GEANS SOFTWARE ANALYSIS DEVELOPMENT AND VALIDATION

Reference Systems Branch
Reconnaissance and Weapon Delivery Division

August 1977

TECHNICAL REPORT AFAL-TR-77-119
FINAL REPORT FOR PERIOD NOVEMBER 1975 TO NOVEMBER 1976

Approved for public release; distribution unlimited.

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 (OI) 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
Project Engineer

JAMES E. BARNES
Project Engineer

FOR THE COMMANDER

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

"If your address has changed, if you wish to be removed from our mailing list, or if the addressee is no longer employed by your organization please notify AFAL/RWA, W-PAFB, OH 45433 to help us maintain a current mailing list".

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

<table>
<thead>
<tr>
<th>REPORT NUMBER</th>
<th>2. GOVT ACCESSION NO.</th>
<th>3. RECIPIENT'S CATALOG NUMBER</th>
</tr>
</thead>
<tbody>
<tr>
<td>AFAL-TR-77-119</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>4. TITLE (and Subtitle)</th>
<th>5. TYPE OF REPORT &amp; PERIOD COVERED</th>
</tr>
</thead>
<tbody>
<tr>
<td>GEANS SOFTWARE ANALYSIS DEVELOPMENT AND VALIDATION</td>
<td>Final Report Nov 75 - Nov 76</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>6. AUTHOR(s)</th>
<th>7. PREPARATION DATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>William Mikulski James E. Barnes</td>
<td>Aug 77</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>8. CONTRACT OR GRANT NUMBER(s)</th>
<th>9. PERFORMING ORGANIZATION NAME AND ADDRESS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Air Force Avionics Laboratory (AFAL/RWA-3)</td>
</tr>
<tr>
<td></td>
<td>Air Force Wright Aeronautical Laboratories</td>
</tr>
<tr>
<td></td>
<td>AFSC, Wright-Patterson AFB, Ohio 45433</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>10. PROGRAM ELEMENT PROJECT, TASK AREA &amp; WORK UNIT NUMBERS</th>
<th>11. CONTROLLING OFFICE NAME AND ADDRESS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project 19277203</td>
<td>Same as block 9 above</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>12. REPORT DATE</th>
<th>13. SECURITY CLASS. (OF THIS REPORT)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aug 77</td>
<td>UNCLASSIFIED</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>14. MONITORING AGENCY NAME &amp; ADDRESS (IF DIFFERENT FROM CONTROLLING OFFICE)</th>
<th>15. SECURITY CLASS. (OF ABSTRACT)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>UNCLASSIFIED</td>
</tr>
</tbody>
</table>

16. DISTRIBUTION STATEMENT (OF THIS REPORT)

Approved for public release; distribution unlimited.

<table>
<thead>
<tr>
<th>17. DISTRIBUTION STATEMENT (OF ABSTRACT ENTERED IN BLOCK 20, IF DIFFERENT FROM REPORT)</th>
<th>18. SUPPLEMENTARY NOTES</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Computer program
High order language
Assembly language

<table>
<thead>
<tr>
<th>20. ABSTRACT (Continue on reverse side if necessary and identify by block number)</th>
</tr>
</thead>
<tbody>
<tr>
<td>The GEANS Software Analysis Development and Validation effort consisted of writing an existing assembly language program in a high order language (HOL) and then comparing the two-load modules for memory requirements and timing. This comparison showed that the program as written in the HOL required more memory and time than the same program as written in assembly language.</td>
</tr>
</tbody>
</table>

DD FORM 1473 EDITION OF 1 NOV 65 IS OBSOLETE

SECURITY CLASSIFICATION OF THIS PAGE (When Data Engr.More Than 6 mo. Before Work Done)

017 670
FOREWORD

This report was prepared by William Mikulski and James E. Barnes 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 19270203. The report covers effort during the period November 1975 to November 1976.
TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>SECTION</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>1</td>
</tr>
<tr>
<td>II</td>
<td>2</td>
</tr>
<tr>
<td>III</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>5</td>
</tr>
<tr>
<td>IV</td>
<td>7</td>
</tr>
<tr>
<td>V</td>
<td>10</td>
</tr>
<tr>
<td>VI</td>
<td>12</td>
</tr>
<tr>
<td>APPENDIX FLOWCHARTS</td>
<td>15</td>
</tr>
<tr>
<td>TABLE</td>
<td>PAGE</td>
</tr>
<tr>
<td>-------</td>
<td>------</td>
</tr>
<tr>
<td>1 Memory Requirements</td>
<td>7</td>
</tr>
<tr>
<td>2 Comparison of GEANS timing in J3B and FOCAP</td>
<td>9</td>
</tr>
</tbody>
</table>
SECTION I
INTRODUCTION

In May 1973 AFAL/RWA-3 undertook the task of converting the Honeywell Gimballed Electrostatic Gyro Navigation System (GEANS) computer program, which was hosted on the Honeywell HDC-601 computer, to run on the Singer/Kearfott SKC-2000 computer. Both the original and converted programs were written in the assembly language of the host computer. Thus, the conversion effort was an assembly language to assembly language conversion, and was completed in December 1975. This effort is described in Technical Report AFAL-TR 77-8, Vol 1 & 2, Conversion of Computer Software for the Gimballed Electrostatic Gyro Navigation System, November 1976.

GEANS is intended for a variety of applications using any of several available computers. The required reprogramming will be most rapidly done using Higher Order Languages (HOL), if a suitable HOL is available. Good HOL's increase programmer productivity, provide readable programs, and, for some languages, allow the same HOL source program to be used on several computers.

The purpose of the investigation described in this report was to evaluate the problems and penalties of using an avionics-oriented HOL to reprogram the GEANS software. The available time and resources limited the effort to applying a readily available HOL, J3B level 0, to only the Alignment and Navigation portions of the GEANS Software. These portions were converted and the resulting computer time and memory required compared with those of the assembly language code.
SECTION II
BACKGROUND OF JOVIAL/J3B

The compiler used was the JOVIAL-3B/O compiler written by Softech Inc. and hosted on the IBM-370/155 at the WPAFB computer center. Following is a short history of JOVIAL-3B:

When the Request for Proposal for the B-1 bomber system was prepared at Aeronautical Systems Division (ASD), it was decided to require the programming of the mission software to be done in a HOL. In October 1971 JOVIAL-73 was selected as the language for the B-1. However the JOVIAL-73 language was not ready in time so in June 1972 a reduced capability HOL of low implementation risks which would satisfy the minimum needs of the B-1 until JOVIAL-73 became available was designed. To achieve low implementation risk, this language was a version of JOVIAL-5, which is a subset of JOVIAL-73. The language was designed by Boeing and three members of the JOVIAL-73 committee, namely RADC, ASD (the B-1 SPO), and ABACUS, Inc. Whenever possible the language was made to conform with what was known of JOVIAL-73 at the time.

The contract for a JOVIAL-3B compiler was finally awarded to a contractor (Softech) who did not use the JOVIAL-5 compiler as a baseline. Thus the need for staying close to JOVIAL-5 was eliminated. Since much more was known about the direction of JOVIAL-73 by the time the JOVIAL-3B specification was finished in October 1972 (by Boeing and Softech), it was very similar to JOVIAL-73.
Two compilers for JOVIAL-3B, one for the IBM-360 and one for the SKC-2000 were written in the Automated Engineering Design (AED) language and hosted on the IBM-360. They were delivered to Boeing in September 1973.

