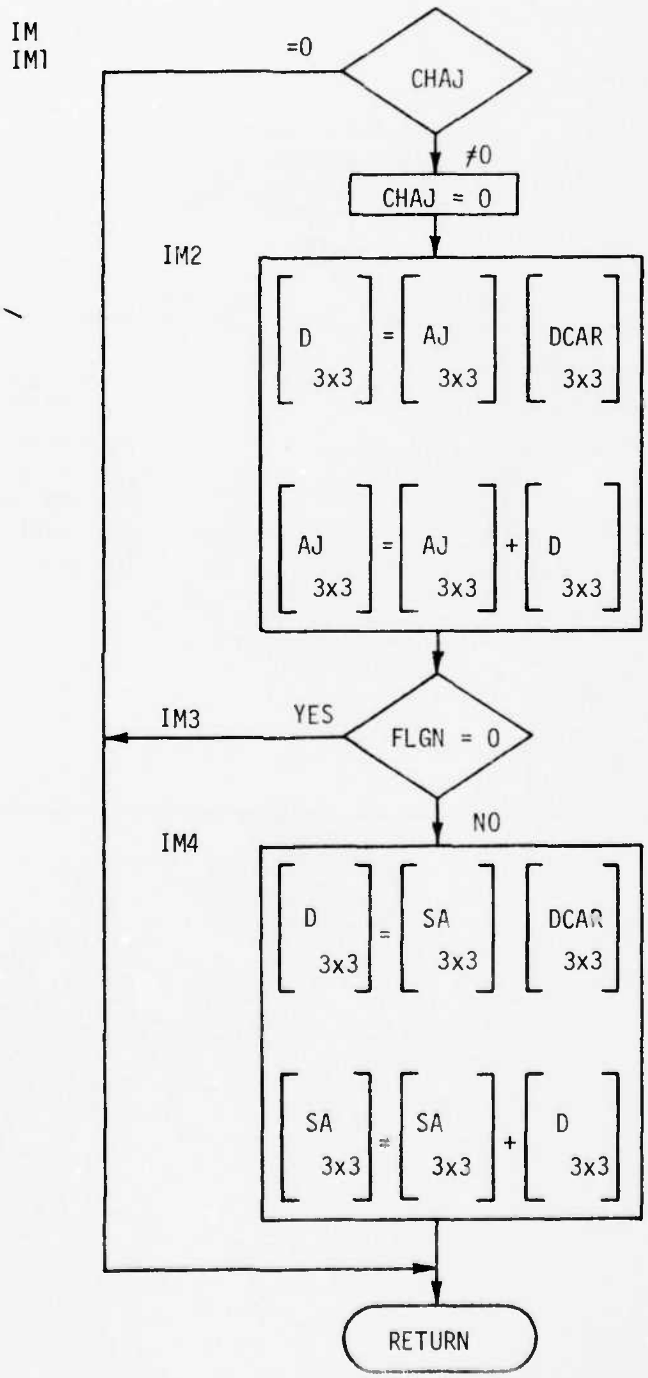
$$
\begin{bmatrix} \phi_x \\ \phi_y \\ \phi_z \end{bmatrix}
=
\underbrace{\overset{\text{IL12}}{\begin{bmatrix} \frac{DC42}{SRT1} & \frac{DC42}{SRT2} & 0 \\ \frac{DC42}{SRT1}- & \frac{DC42}{SRT2} & 0 \\ 0 & 0 & \frac{DC43}{SRT2} \end{bmatrix}}}_{\text{TRANSFORM TO PLATFORM}}
\left\{
\underbrace{DTDC * \overset{\text{IL8}}{\begin{bmatrix} CD16 \\ CD18 \\ CD17 \end{bmatrix}} + \begin{bmatrix} 0 \\ CD28*RATP \\ -CD29*RATM \\ 0 \end{bmatrix}}_{\text{G INDEPENDENT, SPEED INDEPENDENT}}
+
\underbrace{\overset{\text{IL9}}{\begin{bmatrix} GM \\ 3x3 \end{bmatrix}} \begin{bmatrix} DVI \\ DVJ \\ DVK \end{bmatrix}}_{\text{G DEPENDENT, SPEED INDEPENDENT}}
+
\underbrace{\overset{\text{IL10}}{\begin{bmatrix} F1*CD30 \\ \frac{F2*CD32}{SRT2} \\ +(1-DC04)(CD42*RATP-CD43*RATM) \\ F1*CD31 \end{bmatrix}}}_{\text{G INDEPENDENT, SPEED DEPENDENT}}
+
\right.
$$

$$
\left.
\underbrace{\overset{\text{IL11}}{\begin{bmatrix} F1*CD33 & F1*CD34 & F1*CD35 \\ -F2*CD39 & -F2*CD40 & -F2*CD41 \\ F1*CD36 & F1*CD37 & F1*CD38 \end{bmatrix}} \begin{bmatrix} DVI \\ DVJ \\ DVK \end{bmatrix}}_{\text{G \& SPEED DEPENDENT}}
\right\}
$$

IL13

$$
\begin{bmatrix} DCAR \\ 3x3 \end{bmatrix}
=
\begin{bmatrix} 0 & Q_z & \phi_y \\ -\phi_z & 0 & -\phi_x \\ -\phi_y & Q_x & 0 \end{bmatrix}
$$

UPDATE AJ AND SA MATRICES



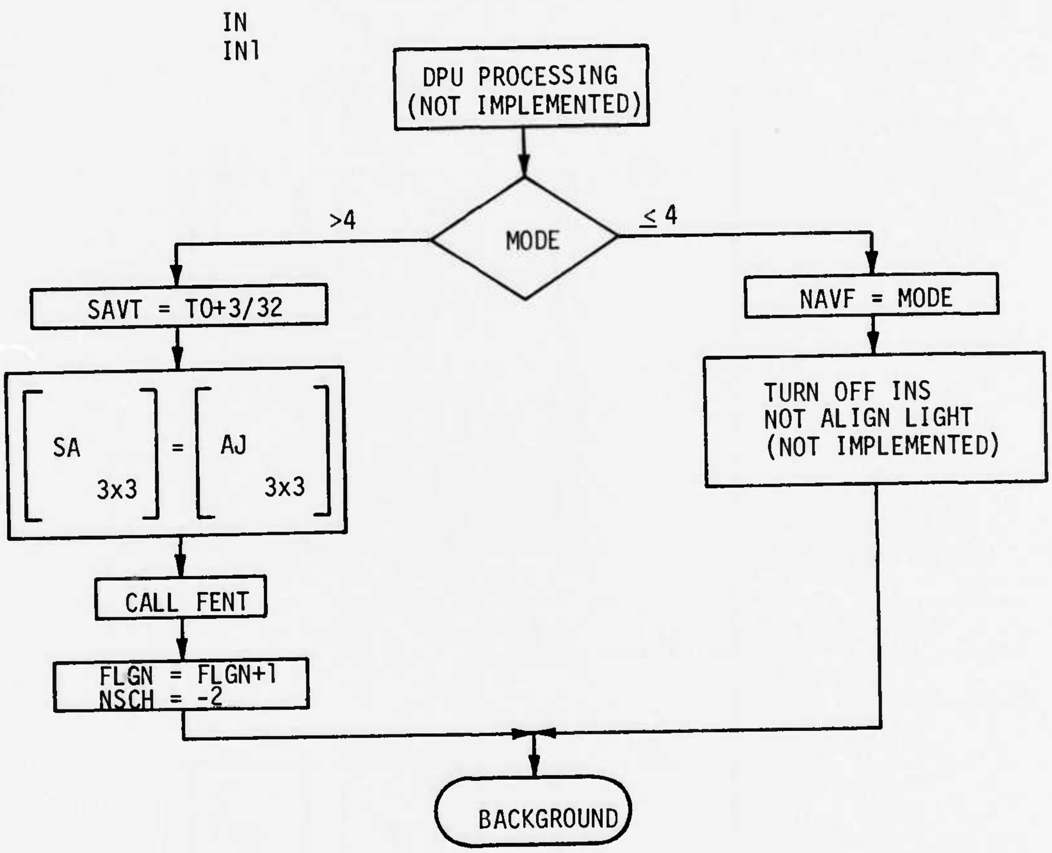
AD-A041 711   AIR FORCE AVIONICS LAB WRIGHT-PATTERSON AFB OHIO   F/G 17/7
CONVERSION OF COMPUTER SOFTWARE FOR THE GIMBALLED ELECTROSTATIC--ETC(U)
FEB 77 W MIKULSKI, W E SHEPHARD

UNCLASSIFIED   AFAL-TR-77-8-VOL-1   NL

2 OF 2
AD
A041711

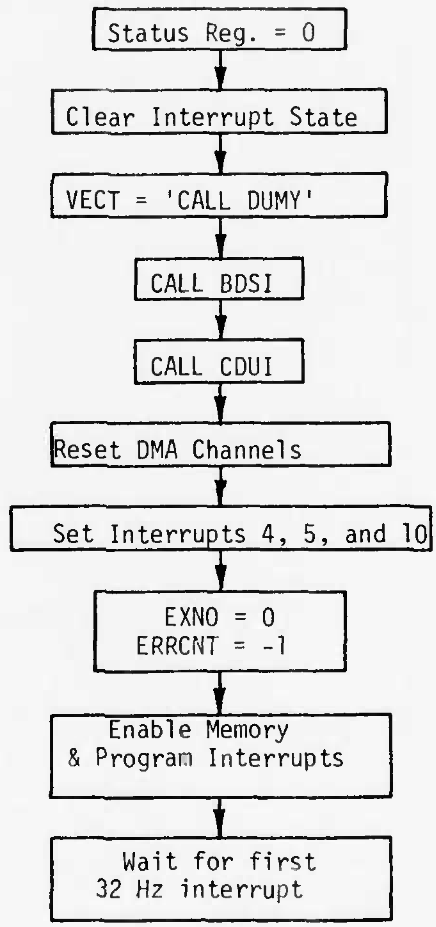
END
DATE
FILMED
8-77

RETURN TO ALIGN DECISION (RTAL)

