NOTICE

THIS DOCUMENT HAS BEEN REPRODUCED FROM MICROFICHE. ALTHOUGH IT IS RECOGNIZED THAT CERTAIN PORTIONS ARE ILLEGIBLE, IT IS BEING RELEASED IN THE INTEREST OF MAKING AVAILABLE AS MUCH INFORMATION AS POSSIBLE
Onboard Utilization of Ground Control Points for Image Correction
Final Report
ONBOARD UTILIZATION OF GROUND CONTROL POINTS FOR IMAGE CORRECTION FINAL REPORT

MARTIN MARIETTA DENVER AEROSPACE Denver, Colorado 80201
V. FOREWORD

This volume of the "Onboard Utilization of Ground Control Points for Image Correction" final report provides a detailed description of the software that was developed to simulate the ground control point navigation system. Three other volumes have been incorporated into the final report. Volume I provides an executive summary, Volume II contains a detailed description of the study and simulator results, and Volume IV is an appendix describing software utilized for image correction accuracy measurement.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Introduction</td>
<td>A-4</td>
</tr>
<tr>
<td>2.</td>
<td>GCP Executive Program (GCP)</td>
<td>A-10</td>
</tr>
<tr>
<td>2.1</td>
<td>Initialize Data Module (INDATA)</td>
<td>A-20</td>
</tr>
<tr>
<td>2.2</td>
<td>Event Sequencer (GCPSEQ)</td>
<td>A-40</td>
</tr>
<tr>
<td>2.2.1</td>
<td>Check for Star Occultation (VISIBLE)</td>
<td>A-52</td>
</tr>
<tr>
<td>2.2.1.1</td>
<td>Perform Occultation Calculation (OCCULT)</td>
<td>A-57</td>
</tr>
<tr>
<td>2.2.1.1.1</td>
<td>Compute Vector to Star (BVECT)</td>
<td>A-64</td>
</tr>
<tr>
<td>2.3</td>
<td>Generate Environment (GENENV)</td>
<td>A-69</td>
</tr>
<tr>
<td>2.3.1</td>
<td>Integrate Actual Navigation State (RUNG)</td>
<td>A-75</td>
</tr>
<tr>
<td>2.3.1.1</td>
<td>Specify Integration Step Size (TREG)</td>
<td>A-81</td>
</tr>
<tr>
<td>2.3.1.2</td>
<td>Compute Position Dynamics (DNAV)</td>
<td>A-86</td>
</tr>
<tr>
<td>2.3.2</td>
<td>Generate Gyro Output (GYROUT)</td>
<td>A-92</td>
</tr>
<tr>
<td>2.3.3</td>
<td>Transformation from Euler Angles (BMAT)</td>
<td>A-100</td>
</tr>
<tr>
<td>2.3.4</td>
<td>Transformation from Quaternians (AMAT)</td>
<td>A-104</td>
</tr>
<tr>
<td>2.4</td>
<td>Process Gyro Data (GYRO)</td>
<td>A-108</td>
</tr>
<tr>
<td>2.4.1</td>
<td>Propagate Attitude (KATT)</td>
<td>A-114</td>
</tr>
<tr>
<td>2.4.1.1</td>
<td>Matrix Multiply (MATAB)</td>
<td>A-120</td>
</tr>
<tr>
<td>2.4.2</td>
<td>Compute State Partials (PDATT)</td>
<td>A-124</td>
</tr>
<tr>
<td>2.4.3</td>
<td>Loud Attitude Transition Matrix (PATT)</td>
<td>A-134</td>
</tr>
<tr>
<td>2.5</td>
<td>Integrate Position State (INTG)</td>
<td>A-138</td>
</tr>
<tr>
<td>2.5.1</td>
<td>Integrate Position Parameters (RKC)</td>
<td>A-143</td>
</tr>
<tr>
<td>2.5.1.1</td>
<td>Compute Solar/Lunar Ephemerides (EPHEM and TRUEA)</td>
<td>A-149</td>
</tr>
<tr>
<td>2.5.1.2</td>
<td>Computer Solar Pressure (SPRESS)</td>
<td>A-158</td>
</tr>
<tr>
<td>2.5.1.3</td>
<td>Compute Solar/Lunar Gravitational Perturbation (GPERT and EQ)</td>
<td>A-164</td>
</tr>
<tr>
<td>2.5.1.4</td>
<td>Compute Navigation Partials (PDNAV)</td>
<td>A-171</td>
</tr>
<tr>
<td>2.6</td>
<td>Generate Measurement (MEASURE)</td>
<td>A-178</td>
</tr>
<tr>
<td>2.6.1</td>
<td>GPS Model (GPS)</td>
<td>A-186</td>
</tr>
<tr>
<td>2.6.2</td>
<td>Star Tracker Model (START)</td>
<td>A-193</td>
</tr>
<tr>
<td>2.6.3</td>
<td>Landmark Tracker Model (LAMKT)</td>
<td>A-199</td>
</tr>
<tr>
<td>2.7</td>
<td>Perform State Estimation (EST)</td>
<td>A-211</td>
</tr>
<tr>
<td>2.7.1</td>
<td>Propagate Covariances (PROP)</td>
<td>A-226</td>
</tr>
<tr>
<td>2.7.2</td>
<td>Compute Measurement Partials (MPART)</td>
<td>A-231</td>
</tr>
<tr>
<td>2.7.2.1</td>
<td>Compute GPS Measurement Partials (HGPS)</td>
<td>A-236</td>
</tr>
<tr>
<td>2.7.2.2</td>
<td>Compute Landmark Tracker Measurement Partials (HLMT)</td>
<td>A-241</td>
</tr>
<tr>
<td>2.7.2.2.1</td>
<td>Compute Partials with Respect to Estimated Transformation from Inertial to Body Coordinates (SPETBI)</td>
<td>A-251</td>
</tr>
<tr>
<td>2.7.2.2.2</td>
<td>Compute Partials of Estimated Measurement Vector in Inertial Space (SPEMI)</td>
<td>A-257</td>
</tr>
<tr>
<td>2.7.2.3</td>
<td>Compute Measurement Partials for Star Tracker K (HSTAR)</td>
<td>A-261</td>
</tr>
<tr>
<td>2.7.3</td>
<td>Make Measurement Updates (UPDATE)</td>
<td>A-270</td>
</tr>
<tr>
<td>2.8</td>
<td>Print Performance Data (OUTDATA)</td>
<td>A-283</td>
</tr>
<tr>
<td>2.9</td>
<td>Plotting and Utility Routines</td>
<td>A-298</td>
</tr>
<tr>
<td>3.</td>
<td>Simulation Mode 2</td>
<td>A-372</td>
</tr>
<tr>
<td>3.1</td>
<td>Landmark (GCP) Extraction Program (LMEX)</td>
<td>A-373</td>
</tr>
<tr>
<td>3.2</td>
<td>Correlation-Scroll Programs</td>
<td>A-375</td>
</tr>
<tr>
<td>3.2.1</td>
<td>Scroll Program (SCRLL)</td>
<td>A-377</td>
</tr>
<tr>
<td>3.2.1.1</td>
<td>Draw a Box (DRAW)</td>
<td>A-379</td>
</tr>
<tr>
<td>3.2.2</td>
<td>Scroll Counter (SCRLL)</td>
<td>A-380</td>
</tr>
<tr>
<td>3.2.3</td>
<td>Correlation (CORR)</td>
<td>A-382</td>
</tr>
<tr>
<td>3.2.3.1</td>
<td>SSDA Correlation Routine (CSDA)</td>
<td>A-384</td>
</tr>
<tr>
<td>Section</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
<td>------</td>
</tr>
<tr>
<td>3.3</td>
<td>Library Routines</td>
<td>A-386</td>
</tr>
<tr>
<td>3.3.1</td>
<td>Draw a Box (DRABOX)</td>
<td>A-387</td>
</tr>
<tr>
<td>3.3.2</td>
<td>Determine the Mean (MEAN)</td>
<td>A-388</td>
</tr>
<tr>
<td>3.3.3</td>
<td>Position Cursor (POS)</td>
<td>A-390</td>
</tr>
<tr>
<td>3.4</td>
<td>I/O Routines</td>
<td>A-392</td>
</tr>
<tr>
<td>3.4.1</td>
<td>Joystick I/O Routines</td>
<td>A-393</td>
</tr>
<tr>
<td>3.4.1.1</td>
<td>Adjust Data (ADJUST)</td>
<td>A-394</td>
</tr>
<tr>
<td>3.4.1.2</td>
<td>Initialize the Joystick (INIJOY)</td>
<td>A-395</td>
</tr>
<tr>
<td>3.4.1.3</td>
<td>Read the Joystick (READJY)</td>
<td>A-396</td>
</tr>
<tr>
<td>3.4.2</td>
<td>Ramtek 9100 I/O Routines</td>
<td>A-397</td>
</tr>
<tr>
<td>3.4.2.1</td>
<td>RM9100 Read Image Window (RAMRDB)</td>
<td>A-398</td>
</tr>
<tr>
<td>3.4.2.2</td>
<td>RM9100 Write Image Window (RAMWRB)</td>
<td>A-399</td>
</tr>
<tr>
<td>3.4.2.3</td>
<td>Reset the RM9100 (RESET)</td>
<td>A-400</td>
</tr>
<tr>
<td>3.4.2.4</td>
<td>Check RM9100 I/O (RMCHCK)</td>
<td>A-401</td>
</tr>
<tr>
<td>3.4.2.5</td>
<td>RM9100 Read Image (RMRDIM)</td>
<td>A-402</td>
</tr>
<tr>
<td>3.4.2.6</td>
<td>Set Up RM9100 (RMSET)</td>
<td>A-403</td>
</tr>
<tr>
<td>3.4.2.7</td>
<td>Reset RM9100 Origin (RMSETO)</td>
<td>A-404</td>
</tr>
<tr>
<td>3.4.2.8</td>
<td>RM9100 Write Image (RMWRIM)</td>
<td>A-405</td>
</tr>
<tr>
<td>3.4.2.9</td>
<td>Issue Wait for Queue I/O (WIO)</td>
<td>A-406</td>
</tr>
</tbody>
</table>
1.0 INTRODUCTION

The Ground Control Point Simulation Program (GCPSIM) has been designed as an analysis tool to predict the performance of the navigation system illustrated in Figure A-1. The system consists of two star trackers, a GPS receiver, a gyro package, and a landmark tracker.
The program has been configured to provide an extensive error analysis rather than simply a covariance analysis. A covariance analysis provides a great deal of information but in many cases this information is inaccurate and misleading. For example, there are cases where the covariance matrix converges over a period of time while the actual state diverges. These types of instability are common in Kalmar filters and without direct knowledge of the state error conclusions can be inaccurate. For this reason, GCPSIM has been designed to model the true state of the vehicle using the dynamic equations of motion. By propagating the true state, it is not only possible to perform a covariance analysis, but to generate the actual state error and measurement residuals as well.

GCPSIM has been designed to accept either simulated GCP measurements or measurements that have been extracted directly from corrected Landsat imagery. These capabilities are included to allow analysis over long periods of time as well as to demonstrate the feasibility.

REPORT FORMAT

The intent of this document is to give visibility to the GCPSIM code. GCPSIM uses top-down structured programming aided by visual control logic representation (VCLR). A total system hierarchy diagram aids in depicting the relationship between the modules.

Each module is presented separately with a general discussion of the function performed followed by requirements, algorithm and process, VCLR, and source code (Fortran IV). The remainder of this section will describe the VCLR presentation aid.

Visual Control Logic Representation (VCLR)

VCLR charts give control logic which is compatible with structured programming. It offers many advantages and flowcharting.

1) Only the standard constructs are used.

2) The total scope and impact of the logic can be seen and easily understood.

3) No extraneous symbols, connections, or notations are used.

VCLR provides visible control logic representation which is a picture of a software design. Its primary purpose is to enable software engineers to express their thinking visually. Concentration is on the control logic of the design.

Standard constructs in visible control logic representations are the same as those for pseudo-code: SEQUENCE, IFTHENELSE, DOWHILE, DOUNTIL and DOCASE. Only the representations differ.
SEQUENCE - A SEQUENCE is simply one standard construct or one single statement followed by another. If "P1" and "P2" are standard constructs or single statements, the sequence would appear in a visible control logic representation as:

```
  P1
  P2
```

IFTHENELSE - Consists of a true/false test, and a path for each state. The true path appears on the left side, under the "T". One of the paths may be a "do-nothing" or "NULL" path. One or both paths must consist of a standard construct or of a single statement. If "Cl" is the condition being tested, "P1" is the true path and "P2" is the false path, the IFTHENELSE construct would be written as:

```
  T  Cl  F
  P1  P2
```

DOWHILE - The DOWHILE is a loop with these characteristics:
1) The counter or other item to be "incremented" is initialized before entering the loop.
2) The test is performed at the beginning of the loop. The conditions which must exist in order for the loop to be executed are the conditions which appear in the DOWHILE test.
3) The item to be executed must be a standard construct or a single statement.
4) The counter is incremented or other increment-like action is generally taken (e.g., another line is read) at the end of the loop.

If "Cl" is the condition which must exist for the loop to be executed, and "P1" is a standard construct or single statement, the DOWHILE would be written as follows:

```
  DOWHILE Cl
  P1
```
DOUNTIL - The DOUNTIL is a loop with these characteristics:
1) The counter or other item to be "incremented" is initialized before entering the loop.
2) The test is performed at the end of the loop. The conditions which must exist in order to exit from the loops are those which appear in the DOUNTIL test.
3) The item to be executed must be a standard construct or a single statement.
4) The counter is incremented or other increment-like action is generally taken (e.g., another line is read) at the beginning of the loop.

If "Cl" is the condition which must exist to exit from the loop, and "Pl" is a standard construct or single statement, the DOUNTIL would be written as follows:

```
    P1
   
  DOUNTIL Cl
```

DOCASE - Use the DOCASE construct when you want to execute a different set of statements for each of several different values of a variable.

If "Cl" is the variable being tested, and if "Cl" may have values of 1, 2 or 3, the construct appears as follows:

```
   DOCASE Cl

    P1    P2    P3    P4
    
    T    C1=1    F
    T    Cl=2    F
    T    Cl=3    F

    P1    P2    P3    P4
```

Example A

Example B

Example "A" is equivalent to the nested IF-THEN-ELSE form shown in "B".
TOTAL SYSTEM HIERARCHY DIAGRAM

The total system hierarchy diagram (Figure A-2) is designed to illustrate the functional structure of GCPSIM. The upper most block is the main program responsible for simulation control. The eleven blocks immediately below this type block represent the subroutines called by the main program. These programs in turn call subroutines or the next level down and so forth.

This report has been separated according to a functional decomposition of the program. Each of the eleven main subroutines are described in separate chapters along with all the subsequent tasks with which they interact. For example, the chapter describing the generate measurement module, MEASURE, is broken into three main sections corresponding to each of the measurement types generated.
Figure A-2
2.0 GCP - Executive Program

The executive program, GCP, is designed to control the entire simulation from data initialization to the end of a run at TSTOP. The main program provides for two modes of operation. The first mode models the measurements from each sensor whereas the second mode replaces the modeled GCP measurements with actual measurements taken from landsat data. Figure A-3 is a VCLR describing the process.

At the top level, the two modes of operation appear to be identical and will be discussed as such. First, simulator initialization is performed within INDATA. This process consists of loading the epoch state and time, and initializing all noise parameters, and setting up the measurement sequence tables. The sequencer, GCPSEQ, determines the type and time of the next measurement to be made. The types of measurements available are GPS, Star Tracker, and landmark tracker. The true vehicle position state is then propagated forward to measurement time by GENENV. GENENV also simulates the gyro output in small increments up to measurement time at TMEAS, and propagates the true attitude state. Within this propagation loop the simulated gyro outputs are processed by GYRO to compensate for gyro bias, scale factor, non-orthogonality, and misalignment, and to update the quaternion. In the second mode of operation this module controls the scrolling of imagery across a video monitor to simulate the sensor front end.

Following true state propagation the appropriate measurement is generated by MEASURE. MEASURE uses a geometry model and the true vehicle state to generate an ideal measurement. The ideal measurement is then corrupted with bias, noise, and misalignment, and compensated for knowledge of these terms. The primary difference between the two modes of operation occurs during measurement generation. In the second mode, the measurement generation module, MEAS, interacts with a separate program running on the PDP 11/70 which performs the GCP extraction.

The estimated state is propagated to measurement time by the module INTG. This propagation is actually part of the extended Kalman filter, but is kept in a separate module for efficiency. INTG also propagates the state transition matrix and the process noise forward in time.

EST uses the simulated measurement, the propagated state estimate, the process noise, and the state transition matrix to generate a new state estimate based on the latest measurement. The new state estimate and covariance matrix are printed, and the entire process beginning with measurement sequencing is repeated until the end of a simulation run identified by TSTOP. Prior to exiting, a plot file is created so that plots of state error, covariance, and measurement residual, may be obtained.
**GCPSIM Executive**

<table>
<thead>
<tr>
<th>1) Modeled GCP Measurements</th>
<th>2) Reconstructed GCP Measurements</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>GCPSIM</strong></td>
<td><strong>GCPSIM</strong></td>
</tr>
<tr>
<td>Read Initial Conditions, Sequence Tables, and Control Parameters</td>
<td>Sequence Operations, Determine Type &amp; Time of Next Measurement</td>
</tr>
<tr>
<td>Print Initial Conditions, Sequence Tables, and Control Parameters</td>
<td>Set New Measurement Flag to TRUE</td>
</tr>
<tr>
<td>Do Case Simulation Mode =</td>
<td>ENVIR Scroll Image Data to Measurement Time</td>
</tr>
<tr>
<td>1) Modeled GCP Measurements</td>
<td>ENVIR Integrate Actual State to Measurement Time, Generate Gyro Output</td>
</tr>
<tr>
<td>GCPSEQ Sequence Operations, Determine Type &amp; Time of Next Measurement</td>
<td>GYRO Process Gyro Output</td>
</tr>
<tr>
<td>Set New Measurement Flag to TRUE</td>
<td>Do Until Measurement Time TIME.EQ.TMEAS</td>
</tr>
<tr>
<td>GENENV Integrate Actual State to Measurement, Generate Gyro Output, Print Gyro Output</td>
<td>ENVIR Integrate Actual State to Measurement Time, Generate Gyro Output</td>
</tr>
<tr>
<td>GYRO Process Gyro Data</td>
<td>GYRO Process Gyro Output</td>
</tr>
<tr>
<td>Do Until Measurement Time TIME.EQ.TMEAS</td>
<td>Do Until Measurement Time TIME.EQ.TMEAS</td>
</tr>
<tr>
<td>MEASURE Generate Measurements &amp; Print</td>
<td>MEAS Generate GPS &amp; ST Measurements, Perform GCP Extraction</td>
</tr>
<tr>
<td>INTG Integrate Filter State to Measurement Time</td>
<td>INTG Integrate Filter State to Measurement Time</td>
</tr>
<tr>
<td>Do Until Measurement Time TIS.EQ.TMEAS</td>
<td>Do Until Measurement Time TIS.EQ.TMEAS</td>
</tr>
<tr>
<td>EST Perform State Estimation</td>
<td>EST Perform State Estimation</td>
</tr>
<tr>
<td>ICALL = 0 (Measurement Output)</td>
<td>ICALL = 0</td>
</tr>
<tr>
<td>OUTDATA Print Output Data</td>
<td>OUTDATA Print Output Data</td>
</tr>
<tr>
<td>Do Until Simulation End TIME.EQ.TSTOP</td>
<td>Do Until Simulation End TIME.EQ.TSTOP</td>
</tr>
<tr>
<td>GCPPLOT Generate Tape Plot</td>
<td>GCPPLOT Generate Tape Plot</td>
</tr>
<tr>
<td>End</td>
<td>End</td>
</tr>
</tbody>
</table>

Figure A-3 GCP VCLR A-11
PROGRAM GCP

INCLUDE 'CONTRL.COM'

COMMON /CONTOL/ MOP,TINT

REAL * TINT

PROGRAM CONTROL DESCRIPTORS FOR MULTIPLE RUNS

MOP MODE OF OPERATION
1 = PREFLIGHT SIMULATION
2 = POSTFLIGHT SIMULATION
3 = MONTE CARLO SIMULATION

TINT NUMBER OF SECONDS OF FULL OPERATION PER CYCLE

INCLUDE 'TIME.COM'

COMMON /TIME/ TIME,TNEXT,TSTOP,TIA,DEL,TIN,DTN,DATED,TZERO

REAL * TIME,TNEXT,TSTOP,TIA,DEL,TIN,DTN,DATED,TZERO

REAL * TMEAS,TRACK,TIS,TISN,DTA,DATER,TPRINT,DTPRINT

REAL * DEL,TLEW,TIA,TSLEW,DEL,TLEW

REAL * TMEAS,TRACK,TIS,TISN,DTA,DATER,TPRINT,DTPRINT

THESE ARE THE TIME REFERENCE FRAMES

TIME ATOMIC TIME SINCE INITIALIZATION (SEC)

TNEXT TIME FOR NEXT POSITION INTEGRATION (SEC)

TSTOP RUN TERMINATION TIME (SEC)

TIA ATTITUDE INTEGRATION TIME (SEC)

DEL STEP SIZE (SEC)

DLE STEP SIZE (SEC)

DTN STEP SIZE (SEC)

DATED DATE OF FLIGHT EPOCH (JD)

DATER DATE OF 1950 EPOCH (JD)

TZERO START TIME IN SECS SINCE DATED

TSLEW TIME NEEDED TO SLEW AND ACQUIRE (SEC)

TIS REAL WORLD REFERENCE TIME (SEC)

TISN TIME FOR NEXT RW POSITION INTEGRATION (SEC)

DTA TOO LARGE AT MEASUREMENT TIME

TPRINT TIME FOR PRINT (SEC)

DTPRINT INCREMENT ON TPRINT (SEC)

INCLUDE 'ASTATE.COM'

COMMON /ASTATE/ DE(4),E(4),WD(3),SF(3),D(3),DD(3)

REAL * DE,E,WD,SF,D,DD

ATTITUDE STATE AND CONSIDERED PARAMETERS

D DIFFERENTIAL OF QUATERNIONS

E QUATERNIONS

WD GYRO DRIFT RATE (RAD/SEC)

SF GYRO SCALE FACTOR

D GYRO NON-ORTHOGONALITY (RAD)

DD GYRO RELATIVE ORIENTATION (RAD)

INCLUDE 'ROTAT.COM'

COMMON /ROTAT/ DTHR(3),DTHEM(3),DTHE(3)

REAL * DTHR,DTHEM,DTHE
GYRO ATTITUDE PARAMETERS

DTHR  REAL WORLD GYRO DATA (RAD)
DTHE  FILTER WORLD GYRO DATA (RAD)
DTHM  FILTER WORLD COMPENSATED GYRO DATA (RAD)

INCLUDE 'NOIS..COM'

COMMON /NOISE/ BWD(3),SWD(3),BSF(3),SSF(3),BD(3),SD(3)

REAL*B WDD,SWD,BSF,SSF,BD,SD,DD,DD,SRM,BRE,SRE

REAL WORLD GYRO MEASUREMENT ERRORS

BWD  SWD   *  GYRO DRIFT (RAD / SEC)
BSF   SSF   *  GYRO SCALE FACTOR
BD    SD    *  GYRO NONORTHOGONALITY (RAD)
DD    DD    *  GYRO RELATIVE ORIENTATION (RAD)

INCLUDE 'MSTATE.COM'

COMMON /MSTATE/ XD(6),X(6),RAOM,RADE

REAL*B XD,X,RAOM,RADE

POSITION STATE AND CONSIDERED PARAMETERS

XD  STATE DERIVATIVES (KM/SEC AND KM/SEC/SEC)
X   STATE POSITION PARAMETERS (KM AND KM/SEC)
RAOM RADIUS OF THE MOON (KM)
RADE EARTH DETECTABLE RADIUS (KM)

INCLUDE 'DEBUG.COM'

COMMON /DEBUG/ ENTER,IDEBUG

USER CONTROLLED PARAMETERS TO VARY DEBUG PRINT LEVEL

ENTER  IF 1, PRINTS WHEN ENTERS MOST SUBROUTINES
IDEBUG 0-10, HIGHER NUMBER MEANS MORE PRINT

INCLUDE 'PHIA..COM'

COMMON /PHIA/ PA(4,4),TA(4,12),PDA(4,16),PHIA(16,16),

COVA(16,16),POA(16,16),QMAX

REAL*B PA,TA,PDA,PHIA,COVA,PDA,QMAX

THESE ARE THE ATTITUDE TRANSITION AND COVARIANCE MATRICES

PA  ATTITUDE STATE TRANSITION MATRIX
TA  PARAMETER TRANSITION MATRIX
PDA DERIVATIVE OF TRANSITION MATRICES
PDA AGGREGATE TRANSITION MATRIX
COVA NEW COVARIANCE MATRIX
POA PREVIOUS COVARIANCE MATRIX
QMAX COVARIANCE NORM MAX

INCLUDE 'PHIN..COM'

COMMON /PHIN/ Pn(6,6),PDN(6,6),PHIN(6,6),COVPN(6,6).
GCP

REAL*8 PN,PDN,PHIN,COVN,PDN

THESE ARE THE NAVIGATION TRANSITION AND COVARIANCE ARRAYS

PN   POSITION STATE TRANSITION MATRIX
PON   DERIVATIVE OF TRANSITION MATRIX
PHIN   AGGREGATE TRANSITION MATRIX
COVN   NEW COVARIANCE MATRIX
PON   PREVIOUS COVARIANCE MATRIX

INCLUDE 'ARRAYS.COM'

COMMON /ARRAYS/ T1(3),T2(3),T3(3),T4(10),T11(3,3),T33(3,3)

REAL*8 T1,T2,T4,T11,T33,T44,T66,T77,T5,T6,T7

THESE ARE TEMPORARY STORAGE ARRAYS FOR USE BY ALL MODULES

T1 - T4   SINGLE DIMENSION ARRAYS
T11 - T77   DUAL DIMENSIONED ARRAYS
T11   DUAL ARRAY: OFF DIAGONAL SET TO ZERO

INCLUDE 'UPDT.COM'

COMMON /UPDT/ QN(6),QA(16),Q(6,6),QDOT(6,6)

REAL*8 QN,QA,Q,QDOT

STATE ESTIMATION PARAMETERS

QN   NAV. DYN. NOISE COVARIANCE DIAGONAL
QA   MIN. VALUES FOR ATT. COVARIANCE DIAGONAL
Q   CONTRIBUTION TO NAV. COV. FOR DYN. NOISE
QDOT   DIFFERENTIAL OF Q

INCLUDE 'RESIDUALS.COM'

COMMON /RESIDUALS/DZHM,DZVL,DZVST1,DZVST2,DZVST3,DXMGPS(6)

REAL*8 DZHM,DZVL,DZVST1,DZVST2,DZVST3,DXMGPS

INCLUDE 'TITL.COM'

COMMON /TITLE/ ATITLE(40)

LOGICAL=1 ATITLE

ATITLE: IS THE TITLE PRINTED AT EACH PRINT TIME
AS WELL AS THE TOP OF EACH PLOT

LOGICAL=1 ATITLE

DIMENSION IDTHEM(3)
OPEN(UNIT=6, RECORDSIZEx132, TYPE=’NEW’)
OPEN(UNIT=12, RECORDSIZEx780, TYPE=’NEW’)

WRITE(5,2)

FORMAT(’INPUT RUN TITLE (MAX. 40 CHAR) : ’,S)

READ(5,4)(ATITLE(I),I=1,40)

FORMAT(40A1)

OPEN(UNIT=13, NAME=’GAIN.DAT’, RECORDSIZEx98, TYPE=’NEW’)

CALL INDATA

WRITE(12,4)(ATITLE(I),I=1,40)

WRITE(6,5)(ATITLE(I),I=1,40)

CALL INICAL

PRINT PARAMETERS

CALL ONAV(TIM.,X,XD,IDUMMY)

SETUP FORCE MODELS

CALL GCPSEQ I SEQUENCE MEASUREMENTS

Flag = FALSE. ! INITIALIZE NEW MEAS. FLAG

COMPUTE TIME OF MEASUREMENT

PROPAGATE TRUE VEHICLE STATE AND PROCESS

GYRO DATA TO MEASUREMENT TIME

CALL GENENV(Flag) ! GENERATE GYRO OUTPUT

CALL CYRO

IF(TIME + .01 .LT. TIME) GO TO 20

CALL MEASURE

CALL INTEGRATE FILTER STATE TO MEASUREMENT TIME

CALL EST
ICALL = 0

PRINT OUTPUT DATA

CALL OUTDATA(ICALL)

PROCESS MEASUREMENTS UNTIL TIME+TSTOP

IF(TIME.LT.TSTOP) GO TO 10

GENERATE PLOT TAPE

CALL PLTDATA

GO TO 110

POSTFLIGHT SIMULATION

IGCP = .FALSE.