The original reduced capability version of JOVIAL-3B has since come to be termed JOVIAL-3B/O (J3B level 0). It has evolved into JOVIAL-3B/1 and, later, JOVIAL-3B/2. The level 2 compiler is the most current version, and can be targeted to the LC-4516D and M362F computers as well as to the SKC-2000 and IBM-370.
SECTION III
INTEGRATION OF THE HOL GEANS PROGRAM WITH EXISTING ASSEMBLY LANGUAGE PROGRAM

1. APPROACH

Coding of the GEANS software in JOVIAL/J3B was separated into three parts. These were functionally different and consisted of:

1) Navigation routines.
2) Alignment routines.
3) Matrix operation subroutines.

Since these were the only portions coded in J3B, it was necessary to link with the existing software written in the SKC-2000 assembly language (FOCAP). To minimize debugging time and effort, the navigation routines were coded in J3B almost line for line from the hand-coded routines. The alignment routines were written based on the flowcharts rather than on the FOCAP code itself. The FOCAP code was checked with the flowcharts to verify the accuracy of the details only. The matrix routines were written in J3B to minimize linkage problems which would have been encountered if the FOCAP written matrix routines were used.

The same labels were used in the J3B code as were used in the FOCAP code, which allowed for greater ease in both timing and debugging. The same data base was also used since the J3B written programs had to link with the hand-written FOCAP programs. Using the same data base caused some problems since the J3B generated code does not use FOCAP "common." This problem was alleviated by modifying the FOCAP-Coded data base to refer to the J3B defined data blocks.
2. PROBLEMS ENCOUNTERED

Since JOVIAL/J3B is target machine independent, it cannot generate input/output code. Because of this, all I/O must be done by handwritten FOCAP routines. The required routines for I/O already existed in the GEANS real-time executive. Therefore, to prevent the additional problem of debugging I/O routines, the J3B code was written to look like the hand-written FOCAP code as far as the executive (hand-written) was concerned. This was done by maintaining the same names for all routines which the executive referenced.

The executive code also had to be modified slightly since J3B generates code using "Page 3", which allows short instructions to be generated which access constants 0 and 1 from core. This required adding code in the executive to set the status word to indicate that a page 3 was in effect.

Another modification to the executive was required for the J3B linkage mechanization. J3B uses a push-down stack arrangement using registers 6 and 14 as stack pointers. The existing GEANS software used register 6 only for the same purpose. Therefore the executive had to be modified to initialize register 14 to the same as register 6 (i.e. to the top of the stack).

The existing math library written for the GEANS in the SKC-2000 had been previously proven to work properly and efficiently. Therefore, it was decided to use this library of math functions rather than the library supplied with the J3B compiler. The only drawback to this
method was that the job linkage and the linkage used by the FOCAP coded library were entirely different. This required additional code to convert from one calling convention and linkage structure to the other. The hand-written math library was written to be most efficient with the hand-written GEANS program. Rather than modify this library, it was decided to keep it intact and add a buffer stage of routines to convert from one linkage convention to the other. The standard J3B linkage convention passes addresses of all arguments in Reg A and Reg B, with all others following the call to the subroutine. The hand-coded library, however, passed the actual argument in the A-B registers in some cases, used index register 4 in other cases and returned the result in the A-B register, or a specified address. This inconsistency in linkage procedures required separate intermediate linkage conversion routines for each routine used in the math library. Standard JOVIAL/J3B linkage convention requires saving index registers 1 thru 5 (XR1-XR5) on the stack, obtaining proper arguments or addresses and placing them at the appropriate location according to the routine being called. Upon return from the subroutine, the results had to be returned to the J3B caller where proscribed by the J3B linkage convention. Also, the index registers (XR1-XR5) had to be restored from the stack and the stack cleaned up to the state it was in when called by the J3B program.
SECTION IV
MEMORY AND TIME REQUIREMENTS

Memory requirements for both the J3B and FOCAP programs were taken from the memory map produced by the Linkage Editor step of the compilation or assembly. Table 1 gives a comparison of these requirements.

<table>
<thead>
<tr>
<th></th>
<th>NAV</th>
<th>ALIGN</th>
<th>INIT</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>J3B</td>
<td>885</td>
<td>1477</td>
<td>744</td>
<td>3106</td>
</tr>
<tr>
<td>FOCAP</td>
<td>1014</td>
<td>836</td>
<td>694</td>
<td>2544</td>
</tr>
</tbody>
</table>

Total memory requirements, including the executive (RTEXEC) and math subroutine library (SUBLIB) plus linkage routines:

<table>
<thead>
<tr>
<th></th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>FOCAP</td>
<td>4524</td>
</tr>
<tr>
<td>J3B</td>
<td>6350</td>
</tr>
</tbody>
</table>
Timing for these programs are listed in Table 2. All major routines were timed using the following method: A Hewlett-Packard 1600S Logic State Analyzer, consisting of a H-P 1600A Logic State Analyzer and a HP 1607 Logic State Analyzer was connected to the address lines of the SKC-2000 at the Computer Control Unit (CCU). The H-P 1600A address compare lines were set to the desired start address of the GEANS program and the H-P 1607 address compare lines were set to the desired stop address. When the address compare lines matched the start address in the SKC-2000 the H-P 1600A generated a trigger which was fed to an electronic counter. The H-P 1607 generated a trigger for the stop address in like manner. The electronic counter recorded the time delay between the start and stop triggers, thus giving the time of execution of the block of code under test. The uncertainty in this method is 100 nanoseconds, the sensitivity of the electronic counter. The accuracy of these measurements is ±100 nanoseconds.
<table>
<thead>
<tr>
<th>SUBROUTINE</th>
<th>TIME IN MICROSECONDS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FOCAP</td>
</tr>
<tr>
<td>IIC</td>
<td>1337</td>
</tr>
<tr>
<td>IID</td>
<td>2620</td>
</tr>
<tr>
<td>IIE</td>
<td>1250</td>
</tr>
<tr>
<td>IIF</td>
<td>8417</td>
</tr>
<tr>
<td>IIG</td>
<td>3460</td>
</tr>
<tr>
<td>IIH</td>
<td>3162</td>
</tr>
<tr>
<td>IIK</td>
<td>1183</td>
</tr>
<tr>
<td>IIM</td>
<td>1636</td>
</tr>
<tr>
<td>IIO</td>
<td>3642</td>
</tr>
<tr>
<td>IIP</td>
<td>13066</td>
</tr>
<tr>
<td>RSET</td>
<td>347</td>
</tr>
<tr>
<td>IIR</td>
<td>45</td>
</tr>
<tr>
<td>FENT</td>
<td>6676</td>
</tr>
<tr>
<td>DECD</td>
<td>2340</td>
</tr>
<tr>
<td>IE</td>
<td>2460</td>
</tr>
<tr>
<td>IF</td>
<td>977</td>
</tr>
<tr>
<td>IG</td>
<td>790</td>
</tr>
<tr>
<td>IH</td>
<td>3883</td>
</tr>
<tr>
<td>IJ</td>
<td>7254</td>
</tr>
</tbody>
</table>
SECTION V
DISCUSSION OF RESULTS

For that portion of GEANS that was written in J3B (Alignment, Navigation and Initialization) the memory requirement was approximately 20% greater for the J3B version than for the FOCAP version. The total memory requirement for J3B was 29% greater than for FOCAP. The J3B compiler produced approximately 80% short instructions (a short instruction is 1/2 word, or 16 bits, long). The FOCAP version produced approximately 15% short instructions.