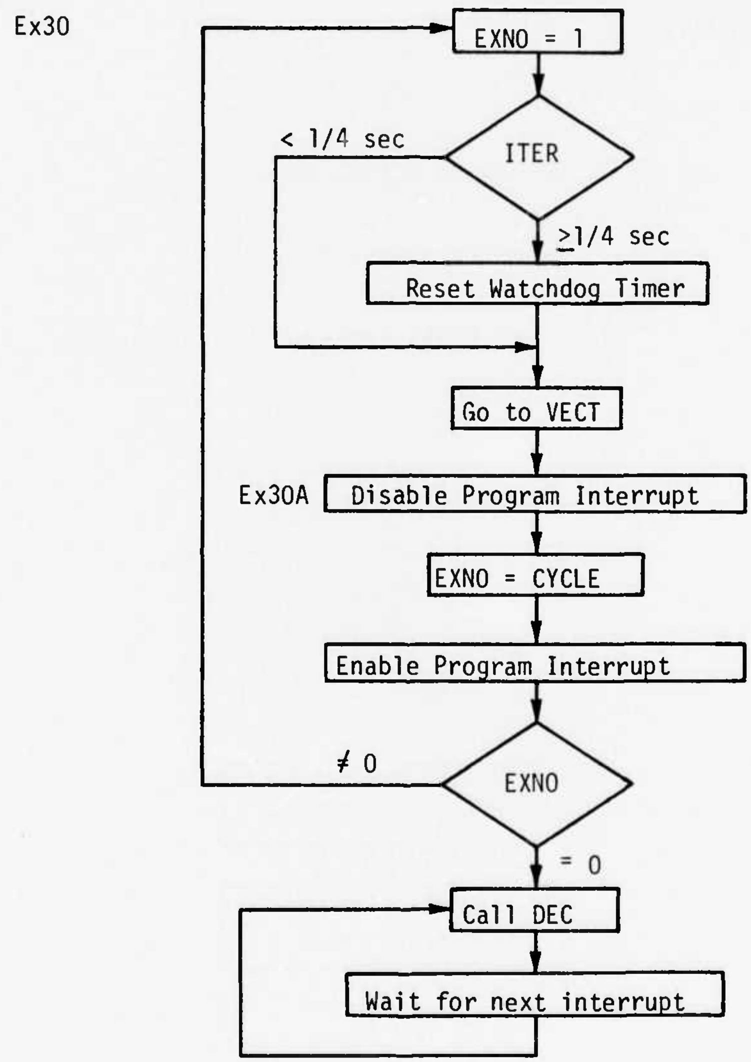


EXECUTIVE INITIALIZATION

EXEC



SKC-2000 Executive



Vector Table

During Alignment

VECT

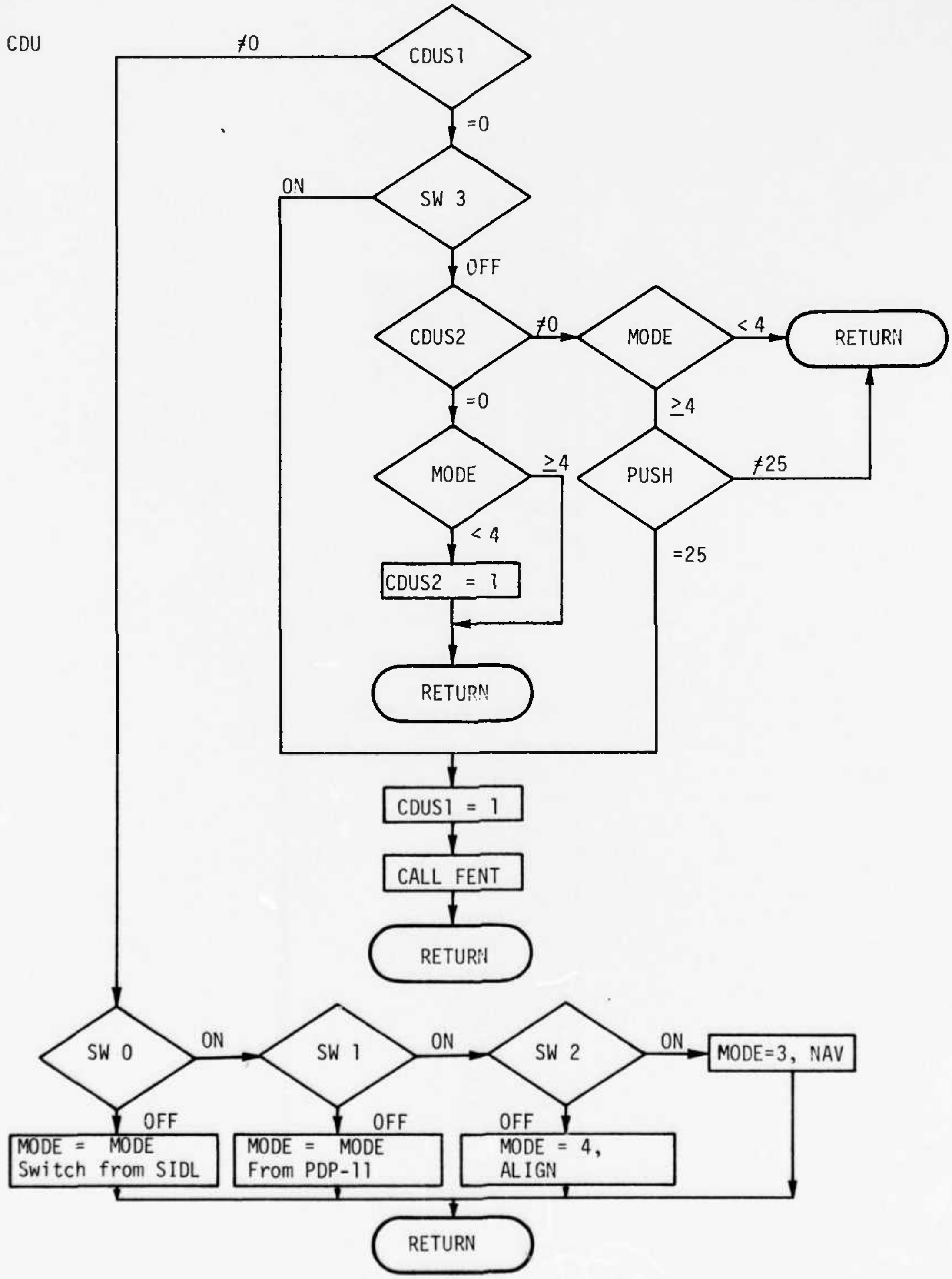
```
CALL DUMY
CALL DECD
CALL CDU
CALL ALIGN (IIA)
CALL SPIN (DUMY)
CALL DUMY
CALL BITE (DUMY)
CALL DUMY
CALL DUMY
CALL DUMY
CALL DUMY
CALL DUMY
CALL DUMY
CALL GASC (DUMY)
GO TO Ex30A
```

During Navigation

VECT

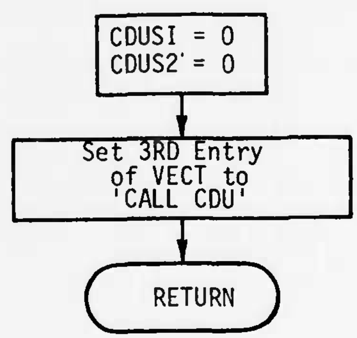
```
CALL DUMY
CALL DECD
CALL CDU
CALL NAV (IA)
CALL SPIN (DUMY)
CALL DUMY
CALL BITE (DUMY)
CALL DUMY
CALL DUMY
CALL DUMY
CALL DUMY
CALL DUMY
CALL DUMY
CALL GASC (DUMY)
GO TO Ex 30A
```

Syncronize SKC-2000 Alignment with Honeywell Alignment.



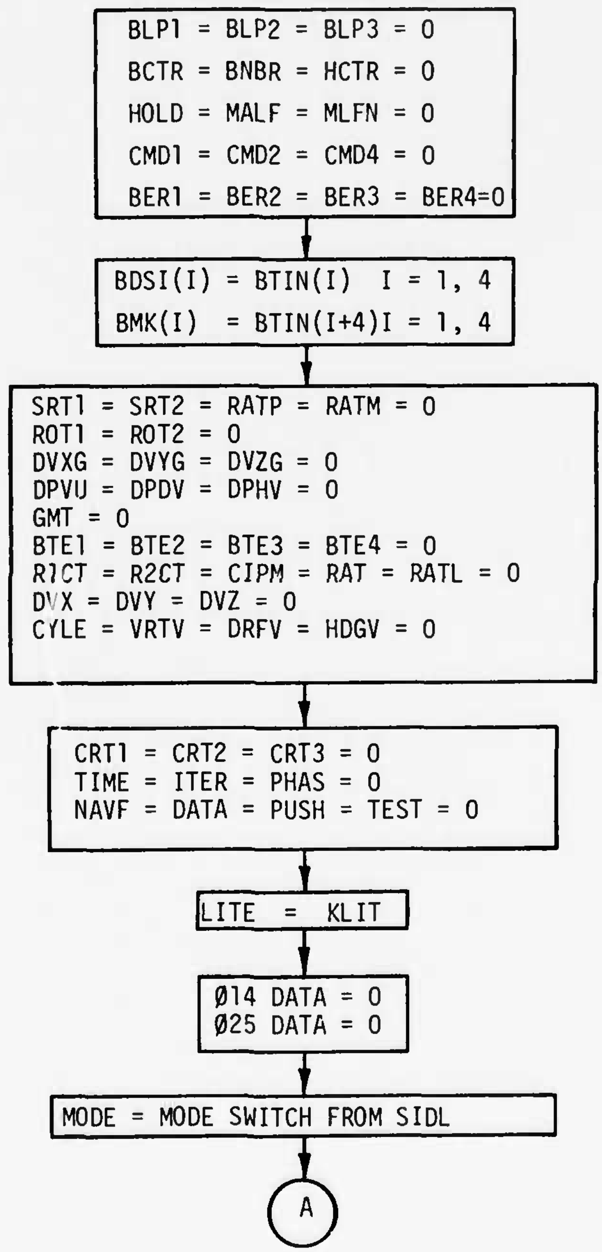
CDU Initialization

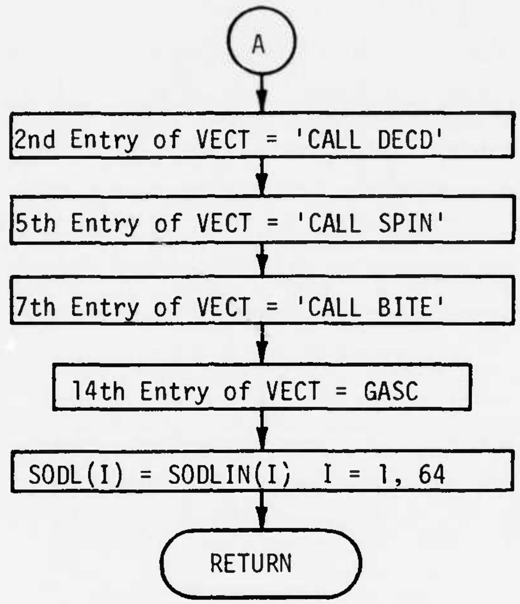
CDUI



Initialize Built-In Test, Data Decode, & Auto Sequencing.

BDSI




A
2nd Entry of VECT = 'CALL DECD'
5th Entry of VECT = 'CALL SPIN'
7th Entry of VECT = 'CALL BITE'
14th Entry of VECT = GASC
SODL(I) = SODLIN(I) I = 1, 64
RETURN

Initialize for Alignment (First Entry)

FENT

$$\begin{bmatrix} A \\ 3\times3 \end{bmatrix} = \begin{bmatrix} SDVI \\ SDVJ \\ SDVK \end{bmatrix} = \begin{bmatrix} VAXI \\ VAYJ \\ VAZK \end{bmatrix} = \begin{bmatrix} DVXI \\ DVYJ \\ DVZK \end{bmatrix} = \begin{bmatrix} DVXG \\ DVYG \\ DVZG \end{bmatrix} = 0$$

SRTI = SRT2 = RATP = RATM = CHA1 = 0

DCON = -8

$$\begin{bmatrix} GM \\ 3\times3 \end{bmatrix} = \begin{bmatrix} CD19 & CD20 & CD21 \\ -CD25 & -CD26 & -CD27 \\ CD22 & CD23 & CD24 \end{bmatrix}$$

A(1, 1) = CD01
A(2, 2) = CD02
A(3, 3) = CD03

$$\begin{bmatrix} AB \\ 3\times3 \end{bmatrix} = \begin{bmatrix} CD07 & CD10 & CD13 \\ CD08 & CD11 & CD14 \\ CD09 & CD12 & CD15 \end{bmatrix} \begin{bmatrix} A \\ 3\times3 \end{bmatrix}$$