CALL GCPSEQ

FLAG = .FALSE.

COMPUTE TIME OF MEASUREMENT

PROPAGATE TRUE VEHICLE STATE AND PROCESS

GYRO DATA TO MEASUREMENT TIME

CALL GENENV(FLAG)

1 GENERATE GYRO OUTPUT

CALL GYRO

IF(TIME+.01.LT.TMEAS) GO TO 220

GENERATE MEASUREMENT

CALL MEAS

INTEGRATE FILTER STATE TO MEASUREMENT TIME

CALL INTG

IF(TIN+.01.LT.TMEAS) GO TO 230

PERFORM STATE ESTIMATION

CALL EST

ICALL = 0

PRINT OUTPUT DATA

CALL OUTDATA(ICALL)

PROCESS MEASUREMENTS UNTIL TIME+TSTOP

IF(TIME.LT.TSTOP) GO TO 40

GENERATE PLOT TAPE

CALL PLTDATA
GCP

0082 110  CONTINUE
0083  CLOSE(UNIT=6,DISPOSE='SAVE')
0084  CLOSE(UNIT=13,DISPOSE='SAVE')
0085  CLOSE(UNIT=12,DISPOSE='SAVE')
0086  CALL EXIT
0087  END

PROGRAM SECTIONS

<table>
<thead>
<tr>
<th>Name</th>
<th>Bytes</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 CODE</td>
<td>454</td>
<td>PIC CON REL LCL SHR EXE RD NO WRT LONG</td>
</tr>
<tr>
<td>1 SPDATA</td>
<td>24</td>
<td>PIC CON REL LCL SHR NOEXE RD NO WRT LONG</td>
</tr>
<tr>
<td>2 LOCAL</td>
<td>184</td>
<td>PIC CON REL LCL NO SHR NOEXE RD WRT LONG</td>
</tr>
<tr>
<td>3 CONTROL</td>
<td>12</td>
<td>PIC JVR REL GBL SHR NOEXE RD WRT LONG</td>
</tr>
<tr>
<td>4 TIME</td>
<td>136</td>
<td>PIC JVR REL GBL SHR NOEXE RD WRT LONG</td>
</tr>
<tr>
<td>5 ASTATE</td>
<td>160</td>
<td>PIC JVR REL GBL SHR NOEXE RD WRT LONG</td>
</tr>
<tr>
<td>6 ROTAT</td>
<td>72</td>
<td>PIC OVR REL GBL SHR NOEXE RD WRT LONG</td>
</tr>
<tr>
<td>7 NOISE</td>
<td>216</td>
<td>PIC OVR REL GBL SHR NOEXE RD WRT LONG</td>
</tr>
<tr>
<td>8 NSSTATE</td>
<td>112</td>
<td>PIC OVR REL GBL SHR NOEXE RD WRT LONG</td>
</tr>
<tr>
<td>9 DEBUG</td>
<td>8</td>
<td>PIC OVR REL GBL SHR NOEXE RD WRT LONG</td>
</tr>
<tr>
<td>10 PHIA</td>
<td>7176</td>
<td>PIC OVR REL GBL SHR NOEXE RD WRT LONG</td>
</tr>
<tr>
<td>11 PHIN</td>
<td>1440</td>
<td>PIC OVR REL GBL SHR NOEXE RD WRT LONG</td>
</tr>
<tr>
<td>12 ARRAYS</td>
<td>1096</td>
<td>PIC OVR REL GBL SHR NOEXE RD WRT LONG</td>
</tr>
<tr>
<td>13 UPDT</td>
<td>752</td>
<td>PIC OVR REL GBL SHR NOEXE RD WRT LONG</td>
</tr>
<tr>
<td>14 RESIDUALS</td>
<td>96</td>
<td>PIC OVR REL GBL SHR NOEXE RD WRT LONG</td>
</tr>
<tr>
<td>15 TITLE</td>
<td>40</td>
<td>PIC OVR REL GBL SHR NOEXE RD WRT LONG</td>
</tr>
</tbody>
</table>

ENTRY POINTS

<table>
<thead>
<tr>
<th>Address</th>
<th>Type</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-00000000</td>
<td>GCP</td>
<td></td>
</tr>
</tbody>
</table>

VARIABLES

<table>
<thead>
<tr>
<th>Address</th>
<th>Type</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>7-00000008</td>
<td>R+B</td>
<td>BRE</td>
</tr>
<tr>
<td>4-00000034</td>
<td>R+B</td>
<td>DATE0</td>
</tr>
<tr>
<td>4-00000070</td>
<td>R+B</td>
<td>DATER</td>
</tr>
<tr>
<td>4-00000000</td>
<td>R+B</td>
<td>DEL</td>
</tr>
<tr>
<td>4-00000000</td>
<td>R+B</td>
<td>DZST1</td>
</tr>
<tr>
<td>14-00000000</td>
<td>R+B</td>
<td>DZST2</td>
</tr>
<tr>
<td>2-00000000</td>
<td>R+B</td>
<td>FLAG</td>
</tr>
<tr>
<td>2-00000000</td>
<td>R+B</td>
<td>IDUMMY</td>
</tr>
<tr>
<td>9-00000000</td>
<td>R+B</td>
<td>MUP</td>
</tr>
<tr>
<td>10-00000000</td>
<td>R+B</td>
<td>OMAX</td>
</tr>
<tr>
<td>8-00000000</td>
<td>R+B</td>
<td>PADM</td>
</tr>
<tr>
<td>7-00000000</td>
<td>R+B</td>
<td>SRE</td>
</tr>
<tr>
<td>4-00000000</td>
<td>R+B</td>
<td>TIME</td>
</tr>
<tr>
<td>4-00000000</td>
<td>R+B</td>
<td>TIN</td>
</tr>
<tr>
<td>4-00000000</td>
<td>R+B</td>
<td>TIS</td>
</tr>
<tr>
<td>4-00000000</td>
<td>R+B</td>
<td>TRACK</td>
</tr>
<tr>
<td>4-00000000</td>
<td>R+B</td>
<td>TSTOP</td>
</tr>
<tr>
<td>4-00000000</td>
<td>R+B</td>
<td>TZERO</td>
</tr>
<tr>
<td>4-00000000</td>
<td>R+B</td>
<td>TPRINT</td>
</tr>
</tbody>
</table>

VAX-11 FORTRAN V2.0-2
<table>
<thead>
<tr>
<th>Address</th>
<th>Type</th>
<th>Name</th>
<th>Bytes</th>
<th>Dimensions</th>
</tr>
</thead>
<tbody>
<tr>
<td>15-00000000</td>
<td>L+1</td>
<td>ATITLE</td>
<td>40 (40)</td>
<td></td>
</tr>
<tr>
<td>7-00000060</td>
<td>R+8</td>
<td>BD</td>
<td>24 (3)</td>
<td></td>
</tr>
<tr>
<td>7-00000090</td>
<td>R+8</td>
<td>BOD</td>
<td>24 (3)</td>
<td></td>
</tr>
<tr>
<td>7-00000030</td>
<td>R+8</td>
<td>BSF</td>
<td>24 (3)</td>
<td></td>
</tr>
<tr>
<td>7-00000200</td>
<td>R+8</td>
<td>BWD</td>
<td>24 (3)</td>
<td></td>
</tr>
<tr>
<td>10-00000000</td>
<td>R+8</td>
<td>COVA</td>
<td>2048 (16, 16)</td>
<td></td>
</tr>
<tr>
<td>11-00000360</td>
<td>R+8</td>
<td>COVN</td>
<td>288 (6, 6)</td>
<td></td>
</tr>
<tr>
<td>5-00000070</td>
<td>R+8</td>
<td>D</td>
<td>24 (3)</td>
<td></td>
</tr>
<tr>
<td>5-00000088</td>
<td>R+8</td>
<td>DD</td>
<td>24 (3)</td>
<td></td>
</tr>
<tr>
<td>5-00000000</td>
<td>R+8</td>
<td>DE</td>
<td>32 (4)</td>
<td></td>
</tr>
<tr>
<td>6-00000030</td>
<td>R+8</td>
<td>DTHE</td>
<td>24 (3)</td>
<td></td>
</tr>
<tr>
<td>6-00000018</td>
<td>R+8</td>
<td>DTHEM</td>
<td>24 (3)</td>
<td></td>
</tr>
<tr>
<td>6-00000000</td>
<td>R+8</td>
<td>DTHR</td>
<td>24 (3)</td>
<td></td>
</tr>
<tr>
<td>14-00000030</td>
<td>R+8</td>
<td>DXMLPS</td>
<td>46 (6)</td>
<td></td>
</tr>
<tr>
<td>5-00000020</td>
<td>R+8</td>
<td>E</td>
<td>32 (4)</td>
<td></td>
</tr>
<tr>
<td>2-00000000</td>
<td>I+4</td>
<td>IDTHEM</td>
<td>12 (3)</td>
<td></td>
</tr>
<tr>
<td>10-00000300</td>
<td>R+8</td>
<td>PA</td>
<td>128 (4, 4)</td>
<td></td>
</tr>
<tr>
<td>10-00000200</td>
<td>R+8</td>
<td>PDA</td>
<td>512 (4, 16)</td>
<td></td>
</tr>
<tr>
<td>10-00000120</td>
<td>R+8</td>
<td>PON</td>
<td>288 (6, 6)</td>
<td></td>
</tr>
<tr>
<td>10-00000400</td>
<td>R+8</td>
<td>PHIA</td>
<td>2048 (16, 16)</td>
<td></td>
</tr>
<tr>
<td>11-00000240</td>
<td>R+8</td>
<td>PHIN</td>
<td>288 (6, 6)</td>
<td></td>
</tr>
<tr>
<td>11-00000000</td>
<td>R+8</td>
<td>PN</td>
<td>288 (6, 6)</td>
<td></td>
</tr>
<tr>
<td>10-00010000</td>
<td>R+8</td>
<td>POA</td>
<td>2048 (16, 16)</td>
<td></td>
</tr>
<tr>
<td>11-00000480</td>
<td>R+8</td>
<td>PON</td>
<td>288 (6, 6)</td>
<td></td>
</tr>
<tr>
<td>13-00000080</td>
<td>R+8</td>
<td>Q</td>
<td>288 (6, 6)</td>
<td></td>
</tr>
<tr>
<td>13-00000000</td>
<td>R+8</td>
<td>QA</td>
<td>288 (6, 6)</td>
<td></td>
</tr>
<tr>
<td>13-00000100</td>
<td>R+8</td>
<td>QDOT</td>
<td>288 (6, 6)</td>
<td></td>
</tr>
<tr>
<td>13-00000000</td>
<td>R+8</td>
<td>QN</td>
<td>48 (6)</td>
<td></td>
</tr>
<tr>
<td>7-00000078</td>
<td>R+8</td>
<td>SD</td>
<td>24 (3)</td>
<td></td>
</tr>
<tr>
<td>7-00000030</td>
<td>R+8</td>
<td>SOD</td>
<td>24 (3)</td>
<td></td>
</tr>
<tr>
<td>5-00000058</td>
<td>R+8</td>
<td>SF</td>
<td>24 (3)</td>
<td></td>
</tr>
<tr>
<td>7-00000048</td>
<td>R+8</td>
<td>SSF</td>
<td>24 (3)</td>
<td></td>
</tr>
<tr>
<td>7-00000018</td>
<td>R+8</td>
<td>SwD</td>
<td>24 (3)</td>
<td></td>
</tr>
<tr>
<td>12-00000000</td>
<td>R+8</td>
<td>T1</td>
<td>24 (3)</td>
<td></td>
</tr>
<tr>
<td>12-00000090</td>
<td>R+8</td>
<td>T11</td>
<td>72 (3, 3)</td>
<td></td>
</tr>
<tr>
<td>12-00000018</td>
<td>R+8</td>
<td>T2</td>
<td>24 (3)</td>
<td></td>
</tr>
<tr>
<td>12-00000030</td>
<td>R+8</td>
<td>T3</td>
<td>24 (3)</td>
<td></td>
</tr>
<tr>
<td>12-00000000</td>
<td>R+8</td>
<td>T33</td>
<td>72 (3, 3)</td>
<td></td>
</tr>
<tr>
<td>12-00000048</td>
<td>R+8</td>
<td>T4</td>
<td>80 (10)</td>
<td></td>
</tr>
<tr>
<td>12-00000128</td>
<td>R+8</td>
<td>T44</td>
<td>128 (4, 4)</td>
<td></td>
</tr>
<tr>
<td>12-00000338</td>
<td>R+8</td>
<td>T5</td>
<td>32 (4)</td>
<td></td>
</tr>
<tr>
<td>12-00000408</td>
<td>R+8</td>
<td>T6</td>
<td>32 (4)</td>
<td></td>
</tr>
<tr>
<td>12-00000148</td>
<td>R+8</td>
<td>T68</td>
<td>288 (6, 6)</td>
<td></td>
</tr>
<tr>
<td>12-00000468</td>
<td>R+8</td>
<td>T7</td>
<td>32 (4)</td>
<td></td>
</tr>
<tr>
<td>12-000002CB</td>
<td>R+8</td>
<td>T77</td>
<td>288 (6, 6)</td>
<td></td>
</tr>
<tr>
<td>10-00000080</td>
<td>R+8</td>
<td>TA</td>
<td>384 (4, 12)</td>
<td></td>
</tr>
<tr>
<td>5-00000040</td>
<td>R+8</td>
<td>WD</td>
<td>24 (3)</td>
<td></td>
</tr>
<tr>
<td>8-00000030</td>
<td>R+8</td>
<td>X</td>
<td>48 (6)</td>
<td></td>
</tr>
<tr>
<td>8-00000000</td>
<td>R+8</td>
<td>XD</td>
<td>48 (6)</td>
<td></td>
</tr>
</tbody>
</table>
GCP

LABELS

<table>
<thead>
<tr>
<th>Address</th>
<th>Label</th>
<th>Address</th>
<th>Label</th>
<th>Address</th>
<th>Label</th>
<th>Address</th>
<th>Label</th>
<th>Address</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-00000005</td>
<td>5'</td>
<td>0-0000000A4</td>
<td>10</td>
<td>0-000000AD</td>
<td>20</td>
<td>0-000000DD</td>
<td>30</td>
<td>0-00000125</td>
<td>40</td>
</tr>
</tbody>
</table>

FUNCTIONS AND SUBROUTINES REFERENCED

DNAXV

INDATA

EST

FOR$CLOSE

FOR$EXIT

FOR$OPEN

GCPSERQ

GENENV

GYRO

Total Space Allocated = 11978 Bytes

COMMAND QUALIFIERS

FORTRAN /L.GCP

/CHECK=(NOBOUNDS,OVERFLOW)

/DEBUG=(NOSYMBOLS,TRACEBACK)

/F77 /NOG_FLOATING /14 /OPTIMIZE /WARNINGS /NOD_LINES /NMACHINE_CODE /CONTINUATIONS=19

COMPILATION STATISTICS

Run Time: 3.00 seconds

Elapsed Time: 12.04 seconds

Page Faults: 196

Dynamic Memory: 51 pages
2.1 INITIALIZE DATA MODULE (INDATA)

The Initialize Data module shall perform the basic functions of the telemetry up-link to the spacecraft. It will initialize and reinitialize both attitude and navigation states; the lunar ephemeris, the command sequence tables, and the time reference. When the sequence tables are reinitialized, a flag is set and the sequencing reverts to the beginning of the tables.

A data file is read so that only changes to the standard base need to be input. In addition to inputing data this module will transform attitude data in terms of Euler angles (more conventional) to identical attitude data in terms of quaternions.

The attitude parameter transformation is:

\[
\begin{align*}
e_0 &= \cos \frac{x}{2} \cos \frac{\theta}{2} \cos \frac{\phi}{2} + \sin \frac{x}{2} \sin \frac{\theta}{2} \sin \frac{\phi}{2} \\
e_1 &= \cos \frac{x}{2} \cos \frac{\theta}{2} \sin \frac{\phi}{2} - \sin \frac{x}{2} \sin \frac{\theta}{2} \cos \frac{\phi}{2} \\
e_2 &= \cos \frac{x}{2} \sin \frac{\theta}{2} \cos \frac{\phi}{2} + \sin \frac{x}{2} \cos \frac{\theta}{2} \sin \frac{\phi}{2} \\
e_3 &= \sin \frac{x}{2} \cos \frac{\theta}{2} \cos \frac{\phi}{2} - \cos \frac{x}{2} \sin \frac{\theta}{2} \sin \frac{\phi}{2}
\end{align*}
\]

The coordinate transformation for the covariance matrix \( P \) is computed by:

\[
Pe_0 - e_3 = \phi P \phi^T
\]

Where, the transition matrix \( \phi \) is computed from the above quaternion - Euler angle relationship:

\[
\phi = \begin{bmatrix}
\begin{array}{ccc}
\frac{3e_1}{3x} & \frac{3e_1}{3\theta} & \frac{3e_1}{3\phi} \\
\frac{3e_1}{3x} & \frac{3e_1}{3\theta} & \frac{3e_1}{3\phi} \\
\frac{3e_2}{3x} & \frac{3e_2}{3\theta} & \frac{3e_2}{3\phi} \\
\frac{3e_3}{3x} & \frac{3e_3}{3\theta} & \frac{3e_3}{3\phi}
\end{array}
\end{bmatrix}
\]
**INDATA VCLR**

<table>
<thead>
<tr>
<th>READ INPUT DATA FROM DATA FILE</th>
</tr>
</thead>
<tbody>
<tr>
<td>SET UP QUATERNIAN FROM EULER ANGLES</td>
</tr>
<tr>
<td>SET UP INITIAL NAVIGATION PROCESS NOISE</td>
</tr>
<tr>
<td><strong>COMPUTE COVARIANCE BY EULER ANGLES</strong></td>
</tr>
<tr>
<td><strong>T</strong></td>
</tr>
<tr>
<td>COMPUTE COVARIANCE BY EULER ANGLES</td>
</tr>
<tr>
<td>PRINT DATA BASE</td>
</tr>
<tr>
<td>INITIALIZE TIMES</td>
</tr>
</tbody>
</table>

Figure A-4
SUBROUTINE INDATA

INCLUDE 'DEBUG.COM'

COMMON /DEBUG/ ENTER,DEBUG

USER CONTROLLED PARAMETERS TO VARY DEBUG PRINT LEVEL

ENTER IF 1, PRINTS WHEN ENTERS MOST SUBROUTINES
DEBUG 0-10, HIGHER NUMBER MEANS MORE PRINT

INCLUDE 'CTRL.COM'

COMMON /CTRL/ MOP,TINT

REAL*B TINT

PROGRAM CONTROL DESCRIPTORS FOR MULTIPLE RUNS

MOP MODE OF OPERATION
1 = PREFLIGHT SIMULATION
2 = POSTFLIGHT SIMULATION
3 = MONTE CARLO SIMULATION

TINT NUMBER OF SECONDS OF FULL OPERATION PER CYCLE

INCLUDE 'ENVIR.COM'

COMMON /ENVIR/ STATE(10),PROFILE(10,4),INIT

REAL*B STATE,PROFILE

REAL STATE

PARAMETERS
STATE VALUES; X,Y,Z,XD,YD,LD,EO,ET,E2,E3

PROFILE ATTITUDE PROFILE-TIME (SEC) VS
INERTIAL ANGULAR RATES (RAD/SEC)

INIT INTEGRATION INITIALIZATION KEY (-1)

INCLUDE 'GFPART.COM'

COMMON/GFPART/ FA(4,4),EA(4,12),FN(6,6)

REAL*B FA,EA,FN

MEASUREMENT AND STATE PARTIALS

FA ATTITUDE STATE PARTIALS
EA CONSIDERED PARAMETERS PARTIALS
FN STATE PARTIALS

INCLUDE 'NOISE.COM'

COMMON /NOISE/ BWD(3),SWD(3),BSF(3),SSF(3),BD(3),SD(3)

REAL*B BWD,SWD,BSF,SSF,BD,SD,BDD,SDD,SRM,BRE,SRE

REAL WORLD GYRO MEASUREMENT ERRORS

BWD SWD * * GYRO DRIFT (RAD/SEC)
BSF SSF * * GYRO SCALE FACTOR
BD SD * * GYRO NONORTHOGONALITY (RAD)
BDD SDD * * GYRO RELATIVE ORIENTATION (RAD)

INCLUDE 'TARGETS.COM'

COMMON /TARGETS/ MTYPE,IS,NS,UFLAG,MCODE,PI,TPI

LOGICAL UFLAG
REAL*8 PI,TPI  

MEASUREMENT SPECIFICATIONS  

MTYPE  MEASUREMENT TYPE  
JFLAG  SET FOR STAR OBSTRUCTION  
MCODE  MEASUREMENT PROCESSING  
PI  PI  
TPI  2*PI  

INCLUDE 'ASTATE.COM'  

COMMON /ASTATE/ DE(4),E(4),WD(3),SF(3),D(3),GO(3)  
REAL*8 DE,E,WD,SF,D,GO  

ATTITUDE STATE AND CONSIDERED PARAMETERS  

D  DIFFERENTIAL OF QUATERNIONS  
E  QUATERNIONS  
WD  GYRO DRIFT RATE (RAD/SEC)  
SF  GYRO SCALE FACTOR  
D  GYRO NON-ORTHOGONALITY (RAD)  
GO  GYRO RELATIVE ORIENTATION (RAD)  

INCLUDE 'FILTER.COM'  

COMMON/FILTER/ IPN(6),IPA(16)  
IPN WAS IPN(11)  

FILTER DATA CONSTANTS  

IPN  ARRAY INDEX OF ESTIMATED POS PARAMETERS  
IPA  ATT PARAMETERS  

INCLUDE 'SEOU.COM'  

COMMON/SEOU/I S(25),I TS(4,5),I AA(3,4),I FLAG 
..,DST,DGPS,IDELAY(11)  
REAL*8 DST,DGPS  
LOGICAL IFLAG  

SEQUENCE OF EVENTS TABLES  

I S  EVENT SCHEDULAR  
I TS  CYCLE SEQUENCER  
I AA  ALTERNATE ACTIONS  
I FLAG  SET WHEN TARGET IS OBSTRUCTED  

INCLUDE 'TMAT.COM'  

COMMON /TMAT/ A(3,3),B(3,3),C(3,3),EM(4,3)  
REAL*8 A,B,C,EM  

TRANSFORMATION MATRICES  

A  INERTIAL TO BODY AXES  
B  GYRO TO BODY AXES  
C  GYRO NON-ORTHOGONAL TO GYRO AXES  
EM  BODY TO QUATERNION AXES  

INCLUDE 'NSTATE.COM'
COMMON /STATE/ XD(6),X(6),RADE,RAOE
REAL*8 XD,X,RADE,RAOE

POSITIVE STATE AND CONSIDERED PARAMETERS

STATE DERIVATIVES (KM/SEC AND KM/SEC/SEC)
STATE POSITION PARAMETERS (KM AND KM/SEC)
RADIUS OF THE MOON (KM)
EARTH DETECTABLE RADIUS (KM)

INCLUDE 'ARRAYS.COM'

COMMON /ARRAYS/ T1(3),T2(3),T3(3),T4(10),T11(3,3),T33(3,3)
REAL*8 T1,T2,T3,T4,T11,T33,T44,T66,T77,T5,T6,T7

THESE ARE TEMPORARY STORAGE ARRAYS FOR USE BY ALL MODULES
SINGLE DIMENSION ARRAYS
DUAL DIMENSIONED ARRAYS
DUAL ARRAY; OFF DIAGONAL SET TO ZERO

INCLUDE 'TIME.COM'

COMMON /TIME/ TIME,TNEXT,TSTOP,TIA,DEL,TIN,DTN,DATEO,TZERO

REAL*8 TRACK,TIS,TISN,DATA,DATER,TPRINT,DTPRINT

INCLUDE 'CONST.COM'

COMMON /CONST/ ATM, RBM, RBE, RBO, RE2, RR2, MM, US, UE, J2, J3, J4, DTU, PK1

REAL*8 ATM, RBM, RBE, RBO, RE2, RR2, MM, US, UE, J2, J3, J4, DTU, PK1

PROGRAM CONSTANTS
<table>
<thead>
<tr>
<th>INDATA</th>
<th>01600</th>
<th>C</th>
<th>ATM</th>
<th>S/C AREA TO MASS RATIO (METEES/KG)</th>
</tr>
</thead>
<tbody>
<tr>
<td>01600</td>
<td>C</td>
<td></td>
<td>RBE</td>
<td>OBSTRUCTION RADIUS OF THE MOON (KM)</td>
</tr>
<tr>
<td>01600</td>
<td>C</td>
<td></td>
<td>R.2</td>
<td>SQUARE OF THE EARTHS RADIUS (KM 2)</td>
</tr>
<tr>
<td>01600</td>
<td>C</td>
<td></td>
<td>RMD</td>
<td>LUNAR RADIUS (KM 2)</td>
</tr>
<tr>
<td>01600</td>
<td>C</td>
<td></td>
<td>UM</td>
<td>LUNAR GRAVITATION CONSTANT (KM 3/SEC 2)</td>
</tr>
<tr>
<td>01600</td>
<td>C</td>
<td></td>
<td>US</td>
<td>SOLAR</td>
</tr>
<tr>
<td>01600</td>
<td>C</td>
<td></td>
<td>U.</td>
<td>EARTH</td>
</tr>
<tr>
<td>01600</td>
<td>C</td>
<td></td>
<td>U2,U3,U4</td>
<td>ZONAL GRAVITATIONAL HARMONIC TERMS</td>
</tr>
<tr>
<td>01600</td>
<td>C</td>
<td></td>
<td>DTU</td>
<td>REGULARIZED TIME STEP SIZE (SEC)</td>
</tr>
<tr>
<td>01600</td>
<td>C</td>
<td></td>
<td>PKI</td>
<td>SOLAR PRESSURE CONSTANT</td>
</tr>
</tbody>
</table>