The J3B version was found to run two to three times slower than the FOCAP version. A number of things contributed to this:

1) The compiler was designed to create a high density of short instructions. This it does, and actually produces more instructions than necessary as a result. It reserves Index Register 5 (XR5) as a base register and loads XR5 with the address of one of several data areas to create short instructions. As a result when some operations are being performed, such as creation of a 3x3 matrix, the compiler loads XR5 once for each data item that is moved. So an index register load is performed (which takes one full word of memory and 2.5 microseconds) to create two short instructions. This sequence is repeated nine times, and is very inefficient both in time and memory usage.

2) The compiler generates code which computes an address offset each time it is used within a loop. For example, if the offset is used four times within a loop it is computed four times, and it only needs to be computed the first time.
3) The special linkage subroutines, those that resolve the J3B/FOCAP linkage differences, consumed 50 microseconds. Each subroutine call in GEANS required 100 microseconds more to execute because of this requirement.

The JOVIAL-3B level 0 compiler is poorly optimized to save time of execution. It will produce a high density of instructions, and with a data base designed to take advantage of this ability some optimization in time and memory is possible. For this effort the FOCAP data base was used, so the shortcomings of the compiler were accentuated. Later versions may be more efficient.
SECTION VI

SUMMARY AND CONCLUSIONS

The GEANS software development effort consisted of reprogramming the GEANS Alignment and Navigation algorithms in a High Order Language (HOL), JOVIAL/J3B, Level 0. The GEANS program was hosted on the Singer/Kearfott SKC-2000 computer and was written in the SKC-2000 assembly language, FOCAP. The purpose of the J3B effort was to compare the memory and time requirements of GEANS as written in FOCAP to GEANS written in J3B. Alignment, navigation and initialization portions of GEANS were coded in J3B. The FOCAP data base, math subroutine library, and real time executive were retained as part of the J3B version.

Final results showed that the J3B version required 29% more memory and two to three times more time than the FOCAP version. The memory requirement of 29% is somewhat misleading because additional code had to be written to resolve linkage convention differences between the FOCAP and J3B versions. In those routines that were coded in J3B directly from FOCAP (i.e. Alignment and Navigation) the J3B version required 20% more memory than FOCAP. A similar case can be made for timing requirements. The particular compiler used for this study (J3B Level 0) is a very inefficient compiler and is very poorly optimized. If GEANS had been written from scratch in J3B, with a properly designed data base and real time executive, a considerable increase in timing and memory efficiency might have been realized over the results of this study.
The results of this study were brought to the attention of Softech, Inc. (designers of J3B), who explained that the latest version of J3B (J3B Level 2) corrects most, if not all, of the deficiencies of J3B level 0. A more efficient compiler would most definitely show better results. The overhead for an efficient compiler of any high order language would be close to 20% more than assembly language.
GEANS NAVIGATION

EXEC

DECODE

NSCH

0

1

2

3

DV BIAS & SCALE FACTOR

LAT AND LONG COMPUTATION

NAVIGATION OUTPUT

GRAVITY MODEL

COMPUTE LOCAL VELOCITY, EARTH RELATIVE VELOCITY AND GROUND SPEED

VERTICAL DAMPING

VELOCITY AND DISTANCE COMPUTATION

RETURN TO ALIGN DECISION

DRIFT COMPENSATION
ALIGNMENT SUB-EXECUTIVE

NAVF = 0
ASCH = ASCH + 1

< 0

= 0

= 1

= 2

= 3

ASCH

BACKGROUND

NCCU = NCCU + 1
NCCD = NCCD + 1

IIF

IIG

NCCD

IIR

NAV1

SAM1

IIC

IIF

IIG

IIR

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE

IIE
LOW PASS FILTER

\[ \begin{align*}
& \text{IFC} = \text{IC} \\
& \text{IFC} 1 \\
& \text{IFC} 2 \\
& \text{ID1} \\
& \text{ID2} \\
& \text{ID3} \\
& \text{ID4}
\end{align*} \]
IF = 1
IF1

\[
\begin{bmatrix}
SDVI \\
SDVJ \\
SDVK
\end{bmatrix} = \begin{bmatrix}
SDVI \\
SDVJ \\
SDVK
\end{bmatrix} + \begin{bmatrix}
DVXI \\
DVXJ \\
DVXK
\end{bmatrix}
\]

IF2

DCON = DCON + 1

IF3

DCON < 0

NO

IF4

DCON = DCSK

IF5

SRT1 = SRT1/DTDC
SRT2 = SRT2/DTDC

IF6

F1 = 1 - (SRT1/DCO4)**2
F2 = 1 - (SRT2/DCO4)**2

IF7

\[
\begin{bmatrix}
DVI \\
DVJ \\
DVK
\end{bmatrix} = \begin{bmatrix}
0 & 0 & 1 \\
\sqrt{2}/2 & \sqrt{2}/2 & 0 \\
-\sqrt{2}/2 & \sqrt{2}/2 & 0
\end{bmatrix} \begin{bmatrix}
SDVI \\
SDVJ \\
SDVK
\end{bmatrix}
\]

IF14

SRT1 = SRT2 = 0
RATP = RATM = 0
SDVI = SDVJ = SDVK = 0
CHAJ = 2

RETURN

21
\[
\begin{align*}
\begin{bmatrix}
\phi_x \\
\phi_y \\
\phi_z
\end{bmatrix} &=
\begin{bmatrix}
DC42 & DC42 & 0 \\
SRT1 & SRT2 & 0 \\
DC42- & DC42 & 0 \\
SRT1 & SRT2 & 0
\end{bmatrix}
\begin{bmatrix}
\phi_x \\
\phi_y \\
\phi_z
\end{bmatrix}
\end{align*}
\]

<table>
<thead>
<tr>
<th>Phase</th>
<th>Matrix</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IIF12</td>
<td>( DTDC )</td>
<td>Transform to platform</td>
</tr>
<tr>
<td>IIF8</td>
<td>( G )</td>
<td>Independent, speed independent</td>
</tr>
<tr>
<td>IIF9</td>
<td>( G )</td>
<td>Dependent, speed independent</td>
</tr>
<tr>
<td>IIF10</td>
<td>( G )</td>
<td>Independent, speed dependent</td>
</tr>
</tbody>
</table>

\[
\begin{bmatrix}
1*CD33 & 1*CD34 & 1*CD35 \\
-F2*CD39 & -F2*CD40 & -F2*CD41 \\
F1*CD36 & F1*CD37 & F1*CD38
\end{bmatrix}
\begin{bmatrix}
DVI \\
DVJ \\
DVK
\end{bmatrix}
\]

G & Speed Dependent

\[
\begin{bmatrix}
DCAz \\
3x3
\end{bmatrix} =
\begin{bmatrix}
0 & 0 & \phi_z \\
-\phi_z & 0 & -\phi_x \\
-\phi_y & 0 & 0
\end{bmatrix}
\]
STHR = \sin (WDT \times (NCCU - .5) - \text{PHA})
.deepEqual
CTHR = \cos (WDT \times (NCCU - .5) - \text{PHA})
.