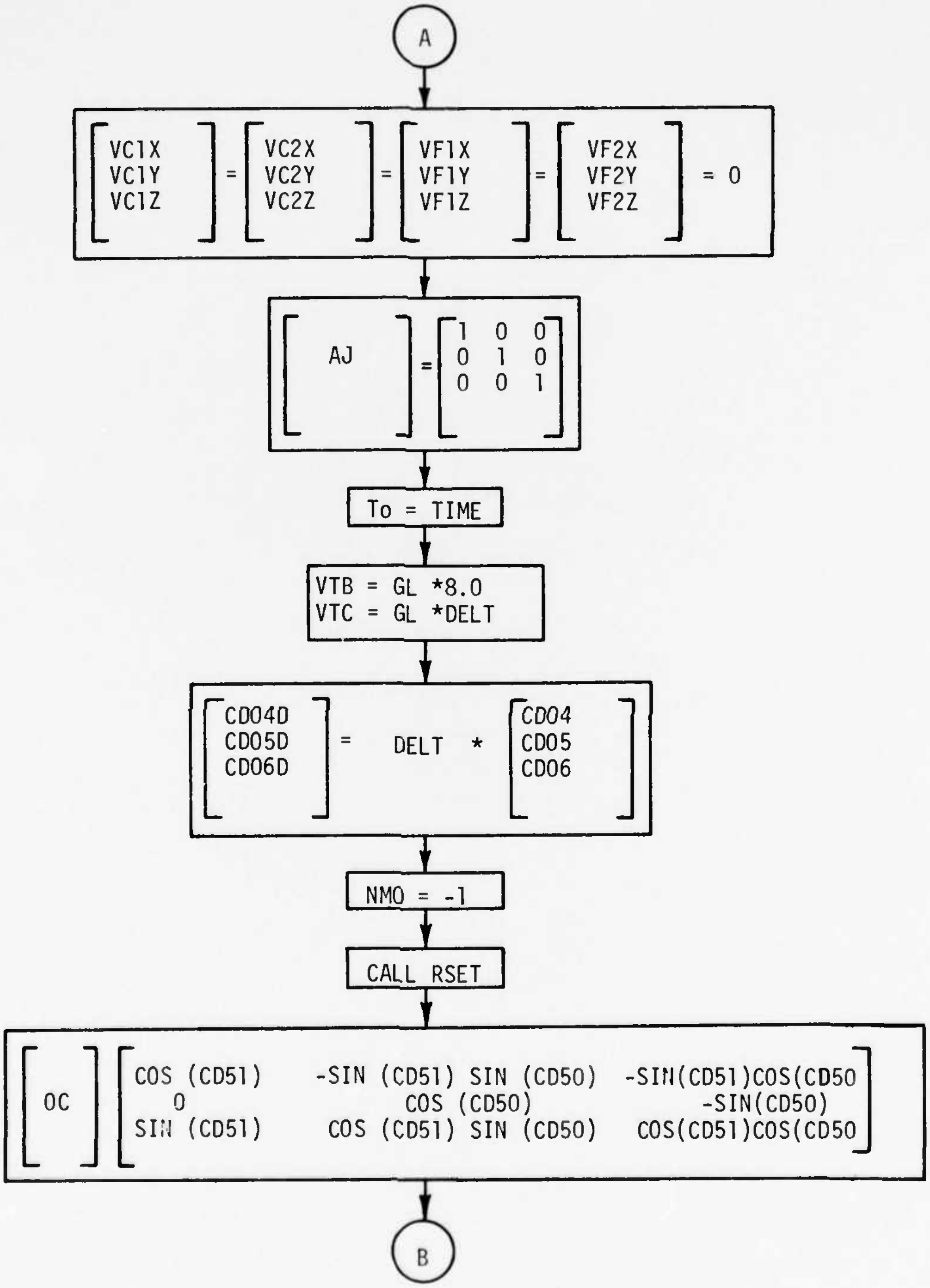
LAT = LATL

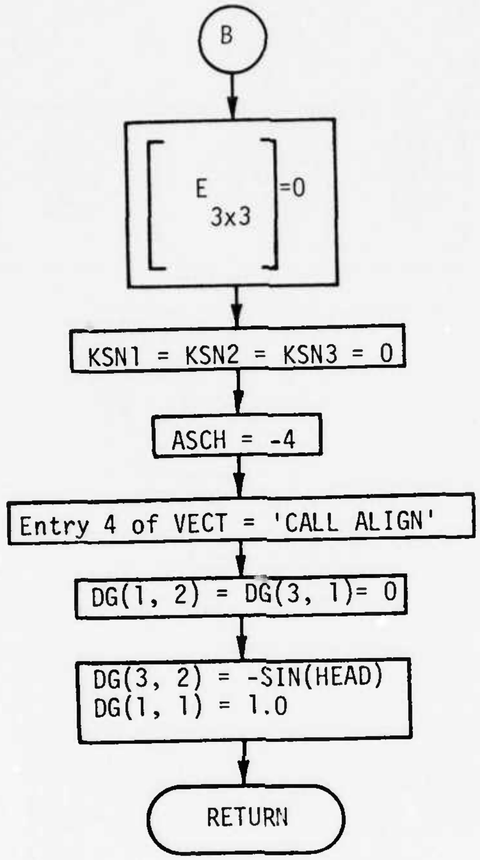
SGDL = SIN (LAT)
CGDL = COS (LAT)