### INCLUDE 'UPDT.COM'

```plaintext
COMMON /UPDT/ QN(6),QA(16),Q(6,6),QDOT(6,6)
REAL*8 QN,QA,Q,QDOT
```

### STATE ESTIMATION PARAMETERS

```plaintext
QN NAV. DYN. NOISE COVARIANCE DIAGONAL
QA MIN. VALUES FOR ATT. COVARIANCE DIAGONAL
Q O CONTRIBUTION TO NAV. COV. FOR DYN. NOISE
QDOT DIFFERENTIAL OF Q
```

### INCLUDE 'PLOT.COM'

```plaintext
COMMON /PLOT/ TP1,TP2
REAL*8 TP1,TP2
```

### PLOTTING INFORMATION

```plaintext
TP1 LOWER ABSCISSA VALUE - TIME (MIN)
TP2 UPPER " " " "
```

### INCLUDE 'PHIA.COM'

```plaintext
COMMON /PHIA/ PA(4,4),TA(4,12),PDA(4,16),PHIA(16,16),COVA(16,16),POA(16,16),QMAX
REAL*8 PA,TA,PDA,PHIA,COVA,POA,QMAX
```

### THESE ARE THE ATTITUDE TRANSITION AND COVARIANCE MATRICES

```plaintext
PA ATTITUDE STATE TRANSITION MATRIX
TA PARAMETER TRANSITION MATRIX
PDA DERIVATIVE OF TRANSITION MATRICES
PHIA AGGREGATE TRANSITION MATRIX
COVA NEW COVARIANCE MATRIX
QMAX PREVIOUS COVARIANCE MATRIX
```

### INCLUDE 'PHIN.COM'

```plaintext
COMMON /PHIN/ PN(6,6),PDN(6,6),PHIN(6,6),COVN(6,6)
REAL*8 PN,PDN,PHIN,COVN,PDN
```

### THESE ARE THE NAVIGATION TRANSITION AND COVARIANCE ARRAYS

```plaintext
01600   0044   C   INCLUDE 'UPDT.COM'
01700   0045   C   COMMON /UPDT/ QN(6),QA(16),Q(6,6),QDOT(6,6)
01700   0046   C   REAL*8 QN,QA,Q,QDOT
01700   0047   C   INCLUDE 'PLOT.COM'
01800   0048   C   COMMON /PLOT/ TP1,TP2
01800   0049   C   REAL*8 TP1,TP2
01800   0050   C   INCLUDE 'PHIA.COM'
01800   0051   C   COMMON /PHIA/ PA(4,4),TA(4,12),PDA(4,16),PHIA(16,16),COVA(16,16),POA(16,16),QMAX
01800   0052   C   REAL*8 PA,TA,PDA,PHIA,COVA,POA,QMAX
01800   0053   C   INCLUDE 'PHIN.COM'
01800   0054   C   COMMON /PHIN/ PN(6,6),PDN(6,6),PHIN(6,6),COVN(6,6)
01800   0055   C   REAL*8 PN,PDN,PHIN,COVN,PDN
01800   0056   C   INCLUDE 'PHIN.COM'
```
STAR TRACKER PARAMETERS

BEFORE EACH CASE THE LAST SUBSCRIPT REFERS TO THE TRACKER USED

BS = BIAS - ACTUAL (RAD)
SS = NOISE STANDARD DEVIATION - ACTUAL (RAD)
TNS = MISALIGNMENT ARRAY - TRANSFORMATION FROM STAR TRACKER TO NOMINAL
TBNS = ORIENTATION ARRAY - TRANSFORMATION FROM NOMINAL TO BODY
BSK = BIAS - KNOWLEDGE (RAD)
SSK = NOISE STANDARD DEVIATION - KNOWLEDGE (RAD)
TNSK = MISALIGNMENT KNOWLEDGE ARRAY - TRANSFORMATION FROM STAR TRACKER TO NOMINAL

INCLUDE 'COMPOSIT.COM'
COMMON /COMPOSIT/ PHI(22,22),QT(22,22),COV(22,22),PO(22,22),IP(22),XT(22),P
REAL*8 PHI,QT,COV,PO,XT,P

PHI = COMPOSIT STATE TRANSITION MATRIX
QT = PROCESS NOISE ARRAY
COV = NEWEST COVARIANCE ARRAY
PO = OLD COVARIANCE ARRAY
IP = ARRAY OF FLAGS INDICATING ESTIMATED AND CONSIDERED PARAMETERS
XT = COMPOSIT ESTIMATED PLUS CONSIDERED STATE VECTOR
F = INITIALIZED TRANSITION MATRIX FOR NEXT INTERVAL

INCLUDE 'LMTPAR.COM'
COMMON /LMTPAR/ AL,LON,LAT,TBNL(3,3),TNL(3,3),BL(2),SL(2),BK(2),SKL(2),TNLK(3,3),TIEO(3,3),SIGGCP,THET
REAL*8 AL,TBNL,TNL,BL,SL,BK,SKL,TNLK,TIEO,SEGCP,LAT,THET

LANDMARK TRACKER PARAMETERS
AL = ALTITUDE OF LANDMARK (KM)
LON = LONGITUDE OF LANDMARK (DEG)
LAT = LATITUDE OF LANDMARK (DEG)
TBNL = ORIENTATION ARRAY FOR LANDMARK TRACKER NOMINAL TO BODY
01300  *  C  TNL = VISALIGNMENT ARRAY - ACTUAL  TRACKER TO NOMINAL
01400  *  C  BL = BIAS - ACTUAL (RAD)
01500  *  C  SL = NOISE STANDARD DEVIATION - ACTUAL (RAD)
01600  *  C  BK = BIAS - KNOWLEDGE (RAD)
01700  *  C  THET = LCON ANGLE (RAD)
01800  *  C  SKL = NOISE STANDARD DEVIATION -KNOWLEDGE (RAD)
01900  *  C  TIEO = INITIAL EARTH FIXED TO INERTIAL
02000  *  C  TNLK = VISALIGNMENT ARRAY KNOWLEDGE  TRACKER TO NOMINAL
02100  *  C  SIGGCP = POSITION UNCERTAINTY DUE TO CLOUDS
02200  *  C
02300  *  C
02400  *  C
02500  *  C
02600  0068
02700  0069
02800  COMMON /MEASOUT/ V=6, PRS(6,6), DMC(2), D/C(2), W(3,2),
02900  RS(2,2,2), DMC, DVCL, LWM(3), RL(2,2), EMXG(6),
03000  EDMS(2), ECS(4), EDH, EDVL
03100  0070
03200  REAL*8 MX, PRS, DMC, DVCL, LWM, RL, EMXG,
03300  EDMS, ECS, EDH, EDVL
03400  *  C
03500  *  C
03600  *  C
03700  *  C
03800  *  C
03900  *  C
04000  *  C
04100  *  C
04200  *  C
04300  *  C
04400  *  C
04500  *  C
04600  *  C
04700  *  C
04800  *  C
04900  *  C
05000  *  C
05100  *  C
05200  *  C
05300  *  C
05400  *  C
05500  *  C
05600  *  C
05700  *  C
05800  *  C
05900  *  C
06000  *  C
06100  *  C
06200  *  C
06300  *  C
06400  *  C
06500  *  C
06600  *  C
06700  *  C
06800  *  C
06900  *  C
07000  *  C
07100  *  C
07200  *  C
07300  *  C
07400  *  C
07500  *  C
07600  *  C
07700  *  C
07800  *  C
07900  *  C
08000  *  C
08100  *  C
08200  *  C
08300  *  C
08400  *  C
08500  *  C
08600  *  C
08700  *  C
08800  *  C
08900  *  C
09000  *  C
09100  *  C
09200  *  C
09300  *  C
09400  *  C
09500  *  C
09600  *  C
09700  *  C
09800  *  C
09900  *  C
10000  *  C
10100  *  C
10200  *  C
10300  *  C
10400  *  C
10500  *  C
10600  *  C
10700  *  C
10800  *  C
10900  *  C
11000  *  C
11100  *  C
11200  *  C
11300  *  C
11400  *  C
11500  *  C
11600  *  C
11700  *  C
11800  *  C
11900  *  C
12000  *  C
12100  *  C
12200  *  C
12300  *  C
12400  *  C
12500  *  C
12600  *  C
12700  *  C
12800  *  C
12900  *  C
13000  *  C
13100  *  C
13200  *  C
13300  *  C
13400  *  C
13500  *  C
13600  *  C
13700  *  C
01300  INCLUDE 'MEASOUT.COM'
01400  COMMON /MEASOUT/ V=6, PRS(6,6), DMC(2), D/C(2), W(3,2),
01500  RS(2,2,2), DMC, DVCL, LWM(3), RL(2,2), EMXG(6),
01600  EDMS(2), ECS(4), EDH, EDVL
01700  REAL*8 MX, PRS, DMC, DVCL, LWM, RL, EMXG,
01800  EDMS, ECS, EDH, EDVL
01900  *  C
02000  *  C
02100  *  C
02200  *  C
02300  *  C
02400  *  C
02500  *  C
02600  *  C
02700  *  C
02800  *  C
02900  *  C
03000  *  C
03100  *  C
03200  *  C
03300  *  C
03400  *  C
03500  *  C
03600  *  C
03700  *  C
03800  0071
03900  INCLUDE 'GPSPAR.COM'
04000  COMMON /GPSPAR/ PB(3), VB(3), PS(3), VS(3), PB(3), VB(3), PSK(3),
04100  VS(3)
04200  REAL*8 PB, VB, PS, VS, PBK, VBK, PSK, VSK
04300  *  C
GPS PARAMETERS
PB = POSITION BIAS - ACTUAL
VB = VELOCITY BIAS - ACTUAL
PS = POSITION NOISE STANDARD DEVIATION - ACTUAL
VS = VELOCITY NOISE STANDARD DEVIATION - ACTUAL
PBK = POSITION BIAS - KNOWLEDGE
VKB = VELOCITY BIAS - KNOWLEDGE
PSK = POSITION NOISE STANDARD DEVIATION - KNOWLEDGE
VSK = VELOCITY NOISE STANDARD DEVIATION - KNOWLEDGE

INCLUDE 'PART.COM'
COMMON /PART/ PX(22), PY(22), PZ(22), PXD(22), PYD(22), PDH5(22,2), PDV5(22,2), PDHS(22), PDVLS(22), PDVL(22)
REAL*8 PX, PY, PZ, PXD, PYD, PDH5, PDV5, PDHS, PDVLS, PDVL

PARTIALS OF THE RESPECTIVE MEASUREMENTS MADE
FOR GPS
PX = PARTIALS OF X POSITION MEASUREMENT
PY = ...
PZ = ...
PXD = ...
PYD = ...

FOR STAR TRACKER M (K IS THE SECOND PARAMETER)
PDHS = PARTIALS OF HORIZONTAL DIFLECT
PDV5 = ...

FOR LANDMARK TRACKER
PDHS = ...
PDVL = ...

INCLUDE 'CLOUD.COM'
COMMON /CLOUD/ CLOTBL(12)
REAL*8 CLOTBL
PCNT = THE PERCENTAGE OF CLOUD COVER

INCLUDE 'TITL.COM'
COMMON /TITL/ ATITL(40)
LOGICAL*1 ATITL

ATITL IS THE TITLE PRINTED AT EACH PRINT TIME AS WELL AS THE TOP OF EACH PLOT

INCLUDE 'MODE.COM'
COMMON /MODE/ MODE(10)

MODE(1) = LANDMARK TRACKER SWEEP MODE
0 = RANDOM
1 = FIXED AT INPUT THET
2 = NO DEFAULT TO STAR TRACKER

MODE(2) = CLOUD SELECTION MODE
0 = RANDOM CLOUD DENSITIES BASED ON INPUT TABLES CLOTBL
1 = FIXED DENSITY AT NO CLOUDS
2 = NO CLOUDS WITH 100% CLOUD
INDATA

01200   * C   COVER FOR A SPECIFIED PERIOD (CLOTBL(11,12))
01300   * C   MODE(3-10) NOT SPECIFIED AT PRESENT
01400   * C
01500   * C
02000   * C
02500   0085   DIMENSION TP(4,3), TPT(3,4), PTP(3,3), PQ(4,4), TMP(3,4)
03000   0086   DIMENSION SDA(16), SDN(6)
03500   C      EQUIVALENCE (T1(1), P1), (T2(2), THE), (T3(3), PHI)
04000   C
04500   0087   REAL*8 A1, A2, A3, A4, A5, A6, A7, A8, C1, C2, C3S1, S2, S3, TSTART, PQ, PTP,
05000   C
05500   0088   SDA, TMP, TP, TP1, C3S1
06000   C
06500   0089   TAPES ATTITUDE PLOT TAPE
07000   C
07500   0090   TAPE7 NAVIGATION PLOT TAPE
08000   0091   OPEN(UNIT=4, TYP = 'OLD')
08500   0092   READ(4,4) (ATTLE(I), I=1,40)
09000   0093   FORMAT(20X,4OA1)
09500   0094   READ(4,60) (A(I,J), I=1,3, J=1,3)
10000   0095   READ(4,10) A
10500   0096   READ(4,10) A
11000   0097   READ(4,60) (B(I,J), I=1,3, J=1,3)
11500   0098   READ(4,45) (B(I,J), I=1,2)
12000   0099   READ(4,10) B
12500   0100   READ(4,45) (BS(I,J), I=1,2, J=1,2)
13000   0101   READ(4,45) (BSKI(I,J), I=1,2, J=1,2)
13500   0102   READ(4,60) (BSPI(I), I=1,3)
14000   0103   READ(4,60) (BWD(I), I=1,3)
14500   0104   READ(4,60) (C(I,J), I=1,3, J=1,3)
15000   0105   READ(4,90) CLOTBL(I), I=1,10
15500   0106   READ(4,45) CLOTBL(11), CLOTBL(12)
16000   0107   READ(4,115) (COV(I,J), I=1,22, J=1,22)
16500   0108   READ(4,130) (COV1(I,J), I=1,16)
17000   0109   READ(4,130) (COV1(I,J), I=1,16)
17500   0110   READ(4,60) (D(I,J), I=1,3)
18000   0111   READ(4,120) DATE0
18500   0112   READ(4,120) DATE0
19000   0113   READ(4,60) (DD(I), I=1,3)
19500   0114   READ(4,70) (DE(I), I=1,4)
20000   0115   READ(4,10) DEL
20500   0116   READ(4,10) DMHCL
21000   0117   READ(4,45) (DMCS(I,J), I=1,2)
21500   0118   READ(4,10) DTA
22000   0119   READ(4,20) DTPS, DTN
22500   0120   READ(4,10) DTTP [NT
23000   0121   READ(4,420) DTPS, DTST
23500   0122   READ(4,10) DTU
24000   0123   READ(4,10) DVCL
24500   0124   READ(4,45) (DVCS(I,J), I=1,2)
25000   0125   READ(4,70) (E(I), I=1,4)
25500   0126   READ(4,70) (E(I), I=1,4, J=1,12)
26000   0127   READ(4,70) (F(I,J), I=1,4, J=1,4)
26500   0128   READ(4,80) (FNI(I,J), I=1,6, J=1,6)
27000   0129   READ(4,140) ICOVD
27500   0130   READ(4,200) IDEBUG, ENTER

VAX-11 FORTRAN V2.0-2  10-Apr-1981 06:56:10
VAX-11 FORTRAN V2.0-2  10-Apr-1981 06:51:31
PDBA0([D11.R,GCP)INDATA.FOR:63
08300 0131 READ(4,106)(IDELAY(I),I=1,11)
08400 0132 READ(4,140) IDLMAX
08500 0133 READ(4,160)(IES(I),I=1,25)
08600 0134 READ(4,210) IFLAG
08700 0135 READ(4,140) ISEG
08800 0136 READ(4,160)(IGCDOT(1,J),I=1,4),J=1,10)
08900 0137 READ(4,140) INIT
09000 0138 READ(4,195)(IP(I),I=1,22)
09100 0139 READ(4,170)(IP(I),I=1,6)
09200 0140 READ(4,140) IS
09300 0141 READ(4,160)((ITS(I,J),I=1,4),J=1,9)
09400 0142 READ(4,120) J2
09500 0143 READ(4,120) J3
09600 0144 READ(4,120) J4
09700 0145 READ(4,210) JFlag
09800 0146 READ(4,10) LAT
09900 0147 READ(4,60)(LWM(I),I=1,3)
10000 0148 READ(4,10) LON
10100 0149 READ(4,140) MCM
10200 0150 READ(4,165)(MODE(I),I=1,10)
10300 0151 READ(4,140) MP
10400 0152 READ(4,60)((MS(I,J),I=1,3),J=1,2)
10500 0153 READ(4,140) MYP
10600 0154 READ(4,80)(MAT(I),I=1,5)
10700 0155 READ(4,115)(P(I,J),I=1,22),J=1,22)
10800 0156 READ(4,70)((PA(I,J),I=1,4),J=1,4)
10900 0157 READ(4,60)(PB(I),I=1,3)
11000 0158 READ(4,50)(PBK(I),I=1,3)
11100 0159 READ(4,70)(POA(I,J),I=1,4),J=1,16)
11200 0160 READ(4,115)(PDH(I,J),I=1,22)
11300 0161 READ(4,115)(PDM(I,J),I=1,22),J=1,2)
11400 0162 READ(4,80)(POV(I),I=1,16)
11500 0163 READ(4,115)(PDV(I,J),I=1,22)
11600 0164 READ(4,115)(PJS(I,J),I=1,22),J=1,2)
11700 0165 READ(4,115)(PH(I,J),I=1,22)
11800 0166 READ(4,110)((PR(I,J),I=1,16),J=1,18)
11900 0167 READ(4,80)(PH(I,J),I=1,16)
12000 0168 READ(4,10) P1
12100 0169 READ(4,120) P1
12200 0170 READ(4,80)(PV(I,J),I=1,6),J=1,6)
12300 0171 READ(4,115)(PP(I,J),I=1,22),J=1,2)
12400 0172 READ(4,110)((PGA(I,J),I=1,16),J=1,16)
12500 0173 READ(4,80)(POV(I,J),I=1,6)
12600 0174 READ(4,90)((POVE(I,J),I=1,10),J=1,4)
12700 0175 READ(4,60)(PS(I),I=1,3)
12800 0176 READ(4,60)(PSK(I),I=1,3)
12900 0177 READ(4,60)(PXM(I),I=1,22)
13000 0178 READ(4,115)(PXY(I,J),I=1,22)
13100 0179 READ(4,115)(PYY(I),I=1,22)
13200 0180 READ(4,115)(PZ(I,J),I=1,22)
13300 0181 READ(4,115)(PXD(I,J),I=1,22)
13400 0182 READ(4,115)(PSD(I,J),I=1,22)
13500 0183 READ(4,80)((Q(I,J),I=1,16),J=1,6)
13600 0184 READ(4,110)(QA(I,J),I=1,16)
13700 0185 READ(4,80)((QDOT(I,J),I=1,16),J=1,6)
13800 0186 READ(4,10) QMAX
13900 0187 READ(4,10) QMAX
| INDATA | 14000 | 0188 | READ(4,800)(ON(I),I=1,6) |
|        | 14100 | 0189 | READ(4,1155)(QT(I,J),I=1,22,J=1,22) |
|        | 14200 | 0190 | READ(4,40)(RADE,RADM,PRG,PRM) |
|        | 14300 | 0191 | READ(4,10)(RBO) |
|        | 14400 | 0192 | READ(4,120)(RE2) |
|        | 14500 | 0193 | READ(4,800)(RGRS(I,J),I=1,6,J=1,6) |
|        | 14600 | 0194 | READ(4,70)((RL(I,J),I=1,2,J=1,2)) |
|        | 14700 | 0195 | READ(4,120)(RDM2) |
|        | 14800 | 0196 | READ(4,45)((RS(I,J,K),I=1,2,J=1,2,K=1,2)) |
|        | 14900 | 0197 | READ(4,60)(SDI(I),I=1,3) |
|        | 15000 | 0198 | READ(4,60)(SDD(I),I=1,3) |
|        | 15100 | 0199 | READ(4,60)(SFI(I),I=1,3) |
|        | 15200 | 0200 | READ(4,10)(SGGCP) |
|        | 15300 | 0201 | READ(4,45)(SKL(I),I=1,2) |
|        | 15400 | 0202 | READ(4,45)(SL(I),I=1,2) |
|        | 15500 | 0203 | READ(3,10)(SRE) |
|        | 15600 | 0204 | READ(4,60)(SSF(I),I=1,3) |
|        | 15700 | 0205 | READ(4,70)((SS(I,J),I=1,2,J=1,2)) |
|        | 15800 | 0206 | READ(4,70)((SSK(I,J),I=1,2,J=1,2)) |
|        | 15900 | 0207 | READ(4,80)(STAT(I),I=1,10) |
|        | 16000 | 0208 | READ(4,60)(SWD(I),I=1,3) |
|        | 16100 | 0209 | READ(4,60)((TBNL(I,J),I=1,3,J=1,3)) |
|        | 16200 | 0210 | READ(4,60)((TNS5(I,J,K),I=1,3,J=1,3,K=1,2)) |
|        | 16300 | 0211 | READ(4,10)(TET) |
|        | 16400 | 0212 | READ(4,60)((TIEO(I,J),I=1,3,J=1,2)) |
|        | 16500 | 0213 | READ(4,60)((TNL(I,J),I=1,3,J=1,3)) |
|        | 16600 | 0214 | READ(4,60)((TNLK(I,J),I=1,3,J=1,3)) |
|        | 16700 | 0215 | READ(4,60)((TNS(I,J,K),I=1,3,J=1,3,K=1,2)) |
|        | 16800 | 0216 | READ(4,60)((TNSK(I,J,K),I=1,3,J=1,3,K=1,2)) |
|        | 16900 | 0217 | READ(4,60)((T1(I),I=1,3)) |
|        | 17000 | 0218 | READ(4,60)((T11(I,J),I=1,3,J=1,3)) |
|        | 17100 | 0219 | READ(4,60)(T2(I),I=1,3) |
|        | 17200 | 0220 | READ(4,60)(T3(I),I=1,3) |
|        | 17300 | 0221 | READ(4,60)((T33(I,J),I=1,3,J=1,3)) |
|        | 17400 | 0222 | READ(4,90)(TT4(I),I=1,10) |
|        | 17500 | 0223 | READ(4,70)((T44(I,J),I=1,4,J=1,4)) |
|        | 17600 | 0224 | READ(4,70)(T5(I),I=1,4) |
|        | 17700 | 0225 | READ(4,70)(T6(I),I=1,4) |
|        | 17800 | 0226 | READ(4,80)((T66(I,J),I=1,6,J=1,6)) |
|        | 17900 | 0227 | READ(4,70)(T7(I),I=1,4) |
|        | 18000 | 0228 | READ(4,80)((T77(I,J),I=1,6,J=1,6)) |
|        | 18100 | 0229 | READ(4,70)((T8(I,J),I=1,4,J=1,12)) |
|        | 18200 | 0230 | READ(4,30)(TIA,TIM,TIN) |
|        | 18300 | 0231 | READ(4,40)(INT,TIS,TISN,TMSAS) |
|        | 18400 | 0232 | READ(4,10)(TNEXT) |
|        | 18500 | 0233 | READ(4,30)(TP1,TP2,TP3) |
|        | 18600 | 0234 | READ(4,10)(TPRINT) |
|        | 18700 | 0235 | READ(4,10)(STOP) |
|        | 18800 | 0236 | READ(4,10)(TZERO) |
|        | 18900 | 0237 | READ(4,120)(UE) |
|        | 19000 | 0238 | READ(4,120)(UM) |
|        | 19100 | 0239 | READ(4,120)(US) |
|        | 19200 | 0240 | READ(4,60)(V8(I),I=1,3) |
|        | 19300 | 0241 | READ(4,60)(VBK(I),I=1,3) |
|        | 19400 | 0242 | READ(4,60)(VS(I),I=1,3) |
|        | 19500 | 0243 | READ(4,60)(VSK(I),I=1,3) |
|        | 19600 | 0244 | READ(4,60)(WD(I),I=1,3) |
INDATA