RDVX = (AK2T \times \text{CTHR}) \times \Delta T

RDVY = (AK2T \times \text{STHR}) \times \Delta T

RDVZ = (AK2T) \times \Delta T
.

DPTO = VT / \sqrt{DVX^2 + DVY^2 + DVZ^2}

\text{SRA} = \text{SRA} + 1. - \text{DPTO}

YA1 = YA1 + \text{TEMO}

YA2 = YA2 + \text{TEMO} \times \text{STHR}

YC2 = YC2 + \text{TEM4} \times \text{STHR}

YCI = YC1 + \text{TEM4} \times \text{CTHR}

YB1 = YB1 + \text{TEM2}

YB2 = YB2 + \text{TEM2} \times (NCCU/8.) \times \Omega GA
.

\text{RETURN}
SUMMING FOR E.P.A. SOLUTION

\[
\begin{bmatrix}
VAXI \\
VAYJ \\
VAZK
\end{bmatrix} = \begin{bmatrix}
VAXI \\
VAYJ \\
VAZK
\end{bmatrix} + \begin{bmatrix}
DVXJ \\
DVYJ \\
DVZK
\end{bmatrix}
\]

SUMMING FOR LOCAL LEVEL SOLUTION

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

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

RETURN
E.P.A. SOLUTION

IJK
IJK2

\[
\begin{align*}
\text{TEM2} &= \frac{VAX}{AK1T} \\
\text{TEMO} &= \frac{VAY}{AK1T} \\
\text{SWT} &= \frac{\text{TEMO}}{\sqrt{\text{TEMO}^2 + \text{TEM2}^2}} \\
\text{CWT} &= \frac{\text{TEM2}}{\sqrt{\text{TEMO}^2 + \text{TEM2}^2}} \\
\text{SX} &= \text{SY} = \text{SZ} = 0 \\
\text{ACM} &= 0 \\
\text{PHA} &= 0 \\
&\text{RETURN}
\end{align*}
\]
LOCAL LEVEL SOLUTION

IIM

IIM1

IIM2

IIM3

\[
\begin{bmatrix}
\text{TEM0} \\
\text{TEM2} \\
\text{TEM4}
\end{bmatrix}
= \frac{1}{\sqrt{\beta}} \times 
\begin{bmatrix}
\text{VAX} \\
\text{VAY} \\
\text{VAZ}
\end{bmatrix}
\]

\[
\text{SZ} = \text{TEM2} \times \text{CGDL}
\]

\[
\text{SX} = -\text{TEM2} \times \text{SGDL}
\]

\[
\text{SY} = -\text{SIGN(TEM4)} \times \text{SQRT(TEM4**2 + TEMO**2)}
\]

\[
\text{ACM} = 0
\]

\[
\text{SWT} = \text{SIN(WOPP*NGC)}
\]

\[
\text{CWT} = \text{COS(WOPP*NGC)}
\]

RETURN
LEAST SQUARES SOLUTION

\[
\begin{align*}
\begin{bmatrix}
X_A \\
2x1
\end{bmatrix} &=
\begin{bmatrix}
a_{MCSI} \\
\end{bmatrix}
\begin{bmatrix}
Y_A \\
2x1
\end{bmatrix} \\
\begin{bmatrix}
X_B \\
2x1
\end{bmatrix} &=
\begin{bmatrix}
b_{MCSI} \\
\end{bmatrix}
\begin{bmatrix}
Y_B \\
2x1
\end{bmatrix} \\
\begin{bmatrix}
X_C \\
2x1
\end{bmatrix} &=
\begin{bmatrix}
c_{MCSI} \\
\end{bmatrix}
\begin{bmatrix}
Y_C \\
2x1
\end{bmatrix}
\end{align*}
\]

\[
\begin{align*}
SX &= X_C(2)/AK1T \\
SZ &= X_A(2)/AK1T \\
SY &= -X_C(1)/AK1T \\
ACM &= SRA/NCCU \\
SX &= SX+SGDL*(SZ*CGDL-XB(1)/VTC-SX*SGDL) \\
SZ &= SZ-CGDL*(SZ*CGDL-XB(1)/VTC-SX*SGDL)
\end{align*}
\]

\[
\begin{align*}
SWT &= \sin(WOPP*NCCU) \\
CWT &= \cos(WOPP*NCCU)
\end{align*}
\]
COMPUTE $\Delta A$ MATRIX AND $\Delta AJ$ MATRIX

\[
\begin{align*}
A &= \begin{bmatrix}
\text{TEM0} & \text{SZ} & 0 \\
-\text{SZ} & \text{TEM4} & 0 \\
0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
\text{TEM2} & 0 & -\text{SY} \\
0 & 1 & 0 \\
\text{SY} & 0 & \text{TEM2}
\end{bmatrix}
\begin{bmatrix}
1 & 0 & 0 \\
0 & \text{TEM0} & \text{SX} \\
0 & -\text{SX} & \text{TEM0}
\end{bmatrix}
\end{align*}
\]

CALL IIR5
CALL ALNO
RETURN
SUBROUTINE RESET

IIQ1

\[
\begin{align*}
SRA &= 0 \\
VAX &= 0 \\
VAY &= 0 \\
VAZ &= 0 \\
\end{align*}
\]

\[
\begin{bmatrix}
Y\text{A1} \\
Y\text{A2}
\end{bmatrix} =
\begin{bmatrix}
Y\text{B1} \\
Y\text{B2}
\end{bmatrix} =
\begin{bmatrix}
Y\text{C1} \\
Y\text{C2}
\end{bmatrix} =
\begin{bmatrix}
X\text{A1} \\
X\text{A2}
\end{bmatrix} =
\begin{bmatrix}
X\text{B1} \\
X\text{B2}
\end{bmatrix} =
\begin{bmatrix}
X\text{C1} \\
X\text{C2}
\end{bmatrix} = 0
\]

IIQ2

\[
N\text{MO} = N\text{MO} + 1
\]

\[
N\text{MO} \geq 4
\]

\[
N\text{MO} < 4
\]

CALL A\text{LNO}

IIQ2A

\[
N\text{MO} = 0
\]

\[
N\text{MO} \geq 8
\]

IIQ2E

IIQ2C

FLGN = 0; INS NOT ALIGN LITE OFF

SAVT = TIME + BB1

SETUP FINE SOL'N SCALING SHIFTS -
FIDDLE WITH BITE BITS FOR IMU & VAC & ROT UNR

\[
N\text{CCD} = -64
\]

SAMI = 1

SET UP COURSE SOL'N SCALING SHIFTS

\[
N\text{CCD} = -1200
\]

MCSI = 0

\[
N\text{CCD} = -1680
\]

MCSI = 1

NCCD = -7200

MCSI = 2

NCCD = -7200

MCSI = 3

IIQ3

SAMI = 2

IIQ3A

NCCU = 0
GO TO NAV DECISION

IIR

ASCH = -1

DPU PROCESSING

BACKGROUND

> 4

= 4

BACKGROUND WILL BE IN AIR ALIGNMENT

< 4

MODE

< 4

FLGN = 0

NMO

> 7

BACKGROUND

≠ 0

IIR2

TMPR = 1

AJ

3x3

SA

3x3

IIR3

ac = OMEG*(TIME-SAVT)