AK1T = CGDL *GL
AK2T = SGDL *GL

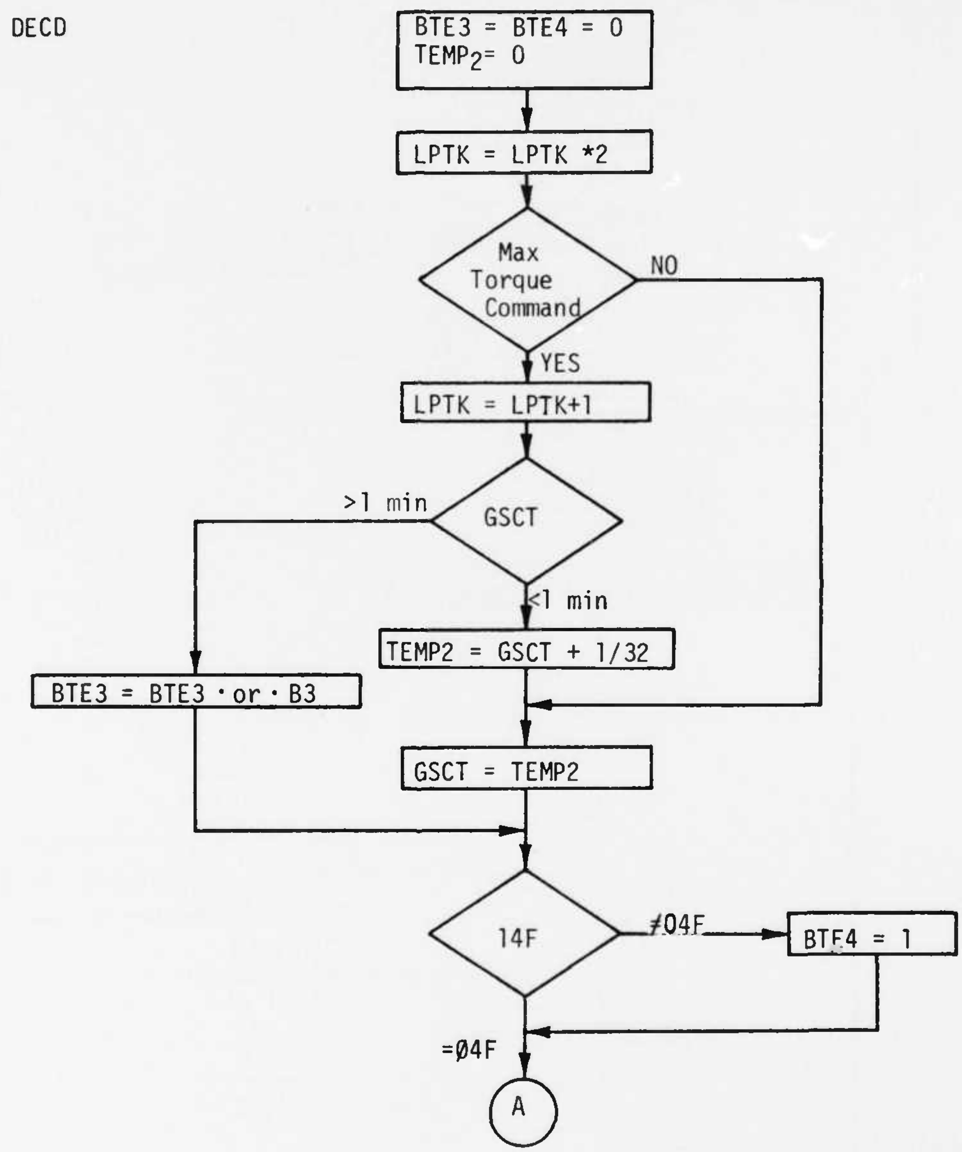
PHA = 0

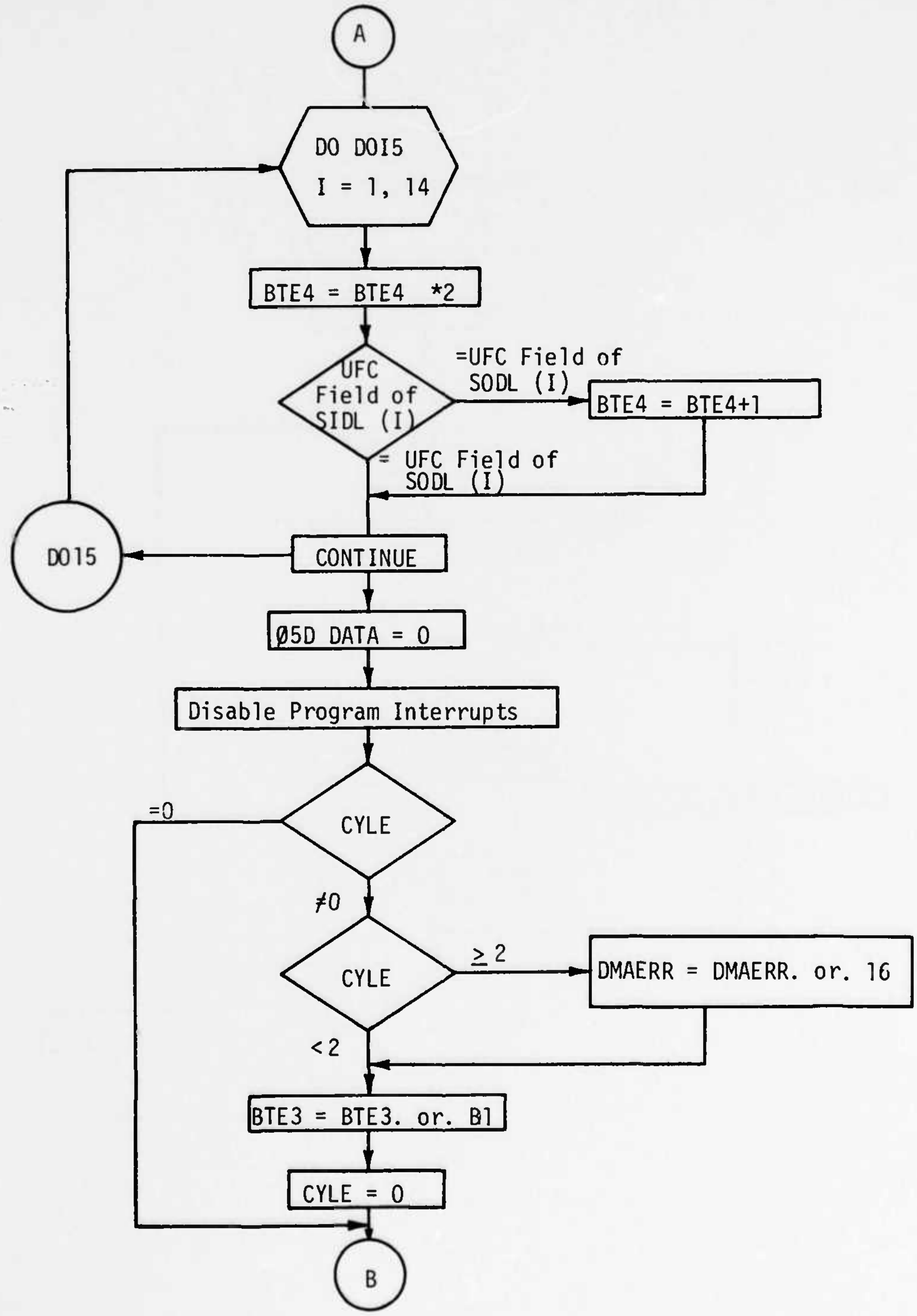
A

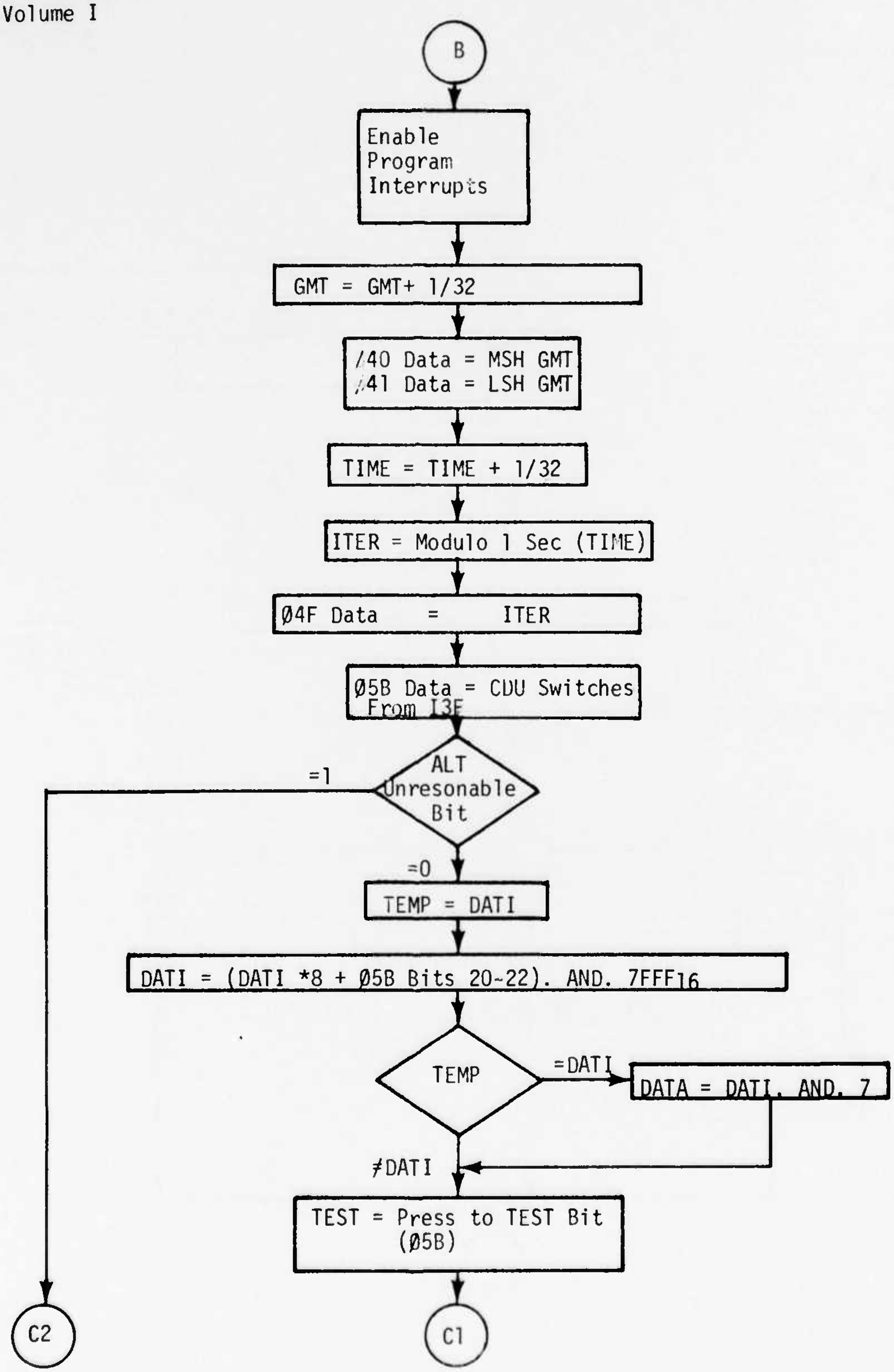

A
[VC1X VC1Y VC1Z] = [VC2X VC2Y VC2Z] = [VF1X VF1Y VF1Z] = [VF2X VF2Y VF2Z] = 0
[AJ] = [1 0 0; 0 1 0; 0 0 1]
To = TIME
VTB = GL *8.0
VTC = GL *DELT
[CD04D CD05D CD06D] = DELT * [CD04 CD05 CD06]
NMO = -1
CALL RSET
[OC] [COS (CD51) -SIN (CD51) SIN (CD50) -SIN(CD51)COS(CD50
0 COS (CD50) -SIN(CD50)
SIN (CD51) COS (CD51) SIN (CD50) COS(CD51)COS(CD50]
B

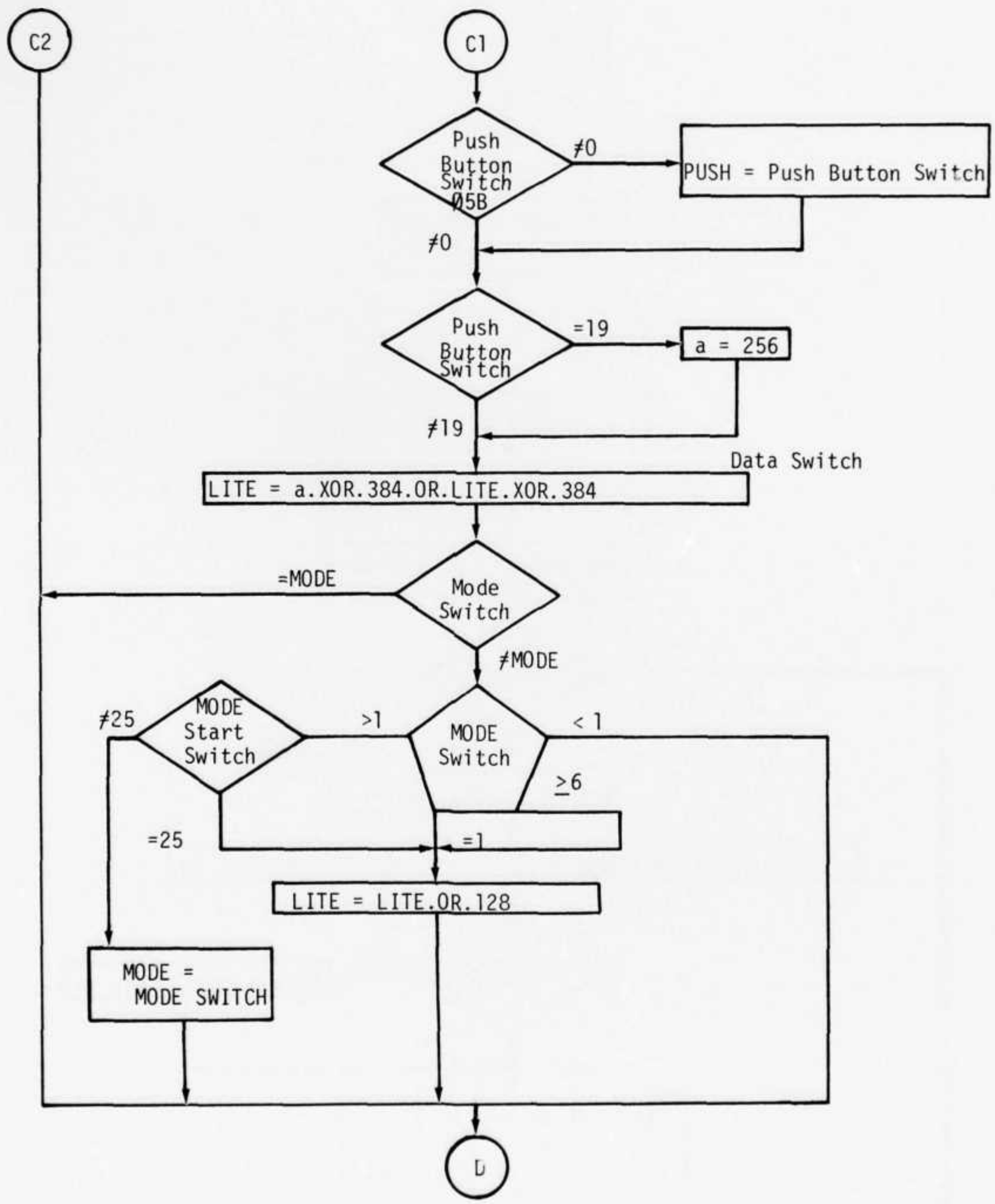

B
$\begin{bmatrix} E_{3x3} \end{bmatrix} = 0$
KSN1 = KSN2 = KSN3 = 0
ASCH = -4
Entry 4 of VECT = 'CALL ALIGN'
DG(1, 2) = DG(3, 1)= 0
DG(3, 2) = -SIN(HEAD)
DG(1, 1) = 1.0
RETURN