10-Apr-1981 06:56:10  VAX-11 FORTRAN V2.0-2  Page 11

19700  0245  READ(4,115)(XT(I),I=1,22)
19800  0246  READ(4,80)(XI(I),I=1,6)
19900  0247  READ(4,90)(XD(I),I=1,6)
20000  0248  10 FORMAT(20X,F20.10)
20100  0249  20 FORMAT(20X,F20.10/,20X,F20.10)
20200  0250  30 FORMAT(20X,F20.10/,20X,F20.10/,20X,F20.10)
20300  0251  40 FORMAT(20X,F20.10/,20X,F20.10/,20X,F20.10/,20X,F20.10)
20400  0252  45 FORMAT(20X,2F20.10)
20500  0253  60 FORMAT(20X,3F20.10)
20600  0254  70 FORMAT(20X,4F20.10)
20700  0255  80 FORMAT(20X,6F20.10)
20800  0256  90 FORMAT(20X,10F20.10)
20900  0257  100 FORMAT(20X,11F20.10)
21100  0258  110 FORMAT(20X,16F20.10)
21200  0259  115 FORMAT(20X,22F20.10)
21300  0260  120 FORMAT(20X,E20.10)
21400  0261  130 FORMAT(20X,5E20.10)
21500  0262  140 FORMAT(20X,3E20.10)
21600  0263  150 FORMAT(20X,3E20.10)
21700  0264  160 FORMAT(20X,4E20.10)
21800  0265  165 FORMAT(20X,10I10)
21900  0266  166 FORMAT(20X,11I10)
22000  0267  170 FORMAT(20X,11I10)
22100  0268  180 FORMAT(20X,12I10)
22200  0269  190 FORMAT(20X,16I10)
22300  0270  195 FORMAT(20X,22I10)
22400  0271  200 FORMAT(20X,110/,20X,110)
22500  0272  210 FORMAT(20X,L7)
22600  0273  REWIND 7
22700  0274  C******************************************************************************
22800  0275  C                           PSI = 361.
22900  0276  C******************************************************************************
23000  0277  T1(3)=361. I TEMPORARY FIX; JACK
23100  0278  DO 400 I=1,16 I TEMPORARY FIX
23200  0279  400 SDA(I)=0. I TEMPORARY FIX
23300  0280  DO 450 I=1,6 I TEMPORARY FIX
23400  0281  450 SDN(I)=0. I TEMPORARY FIX
23500  0282  DO 500 I=1,6
23600  0283  500 DO 500 J=1,6
23700  0284  0(I,J) = 0. I TEMPORARY FIX
23800  0285  C******************************************************************************
23900  0286  IF(Psi.EQ.361.1 GO TO 1000 I TEMPORARY FIX; JACK
24000  0287  C******************************************************************************
24100  0288  IF(T1(3).EQ.361.) GO TO 1000 I TEMPORARY FIX; JACK
24200  0289  C1 = COS(TI(1))
24300  0290  S1 = SIN(TI(1))
24400  0291  C2 = COS(TI(2))
24500  0292  S2 = SIN(TI(2))
24600  0293  C3 = COS(TI(3))
24700  0294  S3 = SIN(TI(3))
24800  0295  A1 = S1+C2+C3
24900  0296  A2 = 1+C1+S2
25000  0297  A3 = 1+C1+S3
25100  0298  A4 = S1+C2+S3
25200  0299  A5 = C1+C2+S3
25300  0300  A6 = S1+S2+S3
25400 0295  A7 = C1+C2+C3
25500 0296  AB = S1+S2+S3
25600 0297  E(1) = A7+AB
25700 0298  E(2) = A5-AB
25800 0299  E(3) = A3*A4
25900 0300  E(4) = A1-A2
26000  
26100 0301  CONTINUE
26200 0302  DO 1100 I=1,6
26300 0303  1100 Q(I,1)=QN(I)
26400 0304  IF(ICOV(I).EQ.0.) GO TO 8000
26500 0305  DO 2200 I=1,6
26600 0306  2200 COV(I,1)=COVN(I,1)
26700 0307  DO 3300 I=1,6
26800 0308  3300 COV(I+6,1+6)=COV(1,1)
26900 0309  IF(SDN(I,1).EQ.0.) GO TO 2500
27000 0310  DO 2000 I=1,6
27100 0311  2000 QDN(I,1) = SDN(I)**2
27200 0312  DO 2100 I=1,6
27300 0313  2100 COV(I,1)=COVN(I,1)
27400 0314  2500 IF(SDA(2).EQ.0.) GO TO 8000
27500 0315  DO 3400 I=1,6
27600 0316  3000 QOA(I,1) = SDA(I)**2
27700 0317  DO 3500 I=1,6
27800 0318  3100 COV(I+6,1+6)=COV(1,1)
27900 0319  DO 4400 I=1,3
28000 0320  J = I+1
28100 0321  4000 PTP(I,1) = SDA(J)**2
28200 0322  TP(1,1) = A2-A1
28300 0323  TP(2,1) = -A3-A4
28400 0324  TP(3,1) = A5-A6
28500 0325  TP(4,1) = A7*A8
28600 0326  TP(1,2) = A4-A3
28700 0327  TP(2,2) = A1-A2
28800 0328  TP(3,2) = A7-A8
28900 0329  TP(4,2) = A5-A6
29000 0330  TP(1,3) = A6-A5
29100 0331  TP(2,3) = A7+A8
29200 0332  TP(3,3) = A1-A2
29300 0333  TP(4,3) = -A3-A4
29400 0334  DO 5000 I=1,3
29500 0335  DO 5100 J=1,4
29600 0336  TP(J,1) = .5*TP(J,1)
29700 0337  5000 PTP(J,1) = TP(J,1)
29800 0338  CALL MATAB(PTP,TP,TMP,3,2,4)
29900 0339  CALL MATAB(PTP,TMP,PO,4,3,4)
30000 0340  DO 6000 I=1,4
30100 0341  DO 6100 J=1,4
30200 0342  6000 QOA(I,1) = PQ(I,1)
30300 0343  DO 7000 I=1,16
30400 0344  DO 7100 J=1,16
30500 0345  7000 POA(I,1) = COVA(I,1)
30600 0346  DO 7500 I=1,6
30700 0347  DO 7600 J=1,6
30800 0348  7500 POA(I,1) = COV(I,1)
30900 0349  8000 WRITE(6,9000) TSTART,TSTOP,E,X
31000 0350  WRITE(6,10000) (COVA(I,1),I=1,16)
WRITE(6,11000) CDNN(1,1),1=1,6
9000 FORMAT(1HM,5X,46HGROUND CONTROL POINT SIMULATION PROGRAM GCPSIM
//5X,25HRUN START AND STOP TIMES .2F10.0/5X,
.25HINIT QUATERNION VALUES .4F10.7/5X,
.25HINIT POSITION VALUES .3EO27.14/30X,5E12.4///)
10000 FORMAT(5X,25HATT PARAMETER VARIANCES ,BE12.4/30X,5E12.4///)
11000 FORMAT(5X,25HNAV PARAMETER VARIANCES ,6E12.4/30X,5E12.4///)
WRITE(6,11200) MGP
11200 FORMAT(4X,' MODE OF OPERATION = ',I1)
WRITE(6,11300) UE.US.UW
11300 FORMAT(4X,' GRAVITATIONAL CONSTANTS -- EARTH, SUN, MOON ',4X,
11400 FORMAT(4X,' REAL EARTH, SUN, MOON ',4X,.
11500 FORMAT(4X,' ESTIMATED RADII -- EARTH, MOON ',1X,.
11600 FORMAT(4X,' ACTION TABLE/20X,5HSTAGE SPECIFICATION/3(29X,11/)20X,
11700 FORMAT(4X,' ALTERNATE ACTION/3IS0X,4/)///)
WRITE(6,11800) ICID,IR,IA
11800 FORMAT(4X,' EVENT CLUSTER/5X,75H
11900 FORMAT(4X,' EVENT START-STOP, MAX NO.
12000 FORMAT(20X,'6HEVENT SEQUENCING/15X,75H
12100 FORMAT(20X,16EVENT CYCLING/25X,50HEVENT START - STOP, MAX NO. CYC
12200 FORMAT(20X,'HEVENT CYCLING/25X,50HEVENT START - STOP, MAX NO. CYC
12300 FORMAT(4X,' ALTERNATE ACTION/3IS0X,4/)///)
### Program Sections

<table>
<thead>
<tr>
<th>Name</th>
<th>Bytes</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 $CODE</td>
<td>9042</td>
<td>PIC CON REL LCL SHR EXE RD NOWRT LONG</td>
</tr>
<tr>
<td>1 $POATA</td>
<td>1149</td>
<td>PIC CON REL LCL SHR NOEKE RD NOWRT LONG</td>
</tr>
<tr>
<td>2 $LOCAL</td>
<td>976</td>
<td>PIC CON REL LCL NOEKR NOEKE RD WRT QUAD</td>
</tr>
<tr>
<td>3 DEBUG</td>
<td>8</td>
<td>PIC OVR REL GBL SHR NOEKE RD WRT LONG</td>
</tr>
<tr>
<td>4 CONTRL</td>
<td>12</td>
<td>PIC OVR REL GBL SHR NOEKE RD WRT LONG</td>
</tr>
<tr>
<td>5 ENVIR</td>
<td>404</td>
<td>PIC OVR REL GBL SHR NOEKE RD WRT LONG</td>
</tr>
<tr>
<td>6 GPART</td>
<td>600</td>
<td>PIC OVR REL GBL SHR NOEKE RD WRT LONG</td>
</tr>
<tr>
<td>7 NOISE</td>
<td>216</td>
<td>PIC OVR REL GBL SHR NOEKE RD WRT LONG</td>
</tr>
<tr>
<td>8 TARGETS</td>
<td>36</td>
<td>PIC OVR REL GBL SHR NOEKE RD WRT LONG</td>
</tr>
<tr>
<td>9 ASTATE</td>
<td>160</td>
<td>PIC OVR REL GBL SHR NOEKE RD WRT LONG</td>
</tr>
<tr>
<td>10 FILTER</td>
<td>88</td>
<td>PIC OVR REL GBL SHR NOEKE RD WRT LONG</td>
</tr>
<tr>
<td>11 SEQU</td>
<td>292</td>
<td>PIC OVR REL GBL SHR NOEKE RD WRT LONG</td>
</tr>
<tr>
<td>12 TMAT</td>
<td>312</td>
<td>PIC OVR REL GBL SHR NOEKE RD WRT LONG</td>
</tr>
<tr>
<td>13 NSTATE</td>
<td>112</td>
<td>PIC OVR REL GBL SHR NOEKE RD WRT LONG</td>
</tr>
<tr>
<td>14 ARAYS</td>
<td>1096</td>
<td>PIC OVR REL GBL SHR NOEKE RD WRT LONG</td>
</tr>
<tr>
<td>15 TIME</td>
<td>136</td>
<td>PIC OVR REL GBL SHR NOEKE RD WRT LONG</td>
</tr>
<tr>
<td>16 CONST</td>
<td>112</td>
<td>PIC OVR REL GBL SHR NOEKE RD WRT LONG</td>
</tr>
<tr>
<td>17 UPDT</td>
<td>752</td>
<td>PIC OVR REL GBL SHR NOEKE RD WRT LONG</td>
</tr>
<tr>
<td>18 PLOT</td>
<td>16</td>
<td>PIC OVR REL GBL SHR NOEKE RD WRT LONG</td>
</tr>
<tr>
<td>19 PHIA</td>
<td>7176</td>
<td>PIC OVR REL GBL SHR NOEKE RD WRT LONG</td>
</tr>
<tr>
<td>20 PHIN</td>
<td>1440</td>
<td>PIC OVR REL GBL SHR NOEKE RD WRT LONG</td>
</tr>
<tr>
<td>21 GGOAT</td>
<td>192</td>
<td>PIC OVR REL GBL SHR NOEKE RD WRT LONG</td>
</tr>
<tr>
<td>22 STARPAR</td>
<td>16264</td>
<td>PIC OVR REL GBL SHR NOEKE RD WRT LONG</td>
</tr>
<tr>
<td>23 COMPOSIT</td>
<td>332</td>
<td>PIC OVR REL GBL SHR NOEKE RD WRT LONG</td>
</tr>
<tr>
<td>24 LMTRP</td>
<td>648</td>
<td>PIC OVR REL GBL SHR NOEKE RD WRT LONG</td>
</tr>
<tr>
<td>25 MEASOUT</td>
<td>112</td>
<td>PIC OVR REL GBL SHR NOEKE RD WRT LONG</td>
</tr>
<tr>
<td>26 GPSPAR</td>
<td>2112</td>
<td>PIC OVR REL GBL SHR NOEKE RD WRT LONG</td>
</tr>
<tr>
<td>27 PART</td>
<td>96</td>
<td>PIC OVR REL GBL SHR NOEKE RD WRT LONG</td>
</tr>
<tr>
<td>28 CLOUD</td>
<td>40</td>
<td>PIC OVR REL GBL SHR NOEKE RD WRT LONG</td>
</tr>
<tr>
<td>29 TITLE</td>
<td>40</td>
<td>PIC OVR REL GBL SHR NOEKE RD WRT LONG</td>
</tr>
<tr>
<td>30 MODE</td>
<td>40</td>
<td>PIC OVR REL GBL SHR NOEKE RD WRT LONG</td>
</tr>
</tbody>
</table>

#### Entry Points

<table>
<thead>
<tr>
<th>Address</th>
<th>Type</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-00000000</td>
<td></td>
<td>INDATA</td>
</tr>
</tbody>
</table>

#### Variables

<table>
<thead>
<tr>
<th>Address</th>
<th>Type</th>
<th>Name</th>
<th>Address</th>
<th>Type</th>
<th>Name</th>
<th>Address</th>
<th>Type</th>
<th>Name</th>
<th>Address</th>
<th>Type</th>
<th>Name</th>
<th>Address</th>
<th>Type</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-00000000</td>
<td>R+8</td>
<td>A1</td>
<td>2-00000028</td>
<td>R+8</td>
<td>A2</td>
<td>2-00000029</td>
<td>R+8</td>
<td>A3</td>
<td>2-00000029</td>
<td>R+8</td>
<td>A4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2-00000240</td>
<td>R+8</td>
<td>A5</td>
<td>2-00000248</td>
<td>R+8</td>
<td>A6</td>
<td>2-00000280</td>
<td>R+8</td>
<td>A7</td>
<td>2-00000280</td>
<td>R+8</td>
<td>A8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2-000002CB</td>
<td>R+8</td>
<td>C2</td>
<td>2-00000070</td>
<td>R+8</td>
<td>C3</td>
<td>2-000000C8</td>
<td>R+8</td>
<td>C4</td>
<td>2-000000C8</td>
<td>R+8</td>
<td>C5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15-00000070</td>
<td>R+8</td>
<td>DATER</td>
<td>15-00000020</td>
<td>R+8</td>
<td>DEL</td>
<td>15-00000024</td>
<td>R+8</td>
<td>DTM</td>
<td>15-00000078</td>
<td>R+8</td>
<td>DTPRINT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11-000000FC</td>
<td>R+8</td>
<td>DTGPS</td>
<td>15-00000030</td>
<td>R+8</td>
<td>DTN</td>
<td>15-00000080</td>
<td>R+8</td>
<td>DTPRINT</td>
<td>21-00000088</td>
<td>R+8</td>
<td>DTPRINT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11-000000EB</td>
<td>R+8</td>
<td>DTST</td>
<td>15-00000060</td>
<td>R+8</td>
<td>DTU</td>
<td>25-0000001E</td>
<td>R+8</td>
<td>DVL</td>
<td>21-00000003</td>
<td>R+8</td>
<td>EDTDL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>25-00000280</td>
<td>R+8</td>
<td>EDVL</td>
<td>2-00000030</td>
<td>I+4</td>
<td>I</td>
<td>2-00000000</td>
<td>I+4</td>
<td>IENTER</td>
<td>11-00000004</td>
<td>L+4</td>
<td>IFLAG</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3-00000042</td>
<td>I+4</td>
<td>IDEBUG</td>
<td>21-0000000C</td>
<td>I+4</td>
<td>IDM MAX</td>
<td>2-00000000</td>
<td>I+4</td>
<td>IENTER</td>
<td>2-00000000</td>
<td>I+4</td>
<td>IFLAG</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Note:** The document contains listings for program sections, entry points, and variables with addresses, types, and names.
<table>
<thead>
<tr>
<th>Address</th>
<th>Type</th>
<th>Name</th>
<th>Bytes</th>
<th>Dimensions</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>R-B</td>
<td>A</td>
<td>72</td>
<td>(3, 3)</td>
</tr>
<tr>
<td>29</td>
<td>L-1</td>
<td>ATITLE</td>
<td>40</td>
<td>(40)</td>
</tr>
<tr>
<td>12</td>
<td>R-B</td>
<td>B</td>
<td>72</td>
<td>(3, 3)</td>
</tr>
<tr>
<td>24</td>
<td>R-B</td>
<td>BO</td>
<td>24</td>
<td>(2)</td>
</tr>
<tr>
<td>7</td>
<td>R-B</td>
<td>BDD</td>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>24</td>
<td>R-B</td>
<td>BKML</td>
<td>16</td>
<td>(2)</td>
</tr>
<tr>
<td>24</td>
<td>R-B</td>
<td>BL</td>
<td>16</td>
<td>(2)</td>
</tr>
<tr>
<td>22</td>
<td>R-B</td>
<td>BS</td>
<td>32</td>
<td>(2, 2)</td>
</tr>
<tr>
<td>22</td>
<td>R-B</td>
<td>BSF</td>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>22</td>
<td>R-B</td>
<td>BSK</td>
<td>32</td>
<td>(2, 2)</td>
</tr>
<tr>
<td>7</td>
<td>R-B</td>
<td>BWB</td>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>12</td>
<td>R-B</td>
<td>C</td>
<td>72</td>
<td>(3, 3)</td>
</tr>
<tr>
<td>28</td>
<td>R-B</td>
<td>CLTBL</td>
<td>96</td>
<td>(12)</td>
</tr>
<tr>
<td>23</td>
<td>R-B</td>
<td>COV</td>
<td>3872</td>
<td>(22, 22)</td>
</tr>
<tr>
<td>19</td>
<td>R-B</td>
<td>CVRA</td>
<td>2048</td>
<td>(16, 16)</td>
</tr>
<tr>
<td>20</td>
<td>R-B</td>
<td>CVN</td>
<td>286</td>
<td>(6, 6)</td>
</tr>
<tr>
<td>9</td>
<td>R-B</td>
<td>D</td>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>9</td>
<td>R-B</td>
<td>DD</td>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>9</td>
<td>R-B</td>
<td>DE</td>
<td>32</td>
<td>(4)</td>
</tr>
<tr>
<td>25</td>
<td>R-B</td>
<td>DMCS</td>
<td>16</td>
<td>(2)</td>
</tr>
<tr>
<td>25</td>
<td>R-B</td>
<td>DVC</td>
<td>16</td>
<td>(2)</td>
</tr>
<tr>
<td>9</td>
<td>R-B</td>
<td>E</td>
<td>32</td>
<td>(4)</td>
</tr>
<tr>
<td>6</td>
<td>R-B</td>
<td>EA</td>
<td>304</td>
<td>(4, 12)</td>
</tr>
<tr>
<td>25</td>
<td>R-B</td>
<td>EDMS</td>
<td>16</td>
<td>(2)</td>
</tr>
<tr>
<td>25</td>
<td>R-B</td>
<td>EDVS</td>
<td>16</td>
<td>(2)</td>
</tr>
<tr>
<td>12</td>
<td>R-B</td>
<td>EM</td>
<td>96</td>
<td>(4, 3)</td>
</tr>
<tr>
<td>25</td>
<td>R-B</td>
<td>EMAG</td>
<td>48</td>
<td>(6)</td>
</tr>
<tr>
<td>6</td>
<td>R-B</td>
<td>FA</td>
<td>128</td>
<td>(4, 4)</td>
</tr>
<tr>
<td>6</td>
<td>R-B</td>
<td>FM</td>
<td>288</td>
<td>(6, 6)</td>
</tr>
<tr>
<td>11</td>
<td>R-B</td>
<td>IAA</td>
<td>48</td>
<td>(3, 4)</td>
</tr>
<tr>
<td>11</td>
<td>R-B</td>
<td>IDELAY</td>
<td>111</td>
<td>(11)</td>
</tr>
<tr>
<td>11</td>
<td>R-B</td>
<td>IES</td>
<td>100</td>
<td>(25)</td>
</tr>
<tr>
<td>21</td>
<td>R-B</td>
<td>IGCPDT</td>
<td>160</td>
<td>(4, 10)</td>
</tr>
<tr>
<td>23</td>
<td>R-B</td>
<td>IIP</td>
<td>88</td>
<td>(22)</td>
</tr>
<tr>
<td>10</td>
<td>R-B</td>
<td>IPA</td>
<td>64</td>
<td>(16)</td>
</tr>
<tr>
<td>10</td>
<td>R-B</td>
<td>IPN</td>
<td>24</td>
<td>(6)</td>
</tr>
<tr>
<td>11</td>
<td>R-B</td>
<td>ITS</td>
<td>60</td>
<td>(4, 5)</td>
</tr>
<tr>
<td>ADDRESS</td>
<td>NAME</td>
<td>VALUE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>--------</td>
<td>-------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>25-00000010</td>
<td>R8 LMU</td>
<td>24 (3)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>30-00000000</td>
<td>R4 MODE</td>
<td>40 (10)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>25-00000000</td>
<td>R8 WM</td>
<td>49 (3, 2)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>25-00000000</td>
<td>R8 MX</td>
<td>49 (6)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>23-00003088</td>
<td>R8 P</td>
<td>3872 (22, 22)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>19-00000000</td>
<td>R8 PA</td>
<td>128 (4, 4)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>26-00000000</td>
<td>R8 PW</td>
<td>24 (3)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>26-00000060</td>
<td>R8 PBK</td>
<td>24 (3)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>19-00000200</td>
<td>R8 PDA</td>
<td>512 (4, 16)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>27-00003060</td>
<td>R8 PDHL</td>
<td>176 (22)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>27-00004200</td>
<td>R8 PDHS</td>
<td>252 (22, 2)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20-00001200</td>
<td>R8 PDN</td>
<td>288 (6, 6)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>27-00007070</td>
<td>R8 PDVL</td>
<td>176 (22)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>27-00000250</td>
<td>R8 PDVS</td>
<td>252 (22, 2)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>23-00003000</td>
<td>R8 PHI</td>
<td>3872 (22, 22)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>19-00003400</td>
<td>R8 PHIA</td>
<td>2048 (16, 16)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20-00002400</td>
<td>R8 PHIM</td>
<td>2048 (6, 6)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20-00000000</td>
<td>R8 PN</td>
<td>288 (6, 6)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>21-00002060</td>
<td>R8 PQ</td>
<td>3872 (22, 22)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>19-00001400</td>
<td>R8 PGA</td>
<td>2048 (16, 16)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20-00004800</td>
<td>R8 PON</td>
<td>288 (6, 6)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2-00000100</td>
<td>R8 PO</td>
<td>128 (4, 4)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5-00000050</td>
<td>R8 PROFILE</td>
<td>320 (10, 4)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>26-00000330</td>
<td>R8 PS</td>
<td>24 (3)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>26-00000090</td>
<td>R8 PSK</td>
<td>24 (3)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2-00000000</td>
<td>R8 PTP</td>
<td>72 (3, 3)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>27-00000060</td>
<td>R8 PX</td>
<td>176 (22)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>27-00000210</td>
<td>R8 PXD</td>
<td>176 (22)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>27-00000080</td>
<td>R8 PY</td>
<td>176 (22)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>27-00002250</td>
<td>R8 PYD</td>
<td>176 (22)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>27-00000160</td>
<td>R8 PZ</td>
<td>176 (22)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>27-00003370</td>
<td>R8 PZD</td>
<td>176 (22)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17-00000030</td>
<td>R8 Q</td>
<td>288 (6, 6)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17-00000030</td>
<td>R8 QA</td>
<td>128 (16)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17-00000100</td>
<td>R8 QDOT</td>
<td>288 (6, 6)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17-00000000</td>
<td>R8 QM</td>
<td>48 (6)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>23-00000070</td>
<td>R8 QT</td>
<td>3872 (22, 22)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>25-00000330</td>
<td>R8 RGPS</td>
<td>288 (6, 6)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>25-00000100</td>
<td>R8 RL</td>
<td>32 (2, 2)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>25-00000140</td>
<td>R8 RS</td>
<td>64 (2, 2, 2)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7-00000018</td>
<td>R8 SD</td>
<td>24 (3)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2-00000118</td>
<td>R8 SDA</td>
<td>128 (16)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7-00000048</td>
<td>R8 SDO</td>
<td>24 (3)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2-00000268</td>
<td>R4 SDN</td>
<td>24 (6)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9-0000058</td>
<td>R8 SF</td>
<td>24 (3)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>24-00000078</td>
<td>R8 SKL</td>
<td>16 (2)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>24-00000088</td>
<td>R8 SL</td>
<td>16 (2)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>22-00000020</td>
<td>R8 SS</td>
<td>32 (2, 2)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7-00000048</td>
<td>R8 SSF</td>
<td>24 (3)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>22-00000180</td>
<td>R8 SSK</td>
<td>32 (2, 2)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5-00000000</td>
<td>R8 STATE</td>
<td>16 (10)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7-00000018</td>
<td>R8 SWO</td>
<td>24 (3)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14-00000000</td>
<td>R8 T1</td>
<td>24 (3)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14-00000098</td>
<td>R8 T11</td>
<td>72 (3, 3)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14-00000016</td>
<td>R8 T2</td>
<td>24 (3)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14-00000038</td>
<td>R8 T3</td>
<td>24 (3)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14-00000060</td>
<td>R8 T33</td>
<td>72 (3, 3)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
FUNCTIONS AND SUBROUTINES REFERENCED

FORSKOPEN MATAB MTHSDCOS MTHSDSIN

Total Space Allocated = 48231 Bytes
COMMAND QUALIFIERS

FORTRAN /L INDATA

/CHECK=(NOBOUNDS,OVERFLOW)
/DEBUG=(NOSYMBOLS,TRACEBACK)
/FF77 /NOG_FLOATING /14 /OPTIMIZE /WARNINGS /NOD_LINES /NOMACHINE_CODE /CONTINUATIONS=19

COMPILATION STATISTICS

Run Time: 22.24 seconds
Elapsed Time: 68.35 seconds
Page Faults: 1281
Dynamic Memory: 298 pages
2.2 Event Sequencer (GCPSEQ)

One of the primary functions of GCPSIM is to analyze the effect of different measurement sequences and to establish a baseline sequence which will meet the operational requirements. The operations sequencer (GCPSEQ) has been designed to allow any sequence of operations to be simulated.

The GCPSEQ module is responsible for controlling the type of measurements performed by the onboard GCP detection system. The possible measurement types are identified by an event code as illustrated in Table A-1. The delay times listed in the table were chosen to be multiples of the cycle time between two consecutive scans of the mirror in the MSS. This would allow the implementation of a scan initialization signal which would be generated by the onboard navigation system to control the scanner of the MSS. Control of the scan mechanism in this manner would eliminate the distortions (as large as 28 pixels) associated with variation of the scan period and would reduce the amount of onboard processing required for image correction. The primary interrupt period of the onboard navigation system will be chosen such that it divides evenly into the scan period.

An event sequencing table has been established to identify series of events to be performed. The overall mission may be broken down into different types of operational sequences defined by partitioning locations in the event sequencing table containing the event codes to be performed. A sample event sequence table is shown in Table A-1. In the table, the first partition contains a sequence which performs a GPS measurement followed by a 10 second wait. The second partition contains a sequence where a GPS measurement is followed by a #1 star tracker measurement, a 10 second wait, a second GPS measurement, and a #2 star tracker measurement.

<table>
<thead>
<tr>
<th>PARTITION NUMBER</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>SEQUENCE NUMBER</td>
<td>1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EVENT CODES</td>
<td>2 8 2 3 8 2 4 2 6 3 8 2 6 4 8 3 6 4 6</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table A-1. Event Sequence Table
Partitions within the event sequence table are defined by an event cycling table. The cycling table defines partitions in terms of the sequence numbers which mark the beginning and end of the sequence within the partition.

In addition to defining partitions the event cycling table defines either the number of repetitions of a partition or the maximum amount of time to be spent cycling through a partition. Operations within a partition are terminated when one of these limits is reached. A sample event cycling table is shown in Table A-2. A partition is defined in terms of the event sequence numbers. In the table, the first event group defines a partition beginning at sequence number 1 and ending with sequence number 2. Operations within the partition will be repeated five times or for two hours depending on which occurs first.

<table>
<thead>
<tr>
<th>PARTITION NUMBER</th>
<th>SEQUENCE START</th>
<th>SEQUENCE STOP</th>
<th>CYCLE REP. LIMIT</th>
<th>CYCLE TIME (HOURS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>7</td>
<td>100</td>
<td>15</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>2</td>
<td>50</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table A-2. Sample Event Cycling Table

GCP measurement are considered to take precedence over all other types of events and will therefore be scheduled on a separate basis. A GCP count table (Table II-IV) containing all information necessary to locate the predicted GCP locator has been implemented in the sequencer.