SWT = SIN(OMEG*(TIME-SAVT))

CWT = COS(OMEG*(TIME-SAVT))

C
\[
\begin{bmatrix}
\mathbf{D} \\
3x3
\end{bmatrix}
= \begin{bmatrix}
\mathbf{CWT} & \mathbf{SWT} & 0 \\
-\mathbf{SWT} & \mathbf{CWT} & 0 \\
0 & 0 & 1
\end{bmatrix}
\]

\[
\begin{bmatrix}
\mathbf{D_1} \\
3x3
\end{bmatrix}
= \begin{bmatrix}
\mathbf{D} \\
3x3
\end{bmatrix}
\begin{bmatrix}
\mathbf{A}\mathbf{J} \\
3x3
\end{bmatrix}
\]

\[
\begin{bmatrix}
\mathbf{A}\mathbf{J} \\
3x3
\end{bmatrix}
= \begin{bmatrix}
\mathbf{D_1} \\
3x3
\end{bmatrix}
\]

RETURN
NAVIGATION INITIALIZATION

ENTRY 04 OF VECT
= CALL NAV

NAVF = 1

TEMO = ATAN(SGDL/(CGDL*KGDL))
SGCL = SIN(TEMO)
S2GC = SGCL**2
CGCL = COS(TEMO)
C2GC = CGCL**2

DELR = ALT-(21385*S2GC*(1+.00503*C2GC))
RAD = DELR+RADE

X = RAD*CGCL
Z = RAD*SGCL

Y = LATB = LONB = 0
VX = VZ = 0
VY = OMGA*X
RXYZ = RAD
LONG = LONL
LGO = LONL
TO = TIME - 3/32
TLPO = TO

RETURN
ALIGNMENT OUTPUT ROUTINE

ALNO

DECFGLG = DECFGLG+1

BTIME = TIME

\[
\begin{bmatrix}
BSX \\
BSY \\
BSZ
\end{bmatrix} = 
\begin{bmatrix}
SX \\
SY \\
SZ
\end{bmatrix}
\]

\[
\begin{bmatrix}
BAJ
\end{bmatrix}_{3x3} = 
\begin{bmatrix}
AJ
\end{bmatrix}_{3x3}
\]

BNMO = NMO

RETURN
NAVIGATION SUB-EXECUTIVE

NSCH = NSCH + 1

CALL IC
CALL ID
CALL IE
CALL IF
CALL IG

CALL IH
CALL IJ

CALL NAVO

CALL IL
CALL IM
CALL RTAL
NSCH = -1

BACKGROUND

=0

1

2

3

IA
ACCELEROMETER BIAS & SCALE FACTOR COMPUTATION AND NON-ORTHOGONALITY COMPENSATION

![Diagram](image)

\[ \begin{bmatrix} \text{DVX}_1 \\ \text{DYY}_J \\ \text{DVZ}_K \end{bmatrix} = \begin{bmatrix} \text{AB} \end{bmatrix}_{3x3}, \begin{bmatrix} \text{DVX}_G \\ \text{DVY}_G \\ \text{DVZ}_G \end{bmatrix} = \begin{bmatrix} \text{CD04D} \\ \text{CD05D} \\ \text{CD06D} \end{bmatrix} \]

\[ \text{DVX}_G = \text{DVY}_G = \text{DVZ}_G = 0 \]

RETURN
ROTATION FROM PLATFORM FRAME TO NAVIGATION FRAME

\[
\begin{bmatrix}
DVX \\
DVY \\
DVZ
\end{bmatrix} = AJ 
\begin{bmatrix}
DVXI \\
DVYJ \\
DVZK
\end{bmatrix}
\]

RETURN
GRAVITY MODEL

\[ P = 1 - (\text{DELUR/RADE}) + (\text{DELUR/RADE})^{**2} \]

\[
\begin{align*}
\text{GDXT} &= \text{GCA}2*(\text{GCA}5-6*\text{S2GC}+9*\text{S2GC}^{**2}) + \text{GCA}0*\text{P}^{**3} - \text{GCA}1((\text{GCA}3-\text{S2GC})*\text{P}^{**5})*X \\
\text{GYDT} &= \text{GCA}2*(\text{GCA}5-6*\text{S2GC}+9*\text{S2GC}^{**2}) + \text{GCA}0*\text{P}^{**3} - \text{GCA}1((\text{GCA}3-\text{S2GC})*\text{P}^{**5})*Y \\
\text{GZDT} &= \text{GCA}2*(\text{GCA}6+\text{GCA}5-10*\text{S2GC}+9*\text{S2GC}^{**2}) + \text{GCA}0*\text{P}^{**3} - \text{GCA}1((\text{GCA}4+\text{GCA}3-\text{S2GC})*\text{P}^{**5})*Z
\end{align*}
\]

RETURN
VERTICAL DAMPING COMPUTATION

IF

IF2

DELR=ALT-(21385*52GC*(1+.00503xC2GC))

IF3

RAD=DELR + RADE

IF1

\[ \phi 46 \text{ DATA } = \text{ ALT - (RAD-RXYZ)} \]

\[
\begin{bmatrix}
LDVX \\
LDVY \\
LDVZ
\end{bmatrix}
= CD52*(RAD-RXYZ) *
\begin{bmatrix}
X \\
Y \\
Z
\end{bmatrix}
\]

RETURN

DOUBLE INTEGRATION FOR VELOCITY AND DISTANCE

IG

IG1

\[
\begin{bmatrix}
VX \\
VY \\
VZ
\end{bmatrix}
= \begin{bmatrix}
VX \\
VY \\
VZ
\end{bmatrix} + \text{DELT} *
\begin{bmatrix}
GXDT \\
GYDT \\
GZDT
\end{bmatrix} + \begin{bmatrix}
DVX \\
DVY \\
DVZ
\end{bmatrix}
\]

\[
\begin{bmatrix}
X \\
Y \\
Z
\end{bmatrix}
= \begin{bmatrix}
X \\
Y \\
Z
\end{bmatrix} + \text{DELT} *
\begin{bmatrix}
LDVX \\
LDVY \\
LDVZ
\end{bmatrix} + \begin{bmatrix}
VX \\
VY \\
VZ
\end{bmatrix}
\]

RETURN
LATITUDE AND LONGITUDE COMPUTATION

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

\[
\begin{bmatrix}
V V \\
V E \\
V N
\end{bmatrix} =
\begin{bmatrix}
CGDL*CCLG & CGDL*SCLG & SGDL \\
-SCLG & CCLG & 0 \\
-SGDL*CCLG & -SGDL*SCLG & CGDL
\end{bmatrix}
\begin{bmatrix}
VX+Y*OMGA \\
VY-X*OMGA \\
VZ
\end{bmatrix}
\]

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

\[
\begin{bmatrix}
\theta_48 \\
\theta_49 \\
\theta_4b
\end{bmatrix} =
\begin{bmatrix}
LSH OF VV \\
LSH OF VE \\
LSH OF VN
\end{bmatrix}
\]

\[
\begin{bmatrix}
\theta_47 \\
\theta_49 \\
\theta_4b
\end{bmatrix} =
\begin{bmatrix}
MSH OF VV \\
MSH OF VE \\
MSH OF VN
\end{bmatrix}
\]

\[
VEL2 =
\begin{bmatrix}
V V \\
V E \\
V N
\end{bmatrix}
\cdot
\begin{bmatrix}
V V \\
V E \\
V N
\end{bmatrix}
\]

\[
GS = SQRT (VEL2-VV**2)
\]