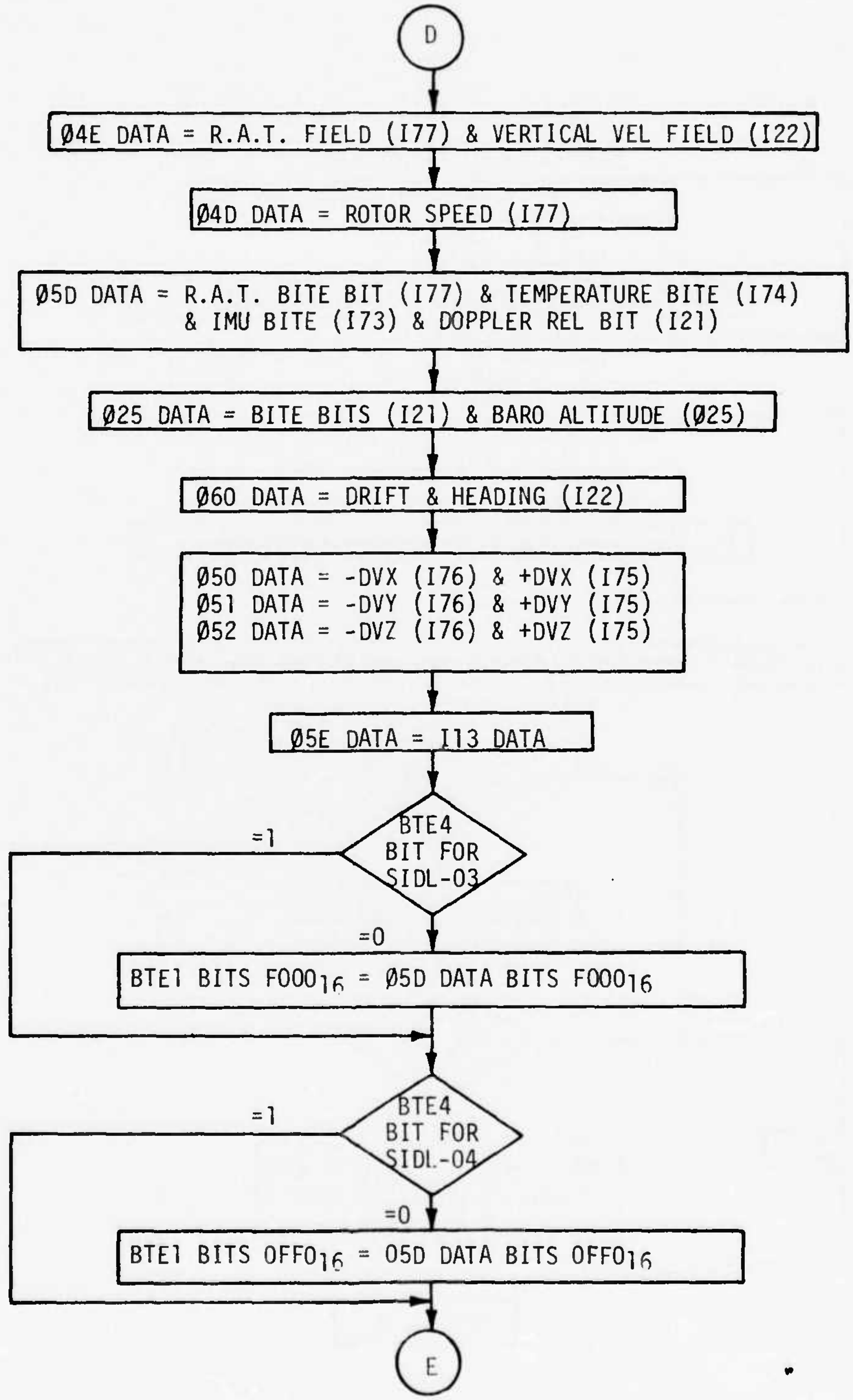
Decode SIDL

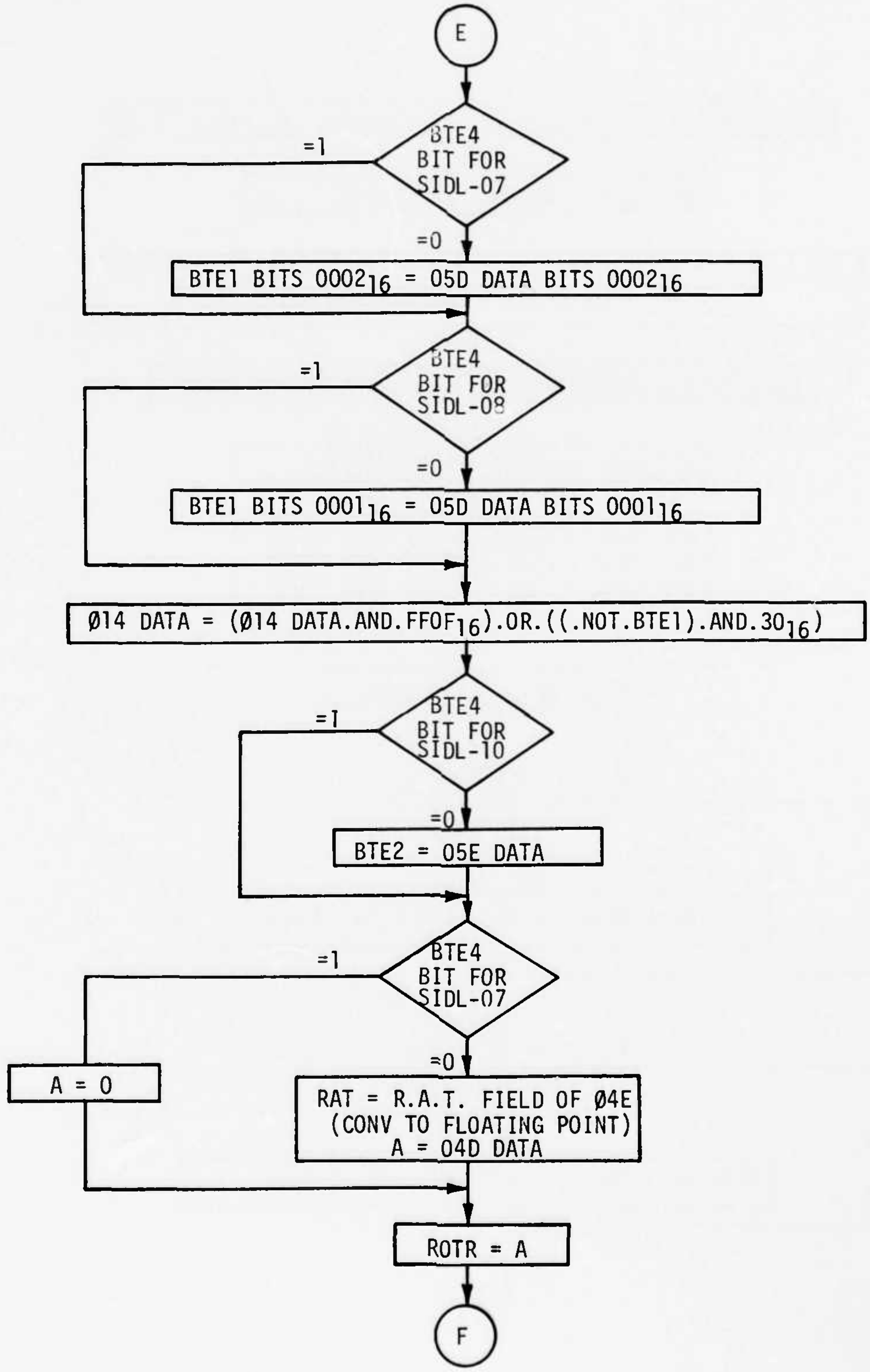

DECD
BTE3 = BTE4 = 0
TEMP2= 0
LPTK = LPTK *2
Max Torque Command
NO
YES
LPTK = LPTK+1
GSCT
>1 min
<1 min
TEMP2 = GSCT + 1/32
BTE3 = BTE3 · or · B3
GSCT = TEMP2
14F
≠04F
BTE4 = 1
=Ø4F
A

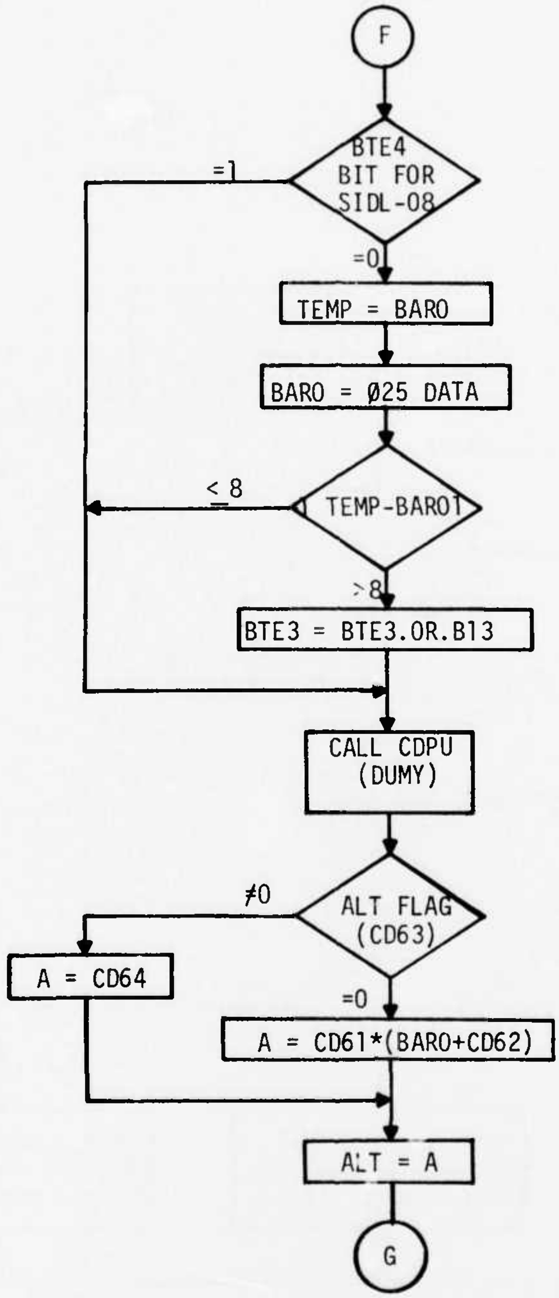

A
DO DOI5
I = 1, 14
BTE4 = BTE4 *2
UFC Field of SIDL (I)
=UFC Field of SODL (I)
BTE4 = BTE4+1
= UFC Field of SODL (I)
D015
CONTINUE
Ø5D DATA = 0
Disable Program Interrupts
=0
CYLE
≠0
CYLE
≥ 2
DMAERR = DMAERR. or. 16
< 2
BTE3 = BTE3. or. B1
CYLE = 0
B

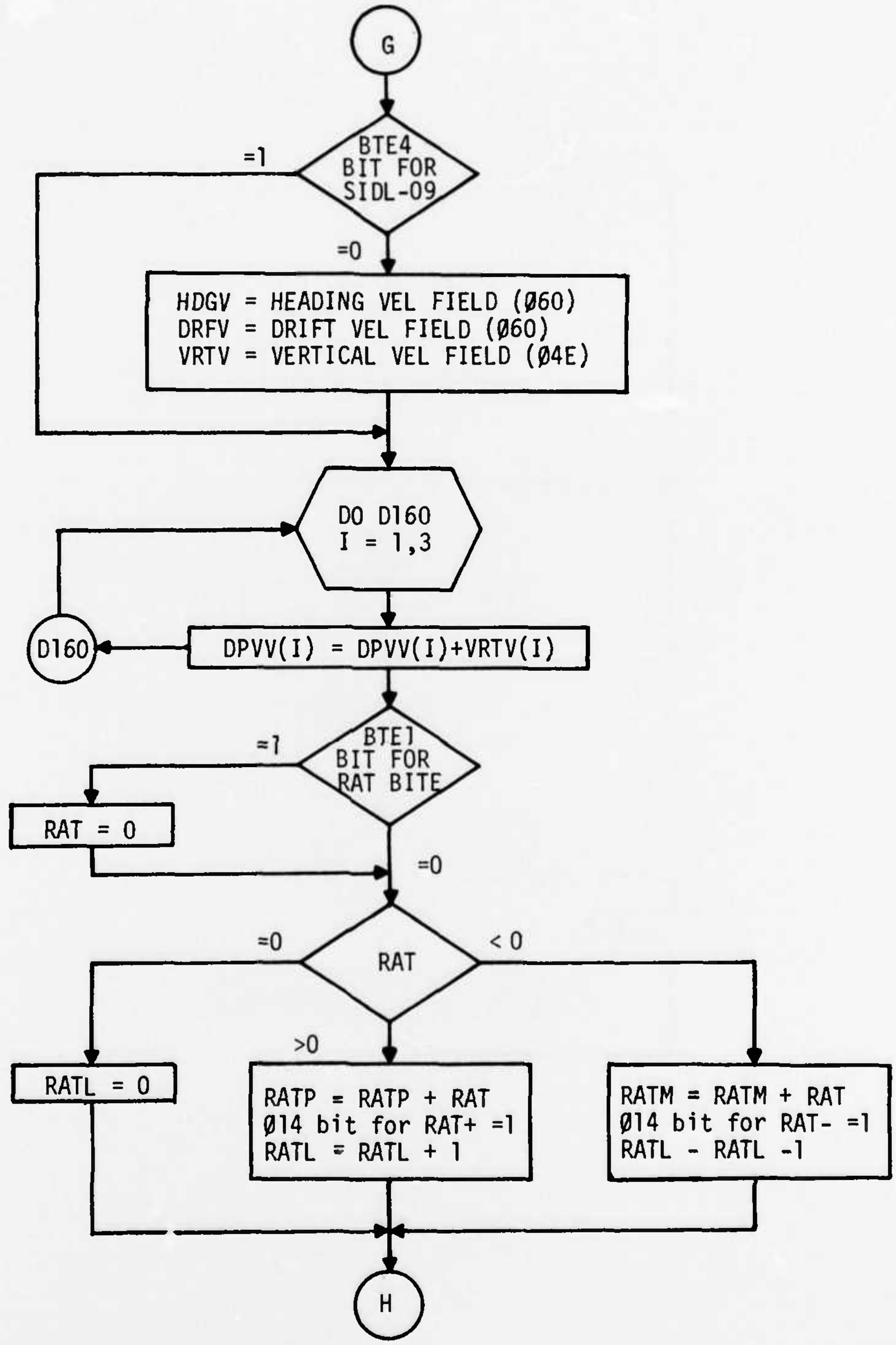

B
Enable Program Interrupts
GMT = GMT+ 1/32
/40 Data = MSH GMT
/41 Data = LSH GMT
TIME = TIME + 1/32
ITER = Modulo 1 Sec (TIME)
Ø4F Data = ITER
Ø5B Data = CDU Switches From I3E
ALT Unresonable Bit
=1
=0
TEMP = DATI
DATI = (DATI *8 + Ø5B Bits 20-22). AND. 7FFF16
TEMP
=DATI
DATA = DATI. AND. 7
≠DATI
TEST = Press to TEST Bit (Ø5B)
C2
C1

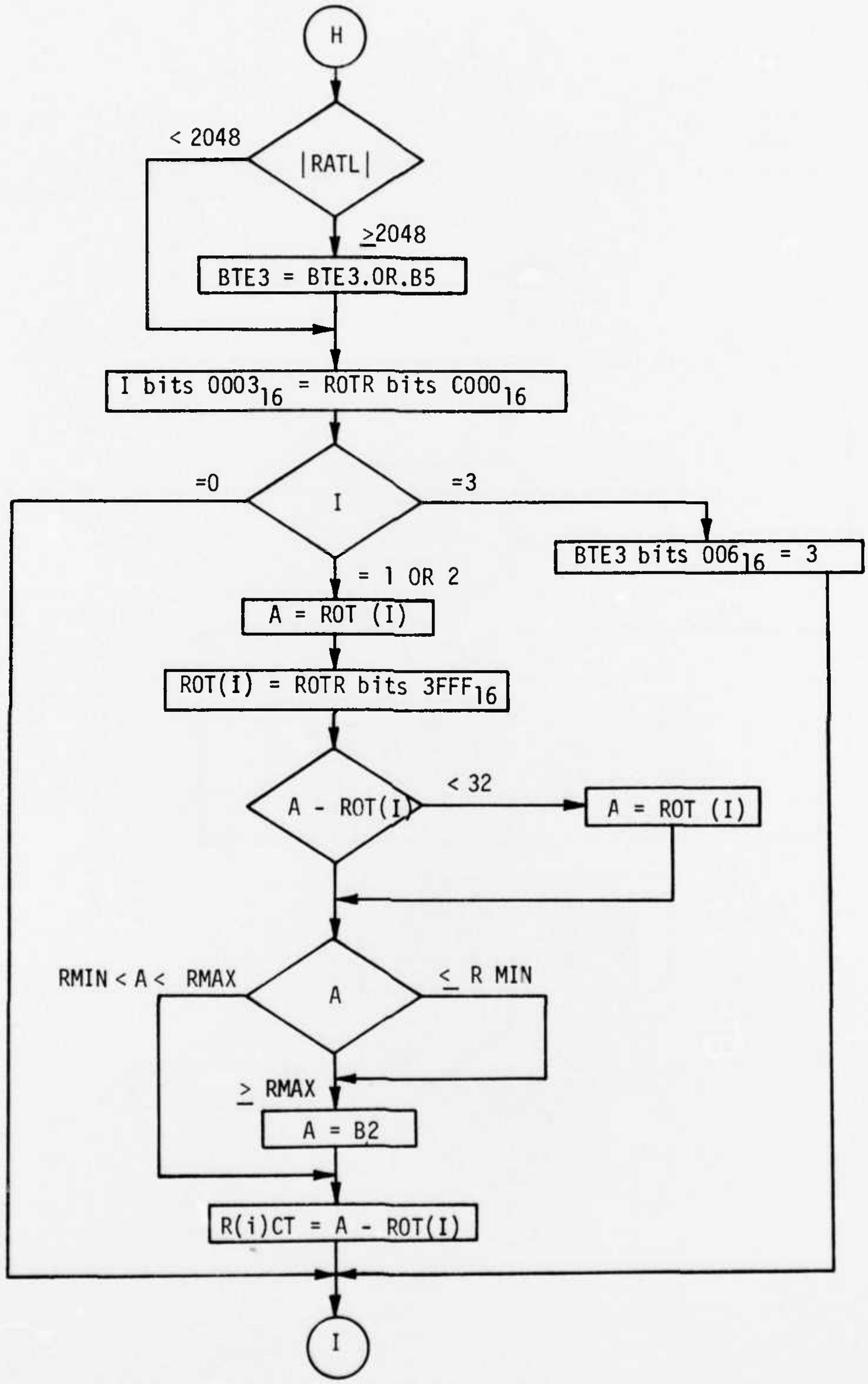

C2
C1
Push Button Switch Ø5B
≠0
PUSH = Push Button Switch
≠0
Push Button Switch
=19
a = 256
≠19
Data Switch
LITE = a.XOR.384.OR.LITE.XOR.384
=MODE
Mode Switch
≠MODE
≠25
MODE Start Switch
>1
MODE Switch
< 1
≥6
=25
=1
LITE = LITE.OR.128
MODE = MODE SWITCH
D