The location and size of the search area is completely defined by the line start, line stop, element start, and element stop indicators contained in the count table as shown in Figure A-5.
In Figure A-5 the two long horizontal lines enclose a diagram of the imaging sensor's field of view over time. The vertical swaths depict one scan of the sensor's instantaneous field of view, and the small squares within the scan line are individual picture elements. The start count contains the number of scans between the present scan line and the first line of the search area. The stop count contains the size of the search area in scan lines. Once it is known that the search area is within the field of view, it is necessary to determine its location. The element start count contains the distance, in picture elements, of the search area from the start of the scan line. The element stop count contains the distance to the end of the search area from the start of the scan line. These four parameters totally describe the position and size of the search area relative to the current position of the sensor's field of view. The center of the search area is coincident with the estimated center of the landmark and its size is a direct function of the uncertainty in satellite position and attitude.
The functional flow of the sequencer is illustrated by the VCLR in Figure A-6. On the first entry into the sequencer, all indices in the sequence table, the cycle repetition table and the GCP count table are initialized. On all other entries to the sequencer, the event sequence index, ISS, is incremented. A test is conducted to determine if the sequence index has exceeded the bounds established by the cycling table. If these bounds have not been exceeded the event code is read from the sequence table. If the limits have been exceeded, the sequence index is set back to the start of the event partition defined by the first column in the cycling table. The cycle repetition index is then incremented to indicate the total number of cycles through that partition. If the number of cycles does not exceed the limits established by the cycling table the event code is obtained directly from the sequence table. If the total number of cycles exceeds the limits, the cycling rep index and the cycle time are reinitialized to zero and the next partition entered by incrementing the event group counters. If the event group counter exceeds the number of partitions set up in the cycling table, the simulation is effectively terminated.

Following extraction of the event code, MCODE, from the sequence table a test is made to determine if the event is a measurement or a delay. If MCODE is greater than five, indicating a delay, the measurement time is found by adding the length of the delay to the current time.

If MCODE denotes a measurement a sequence test is initiated. First, a test is made to determine if a GPS update is to be made. If it is then the measurement time is found by

\[ T_{MEAS} = T_{IME} + DT_{GPS} \]

where: \( DT_{GPS} \) is the time required to make the GPS measurement

If the measurement is not a GPS update a test is made to determine if a measurement is being made by star tracker one. If this is the case, a vector from the star tracker to the star is generated by the subroutine BVECT. BVECT is described later in this chapter. Since the star being observed by the tracker might be obscured by a major body such as the sun, moon, or earth, a test is made by subroutine VISIBLE to determine if the tracker is occulted. VISIBLE is described later in this chapter. If the first star tracker is occulted the process is repeated for the second star tracker, and if both trackers are occulted the sequencer will delay measurements for a period of time. If the measurement code originally indicated a star tracker two sighting, the process is performed in the reverse sequence. Assuming that one of the star trackers is not occulted the measurement time is found by

\[ T_{MEAS} = T_{IME} + DT_{ST} \]

where: \( DT_{ST} \) is the time required to make a star tracker measurement
Following computation of the measurement time, either through delay sequencing or measurement sequencing, a test is performed to determine if there is sufficient time prior to the next GCP sighting for the measurement to be made. First, the number of scan lines to the GCP is computed by

\[
\text{LINLEFT} = \text{LINNUM} - (\text{TIME} - \text{TGCP}) / \text{DTPS}
\]

where:
- \(\text{LINLEFT}\) = number of scan lines remaining to the GCP
- \(\text{LINNUM}\) = number of times between last GCP and next GCT
- \(\text{TIME}\) = current time
- \(\text{TGCP}\) = the time at which last GCP was sighted
- \(\text{DTPS}\) = time required per scan line

If the number of scan lines left to the GCP is larger than the number of lines that will be scanned during the planned measurement, the measurement is made. If there is insufficient time remaining, the measurement code is set to 1 indicating a GCP sighting and TMEAS is computed.
### Sequence Operations

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>First visit to sequencer</td>
</tr>
<tr>
<td>2</td>
<td>Increment event index ISS</td>
</tr>
<tr>
<td>3</td>
<td>Store event index as ISEQ</td>
</tr>
<tr>
<td>4</td>
<td>Sequence index greater than sequence</td>
</tr>
<tr>
<td></td>
<td>Set sequence index to sequence start</td>
</tr>
<tr>
<td></td>
<td>Store sequence index ISS</td>
</tr>
<tr>
<td>5</td>
<td>Increment cycle repetition index</td>
</tr>
<tr>
<td>6</td>
<td>Cycle rep or time exceeds limits</td>
</tr>
<tr>
<td></td>
<td>Increment event group counter</td>
</tr>
<tr>
<td>7</td>
<td>Event group greater than max</td>
</tr>
<tr>
<td></td>
<td>Set cycle rep index to 0</td>
</tr>
<tr>
<td>8</td>
<td>Set cycle time to 0</td>
</tr>
<tr>
<td>9</td>
<td>Set sequence index to sequence start</td>
</tr>
<tr>
<td></td>
<td>Store sequence index ISS</td>
</tr>
</tbody>
</table>

Read event code MCODE from sequence table

<table>
<thead>
<tr>
<th>MCODE</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>.GE. 5</td>
<td>TMEAS = TIME + IDELAY(MCODE)</td>
</tr>
<tr>
<td>.EQ. 2</td>
<td>TMEAS = TIME + DTGPS</td>
</tr>
<tr>
<td>.EQ. 3</td>
<td>TMEAS = MCODE.EQ.4</td>
</tr>
</tbody>
</table>

Generate angle from ST2 to star

<table>
<thead>
<tr>
<th>Visible</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>BVECT</td>
</tr>
<tr>
<td>T</td>
<td>Generate angle from ST1 to star</td>
</tr>
<tr>
<td></td>
<td>Star visible to ST2</td>
</tr>
</tbody>
</table>

Generate angle from ST1 to star

<table>
<thead>
<tr>
<th>Visible</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>BVECT</td>
</tr>
<tr>
<td>T</td>
<td>Generate angle from ST2 to star</td>
</tr>
<tr>
<td></td>
<td>Star visible to ST1</td>
</tr>
</tbody>
</table>

Generate number of lines left prior to GCP sighting

<table>
<thead>
<tr>
<th>Time left prior to GCP sighting is less than TMEAS</th>
</tr>
</thead>
<tbody>
<tr>
<td>MCODE.EQ.1</td>
</tr>
<tr>
<td>.EQ. 1</td>
</tr>
<tr>
<td>T. 5 = time of GCP sighting</td>
</tr>
</tbody>
</table>

Update GCP index

Read number of lines to next GCP

Update GCP time flag

Return
SUBROUTINE GCSEQ
REAL*8 ANGLE, CT, ST, VEC
INCLUDE 'DEBUG.COM'
COMMON /DEBUG/ IENTER, IDEBUG

USER CONTROLLED PARAMETERS TO VARY DEBUG PRINT LEVEL

INTER IF 1, PRINTS WHEN ENTERS MOST SUBROUTINES
IDEBUG 0-10, HIGHER NUMBER MEANS MORE PRINT

INCLUDE 'TARGETS.COM'
COMMON /TARGETS/ MTYPE, IS, NS, JFLAG, MCODE, PI, TPI
LOGICAL UFLAG

MEASUREMENT SPECIFICATIONS

TYPE MEASUREMENT TYPE
UFLAG SET FOR STAR OBSTRUCTION
MCODE * * MEASUREMENT PROCESSING
PI PI
TPI 2*PI

INCLUDE 'SEQU.COM'
COMMON /SEQU/ IS(25), ITS(4,5), IAA(3,4), IFLAG
LOGICAL IFLAG

SEQUENCE OF EVENTS TABLES

IS EVENT SCHEDULER
ITS CYCLE SEQUENCER
IAA ALTERNATE ACTIONS
IFLAG SET WHEN TARGET IS OBSTRUCTED

INCLUDE 'TIME.COM'
COMMON /TIME/ TIME, TNEXT, TSTOP, TIA, DEL, TIN, DTN, DATEO, DATER, TZERO
LOGICAL TMEAS, TRACK, TIS, ITSN, DTA, DATER, TPRINT, DTPRINT

REAL*8 TIME, TNEXT, TSTOP, TIA, DEL, TIN, DTN, DATEO, DATER, TMEAS, TRACK, TIS,
ITSN, DTA, TZERO, DATER, TPRINT, DTPRINT

THESE ARE THE TIME REFERENCE FRAMES

TIME ATOMIC TIME SINCE INITIALIZATION (SEC)
TNEXT TIME FOR NEXT POSITION INTEGRATION (SEC)
TSTOP RUN TERMINATION TIME (SEC)
TIA ATTITUDE INTEGRATION TIME (SEC)
DEL STEP SIZE (SEC)
TIN POSITION INTEGRATION TIME (SEC)
DTN STEP SIZE (SEC)
DATEO DATE OF FLIGHT EPOCH (JD)
DATER DATE OF 1950 EPOCH (JD)
TZERO START TIME IN SEC. SINCE DATEO
TMEAS TIME NEEDED TO S.E.W AND ACQUIRE (SEC)
TIS REAL WORLD REFERENCE TIME (SEC)
THIS PROGRAM DETERMINES EVENT SEQUENCING

THE PROGRAM PARAMETERS ARE:

- **TISN** TIME FOR NEXT RW POSITION INTEGRATION (SEC)
- **DATA** USUALLY + DEL BUT + TSLEW - TIA WHEN DEL TOO LARGE AT MEASUREMENT TIME
- **TPRINT** TIME FOR PRINT (SEC)
- **DTPRINT** INCREMENT ON TPRINT (SEC)

THE LABELED COMMON (SEQ) PARAMETERS ARE:

- **MT** STAR SPECIFICATION TABLE
- **MTYPE** THE TYPE OF MEASUREMENT
- **MSET** THE STAR NUMBER SPECIFICATION
- **MCODE** ACTION

MCPSEQ

- **INCLUDE 'GCPOAT.COM'**
- **COMMON /GCPOAT/ IGCPOAT(4,10),LINNUM,TGCP,IDLMAX,GCPN,DTPS**
- **REAL*8 TGCP,GCPN,DTPS**
- **DIMENSION MT(15),MS(15),VEC(3)**

- **IF (ENTER.EQ.1) WRITE(6,999)**
  - **FORMAT('ENTERING GCPSEQ')**

- **FLAG.EQ. FALS. INDICATES NEW SEQUENCE TABLES**
C IF(IFLAG) GO TO 10
  05700  0022
  05800  0023
  05900  0024
  06000  0025
  06100  0026
  06200  0027
  06300  0028
  06400  0029
  06500  0030
  06600  0031
C
C INCREMENT SEQUENCE
C
07100
C
07200  0032
  07300  0033
  07400  0034
  07500  0035
  07600  0036
  07700  0037
  07800  0038
C
C
07900
C
08100  0040
  08200  0041
C
C
08400
C
08600
C
08800
C
09000
C
09100  0045
  09200  0046
  09300  0047
  09400  0048
  09500  0049
  09600  0050
C
C
09700
C
09900
C C GPS MEASUREMENTS?
10000
C
10200  0051
  10300  0052
  10400  0053
C
10500
C
10600
C
10700
C
10800
C
10900
C
11000  0054
11100
C
11200
STAR TRACKER 2: CHECK TO SEE IF VISIBLE AND GET ANGLE TO STAR

CALL BVECT(VEC.2)

CALL VISIBLE(VEC.4,JFLAG)  ! IS TRACKER OCCULTED

IF(.NOT. JFLAG) GO TO 55

OTHERWISE STAR TRACKER 1 MEASUREMENTS

MCODE=3

CALL BVECT(VEC.1)

CALL VISIBLE(VEC.4,JFLAG)  ! IS TRACKER OCCULTED

IF(.NOT. JFLAG) GO TO 55

GO TO 85

STAR TRACKER 1 MEASUREMENTS

MCODE=4

CALL BVECT(VEC.2)

CALL VISIBLE(VEC.4,JFLAG)  ! IS TRACKER OCCULTED

IF(.NOT. JFLAG) GO TO 55

GO TO 87

SET UP NORMAL DELAY

MCODE=5

TMEAS = TIME + DTST

GO TO 100

MCODE=15

GO TO 87

CHECK FOR TIM TO MAKE GCP MEASUREMENTS

TMEAS = TIME + IDLAY(MCODE-4)

GO TO 100

MCODE=5

TMEAS = TIME + IDLAY(MCODE-4)

GO TO 100

CHECK FOR TIM TO MAKE GCP MEASUREMENTS

TMEAS = TIME + IDLAY(MCODE-4)

GO TO 100

LINLEFT = LIMNUM - (TIME - TGC)/DTPS

IF(LINLEFT .LT. (TMEAS - TIME)/DTPS) GO TO 200

LINLEFT = LIMNUM - (TIME - TGC)/DTPS

IF(LINLEFT .LT. (TMEAS - TIME)/DTPS) GO TO 200
GCPSQ

ARRAYS

<table>
<thead>
<tr>
<th>Address</th>
<th>Type</th>
<th>Name</th>
<th>Bytes</th>
<th>Dimensions</th>
</tr>
</thead>
<tbody>
<tr>
<td>5-00000000A</td>
<td>I+4</td>
<td>IAA</td>
<td>48</td>
<td>(3.4)</td>
</tr>
<tr>
<td>5-00000000B</td>
<td>I+4</td>
<td>IDELAY</td>
<td>44</td>
<td>(11)</td>
</tr>
<tr>
<td>5-00000000C</td>
<td>I+4</td>
<td>IES</td>
<td>100</td>
<td>(25)</td>
</tr>
<tr>
<td>5-00000000D</td>
<td>I+4</td>
<td>IGCPST</td>
<td>160</td>
<td>(4.10)</td>
</tr>
<tr>
<td>5-00000000E</td>
<td>I+4</td>
<td>ITS</td>
<td>80</td>
<td>(4.5)</td>
</tr>
<tr>
<td>2-000000054</td>
<td>I+4</td>
<td>MS</td>
<td>60</td>
<td>(15)</td>
</tr>
<tr>
<td>2-000000018</td>
<td>I+4</td>
<td>MT</td>
<td>60</td>
<td>(15)</td>
</tr>
<tr>
<td>2-000000040</td>
<td>R+8</td>
<td>VEC</td>
<td>24</td>
<td>(3)</td>
</tr>
</tbody>
</table>

LABELS

<table>
<thead>
<tr>
<th>Address</th>
<th>Label</th>
<th>Address</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-00000006A</td>
<td>10</td>
<td>0-000000014E</td>
<td>50</td>
</tr>
<tr>
<td>0-00000011A</td>
<td>45</td>
<td>0-00000017C</td>
<td>55</td>
</tr>
<tr>
<td>0-00000019E</td>
<td>87</td>
<td>0-00000022B</td>
<td>200</td>
</tr>
<tr>
<td>1-000000000</td>
<td>999</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

FUNCTIONS AND SUBROUTINES REFERENCED

BVECT FOREXIT VISIBLE

Total Space Allocated = 1512 Bytes

COMMAND QUALIFIERS

FORTRAN LIST GCP,INDATA,MATAB,OUTDATA,RUNG,NAV,EPHEM,TRUEA,SRESS,OCCLT,VPERT,GCSEQ,VISIBLE,GENENV,TRG,GRD,TATE,BMAT,CMA

/CHECK=(NOBOUNDS,OVERFLOW)
/DEBUG=(NOSYMBOLS,TRACEBACK)
/FORTRAN /NOG_FLOATING /I=4 /OPTIMIZE /WARNING /MOD_LINES /NOMACHINE_CODE /CONTINUATIONS=19

COMPILATION STATISTICS

Run Time: 2.52 seconds
Elapsed Time: 27.61 seconds
Page Faults: 372
Dynamic Memory: 160 pages
2.2.1 CHECK FOR STAR OCCULTATION (VISIBLE)

Subroutine VISIBLE acts as an executive for the determination of whether the target star is occulted by a major body. The bodies checked are the earth, moon, and sun. The occultation geometry is checked by passing to subroutine OCCULT a unit vector to the target star, a vector to the center of the body of concern and the effective radius of the body of concern. OCCULT then returns a .TRUE. in the logic variable L if the star is occulted by the body. Figure A-7 is a VCLR of the process.

Input Variables and Output Variables

VEC = Unit vector to the target star. (Unitless)
RSO = Radius vector to the sun (KM)
RBO = Obstruction radius of the sun (KM)
RSM = Radius vector to the moon (KM)
RBM = Obstruction radius of the moon (KM)
R = Radius vector to the earth (KM)
RBE = Obstructing radius of the earth (KM)
L = Logical flag . TRUE. if occulted.

Calling Routines - GCPSEQ, MEASURE

Routine and Functions Called - OCCULT
Occulted by the sun?
Occulted by the moon?
Occulted by the earth?

Figure A-7
SUBROUTINE VISIBLE(V,EQ,L)
DIMENSION VEC(3)
REAL B, ZETL, Y, V, WAG
LOGICAL L
INCLUDE 'DEBUG.COM'
COMMON /DEBUG/ IENTFR, DEBUG

USER CONTROLLED PARAMETERS TO VARY DEBUG PRINT LEVEL
INTER IF 1, PRINTS WHEN ENTERS MOST SUBROUTINES
IDEBUG 0-10, HIGHER NUMBER MEANS MORE PRINT

INCLUDE 'RVEC.COM'
COMMON /RVEC/ R(3), RM(3), RQ(3), RMS(3), RSO(3), RSS(3), SB(3)
REAL*8 R, RM, RQ, RMS, RSO, RSS, SB

 THESE ARE RADIUS VECTORS IN ECI AND BODY COORDINATES
R EARTH CENTER TO S/C - ECI (KM)
RM MOON - ECI (KM)
RQ SUN - ECI (KM)
RMS SPACECRAFT TO MOON - ECI (KM)
RQO SUN - ECI (KM)
RSS EARTH CENTER TO STAR - ECI
RA ABSOLUTE OF VECTOR R (KM)
R2 SQUARE OF RA (KM 2)
R3 CUBE OF RA (KM 3)
RSM ABSOLUTE OF RMS (KM)

INCLUDE 'CONST.COM'
COMMON /CONST/ ATM, RMB, RBE, RBO, R2, RM, US, UE, J2, J3, J4, DTU, PKI
REAL*8 ATM, RMB, RBE, RBO, R2, RM, US, UE, J2, J3, J4, DTU, PKI

PROGRAM CONSTANTS
ATM S/C AREA TO MASS RATIO (METERS/KG)
RMB OBSTRUCTION RADIUS OF THE MOON (KM)
RBE " EARTH (KM)
RBO " SUN (KM)
R2 SQUARE OF THE EARTHS RADIUS (KM 2)
RM2 " LUNAR RADIUS (KM 2)
UM LUNAR GRAVITATION CONSTANT (KM 3/SEC 2)
US SOLAR " EARTH " " " U ZONAL GRAVITATIONAL HARMONIC TERMS
J2, J3, J4 DTU REGULARIZED TIME STEP SIZE (SEC)
PKI SOLAR PRESSURE CONSTANT

STAR AVAILABILITY

IF (IENTER .EQ. 1) WRITE (6, 999) VEC, NO
FORMAT ('ENTERING VISIBLE ', 3F15.5, 15)
IF (NO .EQ. 1) GO TO 10
C IS THE TARGET OCCULTED BY THE SUN

C

CALL OCCULT(RSO.VEC, RBO, L)

IF(L) GO TO 40

IF(NO.EQ.2) GO TO 20

C IS THE TARGET OCCULTED BY THE MOON

C

CALL OCCULT(RSM.VEC, RBM, L)

IF(L) GO TO 40

IF(NO.EQ.3) GO TO 30

C IS THE TARGET OCCULTED BY THE EARTH

C

CALL OCCULT(R, V, C, RBE, L)

IF(L) GO TO 40

IF(NO.EQ.4) GO TO 50

C IS THE TARGET OCCULTED BY THE S/C

C

V = VMAG(VEC, 3)

IF(VEC(3)/V.GT.ZETL) GO TO 50

40 TRUE.

50 RETURN

END
### ARRAYS

<table>
<thead>
<tr>
<th>Address</th>
<th>Type</th>
<th>Name</th>
<th>Bytes</th>
<th>Dimensions</th>
</tr>
</thead>
<tbody>
<tr>
<td>4-00000000</td>
<td>R+B</td>
<td>R</td>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>4-0000000B</td>
<td>R+B</td>
<td>RM</td>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>4-00000010</td>
<td>R+B</td>
<td>RG</td>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>4-00000018</td>
<td>R+B</td>
<td>RSM</td>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>4-00000028</td>
<td>R+B</td>
<td>RSO</td>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>4-00000040</td>
<td>R+B</td>
<td>RSM</td>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>4-00000058</td>
<td>R+B</td>
<td>RSO</td>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>4-00000078</td>
<td>R+B</td>
<td>RTG</td>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>4-00000098</td>
<td>R+B</td>
<td>SB</td>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>AP-00000004</td>
<td>R+B</td>
<td>VEC</td>
<td>24</td>
<td>(3)</td>
</tr>
</tbody>
</table>

### LABELS

<table>
<thead>
<tr>
<th>Address</th>
<th>Label</th>
<th>Address</th>
<th>Label</th>
<th>Address</th>
<th>Label</th>
<th>Address</th>
<th>Label</th>
<th>Address</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-00000069</td>
<td>10</td>
<td>0-00000085</td>
<td>20</td>
<td>0-000000A1</td>
<td>30</td>
<td>0-000000C0</td>
<td>40</td>
<td>0-000000C4</td>
<td>50</td>
</tr>
</tbody>
</table>

### FUNCTIONS AND SUBROUTINES REFERENCED

- OCCULT
- VMAG

### COMMAND QUALIFIERS

- FORTRAN /LIST GCP,INDATA,MATAB,OUTDATA,RUNG,DNAV,EPHEM,TRUEA,SPRESS,OCULT,GPERT,GCPSEQ,VISIBLE,GENENV,TREG,GRDOUT,RATE,BMAT,CMA
- /CHECK=(NOBOUNDS,OVERFLOW)
- /DEBUG=(NOSYMBOLS,TRACEBACK)
- /F77 /NOG_FLOATING /I4 /OPTIMIZE /WARNINGS /NOD_LINES /NOMACHINE_CODE /CONTINUATIONS=19

### COMPILATION STATISTICS

- Run Time: 1.19 seconds
- Elapsed Time: 13.87 seconds
- Page Faults: 351
- Dynamic Memory: 160 pages
2.2.1.1 SUBROUTINE OCCULT (RSB, RSS, RB, L)

Subroutine OCCULT determines whether the target star is occulted by the major body of concern.

Processing Requirements

The processing uses the dot product of the two vectors RSS and RSB and compares the sine of the included angle with the sine of the angle opposite RB in such a manner as to not require trigonometric functions.

\[ \text{Star} \]

\[ \text{Body} \]

\[ \text{S/C} \]

Figure A-8  Occultation Geometry
In referring to Figure A-8, if \( \alpha \) is greater than \( \beta \), the star will not be occulted. Further it may be seen that \( \beta < 90^\circ \) for all spherical bodies. Therefore, given

\[
\text{RSS} \cdot \text{RSB} = |\text{RSS}| |\text{RSB}| \cos \alpha
\]

if the following condition does not exist the star is occulted

Test A

\[
\text{RSS} \cdot \text{RSB} < 0 , \ \alpha > 90^\circ
\]

A second condition that must exists is that be greater than \( \beta \).

\( \sin \alpha > \sin \beta \) is also a valid condition indicating \( \alpha > \beta \).

and so is

\[
(S\!\!n\!\!a)^2 > (\!\!S\!in\!\!S)^2 \text{ provides } \alpha < 90^\circ
\]

Since

\[
(S\!\!n\!\!a)^2 = 1 - (\cos a)^2 = 1 - \left( \frac{\text{RSS} \cdot \text{RSB}}{|\text{RSS}| |\text{RSB}|} \right)
\]

and

\[
(SinB)^2 = \left( \frac{R_5}{\text{RSB}} \right)^2
\]

Non occultation is therefore implied by

\[
1 - \left( \frac{\text{RSS} \cdot \text{RSB}}{|\text{RSS}| |\text{RSB}|} \right)^2 > \left( \frac{\text{RB}}{|\text{RSB}|} \right)^2
\]

or Test B

\[
|R_{\text{RB}}|^2 - \left( \frac{\text{RSS} \cdot \text{RSB}}{|\text{RSS}|} \right)^2 > (\text{RB})^2
\]

Both of these expressions (A) and (B) are used in the program. Figure A-9 is a VCLR of the process.
Input Requirements

RSS = BSI unit vector to the target star (unitless)
RSB = BCI vector to the body of concern. (KM)
RB  = Obstruction radius of the body of concern. (KM)

Output Requirements

L - Logical variable. TRUE. if the target star is occulted, otherwise FALSE.

Subroutine and Functions Called - VMAG, VDOT
Calling Subroutine - VISIBLE
OCCULT - VCLR

<table>
<thead>
<tr>
<th></th>
<th>INITIALIZE L = TRUE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CALCULATE MAGNITUDE OF VECTOR TO BODY</td>
</tr>
<tr>
<td>T</td>
<td>CRITERION 2 SATISFIED</td>
</tr>
<tr>
<td>F</td>
<td>L = .FALSE.</td>
</tr>
<tr>
<td>T</td>
<td>CRITERION 1 SATISFIED</td>
</tr>
<tr>
<td>F</td>
<td>L = .FALSE.</td>
</tr>
<tr>
<td>NULL</td>
<td>L = .FALSE.</td>
</tr>
<tr>
<td>NULL</td>
<td></td>
</tr>
</tbody>
</table>

Figure A-9
SUBROUTINE OCCULT(S, RSS, RB, L)

C** This routine determines if satellite(s) is being
C** shaded by body B. RSS is ECI vector from satellite to
C** target. RSB is the ECI vector to the body, and RB is the
C** obstructed body radius. L is true for shading
C** the method uses the dot product of RSS and RSB and
C** compares the sine of the included angle with the
C** sine of the angle opposite RB in such a way as to
C** not need trig functions
C** calling routine - VISIBLE
C** called routine - VMAG
C** checked by Jack Wycrs 2 June 1980

C** INCLUDE 'DEBUG.COM'
C** COMMON /DEBUG/ ENTER, IDEBUG

DIMENSION RSB(3), RSS(3)
REAL*8 RA, RB, RDR, RSS, VMAG, VDOT
LOGICAL L
L = .TRUE.
RA = VMAG(RSB, 3)
RDR = VDOT(RSB, RSS, 3) .GE. RB * RB)
L = .FALSE.