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

RETURN
NAVIGATION OUTPUT ROUTINE

NAVO

< 6 MIN

TIME-TLPO

≥ 6 MIN

DECFLG = DECFLG + 1

TLPO = TIME

B TIME = TIME
BLAT = LAT
BLONG = LONG

[BVV
BVE
BVN] = [VV
VE
VN]

[BVX
BVY
BVZ] = [VX
VV
VZ]

RETURN
DRIFT COMPENSATION

IL1

\[
\begin{bmatrix}
SDVI \\
SDVJ \\
SDVK
\end{bmatrix} = \begin{bmatrix}
SDVI \\
SDVJ \\
SDVK
\end{bmatrix} + \begin{bmatrix}
DVXI \\
DVXJ \\
DVXK
\end{bmatrix}
\]

IL2

DCON = DCON + 1

IL3

YES

DCON < 0

NO

IL4

DCON = DCSK

IL5

\begin{align*}
SRT1 &= SRT1/DTDC \\
SRT2 &= SRT2/DTDC
\end{align*}

IL6

\begin{align*}
F1 &= 1 - (SRT1/DC04)^2 \\
F2 &= 1 - (SRT2/DC04)^2
\end{align*}

IL7

\[
\begin{bmatrix}
DVI \\
DVJ \\
DJK
\end{bmatrix} = \begin{bmatrix}
0 & 0 & 1 \\
\sqrt{2}/2 & \sqrt{2}/2 & 0 \\
-\sqrt{2}/2 & \sqrt{2}/2 & 0
\end{bmatrix} \begin{bmatrix}
SDVI \\
SDVJ \\
SDVK
\end{bmatrix}
\]

IL8 -- IL13

DC

SEE NEXT PAGE

IL14

\begin{align*}
SRT1 &= SRT2 = 0 \\
RATP &= RATM = 0 \\
SDVI &= SDVJ = SDVK = 0
\end{align*}

IL15

CHAJ = 0

RETURN
\[
\begin{align*}
\begin{bmatrix}
\varphi_x \\
\varphi_y \\
\varphi_z
\end{bmatrix}
&= 
\begin{bmatrix}
DC42 & DC42 & 0 \\
SRTT & SRT2 & 0 \\
DC42 & DC42 & 0
\end{bmatrix}
\begin{bmatrix}
\varphi_x \\
\varphi_y \\
\varphi_z
\end{bmatrix}
\]
\end{align*}
\]

\[
\text{TRANSFORM TO PLATFORM}
\]

\[
\begin{align*}
\text{IL12} & \quad \text{G INDEPENDENT, SPEED INDEPENDENT} \\
\text{IL8} & \quad \text{G DEPENDENT, SPEED INDEPENDENT} \\
\text{IL9} & \quad \text{G DEPENDENT, SPEED DEPENDENT}
\end{align*}
\]

\[
\begin{align*}
\text{IL11} & \quad \text{G & SPEED DEPENDENT} \\
\text{IL13} & \quad \text{G INDEPENDENT, SPEED DEPENDENT}
\end{align*}
\]

\[
\begin{align*}
\begin{bmatrix}
\varphi_x \\
\varphi_y \\
\varphi_z
\end{bmatrix}
&= 
\begin{bmatrix}
0 & Q_z & 0 \\
-\varphi_z & 0 & \varphi_y \\
-\varphi_y & 0 & \varphi_x
\end{bmatrix}
\begin{bmatrix}
\varphi_x \\
\varphi_y \\
\varphi_z
\end{bmatrix}
\end{align*}
\]

\[
\begin{align*}
\text{DTDC} & \quad \text{G INDEPENDENT, SPEED INDEPENDENT} \\
\text{GM} & \quad \text{G DEPENDENT, SPEED DEPENDENT} \\
\text{DVI} & \quad \text{G DEPENDENT, SPEED DEPENDENT}
\end{align*}
\]

\[
\begin{align*}
\text{DVI} & \quad \text{G INDEPENDENT, SPEED DEPENDENT} \\
\text{DVJ} & \quad \text{G DEPENDENT, SPEED DEPENDENT} \\
\text{DVK} & \quad \text{G DEPENDENT, SPEED DEPENDENT}
\end{align*}
\]

\[
\begin{align*}
\begin{bmatrix}
F1*CD33 & F1*CD34 & F1*CD35 \\
-F2*CD39 & -F2*CD40 & -F2*CD41 \\
F1*CD36 & F1*CD37 & F1*CD38
\end{bmatrix}
\begin{bmatrix}
\varphi_x \\
\varphi_y \\
\varphi_z
\end{bmatrix}
\end{align*}
\]

\[
\begin{align*}
\begin{bmatrix}
F1*CD30 & F2*CD32 & G & SRT2 \\
F1*CD31 & (1-DC04)(CD42*RATP-CD43*RATM) & G & SRT2 \\
\end{bmatrix}
\end{align*}
\]

\[
\begin{align*}
\begin{bmatrix}
F1*CD30 & F2*CD32 & G & SRT2 \\
F1*CD31 & (1-DC04)(CD42*RATP-CD43*RATM) & G & SRT2 \\
\end{bmatrix}
\end{align*}
\]
UPDATE AJ AND SA MATRICES

IM1

= 0

CHAJ

≠ 0

CHAJ = 0

IM2

\[
\begin{bmatrix}
D \\
3x3
\end{bmatrix}
= 
\begin{bmatrix}
AJ \\
3x3
\end{bmatrix}
+ 
\begin{bmatrix}
DCAR \\
3x3
\end{bmatrix}
\]

IM3

YES

FLGN = 0

IM4

NO

\[
\begin{bmatrix}
D \\
3x3
\end{bmatrix}
= 
\begin{bmatrix}
SA \\
3x3
\end{bmatrix}
+ 
\begin{bmatrix}
DCAR \\
3x3
\end{bmatrix}
\]

\[
\begin{bmatrix}
SA \\
3x3
\end{bmatrix}
= 
\begin{bmatrix}
SA \\
3x3
\end{bmatrix}
+ 
\begin{bmatrix}
D \\
3x3
\end{bmatrix}
\]

RETURN
RETURN TO ALIGN DECISION (RTAL)

IN
IN1

DPU PROCESSING
(NOT IMPLEMENTED)

>4

MODE

≤4

SAVT = TO+3/32

SA
3x3

AJ
3x3

CALL FENT

FLGN = FLGN+1

NSCH = -2

NAV = MODE

TURN OFF INS
NOT ALIGN LIGHT
(NOT IMPLEMENTED)

BACKGROUND
EXECUTIVE INITIALIZATION

Status Reg. = 0
Clear Interrupt State

VECT = 'CALL DUMY'

CALL BDSI

CALL CDUI

Reset DMA Channels

Set Interrupts 4, 5, and 10

EXNO = 0
ERRCNT = -1

Enable Memory & Program Interrupts

Wait for first 32 Hz interrupt
SKC-2000 Executive

Ex30

- EXNO = 1

ITER

< 1/4 sec

Reset Watchdog Timer

≥ 1/4 sec

Go to VECT

Ex30A

Disable Program Interrupt

EXNO = CYCLE

Enable Program Interrupt

≠ 0

EXNO

= 0

Call DEC

Wait for next interrupt
Vector Table

During Alignment

CALL DUMMY
CALL DECD
CALL CDU
CALL ALIGN (IIA)
CALL SPIN (DUMY)
CALL DUMMY
CALL BITE (DUMY)
CALL DUMMY
CALL DUMMY
CALL DUMMY
CALL DUMMY
CALL DUMMY
CALL DUMMY
CALL DUMMY
CALL GAS (DUMY)
GO TO Ex30A

During Navigation

CALL DUMMY
CALL DECD
CALL CDU
CALL NAV (IA)
CALL SPIN (DUMY)
CALL DUMMY
CALL BITE (DUMY)
CALL DUMMY
CALL DUMMY
CALL DUMMY
CALL DUMMY
CALL DUMMY
CALL DUMMY
CALL DUMMY
CALL GAS (DUMY)
GO TO Ex 30A
Syncronize SKC-2000 Alignment with Honeywell Alignment.