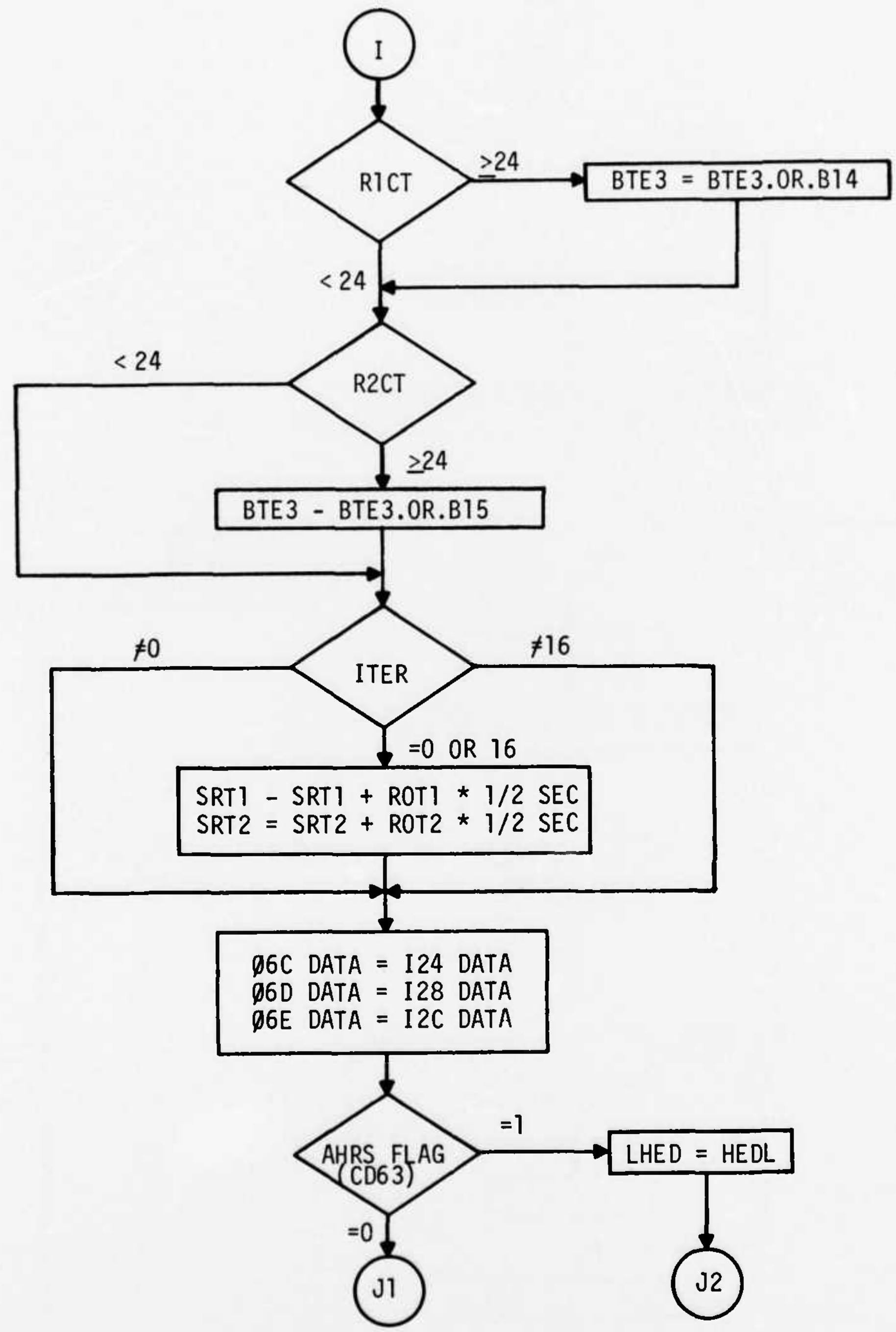

D
Ø4E DATA = R.A.T. FIELD (I77) & VERTICAL VEL FIELD (I22)
Ø4D DATA = ROTOR SPEED (I77)
Ø5D DATA = R.A.T. BITE BIT (I77) & TEMPERATURE BITE (I74) & IMU BITE (I73) & DOPPLER REL BIT (I21)
Ø25 DATA = BITE BITS (I21) & BARO ALTITUDE (Ø25)
Ø60 DATA = DRIFT & HEADING (I22)
Ø50 DATA = -DVX (I76) & +DVX (I75)
Ø51 DATA = -DVY (I76) & +DVY (I75)
Ø52 DATA = -DVZ (I76) & +DVZ (I75)
Ø5E DATA = I13 DATA
BTE4 BIT FOR SIDL-03
=1
=0
BTE1 BITS F000$_{16}$ = Ø5D DATA BITS F000$_{16}$
BTE4 BIT FOR SIDL-04
=1
=0
BTE1 BITS 0FF0$_{16}$ = 05D DATA BITS 0FF0$_{16}$
E

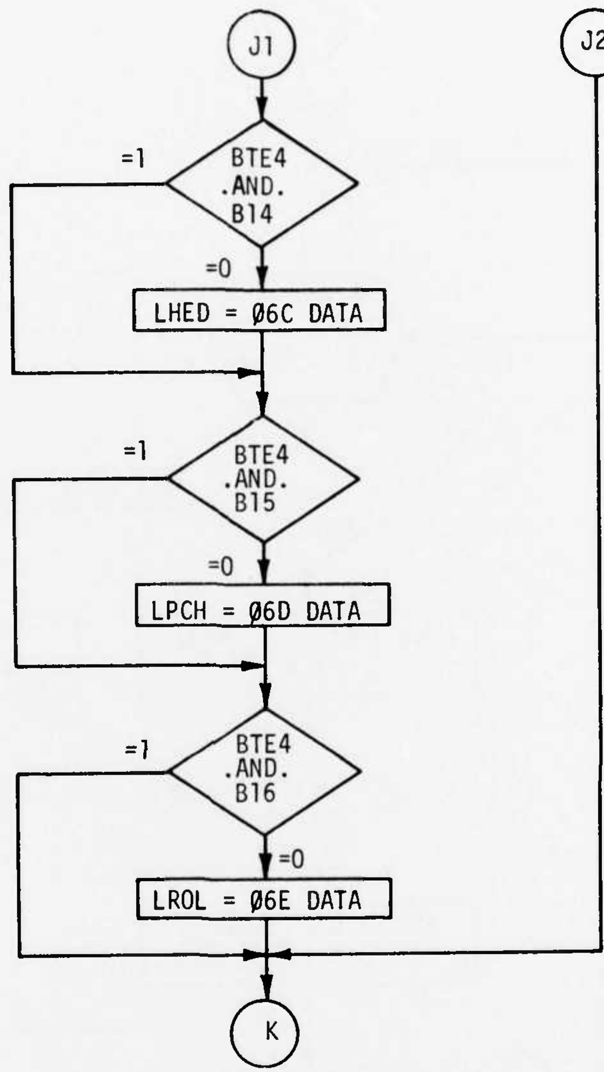

E
BTE4 BIT FOR SIDL-07
=1
=0
BTE1 BITS 0002₁₆ = 05D DATA BITS 0002₁₆
BTE4 BIT FOR SIDL-08
=1
=0
BTE1 BITS 0001₁₆ = 05D DATA BITS 0001₁₆
Ø14 DATA = (Ø14 DATA.AND.FF0F₁₆).OR.((.NOT.BTE1).AND.30₁₆)
BTE4 BIT FOR SIDL-10
=1
=0
BTE2 = 05E DATA
BTE4 BIT FOR SIDL-07
=1
=0
A = 0
RAT = R.A.T. FIELD OF Ø4E
(CONV TO FLOATING POINT)
A = 04D DATA
ROTR = A
F

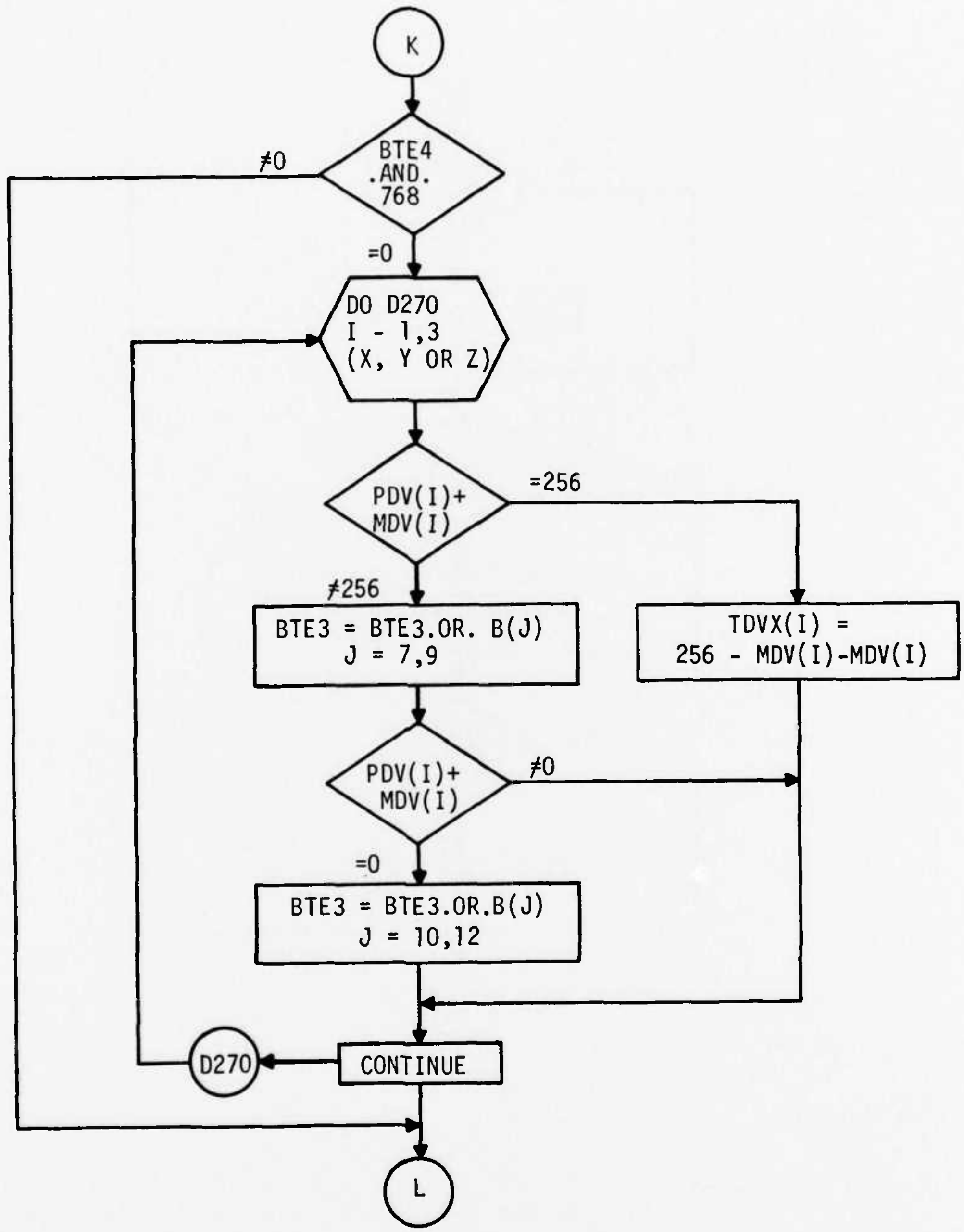

F
BTE4 BIT FOR SIDL-08
=1
=0
TEMP = BARO
BARO = Ø25 DATA
TEMP-BARO
≤ 8
>8
BTE3 = BTE3.OR.B13
CALL CDPU (DUMY)
ALT FLAG (CD63)
≠0
A = CD64
=0
A = CD61*(BARO+CD62)
ALT = A
G