IF(RDR .LE. 0.)L = .FALSE.
IF(IDEBUG .GT. 3) WRITE(6,999) RSS, RSS, RS, RB, L
FORMAT(4X, 'EXITING OCCULT ',7F13.2,L5)
RETURN
END
### PROGRAM SECTIONS

<table>
<thead>
<tr>
<th>Name</th>
<th>Bytes</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 $CODE</td>
<td>6</td>
<td>PIC CON REL LCL</td>
</tr>
<tr>
<td>1 $PCDATA</td>
<td>12</td>
<td>PIC CON REL LCL</td>
</tr>
<tr>
<td>2 $LOCAL</td>
<td>16</td>
<td>PIC CON REL LCL NOSHR NOEXE</td>
</tr>
<tr>
<td>3 DEBUG</td>
<td>20</td>
<td>PIC OVR REL GBL</td>
</tr>
</tbody>
</table>

### ENTRY POINTS

<table>
<thead>
<tr>
<th>Address</th>
<th>Type</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-00000000</td>
<td>OCCULT</td>
<td></td>
</tr>
</tbody>
</table>

### VARIABLES

<table>
<thead>
<tr>
<th>Address</th>
<th>Type</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>3-00000004</td>
<td>1+4</td>
<td>IDEBUG</td>
</tr>
<tr>
<td>AP-00000000</td>
<td>R+B</td>
<td>EB</td>
</tr>
</tbody>
</table>

### ARRAYS

<table>
<thead>
<tr>
<th>Address</th>
<th>Type</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>AP-00000000</td>
<td>R+B</td>
<td>RSB</td>
</tr>
<tr>
<td>AP-00000000</td>
<td>R+B</td>
<td>RSS</td>
</tr>
</tbody>
</table>

### LABELS

<table>
<thead>
<tr>
<th>Address</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-00000000</td>
<td>99'</td>
</tr>
</tbody>
</table>

### FUNCTIONS AND SUBROUTINES REFERENCED

<table>
<thead>
<tr>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>VDOT</td>
</tr>
<tr>
<td>VMAG</td>
</tr>
</tbody>
</table>

Total Space Allocated = 326 Bytes

### COMMAND QUALIFIERS

- FORTRAN /LIST GCP,INDATA,MATAB,OUTDATA,RUNG,DNAV,EPHEM,TRUEA,SPRESS,OCULT,GPERT,GCPSEQ,VISIBLE,GENENV,TREG,GYROUT,RATE,DMAT,CMA
- /CHECK=(NOBOUNDS,OVERFLOW)
- /DEBUG=(NOSYMBOLS,TRACSBACK)
- /F77 /NOG_FLOATING /14 /OPTIMIZE /WARNINGS /MOD_LINES /NOMACHINE_CODE /CONTINUATIONS=19
OCCULT

COMPILATION STATISTICS

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Run Time:</td>
<td>0.87 seconds</td>
</tr>
<tr>
<td>Elapsed Time:</td>
<td>9.62 seconds</td>
</tr>
<tr>
<td>Page Faults:</td>
<td>351</td>
</tr>
<tr>
<td>Dynamic Memory:</td>
<td>160 pages</td>
</tr>
</tbody>
</table>
2.2.1.1.1 Compute vector to star (BVECT)

The function of this module is to compute a unit vector in inertial space pointing along the boresight of the Kth star tracker where K is an input. The process, illustrated by the VCLR in Figure A-10, begins by establishing a unit vector in star tracker coordinates defined by

\[
U = \begin{bmatrix}
0 \\
0 \\
1
\end{bmatrix}
\]

Note that the z axis of the star trackers point along the boresight. This vector is transformed into body coordinates using a two step process. First, the vector is transformed from star tracker coordinates to nominal star tracker coordinates to account for known misalignments. The resulting vector is transformed from nominal star tracker coordinates to body coordinates to account for orientation of the sensor.

The transformation from body coordinates to inertial coordinates is computed by the subroutine AMAT. This transformation matrix is then used to transform the boresight vector into inertial coordinates.
<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Form unit vector in star tracker coordinates along the sensor boresight</td>
</tr>
<tr>
<td>2</td>
<td>Transform unit vector into nominal star tracker coordinates to account for known misalignment</td>
</tr>
<tr>
<td>3</td>
<td><strong>AMAT</strong> Compute transformation from body to inertial coordinates</td>
</tr>
<tr>
<td>4</td>
<td>Transform unit vector in body coordinates into inertial coordinates</td>
</tr>
<tr>
<td>5</td>
<td>Return vector to star in inertial coordinates</td>
</tr>
</tbody>
</table>

Figure A-10
SUBROUTINE VECT(VECT,K)

THE FUNCTION OF SUBROUTINE VECT IS TO ESTABLISH A UNIT
VECTOR ALONG THE INSTRUMENT BORESIGHT FOR THE KTH STAR
TRACKER. THE VECTOR WILL BE KNOWN IN INERTIAL SPACE.

INPUT PARAMETERS

K = AN INTEGER DEFINING THE STAR TRACKER
    TO BE USED

OUTPUT PARAMETERS

VECT = A UNIT VECTOR KNOWN IN INERTIAL SPACE
       DEFINING THE BORESIGHT OF THE
       KTH STAR TRACKER

WRITTEN BY JACK MYERS - 2JULY1980
EXT 4443

--------------------------------------------

0002 INCLUDE 'ENVIR.COM'
0003 COMMON /ENVIP/ STATE(10),PROFILE(10,4),INIT
0004 REAL*8 STATF.PHFILE

REAL WORLD STATE PARAMETERS

STATE STATE VALUES: X,Y,Z,XD,YD,ZD,E0,E1,E2,E3
PROFILE ATTITUDE PROFILE TIME (SEC) VS
INERTIAL ANGULAR RATES (RAD/SEC)
INIT INTEGRATION INITIALIZATION KEY (-1)

0005 INCLUDE 'STARPAR.COM'
0006 COMMON /STARPAR/ BS(2,2),SS(2,2),TNS(2,2),TBNS(3,3),TBNS(3,3,2),
               BSK(2,2),SSK(2,2),TNSK(3,3,2)
0007 REAL*8 BS,SS,TNS,TBNS,BSK,SSK,TNSK

STAR TRACKER PARAMETERS

IN EACH CASE THE LAST SUBSCRIPT REFERS TO THE
TRACK R USED
BS = BIAS - ACTUAL (RAD)
SS = NOISE STANDARD DEVIATION - ACTUAL (RAD)
TNS = MISALIGNMENT ARRAY - TRANSFORMATION FROM
STAR TRACKER TO NOMINAL
TBNS = ORIENTATION ARRAY - TRANSFORMATION FROM
STAR TRACKER TO BODY
BSK = BIAS - KNOWLEDGE (RAD)
SSK = NOISE STANDARD DEVIATION -KNOWLEDGE (RAD)
TNSK = MISALIGNMENT KNOWLEDGE ARRAY - TRANSFORMATION
FROM STAR TRACKER TO NOMINAL

0008 EQUIVALENCE (Q,STATE(7))
0009 DIMENSION UNIT(3),UNIT2(3),AI(3,3),BI(3,3),VECT(3)
0010 REAL*8 VECT,UNIT,UNIT2,AI,BI,0

--------------------------------------------

GENERATE A UNIT VECTOR IN TRACKER COORDINATES

UNIT(1) = 0.
### PROGRAM SECTIONS

<table>
<thead>
<tr>
<th>Name</th>
<th>Bytes</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 SCODE</td>
<td>143</td>
<td>PIC CON REL LCL</td>
</tr>
<tr>
<td>1 SDATA</td>
<td>8</td>
<td>PIC CON REL LCL</td>
</tr>
<tr>
<td>2 $LOCAL</td>
<td>316</td>
<td>PIC CON REL LCL</td>
</tr>
<tr>
<td>3 ENVIR</td>
<td>404</td>
<td>PIC OVR REL GBL</td>
</tr>
<tr>
<td>4 STARPAR</td>
<td>560</td>
<td>PIC OVR REL GBL</td>
</tr>
</tbody>
</table>

### ENTRY POINTS

<table>
<thead>
<tr>
<th>Address</th>
<th>Type</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-00000000</td>
<td>BVECT</td>
<td></td>
</tr>
</tbody>
</table>

### VARIABLES

<table>
<thead>
<tr>
<th>Address</th>
<th>Type</th>
<th>Name</th>
<th>Address</th>
<th>Type</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-00000000</td>
<td>R*B</td>
<td>A1</td>
<td>3-00000190</td>
<td>I*4</td>
<td>INIT</td>
</tr>
<tr>
<td>3-00000030</td>
<td>R*B</td>
<td>Q</td>
<td>2-000000C4</td>
<td>I*4</td>
<td>J</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>AP-000000B0</td>
<td>I*4</td>
<td>K</td>
</tr>
</tbody>
</table>

### ARRAYS

<table>
<thead>
<tr>
<th>Address</th>
<th>Type</th>
<th>Name</th>
<th>Bytes</th>
<th>Dimensions</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-00000030</td>
<td>R*B</td>
<td>A1</td>
<td>72</td>
<td>(3, 3)</td>
</tr>
<tr>
<td>2-00000078</td>
<td>R*B</td>
<td>B1</td>
<td>72</td>
<td>(3, 3)</td>
</tr>
<tr>
<td>4-00000000</td>
<td>R*B</td>
<td>BS</td>
<td>32</td>
<td>(2, 2)</td>
</tr>
<tr>
<td>4-00000160</td>
<td>R*B</td>
<td>BSK</td>
<td>32</td>
<td>(2, 2)</td>
</tr>
<tr>
<td>3-00000050</td>
<td>R*B</td>
<td>PROFILE</td>
<td>320</td>
<td>(10, 4)</td>
</tr>
<tr>
<td>4-00000020</td>
<td>R*B</td>
<td>SS</td>
<td>32</td>
<td>(2, 2)</td>
</tr>
<tr>
<td>4-00000180</td>
<td>R*B</td>
<td>SSK</td>
<td>32</td>
<td>(2, 2)</td>
</tr>
<tr>
<td>3-00000000</td>
<td>R*B</td>
<td>STATE</td>
<td>80</td>
<td>(10)</td>
</tr>
<tr>
<td>4-00000000</td>
<td>R*B</td>
<td>TBNS</td>
<td>144</td>
<td>(3, 3, 2)</td>
</tr>
</tbody>
</table>
BVECT

4-00000000 R=B TNS 144 (3, 3, 2)
4-00000010 R=B TNSK 144 (3, 3, 2)
2-00000000 R=B UNIT 24 (3)
2-00000010 R=B UNIT2 24 (3)
AP-00000000 R=B VECT 24 (3)

LABELS

Address Label
** 10

FUNCTIONS AND SUBROUTINES REFERENCED

AMAT MATAB

Total Space Allocated = 1431 Bytes

COMMAND QUALIFIERS

FORTRAN LIST GCP,INDATA,MATAB,OUTDATA,RUNG,DNAV,EPHEM,TRUEA,SPRESS,OCULT,GPERT,GCPSEQ,VISIBLE,GENENV,TREG,GRYOUT,RATE,BMAT,CMA

/CHECK=(NOBOUNDS,OVERFLOW)
/DEBUG=(NOSYMBOLS,TRACEBACK)
/FT7 /NOG_FLOATING /14 /OPTIMIZE /WARNINGS /NOD_LINES /NOMACHINE_CODE /CONTINUATIONS=19

COMPILATION STATISTICS

Run Time: 1.04 seconds
Elapsed Time: 13.93 seconds
Page Faults: 341
Dynamic Memory: 160 pages
2.3 Generate Environment (GENENV)

The generate environment module (GENENV) is responsible for propagating the true vehicle navigation and attitude states and for generating the gyro output between measurements. The true vehicle state is used to generate the measurements and for determining the error in the estimated state. The gyro output, which is processed by the executive through a call to GYRO, is used to propagate the vehicle attitude between measurements.

GENENV is called by the executive module (CCP). During the first call, GENENV is called by the true navigation state from present time up to measurement time using the real-world integrator, RUNG. This process is performed in small increments determined by the default integration interval DEL.

During all subsequent calls, GENENV simply generates the gyro output over a specified sampling period, DTA, by calling GYROUT. Before exiting, GENENV updates the vehicle time word, TIME.

A VCLR of GENENV is contained in Figure A-11.
New Measurement Sequence

Compute normal integration time using a fixed step size (TISN)

Compute integration step size (DT) as the minimum of TMEAS and TISN minus the time to which position has been integrated

Integrate true vehicle state from TIS to TIS+DT

Do until position is integrated to measurement time TIS=TMEAS

Set old measurement sequence flag

Compute total remaining gyro interval

Compute gyro output interval (DTA) as the minimum of the remaining interval and normal sampling interval

Generate Gyro output GYROUT

Update attitude integrator time flag TIA

Update atomic time flag

Figure A-11
SUBROUTINE GENENV(FLAG)

LOGICAL FLAG

REAL*8 DGYRO, DT, DSTAT

INCLUDE 'DEBUG.COM'

USER CONTROLLED PARAMETERS TO VARY DEBUG PRINT LEVEL

IF 1, PRINTS WHEN ENTERS MOST SUBROUTINES

DEBUG 0-10, HIGHER NUMBER MEANS MORE PRINT

INCLUDE 'TIME.COM'

COMMON /TIME/ TIME, TNEXT, TSTOP, TIA, DEL, TIN, DTA, DATEO, TZERO

COMMON /TIME/ TIME, TNEXT, TSTOP, TIA, DEL, TIN, DTA, DATEO, TZERO

REAL*8 TIME, TMEAS, TTRACK, TIS, TISN, DTA, DATET, DTPRINT, DTIME

THESE ARE THE TIME REFERENCE FRAMES

TMEASURE ATOMIC TIME SINCE INITIALIZATION (SEC)

TSTOP TIME FOR NEXT POSITION INTEGRATION (SEC)

TIA ATTITUDE INTEGRATION TIME (SEC)

DEL STEP SIZE (SEC)

TIN POSITION INTEGRATION TIME (SEC)

DTN STEP SIZE (SEC)

DATEO DATE OF FLIGHT EPOCH (JD)

DATE DATE OF 1950 EPOCH (JD)

TA00 START TIME IN SEC. SINCE DATEO

TELE# TIME NEEDED TO SLEW AND ACQUIRE (SEC)

TIS REAL WORLD REFERENCE TIME (SEC)

TISN TIME FOR NEXT RW POSITION INTEGRATION (SEC)

D TA USUALLY + DEL BUT + TSLW - TIA WHEN DEL

TSLW TOO LARGE AT MEASUREMENT TIME

TPRINT TIME FOR PRINT (SEC)

DTPRINT INCREMENT ON TPRINT (SEC)

INCLUDE 'TARGETS.COM'

COMMON /TARGETS/ MTYPE, IS, NS, UFLAG, MQODE, PI, TPI

LOGICAL UFLAG

REAL*8 PI, TPI

MEASUREMENT SPECIFICATIONS

MTYPE MEASUREMENT TYPE

MQODE SET FOR STAR OBSTRUCTION

MQODE MEASUREMENT PROCESSING

PI PI

TPI 2*PI

INCLUDE 'ENVIR.COM'

COMMON /ENVIR/ STATE(10), PROFILE(10,4), INIT

REAL*8 STATE, PROFILE

REAL WORLD STATE PARAMETERS
INCLUDE 'CONTRI.COM'
COMMON /CTRL/ MOP, TINT
REAL P TINT

PROGRAM CONTROL DESCRIPTORS FOR MULTIPLE RUNS

MOP MODE OF OPERATION
1 = PREFLIGHT SIMULATION
2 = POSTFLIGHT SIMULATION
3 = MONTE CARLO SIMULATION

TINT NUMBER OF SECONDS OF FULL OPERATION PER CYCLE

DIMENSION OSTATE(10)

IF(FLAG) GO TO 10
C---------------------------------------------------------------
C THIS IS A NEW MEASUREMENT
C---------------------------------------------------------------
C INTEGRATE TRU STATE TO MEASUREMENT TIME
C---------------------------------------------------------------
S CALL TREG(TIS,TISN)  I TISN=TIS+DTU
DT = MIN(TISN,TMEAS)-TIS
IF(DT,GT,.01) CALL RUNG(INIT,STATE,OSTATE,TIS,DT)
IF(TIS+.01,LT,TMEAS) GO TO S
FLAG = .TRUE.
C---------------------------------------------------------------
C GENERATE GYRO OUTPUT
C---------------------------------------------------------------
10 DGYRO=TMEAS - TIA  I COMPUTE INTERVAL REMAINING
DTA = MIN(DEL,DGYRO)  I COMPUTE OUTPUT INCREMENT
CALL GYROUT  I GENERATE GYRO OUTPUT
TIA+DTA  I UPDATE GYRO TIME WORD
TIME = TIA
RETURN
END
### Program Sections

<table>
<thead>
<tr>
<th>Name</th>
<th>Bytes</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 CODE</td>
<td>139</td>
<td>PIC CON REL LCL SHR EXE RD NOWRT LONG</td>
</tr>
<tr>
<td>2 LOCAL</td>
<td>132</td>
<td>PIC CON REL LCL NO SHR NOEKE RD WRT QUAD</td>
</tr>
<tr>
<td>3 DEBUG</td>
<td>8</td>
<td>PIC OVR REL GBL SHR NOEKE RD WRT LONG</td>
</tr>
<tr>
<td>4 TIME</td>
<td>136</td>
<td>PIC OVR REL GBL SHR NOEKE RD WRT LONG</td>
</tr>
<tr>
<td>5 TARGETS</td>
<td>36</td>
<td>PIC OVR REL GBL SHR NOEKE RD WRT LONG</td>
</tr>
<tr>
<td>6 ENVIR</td>
<td>404</td>
<td>PIC OVR REL GBL SHR NOEKE RD WRT LONG</td>
</tr>
<tr>
<td>7 CTRL</td>
<td>12</td>
<td>PIC OVR REL GBL SHR NOEKE RD WRT LONG</td>
</tr>
</tbody>
</table>

### Entry Points

<table>
<thead>
<tr>
<th>Address</th>
<th>Type</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td>GENENV</td>
</tr>
</tbody>
</table>

### Variables

<table>
<thead>
<tr>
<th>Address</th>
<th>Type</th>
<th>Name</th>
<th>Address</th>
<th>Type</th>
<th>Name</th>
<th>Address</th>
<th>Type</th>
<th>Name</th>
<th>Address</th>
<th>Type</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>4-0000000B R+B DATEDO</td>
<td>4-0000000070 R+B DATER</td>
<td>4-000000020 R+B DEL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2-00000008 R+B DT</td>
<td>4-0000000098 R+B DATA</td>
<td>4-000000030 R+B DTN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5-00000004 L+4 FLAG</td>
<td>3-000000004 L+4 IDDEBUG</td>
<td>3-000000000 I+4 ENTER</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5-00000000 C+4 IS</td>
<td>5-000000000 C+4 JFLAG</td>
<td>5-000000010 I+4 MCODE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5-00000000 C+4 MTYPE</td>
<td>5-000000008 C+4 NS</td>
<td>5-000000014 R+B PI</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4-0000000C R+B TIME</td>
<td>4-000000028 R+B TIN</td>
<td>7-000000004 R+B TINT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4-0000000E R+B TISM</td>
<td>4-000000048 R+B TIMEAS</td>
<td>4-000000008 R+B TNEXT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4-00000078 R+B TPRINT</td>
<td>4-000000050 R+B TRACK</td>
<td>4-000000010 R+B TSTOP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Arrays

<table>
<thead>
<tr>
<th>Address</th>
<th>Type</th>
<th>Name</th>
<th>Bytes</th>
<th>Dimensions</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-00000000 R+B DSTATE</td>
<td>80</td>
<td>(10)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6-00000005 R+B PROFILE</td>
<td>320</td>
<td>(10, 4)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6-00000000 R+B STATE</td>
<td>80</td>
<td>(10)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Labels

<table>
<thead>
<tr>
<th>Address</th>
<th>Label</th>
<th>Address</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-00000000</td>
<td>5</td>
<td>0-0000005D</td>
<td>10</td>
</tr>
</tbody>
</table>

### Functions and Subroutines Referenced

<table>
<thead>
<tr>
<th>Function</th>
<th>Subroutine</th>
</tr>
</thead>
<tbody>
<tr>
<td>GYROUT</td>
<td>RUNG</td>
</tr>
<tr>
<td></td>
<td>TREG</td>
</tr>
</tbody>
</table>

Total Space Allocated = 887 Bytes
GENENV


COMMAND QUALIFIERS

FORTRAN /LIST GCP,INDATA,MATAB,OUTDATA,RUNG,DNAV,EPHEM,TRUEA,SPRESS,OCULT,GPERT,GCPSEQ,VISIBLE,GENENV,TREG,GYROUT,RATE,BMAT,CMA
/CHECK=(NOBOUNDS,OVERFLOW)
/DEBUG=(NOSYMBOLS,TRACEBACK)
/FF77 /NOG_FLOATING /14 /OPTIMIZE /WARNINGS /NOD_LINES /NOMACHINE_CODE /CONTINUATIONS=19

COMPILATION STATISTICS

Run Time: 1.10 seconds
Elapsed Time: 19.39 seconds
Page Faults: 340
Dynamic Memory: 160 pages
2.3.1 INTEGRATE ACTUAL NAVIGATION STATE (RUNG)

The position state is advanced in time by numerical integration of the equations of motion. The second order equations of motion are composed of the external forces acting on the spacecraft. The external forces consist of geopotential, lunar gravitation and solar gravitation and radiation pressure.

A study showed that the Runge Kutta Gill (RKG) 4th order numerical integration method is optimal for this application. It is self-starting, handles variable step sizes, and is sufficiently accurate.

The Runge Kutta Gill (RKG) method for numerically integrating differential equations is described here.

The change in the value of the function during the computing interval is calculated by

$$
\Delta y = \frac{1}{6}\{k_1 + 2(1-k)k_2 + 2(1+k)k_3 + k_4\}
$$

where

$$
k_1 = hf(t_n, y_n) \quad \quad k = \sqrt{2}/2
$$
$$
k_2 = hf(t_n + \frac{h}{2}, y_n + \frac{h}{2}k_1)
$$
$$
k_3 = hf(t_n + \frac{h}{2}, y_n + (-\frac{h}{2} + k)k_1 + (1-k)k_2)
$$
$$
k_4 = hf(t_n + h, y_n - k\cdot k_2 + (1+k)k_3)
$$

$h =$ computing interval (seconds)

$t_n =$ time of beginning of computing interval (seconds)

$y_n =$ value of function at beginning of computing interval

The derivative function $f$ shall be evaluated four times to calculate the change in the function being integrated during the computing interval.

At first appearance, the software algorithm has no relation to the mathematical description of the integrator. The following is intended to show that the two are indeed identical.

Proceeding through the outer do loop for $K = 1$ to 4 we find the following results:

$K = 1$

$$
f = f(t_n, y_n)
$$
$$
t_1 = \frac{h}{2}f
$$
$$
y_1 = y_0 + \frac{h}{2}dtf
$$
$$
q_1 = f
$$

$K = 2$

$$
f' = f(t_n + \frac{dt}{2}, y_0 + \frac{dt}{2}f)
$$
$$
t_2 = (1 - \frac{\sqrt{2}}{2})(f' - f)
$$
$$
y_2 = y_0 + \frac{dt}{2}f + dt(1 - \frac{\sqrt{2}}{2})(f' - f)
$$
$$
= y_0 + dt(-\frac{1}{4} + \frac{\sqrt{2}}{2})f + dt(1 - \frac{\sqrt{2}}{2})f'
$$
$$
q_2 = f + 3 (1 - \frac{\sqrt{2}}{2})(f' - f) - (1 - \frac{\sqrt{2}}{2})f'
$$
$$
= f - 3(1 - \frac{\sqrt{2}}{2})f + 2(1 - \frac{\sqrt{2}}{2})f'
$$

A-75
\[ K = 3 \]

\[ f'' = f(t_n + \frac{dt}{2}, y + \frac{dt}{2}(-2 + \sqrt{2})f + \frac{dt}{2}(1 - \sqrt{2})f') \]

\[ t_3 = \left(1 + \frac{\sqrt{2}}{2}\right) \left(f'' - f + 3(1 - \frac{\sqrt{2}}{2})f - 2(1 - \frac{\sqrt{2}}{2})f'\right) \]

\[ y_3 = y_0 + \frac{dt}{2}(-1 + \sqrt{2})f + \frac{dt}{2}(1 - \frac{\sqrt{2}}{2})f' + \frac{dt}{2}(1 + \sqrt{2}) \left(f'' - f + 3(1 - \frac{\sqrt{2}}{2})f - 2(1 - \frac{\sqrt{2}}{2})f'\right) \]

\[ = y_0 - \frac{dt}{2} \sqrt{2}f' + \frac{dt}{2} \left(1 + \frac{\sqrt{2}}{2}\right)f'' \]

\[ q_3 = f - 3(1 - \frac{\sqrt{2}}{2})f + 2(1 - \frac{\sqrt{2}}{2})f' + 3 \left(1 + \frac{\sqrt{2}}{2}\right) \left(f'' - f + 3(1 - \frac{\sqrt{2}}{2})f - 2(1 - \frac{\sqrt{2}}{2})f'\right) \]

\[ = -\frac{1f}{2} - (1 + \frac{\sqrt{2}}{2})f' + 2(1 + \frac{\sqrt{2}}{2})f'' \]

\[ K = 4 \]

\[ f''' = f(t_n + dt, y_0 - \frac{dt}{2} \sqrt{2}f' + \frac{dt}{2} \left(1 + \frac{\sqrt{2}}{2}\right)f'') \]

\[ t_4 = \frac{1}{6} \left(f''' - 2 - \frac{1f}{2} - (1 + \frac{\sqrt{2}}{2})f' + 2(1 + \frac{\sqrt{2}}{2})f''\right) \]

\[ = \frac{1}{6} \left(f''' + f + 2(1 + \frac{\sqrt{2}}{2})f' - 4(1 + \frac{\sqrt{2}}{2})f''\right) \]

\[ y_4 = y_0 - \frac{dt}{2} \sqrt{2}f' + \frac{dt}{2} \left(1 + \frac{\sqrt{2}}{2}\right)f'' + \frac{1}{6} \left(dt f + 2dt(1 + \frac{\sqrt{2}}{2})f' - 4dt(1 + \frac{\sqrt{2}}{2})f'' + dt f''\right) \]

\[ = y_0 - \frac{1}{6} \left(dt f + 2dt(1 - \frac{\sqrt{2}}{2})f' + 2dt(1 + \frac{\sqrt{2}}{2})f'' + dt f''\right) \]

Making the substitutions

\[ K_1 = dt f \]
\[ K_2 = dt f' \]
\[ K_3 = dt f'' \]
\[ K_4 = dt f''' \]

Yields

\[ y - y_0 = \Delta y = \frac{1}{6} \left(K_1 + 2(1 - k)K_2 + 2(1 + k)K_3 + K_4\right) \]

which is identical to the original algorithm.
### COMPUTE POSITION DYNAMICS

<table>
<thead>
<tr>
<th>T</th>
<th>K = 1 OR 3</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>T = T + DT/2</td>
<td>NULL</td>
<td></td>
</tr>
</tbody>
</table>

\[ TP = AA(K) \cdot (ZD(I) - BB(K) \cdot Q(I)) \]

\[ W = Z(I) \]

\[ Z(I) = Z(I) + TP \cdot DT \]

\[ TP = (Z(I) - W)/DT \]

\[ Q(I) = Q(I) + 3*T - CC(K) \cdot ZD(I) \]

Where:
- \( AA = .5, 1 - \sqrt{2}/2, 1 + \sqrt{2}/2, .5/3 \)
- \( BB = 2, 1, 1, 2 \)
- \( CC = .5, 1 - \sqrt{2}/2, 1 + \sqrt{2}/2, .5 \)

**Figure A-12**
SUBROUTINE RUNG(INIT,ZD,T1,DT)

C*******************************************************************************
C THIS ROUTINE INTEGRATES THE SIX REAL WORLD NAVIGATION
C STATES FROM TIME T1 TO T1 + DT USING A RUNGE KUTTA GILL
C FORMULATION. STATE PARTIALS,ZD , ARE GENERATED BY DNAV
C AND INTEGRATED TO FORM Z.
C*******************************************************************************