1. CDU
   - #0
2. CDUS1
   - = 0
3. SW 3
   - ON
4. SW 3
   - OFF
5. CDUS2
   - #0
6. CDUS2
   - = 0
7. MODE
   - < 4
8. MODE
   - ≥ 4
9. MODE
   - ≥ 4
10. MODE
    - < 4
11. CDUS2
    - = 1
12. Return
13. CDUS1
    - = 1
14. CALL FENT
15. RETURN
16. SW 0
    - ON
17. SW 1
    - ON
18. SW 2
    - ON
19. MODE = MODE
    - OFF
20. Mode = MODE
    - OFF
21. MODE = 4
    - OFF
22. MODE = 3, NAV
23. RETURN

50
CDU Initialization

CDUI

CDUS1 = 0
CDUS2' = 0

Set 3RD Entry of VECT to 'CALL CDU'

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

BDSI

BLP1 = BLP2 = BLP3 = 0
BCTR = BNBR = HCTR = 0
HOLD = MALF = MLFN = 0
CMD1 = CMD2 = CMD4 = 0
BER1 = BER2 = BER3 = BER4 = 0

BDSI(I) = BTIN(I) I = 1, 4
BMK(I) = BTIN(I+4) I = 1, 4

SRT1 = SRT2 = RATP = RATM = 0
ROT1 = ROT2 = 0
DVXG = DVYG = DVZG = 0
DPVU = DPDV = DPHV = 0
GMT = 0
BTE1 = BTE2 = BTE3 = BTE4 = 0
R1CT = R2CT = CIPM = RAT = RATL = 0
DVX = DVY = DVZ = 0
CYLE = VRTV = DRFV = HDGV = 0

CRT1 = CRT2 = CRT3 = 0
TIME = ITER = PHAS = 0
NAVF = DATA = PUSH = TEST = 0

LITE = KLIT

Ø14 DATA = 0
Ø25 DATA = 0

MODE = MODE SWITCH FROM SIDL

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)

\[
\begin{bmatrix}
A_{3 \times 3}
\end{bmatrix}
= \begin{bmatrix}
SDVI \\
SDVJ \\
SDVK
\end{bmatrix}
= \begin{bmatrix}
VAXI \\
VAYJ \\
VAZK
\end{bmatrix}
= \begin{bmatrix}
DVXI \\
DVYJ \\
DVZK
\end{bmatrix}
= \begin{bmatrix}
DVXG \\
DVYG \\
DVZG
\end{bmatrix} = 0
\]

\[
SRTI = SRT2 = PATP = RATM = CHA1 = 0
\]

\[
DCON = -8
\]

\[
\begin{bmatrix}
GM_{3 \times 3}
\end{bmatrix}
= \begin{bmatrix}
CD19 & CD20 & CD21 \\
CD25 & CD26 & CD27 \\
CD22 & CD23 & CD24
\end{bmatrix}
\]

\[
A(1, 1) = CD01 \\
A(2, 2) = CD02 \\
A(3, 3) = CD03
\]

\[
\begin{bmatrix}
AB_{3 \times 3}
\end{bmatrix}
= \begin{bmatrix}
CD07 & CD10 & CD13 \\
CD08 & CD11 & CD14 \\
CD09 & CD12 & CD15
\end{bmatrix}
\begin{bmatrix}
A_{3 \times 3}
\end{bmatrix}
\]

\[
LAT = LATL
\]

\[
SGDL = SI: (LAT) \\
CGDL = COS (LAT)
\]

\[
AK1T = CGDL * GL \\
AK2T = SGDL * GL
\]

\[
PHA = 0
\]
\[
\begin{bmatrix}
VC_{1X} \\
VC_{1Y} \\
VC_{1Z}
\end{bmatrix}
= 
\begin{bmatrix}
VC_{2X} \\
VC_{2Y} \\
VC_{2Z}
\end{bmatrix}
= 
\begin{bmatrix}
VF_{1X} \\
VF_{1Y} \\
VF_{1Z}
\end{bmatrix}
= 
\begin{bmatrix}
VF_{2X} \\
VF_{2Y} \\
VF_{2Z}
\end{bmatrix} = 0
\]

\[
\begin{bmatrix}
A_{1J}
\end{bmatrix}
= 
\begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1
\end{bmatrix}
\]

To = TIME

\[VTB = GL \times 8.0\]
\[VTC = GL \times \text{DELT}\]

\[
\begin{bmatrix}
C_{D04D} \\
C_{D05D} \\
C_{D06D}
\end{bmatrix}
= \text{DELT} \times 
\begin{bmatrix}
C_{D04} \\
C_{D05} \\
C_{D06}
\end{bmatrix}
\]

NMO = -1

CALL RSET

\[
\begin{bmatrix}
C_{D05} & -\sin(C_{D05}) & \sin(C_{D05}) \\
\cos(C_{D05}) & -\sin(C_{D05}) \cos(C_{D05}) \\
\sin(C_{D05}) & \cos(C_{D05}) & \sin(C_{D05}) \cos(C_{D05})
\end{bmatrix}
\]
\[ E_{3 \times 3} = 0 \]

\[ KSN1 = KSN2 = KSN3 = 0 \]

\[ ASCH = -4 \]

Entry 4 of VECT = 'CALL ALIGN'

\[ DG(1, 2) = DG(3, 1) = 0 \]

\[ DG(3, 2) = -\sin(\text{HEAD}) \]

\[ DG(1, 1) = 1.0 \]

RETURN
Decode SIDL

DECD

BTE3 = BTE4 = 0
TEMP2 = 0

LPTK = LPTK * 2

Max Torque Command

NO

YES

LPTK = LPTK + 1

>1 min

GSCT

<1 min

TEMP2 = GSCT + 1/32

BTE3 = BTE3 or B3

GSCT = TEMP2

14F

=04F

A

BTE4 = 1
A

DO DO15
I = 1, 14

BTE4 = BTE4 * 2

= UFC Field of SIDL (I)
BTE4 = BTE4 + 1

CONTINUE

D5D DATA = 0

Disable Program Interrupts

= 0

CYLE

#0

CYLE

≥ 2

DMAERR = DMAERR or 16

BTE3 = BTE3 or B1

< 2

CYLE = 0

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)