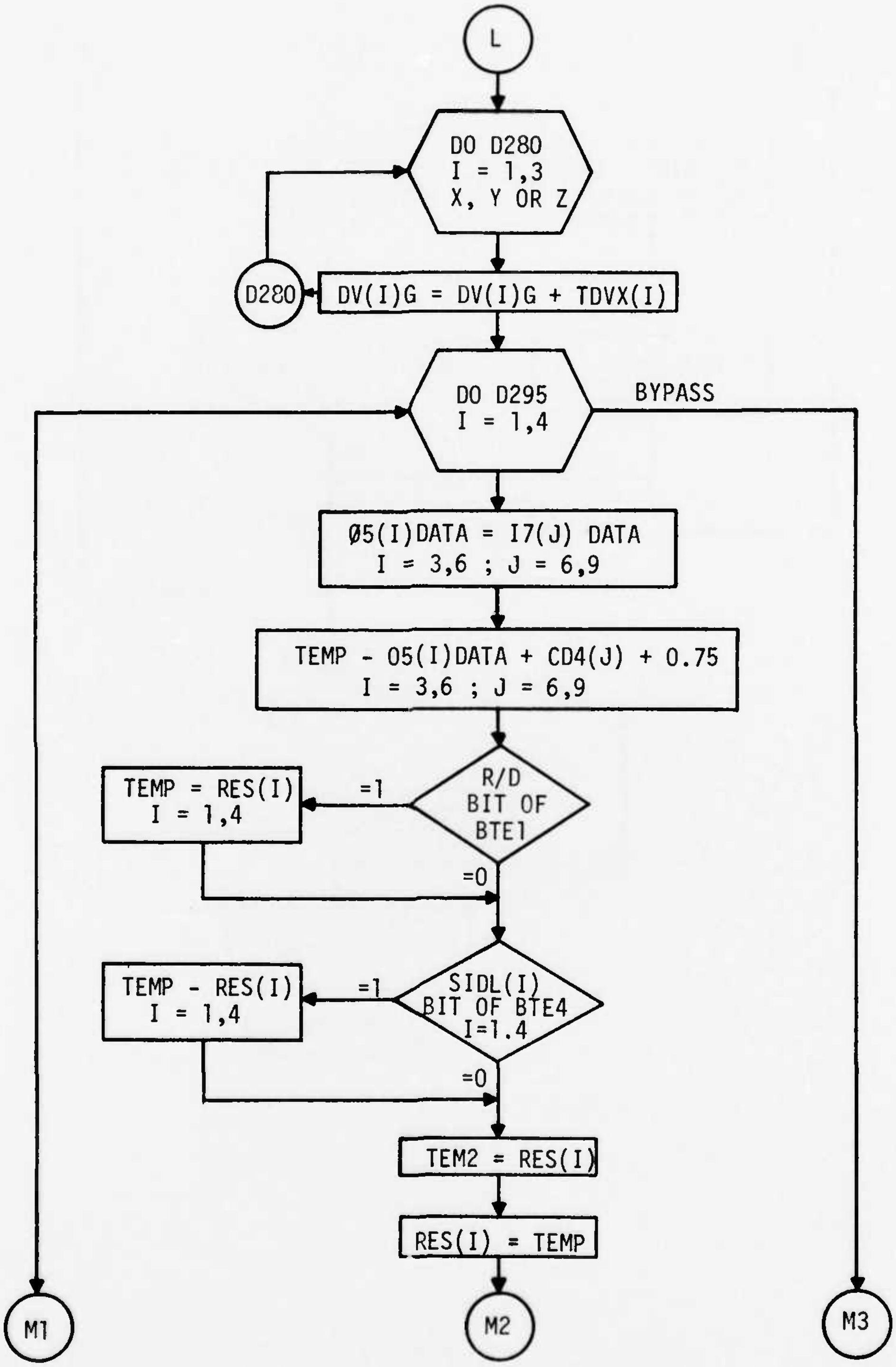

G
BTE4 BIT FOR SIDL-09
=1
=0
HDGV = HEADING VEL FIELD (Ø60)
DRFV = DRIFT VEL FIELD (Ø60)
VRTV = VERTICAL VEL FIELD (Ø4E)
DO D160 I = 1,3
D160
DPVV(I) = DPVV(I)+VRTV(I)
BTE1 BIT FOR RAT BITE
=1
RAT = 0
=0
RAT
=0
< 0
>0
RATL = 0
RATP = RATP + RAT
Ø14 bit for RAT+ =1
RATL = RATL + 1
RATM = RATM + RAT
Ø14 bit for RAT- =1
RATL - RATL -1
H

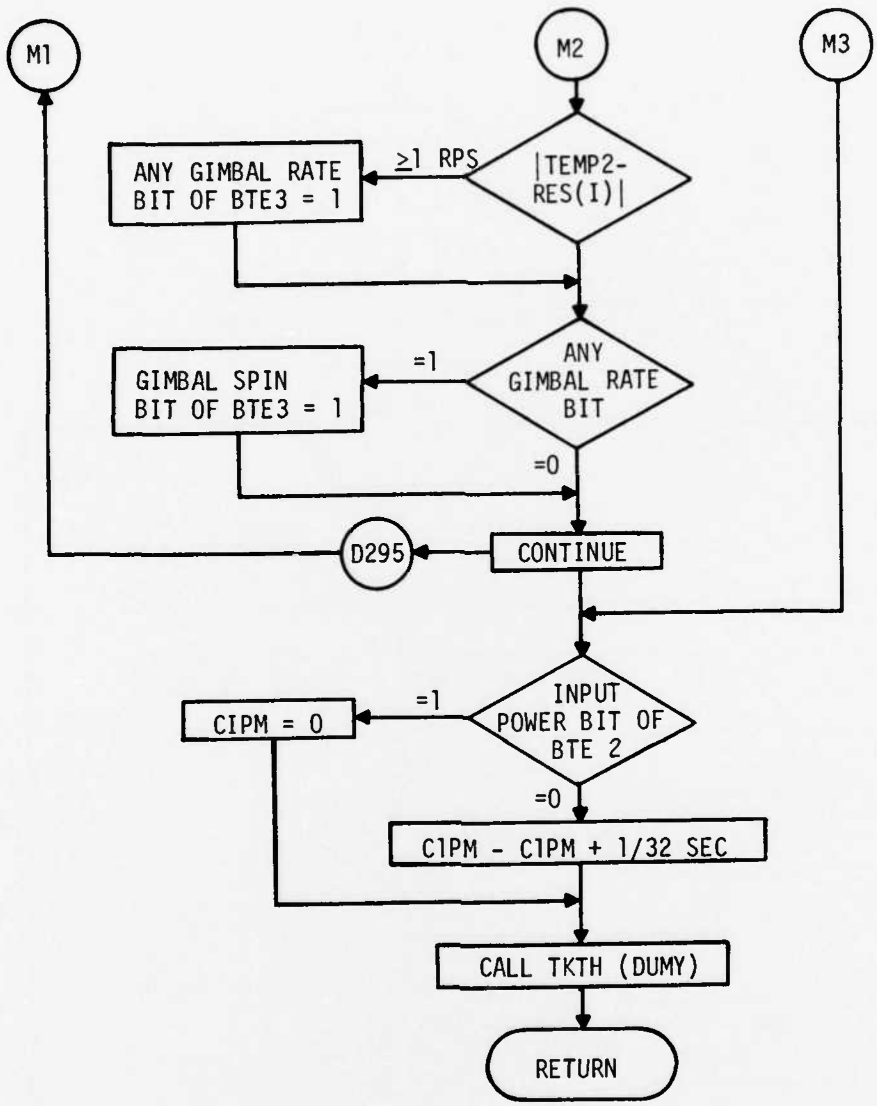

H
|RATL|
< 2048
≥2048
BTE3 = BTE3.OR.B5
I bits 0003₁₆ = ROTR bits C000₁₆
I
=0
=3
= 1 OR 2
BTE3 bits 006₁₆ = 3
A = ROT (I)
ROT(I) = ROTR bits 3FFF₁₆
A - ROT(I)
< 32
A = ROT (I)
A
RMIN < A < RMAX
≤ R MIN
≥ RMAX
A = B2
R(i)CT = A - ROT(I)
I


I
R1CT
>24
BTE3 = BTE3.OR.B14
< 24
R2CT
< 24
>24
BTE3 - BTE3.OR.B15
ITER
≠0
≠16
=0 OR 16
SRT1 - SRT1 + ROT1 * 1/2 SEC
SRT2 = SRT2 + ROT2 * 1/2 SEC
Ø6C DATA = I24 DATA
Ø6D DATA = I28 DATA
Ø6E DATA = I2C DATA
AHRS FLAG (CD63)
=1
LHED = HEDL
=0
J1
J2

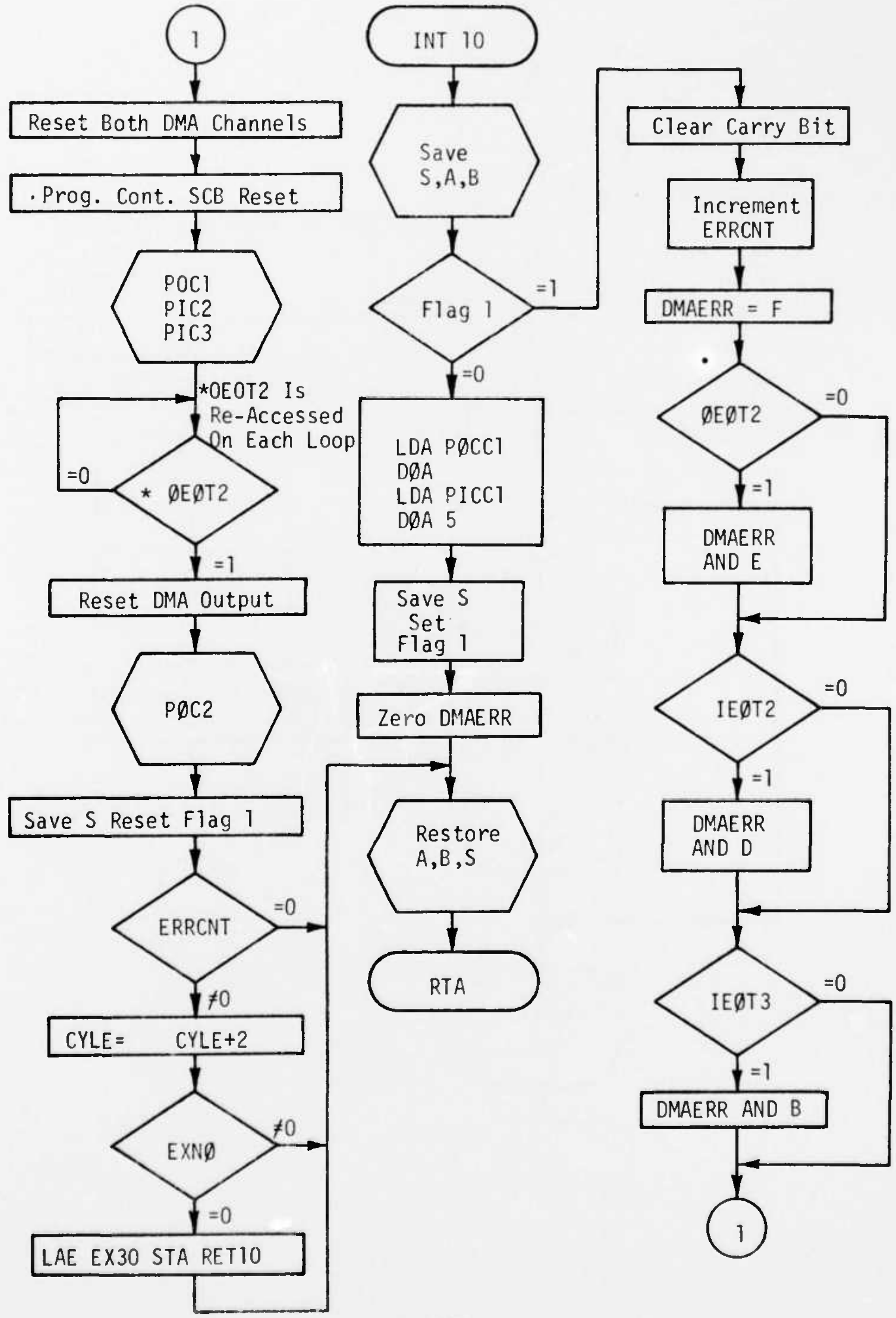

J1
J2
=1
BTE4 .AND. B14
=0
LHED = Ø6C DATA
=1
BTE4 .AND. B15
=0
LPCH = Ø6D DATA
=1
BTE4 .AND. B16
=0
LROL = Ø6E DATA
K


K
BTE4 .AND. 768
≠0
=0
DO D270
I - 1,3
(X, Y OR Z)
PDV(I)+ MDV(I)
=256
≠256
BTE3 = BTE3.OR. B(J)
J = 7,9
TDVX(I) =
256 - MDV(I)-MDV(I)
PDV(I)+ MDV(I)
≠0
=0
BTE3 = BTE3.OR.B(J)
J = 10,12
D270
CONTINUE
L