C*******************************************************************************
C INCLUDE 'DEBUG.COM'
C*******************************************************************************

C USER CONTROLLED PARAMETERS TO VARY DEBUG PRINT LEVEL
C*******************************************************************************

COMMON /DEBUG/IEN,IDEBUG

DIMENSION Z(6,ZD(6))
DIMENSION AA(4),BB(4),CC(4),Q(7)
REAL*8 DT,T1,AA, BB,CC,Q,ZD
 DATA AA / .5, .292893, .1,.707107, .1/.666667/
 DATA BB /2. ,1.,1.1/, .2 /
 DATA CC /5.,292893,1.707107,.5/

DIMENSION Z(6),ZD(6)

C*******************************************************************************
C INITIALIZE O ARRAY
C*******************************************************************************

DO 10 I=1,6
  Z(I) = 0.
10 CONTINUE

C*******************************************************************************
C COMPUTE FOUR TERMS OF RUNGE KUTTA INTEGRATOR
C*******************************************************************************

DO 30 K=1,4
  CALL DNAV(T1,ZD,IDUMPY) ! COMPUTE POS, DYNAMIC PARTIALS
  IF(K.EQ.1.OR.K.EQ.2).T. T = T + DT/2.
30 CONTINUE

C*******************************************************************************
C INTEGRATE SIX NAVIGATION STATES
C*******************************************************************************

DO 30 I=1,6
  T = AA(K)*(ZD(I)-BB(K)*Q(I))
  Z(I) = Z(I)+T*DT
  Q(I) = Q(I)+3.*T-CC(K)*ZD(I)
30 CONTINUE

RETURN
END
### PROGRAM SECTIONS

<table>
<thead>
<tr>
<th>Name</th>
<th>Bytes</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 $CODE</td>
<td>169</td>
<td>Pic CON REL LCL SHR EKE RD NOWRT LONG</td>
</tr>
<tr>
<td>2 $LOCAL</td>
<td>232</td>
<td>Pic CON REL LCL NOHDR NOEKE RD WR Quad</td>
</tr>
<tr>
<td>3 DEBUG</td>
<td>8</td>
<td>Pic OVR REL GBK SHR NOEKE RD WR Quad</td>
</tr>
</tbody>
</table>

### ENTRY POINTS

<table>
<thead>
<tr>
<th>Address Type</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-00000000</td>
<td>RUNG</td>
</tr>
</tbody>
</table>

### VARIABLES

<table>
<thead>
<tr>
<th>Address Type</th>
<th>Name</th>
<th>Address Type</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>AP-000000140 R=B</td>
<td>DT</td>
<td>2-000000A0</td>
<td>I=4</td>
</tr>
<tr>
<td>3-000000000</td>
<td>I=4</td>
<td>ENTER</td>
<td>AP-00000040</td>
</tr>
<tr>
<td>AP-000000100 R=B</td>
<td>TI</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### ARRAYS

<table>
<thead>
<tr>
<th>Address Type</th>
<th>Name</th>
<th>Bytes</th>
<th>Dimensions</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-000000000 R=B</td>
<td>AA</td>
<td>32</td>
<td>(4)</td>
</tr>
<tr>
<td>2-000000200 R=B</td>
<td>BB</td>
<td>32</td>
<td>(4)</td>
</tr>
<tr>
<td>2-000000400 R=B</td>
<td>CC</td>
<td>32</td>
<td>(4)</td>
</tr>
<tr>
<td>2-000000600 R=B</td>
<td>Q</td>
<td>56</td>
<td>(7)</td>
</tr>
<tr>
<td>AP-000000000 R=B</td>
<td>Z</td>
<td>48</td>
<td>(6)</td>
</tr>
<tr>
<td>AP-0000000C0 R=B</td>
<td>ZD</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### LABELS

<table>
<thead>
<tr>
<th>Address</th>
<th>Label</th>
<th>Address</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>**</td>
<td>10</td>
<td>**</td>
<td>30</td>
</tr>
</tbody>
</table>

### FUNCTIONS AND SUBROUTINES REFERENCED

DNAV

Total Space Allocated = 409 Bytes
COMAND QUALIFIERS

FORTRAN /LIST GCP,INDATA,MATAB,OUTDATA,RUNG,ONAV,EPHEM,TRUEA,SPRESS,OCCLUD,GPRT,GPSEQ,VISIBLE,GENENV,TREG,GYROUT,RATE,BMAT,CMA

/’CHECK=’(NOBOUNDS,OVERFLOW)
/’DEBUG=’(NOSYMBOLS,TRACEBACK)
/’F77 /NOG_FLOATING /14 /OThm/PZ dWARS /NOd_LINES /NonMACHINE_CODE /CONTINUATIONS=19

COMPILATION STATISTICS

Fun Time: 1.02 seconds
Elapsed Time: 13.59 seconds
Page Faults: 332
Dynamic Memory: 160 pages
2.3.1.1 SPECIFY INTEGRATION STEP SIZE (TREG)

This module generates the endpoint of the integration period using the default interval DTV.
SPECIFY INTEGRATION STEP

**Figure A-13**

<table>
<thead>
<tr>
<th>COMPUTE MAXIMUM TIME FOR NEXT POSITION INTEGRATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T_{\text{next}} = \text{TIME} + \text{DTV}$</td>
</tr>
</tbody>
</table>
SUBROUTINE TREG(TIME,TNEXT)

INCLUDE 'DEBUG.COM'

COMMON /DEBUG/

USER CONTROLLED PARAMETERS TO VARY DEBUG PRINT LEVEL

INFER IF 1, PRINTS WHEN ENTERS MOST SUBROUTINES
IDEBUG 0-10, HIGHER NUMBER MEANS MORE PRINT

INCLUDE 'RVEC.COM'

COMMON /RVEC/ R1,R2,R3,RSM,RRA,RSO,RSS,RA,RA,2,RA,3,RA,MA,RTG

REAL A,RM,RO,RSM,RMG,RSS,SB,RA,2,RA,3,RA,MA,RTG

THESE ARE RADIUS VECTORS IN ECI AND BODY COORDINATES

R = EARTH CENTER TO S/C - ECI (KM)
RM = MOON - ECI (KM)
RO = SUN - ECI (KM)
RSM = SPACECRAFT TO MOON - ECI (KM)
RSM = ABSOLUTE OF VECTOR R (KM)
RSM = CUBE OF RA (KM 3)
RSM = ABSOLUTE OF RSM (KM)

REAL A,ATM,RBE,RBE,RBE,BBE,RBE

PROGRAM CONSTANTS

ATM = S/C AREA TO MASS RATIO (METERS/KG)
RM = OBSTRUCTION RADIUS OF THE MOON (KM)
RE = EARTH (KM)
RBO = SUN (KM)
R2 = SQUARE OF THE EARTHS RADIUS (KM 2)
RM2 = LUNAR RADIUS (KM 2)
LA = LUNAR GRAVITATION CONSTANT (KM 3/SEC 2)
US = SOLAR
U. = EARTH
J2,J3,J4 = ZONAL GRAVITATIONAL HARMONIC TERMS
DTU = REGULARIZED TIME STEP SIZE (SEC)
PKI = SOLAR PRESSURE CONSTANT

REAL TIME,TNEXT

TNEXT = TIME + DTU
RETURN
END
Program Sections

<table>
<thead>
<tr>
<th>Name</th>
<th>Bytes</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 CODE</td>
<td>13</td>
<td>PIC CON REL LCL</td>
</tr>
<tr>
<td>1 DEBUG</td>
<td>8</td>
<td>PIC OVR REL GBL</td>
</tr>
<tr>
<td>2 RVEC</td>
<td>224</td>
<td>PIC OVR REL GBL</td>
</tr>
<tr>
<td>3 CONST</td>
<td>112</td>
<td>PIC OVR REL GBL</td>
</tr>
</tbody>
</table>

Entry Points

<table>
<thead>
<tr>
<th>Address</th>
<th>Type</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-00000000</td>
<td>TREG</td>
<td></td>
</tr>
</tbody>
</table>

Variables

<table>
<thead>
<tr>
<th>Address</th>
<th>Type</th>
<th>Name</th>
<th>Bytes</th>
<th>Dimensions</th>
</tr>
</thead>
<tbody>
<tr>
<td>5-00000000</td>
<td>R=B</td>
<td>ATM</td>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>5-00000018</td>
<td>R=B</td>
<td>RM</td>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>5-00000030</td>
<td>R=B</td>
<td>RO</td>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>5-00000048</td>
<td>R=B</td>
<td>RSM</td>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>5-00000060</td>
<td>R=B</td>
<td>RSO</td>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>5-00000078</td>
<td>R=B</td>
<td>RSS</td>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>5-00000090</td>
<td>R=B</td>
<td>SB</td>
<td>24</td>
<td>(3)</td>
</tr>
</tbody>
</table>

Arrays

<table>
<thead>
<tr>
<th>Address</th>
<th>Type</th>
<th>Name</th>
<th>Bytes</th>
<th>Dimensions</th>
</tr>
</thead>
<tbody>
<tr>
<td>4-00000000</td>
<td>R=B</td>
<td>R</td>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>4-00000018</td>
<td>R=B</td>
<td>RM</td>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>4-00000030</td>
<td>R=B</td>
<td>RO</td>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>4-00000048</td>
<td>R=B</td>
<td>RSM</td>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>4-00000060</td>
<td>R=B</td>
<td>RSO</td>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>4-00000078</td>
<td>R=B</td>
<td>RSS</td>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>4-00000090</td>
<td>R=B</td>
<td>SB</td>
<td>24</td>
<td>(3)</td>
</tr>
</tbody>
</table>

Total Space Allocated = 357 Bytes

Command Qualifiers

FORTRAN /LIST GCP,INDATA,MAT,OUTDATA,RUNG,DNAV,Ephem,TRUEA,SPRESS,OCULT,GPERT,GCPSEQ,VISIBLE,GENENV,TREG,GROUT,RATE,BMAT,CMX
/CHECK=(NOBOUNDS,OVERFLOW)
/DEBUG=(NOSYMBOLS,TRACEBACK)
/IF77 /NOG_FLOATING /14 /OPTIMIZE /WARNINGS /NOD_LINES /NOWINAME_CODE /CONTINUATIONS=19
Compilation Statistics

- Run Time: 0.66 seconds
- Elapsed Time: 12.52 seconds
- Page Faults: 271
- Dynamic Memory: 150 pages
2.3.1.2 COMPUTE POSITION DYNAMICS (DNAV)

This module computes the forces acting on the spacecraft and generates the differential equation to be integrated by the state propagator.

The position of the spacecraft is calculated by solving three simultaneous, second order differential equations:

\[
\ddot{x}_1 = -\frac{\mu}{R^3} g_1(t, x) + a_1(t, x)
\]
\[
\ddot{x}_2 = -\frac{\mu}{R^3} g_2(t, x) + a_2(t, x)
\]
\[
\ddot{x}_3 = -\frac{\mu}{R^3} g_3(t, x) + a_3(t, x)
\]

where

\[
x = [x_1, x_2, x_3]^T
\]
\[
\mu = \text{Earth gravity constant } (3.985974204E + 05 \text{ Km}^3/\text{sec}^2)
\]
\[
R = (x_1^2 + x_2^2 + x_3^2)^{1/2}
\]
\[
x_1, x_2, x_3 = \text{Coordinate of spacecraft}
\]
\[
g_1, g_2, g_3 = \text{Accelerations caused by zonal harmonics of earth gravity}
\]
\[
a_1, a_2, a_3 = \text{Solar radiation pressure perturbations, sun and moon gravity}
\]

The zonal harmonic accelerations are computed by:

\[
g_1 = -\mu \frac{x_1}{R^3} F_1
\]
\[
g_2 = -\mu \frac{x_2}{R^3} F_1
\]
\[
g_3 = -\frac{\mu}{x^2} \frac{x_3}{R} F_1 - F_2
\]

where

\[
F_1 = \left| \frac{\Re}{R} \right|^2 \cdot f_1 J_2 + \left| \frac{\Re}{R} \right|^3 \cdot f_3 J_3 + \left| \frac{\Re}{R} \right|^4 \cdot f_4 J_4
\]
\[
F_2 = \left| \frac{\Re}{R} \right|^2 \cdot f_1 J_2 + \left| \frac{\Re}{R} \right|^3 \cdot f_2 J_3 + \left| \frac{\Re}{R} \right|^4 \cdot f_3 J_4
\]

and

\[
f_1 = -3 \frac{x_3}{R}
\]
\[
f_2 = -7.5 \left( \frac{x_3}{R} \right)^2 + 1.5
\]
\[
f_3 = -17.5 \left( \frac{x_3}{R} \right)^3 + 7.5 \left( \frac{x_3}{R} \right)
\]
\[
f_4 = -39.375 \left( \frac{x_3}{R} \right)^4 + 26.25 \left( \frac{x_3}{R} \right)^2 - 1.875
\]

\[
J_2 = 1082.7E -6 \text{ (Harmonic term in earth gravity model)}
\]
\[
J_3 = -2.56E -6 \text{ (Harmonic term in earth gravity model)}
\]
\[
J_4 = -1.58E -6 \text{ (Harmonic term in earth gravity model)}
\]

A-86
The process shown in Figure A-14 begins by computing the Julian day corresponding to current time. This parameter is required to compute the solar and lunar ephemerides. Given the locations of the sun, moon, and spacecraft, radius vectors are generated to each of these bodies. These radius vectors are then used to compute solar pressure, lunar and solar gravitational perturbation, and local gravitational effects. These forces are then used to generate the 2nd order differential equations of motion previously discussed.
**COMPUTE POSITION DYNAMICS**

<table>
<thead>
<tr>
<th><strong>COMPUTE JULIAN DAY</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>COMPUTE SOLAR/LUNAR EPHEMERIDES</strong></td>
</tr>
<tr>
<td><strong>COMPUTE RADIUS VECTOR FROM S/C TO EARTH</strong></td>
</tr>
<tr>
<td><strong>COMPUTE RADIUS VECTOR FROM S/C TO MOON</strong></td>
</tr>
<tr>
<td><strong>COMPUTE RADIUS VECTOR FROM S/C TO SUN</strong></td>
</tr>
<tr>
<td><strong>COMPUTE SOLAR RADIATION PRESSURE</strong></td>
</tr>
<tr>
<td><strong>COMPUTE SOLAR/LUNAR FURTURBATION</strong></td>
</tr>
<tr>
<td><strong>COMPUTE GEOPOTENTIAL - 4 ZONAL HARMONICS</strong></td>
</tr>
<tr>
<td><strong>COMPUTE SECOND-ORDER EQUATIONS OF MOTION</strong></td>
</tr>
</tbody>
</table>

*Figure A-14*
SUBROUTINE DNAVIT(ierr, JX, JY, ICALL)

DIMENSION X(6), JD, JAS(3), JASG(3)

REAL X, JD, JAS, JAST, JF2, JDATE, JF3, JF4, JF, JER, JER2, JZ, JZ2

REAL VMAG

INCLUDE 'DEBUG.COM'

COMMON /DEBUG, /ENTER, IDEBUG

USER CONTROLLED PARAMETERS TO VARY DEBUG PRINT LEVEL

INTER IF 1, PRINTS WHEN ENTERS MOST SUBROUTINES

IDEBUG 0-10, HIGHER NUMBER MEANS MORE PRINT

INCLUDE 'TIME.COM'

COMMON /TIME, /TIME, /NEXT, /TSTOP, /TIA, /DEL, /TIN, /DTN, /DATED, /TZERO

TIME, /TRACK, /TIS, /TISN, /DTA, /DATE, /ITPRINT, /DTPRINT

REAL TIME, /NEXT, /TSTOP, /TIA, /DEL, /TIN, /DTN, /DATED, /MEAS, /TRACK, /TIS,

/TISN, /DTA, /TZERO, /DATE, /ITPRINT, /DTPRINT

THESE ARE THE TIME REFERENCE FRAMES

TIME ATOMIC TIME SINCE INITIALIZATION (SEC)

TNEG TIME FOR NEXT POSITION INTEGRATION (SEC)

TSTOP RUN TERMINATION TIME (SEC)

TIA ATTITUDE INTEGRATION TIME (SEC)

DEL STEP SIZE (SEC)

TIN POSITION INTEGRATION TIME (SEC)

DTN STEP SIZE (SEC)

DATED DATE OF FLIGHT EPOCH (JD)

DATE DATE OF 1950 EPOCH (JD)

TZERO START TIME IN SECS. SINCE DATED

TSLEW TIME NEEDED TO SLEW AND ACQUIRE (SEC)

TIS REAL WORLD REFERENCE TIME (SEC)

TISN TIME FOR NEXT RW POSITION INTEGRATION (SEC)

DTA USUALLY + DEL BUT AT SLEW - TIA WHEN DEL

TLEW LARGE MEASUREMENT TIME

TPRINT TIME FOR PRINT (SEC)

DTPRINT INCREMENT ON TPRINT (SEC)

INCLUDE 'CONST.COM'


REAL+8 ATM, /RBV, /RE, /RBO, /RE2, /RM2, /UM, /US, /UE, /J2, /J3, /J4, /DTU, /PK1

PROGRAM CONSTANTS

ATM S/C AREA TO MASS RATIO (METERS/KG)

RBV OBSTRUCTION RADIUS OF THE MOON (KM)

RE EARTH (KM)

RB3 SUN (KM)

R2 SQUARE OF THE EARTHS RADIUS (K M 2)

RM2 LUNAR RADIUS (K M 2)

UM LUNAR GRAVITATION CONSTANT (K M 3/SEC 2)

US SOLAR (K M 3/SEC 2)

U EARTH (K M 3/SEC 2)

J2, J3, J4 ZONAL GRAVITATIONAL HARMONIC TERMS

DTU REGULARIZED TIME STEP SIZE (SEC)

PKI SOLAR PRESSURE CONSTANT
00800 00:3 C INCLUDE 'RVEC.CC'
00800 00:4 COMMON /RVEC/ R(I),R(I+1),R(I+2),R(I+3),RSM(I),RSM(I+1),RSM(I+2),RSM(I+3),SB(I),SB(I+1),SB(I+2),SB(I+3)
00900 01:5 REAL*8 R,RM,RO,RSM,RSO,RSS,SB,RA,R2,R3,RSMA,RTG
00900 00:3 THESE ARE RADIUS VECTORS IN ECI AND BODY COORDINATES

R = EARTH CENTER TO S/C - ECI (KM)
RM = MOON - ECI (KM)
RO = SUN - ECI (KM)
R0 = SPACECRAFT TO MOON - ECI (KM)
R = EARTH CENTER TO STAR - ECI
R = ABSOLUTE OF VECTOR R (KM)
R = SQUARE OF RA (KM^2)
R3 = CUBE OF RA (KM^3)
R3 = RA = VMAG(3)
RSMA = ABSOLUTE OF RSM (KM)

C------- COMPUTE JULIAN DAY -----------------------

00900 00:6 DO 10 1=1,3

10 R(I) = -X(I)
RSM(I) = RO(I)+R(I)
RM(I) = RM(I)+R(I)

C--- SOLAR PRESSURE COMPUTATIONS

01400 00:9 CALL SPRES(ASP)

01700 01:1 IF (DEBUG.GT.3) WRITE(6,998) TIN,ASP
01900 01:2 998 FORMAT(9X,'ACC. DUE TO SOLAR PRESSURE (KM/SEC^2) AT TIME ',F7.1,3E22.14)
02100 01:3 CALL SPRES(ASP)

C------ SOLAR LUNAR GRAVITATIONAL PERTURBATIONS

02300 01:5 CALL GPERT(X,ASL)

02700 01:7 IF (DEBUG.GT.3) WRITE(6,996) TIN,ASL
02900 01:8 996 FORMAT(9X,'ACC. DUE TO MOON AND SUN AT TIME ',F7.1,12X,3E22.14)
03100 01:9 CALL GPERT(X,ASL)

C------ GEODETIC - 4 ZONAL HARMONICS

03300 01:1 ZR = X(3)/RA
03500 01:3 ZR2 = ZR*ZR
03700 01:5 F1 = -3.*ZR
03900 01:7 F2 = -7.5*ZR^2+1.5
04100 01:9 F3 = ZR*(7.5-17.5*ZR2)
DNAV

04700 0637
04800 0038
04900 0039
05000 0040
05100 0041
05200
05300
05400
05500 0042
05600 0043
05700 0044
05800 0045
05900
06000 0046
06100 0047
06200 0048
06300 0049
06400 0050
06500 0051
06600
06700 0052
06800 0053

C COMPUTE STATE PARTIALS

C-----------------------------------------------

05500 0042
05600 0043
05700 0044
05800 0045
05900
06000 0046
06100 0047
06200 0048
06300 0049
06400 0050
06500 0051
06600
06700 0052
06800 0053

ENTRY POINTS

Address Type Name
0-00000000 DNAV

VARIABLES

Address Type Name Address Type Name Address Type Name Address Type Name
5-00000000 R8 ATM 2-00000040 R8 DATE 4-00000038 R8 DATE0 4-00000070 R8 DATER
4-00000020 R8 DEL 4-00000058 R8 DTA 4-00000030 R8 DTN 4-00000080 R8 DPRINT
5-00000060 R8 DTU 2-00000059 R8 F1 2-00000030 R8 F2 2-00000048 R8 F3
2-00000050 R8 F4 2-00000088 I4 I AP-000000090 I4 ICALL 3-00000004 I4 IDEBUG
3-00000000 I4 ENTER 5-00000050 R8 J2 5-00000050 R8 J3 5-00000058 R8 J4
5-00000068 R8 PKI 6-00000080 R8 R2 6-00000008 R8 R3 6-00000048 R8 RA
5-000000C10 R8 RBE 5-00000008 R8 RBD 5-00000018 R8 RBO 5-00000020 R8 RE2
2.3.2 GENERATE GYRO OUTPUT (GYROUT)

The spacecraft attitude is determined from a tabular input of the S/C Euler angle rates as a function of time. This tabulation is called an attitude rate profile array. The gyro errors must be added to attitude rates calculated from the attitude rate profile to make these data realistic. Since the gyro drift effects are accumulative the gyro attitude diverges significantly from the profile attitude permitting only the determination of gyro updates to an accumulated gyro state.

The gyro drift error is accumulative and causes the output data to drift from the nominal attitude profile. To accommodate this drift effect a gyro attitude state is maintained and updated with the attitude profile. This effect is illustrated below.

Procedure:

1. Find the attitude rate from the profile table at time $T_1$.
2. Compute the change in rate since previous time.
3. Update the gyro rate and multiply the new rate by the scale factor bias error and then add a drift bias error
   \[ W = SF_{be} \cdot (W_0 + \Delta W) + W_{db} \]
4. Account for the random errors
   \[ W' = SF_{re} \cdot W + W_{dr} \]
5. Compute the incremental gyro output data
   \[ \Delta W' = (W'_f - W'_i) / DT \]
6. Add random error to the gyro misalignment term.
7. Transform to output frame.
**GENERATE GYRO OUTPUT MODULE**

1. Find actual attitude rate from profile rate table
2. LaGrange interpolation module (Module 3.3.1)
3. Find incremental change in attitude rate
4. Compute random and bias values of gyro drift and scale factor
5. Compute sensed attitude rate
6. Find incremental change in actual attitude rate
7. Add increment to sensed attitude rate
8. Adjust for scale factor and drift biases
9. Adjust for scale factor and drift variances
10. Multiply rate by incremental time (DTA)
11. Find rate gyro output
12. Find random value of nonorthogonal coefficient

Do until all three components of gyro output are computed

13. Compute nonorthogonal transformation matrix
14. Transform gyro output to update quaternions
15. Normalize quaternion coefficients

*Figure A-15*

A-94
SUBROUTINE GYROUT
DIMENSION TEMP(3,3)
REAL GD,DUMMY,OMEG,TEMP,GAUSS
INCLUDE 'ARRAYS.COM'

COMMON /ARRAYS/ T1(3),T2(3),T3(3),T4(10),T11(3,3),T33(3,3)
.,,T4(4,4),T66(6,6),T77(6,6),T5(4),T6(4),T7(4)

REAL*8 T1,T2,T3,T4,T11,T33,T44,T66,T77,T5,T6,T7

THESE ARE TEMPORARY STORAGE ARRAYS FOR USE BY ALL MODULES

T1 - T4 SINGLE DIMENSION ARRAYS
T11 - T77 DUAL DIMENSIONED ARRAYS
T11 DUAL ARRAY: OFF DIAGONAL SET TO ZERO

INCLUDE 'TIME.COM'

COMMON /TIME/ TIME,TNEXT,TSTOP,TIA,DEL,TIN,DTN,DTE0,TZERO

TIME,TNEXT,TSTOP,TIA,DEL,TIN,DTN,DTE0,TZERO

THESE ARE THE TIME REFERENCE FRAMES

TIME ATOMIC TIME SINCE INITIALIZATION (SEC)
TNEXT TIME FOR NEXT POSITION INTEGRATION (SEC)
TSTOP RUN TERMINATION TIME (SEC)
TIA ATTITUDE INTEGRATION TIME (SEC)
DEL " STEP SIZE (SEC)
TIN POSITION INTEGRATION TIME (SEC)
DTN " STEP SIZE (SEC)
DATE0 DATE OF FLIGHT EPOCH (JD)
DATER DATE OF 1950 EPOCH (JD)
TZERO START TIME IN SECS. SINCE DATE0
TSLEW TIME NEEDED TO SLEW AND ACQUIRE (SEC)
TIS REAL WORLD REFERENCE TIME (SEC)
TISN TIME FOR NEXT RW POSITION INTEGRATION (SEC)
DTA USUALLY + DEL BUT + TSLEW - TIA WHEN DEL

OVER LARGE AT MEASUREMENT TIME

TPRINT TIME FOR PRINT (SEC)
DTPRINT INCREMENT ON TPRINT (SEC)

INCLUDE 'TMAT.COM'

COMMON /TMAT/ A(3,3),B(3,3),C(3,3),EM(4,3)

REAL*8 A,B,C,EM

TRANSFORMATION MATRICES

A INERTIAL TO BODY AXES
B GYRO TO BODY AXES
C GYRO NON-ORTHOGONAL TO GYRO AXES
EM BODY TO QUATERNIAN AXES

INCLUDE 'NOISE.COM'

COMMON /NOISE/ BWD(3),SWD(3),BSF(3),SSF(3),BD(3),SD(3)

, ,BDD(3),SDU(3),SRM,BRE,SRE

REAL*8 BWD,SWD,BSF,SSF,BD,SD,BDD,SDU,SRM,BRE,SRE
REAL WORLD GYRO MEASUREMENT ERRORS

- GYRO DRIFT (RAD / SEC)
- GYRO SCALE FACTOR
- GYRO NONORTHOGONALITY (RAD)
- GYRO RELATIVE ORIENTATION (RAD)

INCLUDE 'ASTATE.COM'