04F Data = ITER

05B Data = CUU Switches From 13F

ALT Unresonable Bit

=1

=0

TEMP = DATI

DATI = (DATI * 8 + 05B Bits 20-22). AND. 7FFFF

TEMP

DATA = DATI, AND. 7

TEST = Press to TEST Bit (05B)
C2 \[\rightarrow\] C1

Push Button Switch \(\#0\) \[\rightarrow\] PUSH = Push Button Switch

Push Button Switch \(\#0\) \[\rightarrow\] Push Button Switch = 19 \[\rightarrow\] a = 256

Data Switch

LITE = a.XOR.384.OR.LITE.XOR.384

=MODE

Mode Switch

\#MODE

MODE Start Switch \(\#25\) \[\rightarrow\] >1 \[\rightarrow\] MODE Switch \(\#25\) \[\rightarrow\] =25 \[\rightarrow\] =1

LITE = LITE.OR.128

MODE = MODE SWITCH

D
04E DATA = R.A.T. FIELD (177) & VERTICAL VEL FIELD (122)

04D DATA = ROTOR SPEED (177)

05D DATA = R.A.T. BITE BIT (177) & TEMPERATURE BITE (174) & IMU BITE (173) & DOPPLER REL BIT (121)

025 DATA = BITE BITS (121) & BARO ALTITUDE (025)

060 DATA = DRIFT & HEADING (122)

050 DATA = -DVX (176) & +DVX (175)
051 DATA = -DVY (176) & +DVY (175)
052 DATA = -DVZ (176) & +DVZ (175)

05E DATA = 113 DATA

BTE4 BIT FOR SIDL-03

BTE1 BITS F00016 = 05D DATA BITS F00016

BTE4 BIT FOR SIDL-04

BTE1 BITS OFF016 = 05D DATA BITS OFF016

E
\[
\begin{align*}
\text{BTE1 BITS 000216} &= 05D \text{ DATA BITS 000216} \\
\text{BTE1 BITS 000116} &= 05D \text{ DATA BITS 000116} \\
\phi_{14} \text{ DATA} &= (\phi_{14} \text{ DATA.AND.FFOF16}) \text{.OR.}((.\text{NOT.BTE1}).\text{AND.3016}) \\
\text{BTE2} &= 05E \text{ DATA} \\
\text{RAT} &= \text{R.A.T. FIELD OF } \phi_{4E} \\
&\quad \text{(CONV TO FLOATING POINT)} \\
&\quad \text{A = 04D DATA} \\
\text{ROTR} &= A
\end{align*}
\]
F

=1
BTE4  
BIT FOR  
SIDL-08

=0

TEMP = BARO

BARO = 025 DATA

< 8

< 8

BTE3 = BTE3.OR.B13

> 8

CALL CDPU (DUMMY)

#0

ALT FLAG (CD63)

A = CD64

=0

A = CD61*(BARO+CD62)

ALT = A

G

63
HDGV = HEADING VEL FIELD (060)
DRFV = DRIFT VEL FIELD (060)
VRTV = VERTICAL VEL FIELD (04E)

DO D160
I = 1,3

DPVV(I) = DPVV(I) + VRTV(I)

BTE1 = BIT FOR RAT BITE

RAT = 0

=R = 0 < 0

RATL = 0

RATP = RATP + RAT
014 bit for RAT+ =1
RATL = RATL + 1

RATM = RATM + RAT
014 bit for RAT- =1
RATL - RATL -1
```
MrML 1K-11

< 2048
RATL

≥ 2048
BTE3 = BTE3.OR.B5

I bits 000316 = ROTR bits C00016

= 0

= 3

= 1 OR 2

A = ROT (I)

ROT(I) = ROTR bits 3FFF16

A - ROT(I) < 32

A = ROT (I)

RMIN < A < RMAX

A ≤ RMIN

A > RMAX

A = B2

R(i)CT = A - ROT(I)

I
```
I

R1CT >24 BTE3 = BTE3.OR.B14
< 24

< 24

R2CT

≥24 BTE3 = BTE3.OR.B15

#0

ITER

#16

= 0 OR 16

SRT1 = SRT1 + ROT1 * 1/2 SEC
SRT2 = SRT2 + ROT2 * 1/2 SEC

Ø6C DATA = 124 DATA
Ø6D DATA = 128 DATA
Ø6E DATA = 12C DATA

AHRS FLAG (CD63)

= 1 LHED = HEDL

= 0 J1

J2
K

BTE4 .AND. 768

=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

CONTINUE

L
ANY GIMBAL RATE
BIT OF BTE3 = 1

≥1 RPS

|M2|

ANY GIMBAL RATE
BIT = 1

BIT OF BTE3 = 1

=0

D295

CONTINUE

INPUT
POWER BIT OF
BTE 2

CIPM = 0

=1

=0

CIPM = CIPM + 1/32 SEC

CALL TKTH (DUMY)

RETURN
INTERRUPT 10 ROUTINE [32 HZ]

1. Reset Both DMA Channels
   - Prog. Cont. SCB Reset

   P0C1
   PIC2
   PIC3

   OEOT2 Is Re-Accessed On Each Loop
   =0
   OEOT2
   =1

   Reset DMA Output
   P0C2

   Save S Reset Flag 1

   ERRCNT
   =0
   CYLE = CYLE + 2

   EXNØ
   =0

   LAE EX30 STA RET1O

2. INT 10
   Save S, A, B

   Flag 1
   =0

   LDA P0CC1
   DØA
   LDA PICC1
   DØA 5

   Save S
   Set Flag 1

   Zero DMAERR

   Restore A, B, S

   RTA

   DMAERR
   AND E

   IEOT2
   =0

   IEOT2
   =1

   DMAERR
   AND D

   IEOT3
   =0

   DMAERR
   AND B

   1

3. Clear Carry Bit
   Increment ERRCNT
   DMAERR = F

   OEOT2
   =0

   IEOT2
   =1

   DMAERR
   AND E

   IEOT2
   =0

   IEOT3
   =1

   DMAERR
   AND B

   1
INTERRUPT 5 ROUTINE

INT 05

Save S, A, B

DPI

* PICS Is Snap-Shot

* IFTi

IPEi

Any = 1

A11 = 0

IEOT2

=0

=1

Modify Input Control Word

LDA PICC2

DMA 5

DMAERR AND D

JS TORK

Restore A, B, S

RTA

Clear Carry Bit

INCREMENT ERRCNT

DMAERR LOR 2

From PICS Construct PICC For Error Reset

Reset Error Interrupt
**INTERRUPT 4 ROUTINE**

1. \( \text{INT 04} \)
2. Save \( S, A, B \) DPI
3. Clear Carry Bit

*PÖCS* is Snap-Shot

- Any = 1
  - From PÖCS Construct PÖCC For Error Reset
  - Reset Error Interrupt
  - Increment ERRCNT
  - DMAERR LØR 1
  - DMAERR LØR 4
  - Increment ERRCNT

- \( E\)ØT = 0
  - DMAERR AND E
  - IE\( E\)ØT = 0
    - DMAERR AND 3
    - Reset DMA Channels

- \( E\)ØT = 1
  - RTA
  - Restore A, B, S
  - LAE-EX10 STA RET 04

- \( E\)ØT = 1
  - PÖC1
  - PÖC2
  - PIC2
  - PIC3

- Save S Reset Flag 1

73