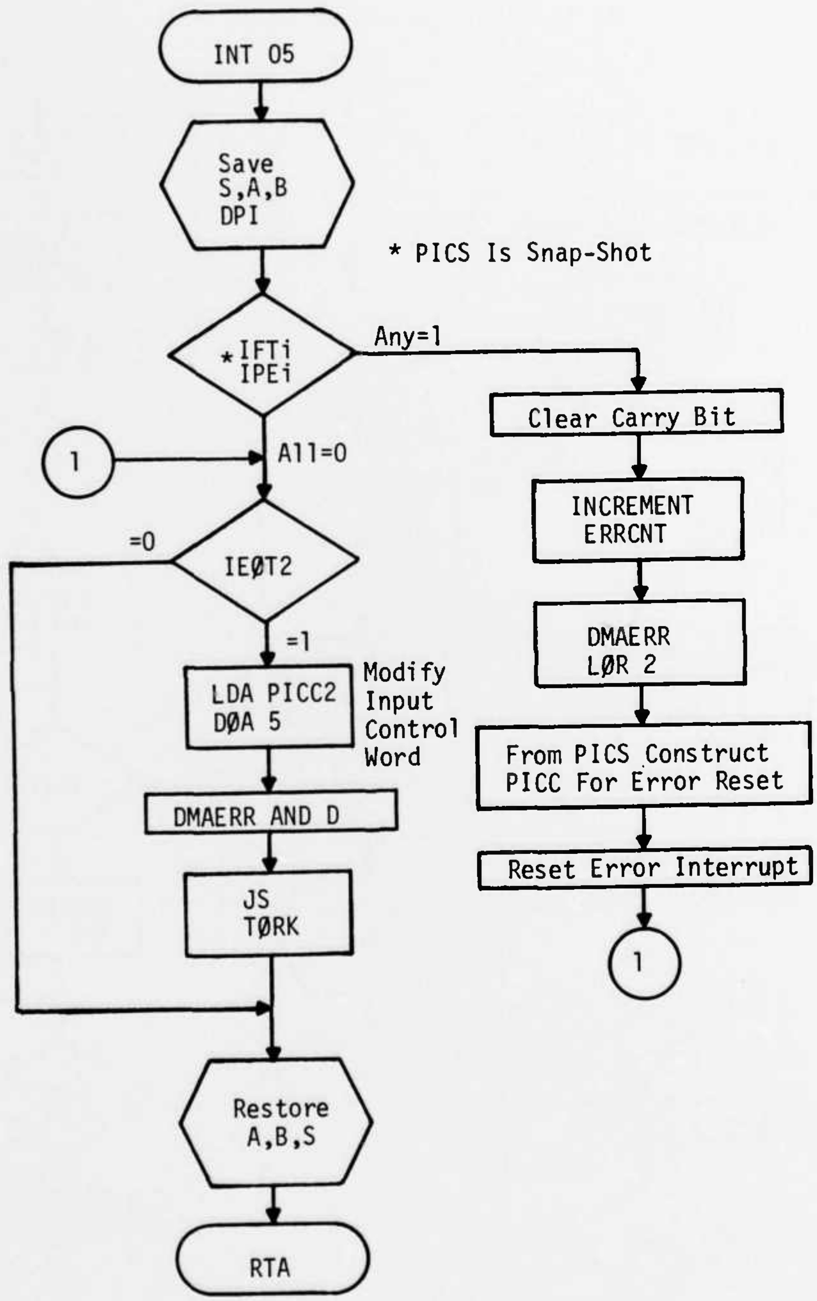

L
DO D280
I = 1,3
X, Y OR Z
D280
DV(I)G = DV(I)G + TDVX(I)
DO D295
I = 1,4
BYPASS
Ø5(I)DATA = I7(J) DATA
I = 3,6 ; J = 6,9
TEMP - 05(I)DATA + CD4(J) + 0.75
I = 3,6 ; J = 6,9
R/D
BIT OF
BTE1
=1
TEMP = RES(I)
I = 1,4
=0
SIDL(I)
BIT OF BTE4
I=1.4
=1
TEMP - RES(I)
I = 1,4
=0
TEM2 = RES(I)
RES(I) = TEMP
M1
M2
M3


M1
M2
M3
ANY GIMBAL RATE BIT OF BTE3 = 1
≥1 RPS
|TEMP2-RES(I)|
GIMBAL SPIN BIT OF BTE3 = 1
=1
ANY GIMBAL RATE BIT
=0
D295
CONTINUE
CIPM = 0
=1
INPUT POWER BIT OF BTE 2
=0
CIPM - CIPM + 1/32 SEC
CALL TKTH (DUMY)
RETURN

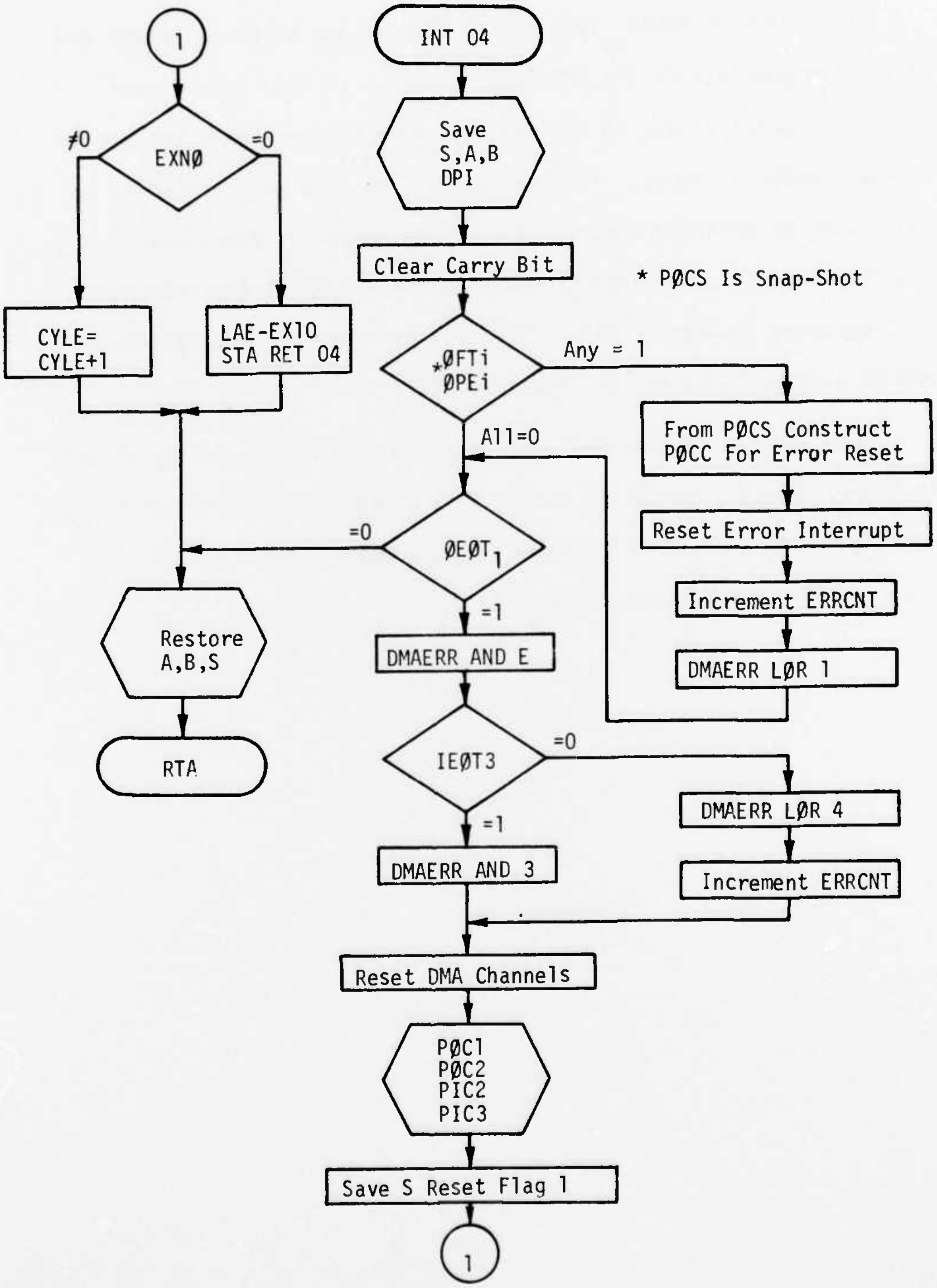
INTERRUPT 10 ROUTINE [32 HZ]



INTERRUPT 5 ROUTINE



INTERRUPT 4 ROUTINE



APPENDIX C

TIMING DATA FOR PROGRAM SEGMENTS

The following table gives timing data on the GEANS Alignment and Navigation routines in the SKC-2000 computer. A test program was written in which a call to the routine under test was made repeatedly a fixed number of times. Then the actual run time of the routine was calculated by dividing elapsed time by the number of executions. Elapsed time was determined by counting the number of 5ms interrupts that occurred during the test. The test was repeated several times and an average taken for the final result.

Each routine in the math subroutine library uses a memory stack to store temporary variables during it's execution. The timing tests were done with this stack resident in core memory and in LSI (Large Scale Integrated circuit) memory.

ROUTINE	AVERAGE EXECUTION TIME IN MILLISECONDS	
	STACK IN CORE	STACK IN LSI
IC (Accel Bias & Scale Factor)	1.733	1.436
ID (Transform Accel to Gyro Co-ordinates)	1.723	1.44
IE (Gravity Model)	3.323	2.737
IF (Vertical Damping & DR)	1.125	0.913
IG (Double Integration for Position	1.481	1.253
IH (Lat & Long Computation)	6.757	5.509
IJ (Local Level Co-ordinates & Ground Speed)	10.149	8.484
IL (Drift Compensation)	12.673	10.458
IM (Update AS & SA Matricies)	9.07	7.283
RTAL (Return to Align Decision)	8.82	7.54
FENT (First Entry)	8.446	7.202
IID (Low Pass Filter)	4.425	3.799
IIH (Reference Profile)	3.372	2.76
IIK (EPA Solution)	1.824	1.507
IIM (local Level Solution)	1.843	1.513
IIO (Least Squares Solution	8.045	6.575
IIP (Compute ΔA & ΔAJ Matricies)	19.613	15.906
IIR (Go to Nav Decision)	6.064	5.142
IA (NAV Sub-executive)		
Branch 1	22.351	17.781
Branch 2	28.351	22.060
Branch 3	10.698	14.246
Branch 4	27.692	34.384
IIA (ALIGN Sub-executive)		
Branch 1	5.452	4.494
Branch 2	11.649	9.547
Branch 3	10.670	8.70
Branch 4	9.077	7.496

BIBLIOGRAPHY

1. PDP-11 Paper Tape Software Programming Handbook
   Digital Equipment Corp., Maynard, Mass. 1970

2. Special Electronics Assembly Board System Manual
   Duane Hague, AFAL/RWA-4 April 1975

3. AN/ASN-101 Optimization Program Interim Report
   Technical Report AFAL-TR-2043-IT-2
   Russ Sittloh, et al Honeywell, Inc. Jan 1973

4. Computer Program Development Specification
   for the Standard Precision Navigator/GEANS
   Inertial Navigation System (Configuration Item
   Number DQG8156A1) September 1974

☆ U. S. GOVERNMENT PRINTING OFFICE: 1977 — 757-001/436