COMMON /ASTATE/ DE(4),E(4),WD(3),SF(3),D(3),DD(3)
REAL*B DE,E,WD, SF, D, DD

ATTITUDE STATE AND CONSIDERED PARAMETERS

- DIFFERENTIAL OF QUATERNIONS
- QUATERNIONS
- GYRO DRIFT RATE (RAD/SEC)
- GYRO SCALE FACTOR
- GYRO NON-ORTHOGONALITY (RAD)
- GYRO RELATIVE ORIENTATION (RAD)

INCLUDE 'ENVIR.COM'

COMMON /ENVIR/ STATE(10),PROFILE(10,4),INIT
REAL*B STATE, PROFILE

REAL WORLD STATE PARAMETERS

STATE STATE VALUES: X,Y,Z,XD,YD,ZD,E0,E1,E2,E3
PROFILE ATTITUDE PROFILE-TIME (SEC) VS
INIT INTEGRATION INITIALIZATION KEY (-1)

INCLUDE 'ROTAT.COM'

COMMON /ROTAT/ DTHR(3),DTHEM(3),DTHE(3)
REAL*B DTHR, DTHEM, DTHE

GYRO ATTITUDE PARAMETERS

DTHR REAL WORLD GYRO DATA (RAD)
DTHEM FILTER WORLD GYRO DATA (RAD)
DTHE FILTER WORLD COMPENSATED GYRO DATA (RAD)

DIMENSION OMEG(3), DUMMY(3)

C***************************************************************************
C INTERPOLATE TO FIND INERTIAL RATES BETWEEN TIMES IN TABLE
C***************************************************************************
DO 200 I = 1, 110
  IF (PROFILE(I,1) .LT. 0.01) GO TO 200
  IF (TIME,GT.PROFILE(I,1)) GO TO 200
  DO 100 J = 2, 4
    OMEG(I,J-1) = PROFILE(I,J-1)
  100 CONTINUE
GYROUT

02500 0033 GO TO 205
02600 0034 CONTINUE
02700 0035 200 IF (ENTER.EQ.2) WRITE(6,125)
02800 0036 125 FORMAT(1,ENTER.EQ.2))
02900 0037 IF (ENTER.EQ.2) WRITE(6,150) TIME,OWEG
03000 0038 150 FORMAT(13,TIME,'E5.1,OWEG*1.3(1X,E16.9))
03100 C******************************************************************************
03200 C COMPUTE TRANSFORMATION MATRIX FROM GYRO TO BODY COORDINATES
03300 C******************************************************************************
03400 0039 CALL EVAT(200,11)
03500 C******************************************************************************
03600 C COMPUTE TRANSFORMATION MATRIX ACCOUNTING FOR GYRO NONORTHOGONALITY
03700 0040 CALL CMAT(FSF,BD,T33)
03800 C******************************************************************************
04000 0041 IF (ENTER.EQ.2) WRITE(6,175) ((T(1:1,J)),J=1,3),I=1,3)((T33(1:1,J)),J=1,3),I=1,3)
04200 0042 175 FORMAT(13,B15.3(1X,E16.9)))/',' C15.3(1X,E16.9)')/
04300 C******************************************************************************
04400 C FIND INVERSE OF D INVERSE OF B INVERSE OF B+C, STORE IN TEMP
04500 C******************************************************************************
04600 0043 CALL MATAB(T11,T33,TEMP,3,3,3)
04700 0044 CALL MIN/3(TEMP,TEMP)
04800 C******************************************************************************
04900 C FIND REAL WORLD ANGLE IN GYRO ANGLES
05000 C******************************************************************************
05100 C******************************************************************************
05200 0045 CALL MATAB(TEMP,OWEG,T11,3,3,1)
05300 C******************************************************************************
05400 C CORRUPT REAL WORLD GYRO ANGLES WITH BIAS
05500 C******************************************************************************
05600 0046 DO 300 I = 1,3
05700 0047 300 DTHRI(I) = (B(I)-I + T1(1))*DTA
05800 C******************************************************************************
05900 0048 IF (ENTER.EQ.2) WRITE(6,180) ((TEV(I,J)),J=1,3),I=1,3),DTHI,DTHM
06000 0049 180 FORMAT(13,B15.3(1X,E16.9)))/',' C15.3(1X,E16.9)')/
06100 C REAL WORLD CHANGES IN GYRO ANGLES:3(1X,E16.9))/,' C15.3(1X,E16.9)')/
06200 C FILTER CHANG 3 IN GYRO ANGLES:3(1X,E16.9))/,' C15.3(1X,E16.9)')/
06300 C******************************************************************************
06400 C FIND FILTER WORLD CHANGE IN GYRO ANGLES BY ADDING RANDOM ERROR
06500 C******************************************************************************
06600 C******************************************************************************
06700 0050 DO 400 J = 1,3
06800 0051 400 GO = GAUSS(0.0,5011))
06900 C******************************************************************************
07000 C******************************************************************************
07100 C UPDATE REAL WORLD QUATERNION
07200 C******************************************************************************
07300 C******************************************************************************
07400 C******************************************************************************
07500 0053 IF (ENTER.EQ.2) WRITE(6,500)
07600 0054 500 FORMAT(13,CALL MATT FOR REAL WORLD )
07700 0055 CALL MATT(DTHI,BDC,BD,BSF,BO,DUMMY,STATE(7))
07800 C******************************************************************************
07900 0056 RETURN
08000 0057 END
## PROGRAM SECTIONS

<table>
<thead>
<tr>
<th>Name</th>
<th>Bytes</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 SCODE</td>
<td>576</td>
<td>PIC CON REL LCL SHR EXE RD NOWRT LONG</td>
</tr>
<tr>
<td>1 SPDATA</td>
<td>271</td>
<td>PIC CON REL LCL SHR NOEXE RD NOWRT LONG</td>
</tr>
<tr>
<td>2 SLOCAL</td>
<td>328</td>
<td>PIC CON REL LCL NOSHR NOEXE RD WRT QUAD</td>
</tr>
<tr>
<td>3 ARRAYS</td>
<td>1096</td>
<td>PIC OVR REL GBL SHR NOEXE RD WRT LONG</td>
</tr>
<tr>
<td>4 TIME</td>
<td>136</td>
<td>PIC OVR REL GBL SHR NOEXE RD WRT LONG</td>
</tr>
<tr>
<td>5 TMAT</td>
<td>312</td>
<td>PIC OVR REL GBL SHR NOEXE RD WRT LONG</td>
</tr>
<tr>
<td>6 NOISE</td>
<td>216</td>
<td>PIC OVR REL GBL SHR NOEXE RD WRT LONG</td>
</tr>
<tr>
<td>7 ASTATE</td>
<td>160</td>
<td>PIC OVR REL GBL SHR NOEXE RD WRT LONG</td>
</tr>
<tr>
<td>8 ENVIR</td>
<td>40</td>
<td>PIC OVR REL GBL SHR NOEXE RD WRT LONG</td>
</tr>
<tr>
<td>9 ROTAT</td>
<td>72</td>
<td>PIC OVR REL GBL SHR NOEXE RD WRT LONG</td>
</tr>
</tbody>
</table>

## ENTRY POINTS

<table>
<thead>
<tr>
<th>Address</th>
<th>Type</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-00000000</td>
<td></td>
<td>GYROUT</td>
</tr>
</tbody>
</table>

## VARIABLES

<table>
<thead>
<tr>
<th>Address</th>
<th>Type</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>4-00000003</td>
<td>R+B</td>
<td>BRE</td>
</tr>
<tr>
<td>4-00000030</td>
<td>R+B</td>
<td>DTA</td>
</tr>
<tr>
<td>4-00000000</td>
<td>1+4</td>
<td>I</td>
</tr>
<tr>
<td>4-00000000</td>
<td>R+B</td>
<td>SRE</td>
</tr>
<tr>
<td>4-00000058</td>
<td>R+B</td>
<td>TIN</td>
</tr>
<tr>
<td>4-00000078</td>
<td>R+B</td>
<td>THROT</td>
</tr>
<tr>
<td>4-00000038</td>
<td>R+B</td>
<td>DATED</td>
</tr>
<tr>
<td>4-00000060</td>
<td>R+B</td>
<td>DTPAINT</td>
</tr>
<tr>
<td>4-00000000</td>
<td>A+B</td>
<td>A</td>
</tr>
<tr>
<td>4-00000000</td>
<td>A+B</td>
<td>B</td>
</tr>
<tr>
<td>4-00000000</td>
<td>A+B</td>
<td>BD</td>
</tr>
<tr>
<td>4-00000000</td>
<td>A+B</td>
<td>BDD</td>
</tr>
<tr>
<td>4-00000000</td>
<td>A+B</td>
<td>BSF</td>
</tr>
<tr>
<td>4-00000000</td>
<td>A+B</td>
<td>BWD</td>
</tr>
<tr>
<td>4-00000000</td>
<td>A+B</td>
<td>C</td>
</tr>
<tr>
<td>4-00000000</td>
<td>A+B</td>
<td>D</td>
</tr>
<tr>
<td>4-00000000</td>
<td>A+B</td>
<td>DD</td>
</tr>
<tr>
<td>4-00000000</td>
<td>A+B</td>
<td>DE</td>
</tr>
<tr>
<td>4-00000000</td>
<td>A+B</td>
<td>DTHE</td>
</tr>
<tr>
<td>4-00000000</td>
<td>A+B</td>
<td>DTHEM</td>
</tr>
<tr>
<td>4-00000000</td>
<td>A+B</td>
<td>DTHR</td>
</tr>
<tr>
<td>4-00000000</td>
<td>A+B</td>
<td>DUMMY</td>
</tr>
<tr>
<td>4-00000000</td>
<td>A+B</td>
<td>E</td>
</tr>
<tr>
<td>4-00000000</td>
<td>A+B</td>
<td>EM</td>
</tr>
<tr>
<td>4-00000000</td>
<td>A+B</td>
<td>QMEG</td>
</tr>
</tbody>
</table>

## ARRAYS

<table>
<thead>
<tr>
<th>Address</th>
<th>Type</th>
<th>Name</th>
<th>Bytes</th>
<th>Dimensions</th>
</tr>
</thead>
<tbody>
<tr>
<td>5-000000000</td>
<td>R+B</td>
<td>A</td>
<td>72</td>
<td>(3, 3)</td>
</tr>
<tr>
<td>5-000000000</td>
<td>R+B</td>
<td>B</td>
<td>72</td>
<td>(3, 3)</td>
</tr>
<tr>
<td>5-000000000</td>
<td>R+B</td>
<td>BD</td>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>5-000000000</td>
<td>R+B</td>
<td>BDD</td>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>5-000000000</td>
<td>R+B</td>
<td>BSF</td>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>5-000000000</td>
<td>R+B</td>
<td>BWD</td>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>5-000000000</td>
<td>R+B</td>
<td>C</td>
<td>72</td>
<td>(3, 3)</td>
</tr>
<tr>
<td>5-000000000</td>
<td>R+B</td>
<td>D</td>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>5-000000000</td>
<td>R+B</td>
<td>DD</td>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>5-000000000</td>
<td>R+B</td>
<td>DE</td>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>5-000000000</td>
<td>R+B</td>
<td>DTHE</td>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>5-000000000</td>
<td>R+B</td>
<td>DTHEM</td>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>5-000000000</td>
<td>R+B</td>
<td>DTHR</td>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>5-000000000</td>
<td>R+B</td>
<td>DUMMY</td>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>5-000000000</td>
<td>R+B</td>
<td>E</td>
<td>32</td>
<td>(4)</td>
</tr>
<tr>
<td>5-000000000</td>
<td>R+B</td>
<td>EM</td>
<td>96</td>
<td>(4, 3)</td>
</tr>
<tr>
<td>2-000000000</td>
<td>R+B</td>
<td>QMEG</td>
<td>24</td>
<td>(3)</td>
</tr>
</tbody>
</table>
GYROUT

8-00000050 R*8 PROFILE 320 (10, 4)
6-00000078 R*8 SD 24 (3)
6-0000008B R*8 SDD 24 (3)
7-0000005B R*8 SF 24 (3)
6-00000058 R*8 SIF 24 (5)
8-00000000 R*8 STATE 80 (10)
6-00000018 R*8 S#D 24 (3)
3-00000000 R*8 T1 24 (3)
3-00000098 R*8 T11 72 (3, 3)
3-00000018 R*8 T2 24 (3)
3-00000030 R*8 T3 24 (3)
3-00000060 R*8 T33 72 (3, 3)
3-00000068 R*8 T4 80 (10)
3-00000128 R*8 T44 126 (4, 4)
3-00000368 R*8 T5 32 (4)
3-00000408 R*8 T6 32 (4)
3-0000014B R*8 T66 288 (6, 6)
3-00000428 R*8 T7 32 (4)
3-000002CB R*8 T77 288 (6, 3)
2-00000000 R*8 TEMP 72 (3, 3)
7-00000040 R*8 WD 24 (3)

LABELS

<table>
<thead>
<tr>
<th>Address</th>
<th>Label</th>
<th>Address</th>
<th>Label</th>
<th>Address</th>
<th>Label</th>
<th>Address</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-00000000</td>
<td>100</td>
<td>1-00000000</td>
<td>125</td>
<td>1-00000013</td>
<td>150</td>
<td>1-00000033</td>
<td>175</td>
</tr>
<tr>
<td>0-00000088</td>
<td>205</td>
<td>**</td>
<td>300</td>
<td>**</td>
<td>400</td>
<td>1-00000062</td>
<td>180</td>
</tr>
<tr>
<td>0-00000083</td>
<td>200</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

FUNCTIONS AND SUBROUTINES REFERENCED

BMAT CMAT GAUSS KATT MATAB MINV3

Total Space Allocated = 3571 Bytes

COMMAND QUALIFIERS

FORTRAN /LIST GCP,INDATA,MATAB,OUTDATA,RUNG,NAV,EPHEM,TRUEA,SPRESS,OCault,GPERt,GCPSEQ,VISIBlE,GENENV,TREG,GYROUT,RATE,BMAT,CMA

/PROC=(NOBOUNDS,OVERFLOW)
/DEBUG=(NOSYMBOLS,TRACEBACK)
/F77 /NOG_FLOATING /I4 /OPTIMIZE /WARNINGS /NOO_LINES /NOMACHINE_CODE /CONTINUATIONS=19

Compilation Statistics

Run Time: 2.92 seconds
Elapsed Time: 40.70 seconds
Page Faults: 406
Dynamic Memory: 160 pages
2.3.3 Transformation from Euler Angles (BMAT)

Subroutine BMAT uses DD(3) an array of Euler angles to generate the direction cosine array B(3,3). The B matrix is an array used for transformation of data from the gyro coordinate set to the body coordinate set.

Processing Requirements

The processing generates a standard Euler angle transformation with a rotational sequence starting from the gyro coordinates first about the z gyro axes, then the y axis, followed by the x axis. All rotations are considered positive.

Input Requirements

DD = Euler angle array

DD(1) = x-axis rotation (rad)
DD(2) = y-axis rotation (rad)
DD(3) = z-axis rotation (rad)

Output Requirements

B = Direction cosine transformation from gyro axes to body axes (unitless)

Subroutines and Function Called - SIN, COS (Math Package)

Calling Subroutines - GYROUT, KATT
"B" TRANSFORMATION MATRIX

FIND SINE AND COSINE OF EACH OF THE THREE COMPONENTS OF GYRO RELATIVE ORIENTATION

COMPUTE ELEMENTS OF B MATRIX USING THE ABOVE, WHERE B TRANSFORMS DATA FROM GYRO TO Sextant COORDINATES

Figure A-16
SUBROUTINE BMAT(DD,B)

THIS ROUTINE COMPUTES THE (3,3) MATRIX
THE B MATRIX TRANSFORMS DATA FROM GYRO TO BODY COORDINATE

INCLUDE 'DEBUG.COM'

COMMON /DEBUG/:ENTER, IDBUG

C
C USER CONTROLLED PARAMETERS TO VARY DEBUG PRINT LEVEL
C
INTER IF 1, PRINTS WHEN ENTERS MOST SUBROUTINES
IDBUG 0-10, HIGHER NUMBER MEANS MORE PRINT

DIMENSION DD(3),B(3,3)

REAL*8 DD,B,CDX,CDY,CDZ,SDX,SDY,SDZ

CDZ = COS(DO(3))
SDZ = SIN(DO(3))

CDY = COS(DO(2))
SDY = SIN(DO(2))

CDX = COS(DO(1))

SDX = SIN(DO(1))

B(1,1) = CDZ*CDY
B(1,2) = SDZ*CDY
B(1,3) = -SDY

B(2,1) = CDZ*SDY*SDX-SDZ*CDX
B(2,2) = SDZ*SDY+SDX*CDZ*CDX
B(2,3) = CDY*SDX

B(3,1) = CDZ*SDY+CDX*SDZ*SDX
B(3,2) = SDZ*SDY*CDX-CDZ*SDX
B(3,3) = CDY*CDX

RETURN
END
**PROGRAM SECTIONS**

<table>
<thead>
<tr>
<th>Name</th>
<th>Bytes</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>CODE</td>
<td>223</td>
<td>PIC CON REL LCL SHR EXE RD NOWRT LONG</td>
</tr>
<tr>
<td>LOCAL</td>
<td>88</td>
<td>PIC CON REL LCL NOSHR NOEXE RD WRT QUAD</td>
</tr>
<tr>
<td>DEBUG</td>
<td>8</td>
<td>PIC OVR REL GBL SHR NOEXE RD WRT LONG</td>
</tr>
</tbody>
</table>

**ENTRY POINTS**

<table>
<thead>
<tr>
<th>Address</th>
<th>Type</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-00000000</td>
<td>BMAT</td>
<td></td>
</tr>
</tbody>
</table>

**VARIABLES**

<table>
<thead>
<tr>
<th>Address</th>
<th>Type</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-00000000</td>
<td>R+8</td>
<td>CDX</td>
</tr>
<tr>
<td>2-00000000</td>
<td>R+8</td>
<td>CDY</td>
</tr>
<tr>
<td>2-00000000</td>
<td>R+8</td>
<td>SDO</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Address</th>
<th>Type</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>3-00000000</td>
<td>I+4</td>
<td>ENTER</td>
</tr>
</tbody>
</table>

**ARRAYS**

<table>
<thead>
<tr>
<th>Address</th>
<th>Type</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>AP-00000000</td>
<td>R+8</td>
<td>B</td>
</tr>
<tr>
<td>AP-00000000</td>
<td>R+8</td>
<td>DD</td>
</tr>
</tbody>
</table>

**FUNCTIONS AND SUBROUTINES REFERENCED**

- MTHSOCOS
- MTHSOSIN

Total Space Allocated = 319 Bytes

**COMMAND QUALIFIERS**

- FORTRAN /LIST GCPI,INDATA,MATAB,OUTDATA,RUNG,DNAV,EPHEM,TRUEA,SPRESS,OCCLUT,GPERT,GCPS,VISIBL,GENENV,TREG,GRUUT,RATE,BMAT,CMA
- /CHECK=(NOBOUND,OVERFLOW)
- /DEBUG=(NOSYMBOLS,TRACEBACK)
- /F77 /NOG_FLOATING /14 /OPTIMIZE /WARNINGS /NOM_LINES /NOMACHINE_CODE /CONTINUATIONS=19

**COMPILED STATISTICS**

- Run time: 1.03 seconds
- Elapsed time: 14.26 seconds
- Page Faults: 313
- Dynamic Memory: 160 pages
2.3.4 Transformation from Quaternians (AMAT)

Subroutine AMAT completes the direction cosine matrix $A(3,3)$ from the quaternion array $E(4)$. The resulting $A$ matrix may be used to transform data from the ECI to body coordinates.

Processing Requirements

The standard transformation equations from quaternion elements to $A$ matrix elements are used. Refer to Motensen, RE "Strapdown Guidance Error Analysis" IEEE Transactions on Aerospace and Electronic Systems May 1974, pp 451-457

Input Requirements

$E =$ The quaternion array describing the body attitude in inertial space (unitless)

Output Requirements

$A =$ Direction cosine matrix for transformation of data from ECI to body coordinator. (unitless)

Subroutine and Functions Called - None

Calling Subroutines - BVECT, LAMKT, HLMT, HSTAR, EST

Mathematical Specification

The transformation from a quaternion array to a direction cosine array is given by

$$A = \begin{bmatrix} E_1^2 + E_2^2 - E_3^2 - E_4^2 & 2(E_2E_3 + E_1E_4) & 2(E_2E_4 - E_1E_3) \\ 2(E_2E_3 - E_1E_4) & E_1^2 - E_2^2 + E_3^2 - E_4^2 & 2(E_1E_2 + E_3E_4) \\ 2(E_2E_4 + E_1E_3) & 2(E_3E_4 - E_1E_2) & E_1^2 - E_2^2 - E_3^2 + E_4^2 \end{bmatrix}$$

Where the quaternion array is given by

$$E = \begin{bmatrix} E_1 \\ E_2 \\ E_3 \\ E_4 \end{bmatrix}$$
"A" TRANSFORMATION MATRIX

<table>
<thead>
<tr>
<th>COMPUTE E COMPONENTS AS FUNCTIONS OF PRESENT QUATERNION</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMPUTE A MATRIX USING E COMPONENTS, WHERE A IS MATRIX REQUIRED TO TRANSFORM DATA FROM ECI TO SEXTANT COORDINATES</td>
</tr>
</tbody>
</table>
THIS ROUTINE COMPUTES THE A(6,3) MATRIX
THE A MATRIX TRANSFORMS DATA FROM ECI TO SEXTANT COORDINATES

INPUT VARIABLES
F = QUATERNIAN TRANSFORMATION FROM INERTIAL TO BODY

OUTPUT VARIABLES
A = DIRECTION COSINE TRANSFORMATION FROM INERTIAL TO BODY

CODE CHECKED BY JACK MYERS 2 JUNE 1980

INCLUDE 'DEBUG.COM'

COMMON /DEBUG/, TENER, DEBUG
USER CONTROLLED PARAMETERS TO VARY DEBUG PRINT LEVEL

IF 1, PRINTS WHEN ENTERS MOST SUBROUTINES
0-10, HIGHER NUMBER MEANS MORE PRINT

E00 = E(1)*E(1)
E11 = E(2)*E(2)
E22 = E(3)*E(3)
E33 = E(4)*E(4)
E01 = E(1)*E(2)
E02 = E(1)*E(3)
E03 = E(1)*E(4)
E12 = E(2)*E(3)
E13 = E(2)*E(4)
E23 = E(3)*E(4)
A(1,1) = E00=E11=E22=E33
A(1,2) = 2.*(E12+E03)
A(1,3) = 2.*(E13-E02)
A(2,1) = 2.*(E12-E03)
A(2,2) = E00=E11=E22=E33
A(2,3) = 2.*(E01+E23)
A(3,1) = 2.*(E01+E23)
A(3,2) = 2.*(E23=E01)
A(3,3) = E00=E11=E22=E33
RETURN
END
**PROGRAM SECTIONS**

<table>
<thead>
<tr>
<th>Name</th>
<th>Bytes</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 $CODE</td>
<td>216</td>
<td>PIC CON REL LCL SHR EXE RD NOWRT LONG</td>
</tr>
<tr>
<td>2 SLOCAL</td>
<td>120</td>
<td>PIC CON REL LCL NO SHR NO EXE RD WRT QUAD</td>
</tr>
<tr>
<td>3 DEBUG</td>
<td>8</td>
<td>PIC OVR REL GBL SHR NO EXE RD WRT LONG</td>
</tr>
</tbody>
</table>

**ENTRY POINTS**

<table>
<thead>
<tr>
<th>Address</th>
<th>Type</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-00000000</td>
<td>AMAT</td>
<td></td>
</tr>
</tbody>
</table>

**VARIABLES**

<table>
<thead>
<tr>
<th>Address</th>
<th>Type</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-00000000 R+B</td>
<td>E00</td>
<td></td>
</tr>
<tr>
<td>2-00000020 R+B</td>
<td>E01</td>
<td></td>
</tr>
<tr>
<td>2-00000038 R+B</td>
<td>E12</td>
<td></td>
</tr>
<tr>
<td>2-00000048 R+B</td>
<td>E23</td>
<td></td>
</tr>
</tbody>
</table>

**ARRAYS**

<table>
<thead>
<tr>
<th>Address</th>
<th>Type</th>
<th>Name</th>
<th>Bytes</th>
<th>Dimensions</th>
</tr>
</thead>
<tbody>
<tr>
<td>AP-000000000</td>
<td>R+B A</td>
<td></td>
<td>72</td>
<td>(3, 3)</td>
</tr>
<tr>
<td>AP-000000000</td>
<td>R+B E</td>
<td></td>
<td>32</td>
<td>(4)</td>
</tr>
</tbody>
</table>

Total Space Allocated = 344 Bytes

**COMMAND QUALIFIERS**

```
FORTRAN /LIST GCP, INDATA, MATAB, OUTDATA, RUNG, OHAY, EPHEM, TRUEA, SPRESS, OCCULT, GPERT, GCSEQ, VISIBLE, SENENV, TREG, GYROUT, RATE, BMAT, CMA
/SAVE /NOBOUNDS, OVERFLOW
/DEBUG /NOSYMBO.S, TRACEBACK) /F77 /NOG_FLOATING /14 /OPTIMIZE /WARNINGS /MOD_LINES /NOMACHINE_CODE /CONTINUATIONS=19
```

**COMPILETION STATISTICS**

- Run Time: 1.18 seconds
- Elapsed Time: 11.63 seconds
- Page Faults: 310
- Dynamic Memory: 160 pages
2.4 PROCESS GYRO DATA (GYRO)

The general function of the Process Gyro Data Module is to process the angular changes sensed by the strapdown gyros at evenly spaced time intervals determined by one scan line of the science sensor.

This data is corrected for gyro considered parameters; drift, nonorthogonality, scale factor, and misalignment. It is then used to update the quaternion so that the attitude of the spacecraft can be known as accurately as possible at all times. Then, it is used to propagate the attitude state transition matrix. This matrix is used each time an attitude measurement is taken to propagate the covariance matrix.

The major input to the Process Gyro Data Module is the angular changes sensed by the gyros during each sample period. The length of this period defaults to 0.1 second, which is assumed to be the time it takes for one scan line of the MLA.

The input data consists of three variables which correspond to the angular changes sensed by the gyros about the three orthogonal gyro axes during the most recent period.

Other input data required by the Process Gyro Data Module are the initial value of the attitude quaternion and the time at which it is valid (the last attitude reference measurement time). The attitude quaternion is initially set with a default accuracy of 0.125 degrees about each axis.

The attitude quaternion, which is propagated by the Process Gyro Data Module, is later updated by the State Estimation Module when an attitude reference measurement is processed. It is also used by the State Estimation Module for calculating the predicted attitude reference measurement when updating the attitude quaternion. The three components of the attitude quaternion are in the closed interval [-1, +1].

The attitude state transition matrix which is propagated in the Process Gyro Data Module is later used in the State Estimation Module to determine an updated covariance matrix after an attitude reference measurement is processed. At this point, the attitude state transition matrix is reset to the identity.

The Process Gyro Data Module processes the gyro data to eliminate gyro systematic errors (process noise). The systematic errors include the constant drift rate, the scale factor, the non-orthogonality of the gyro input axes, and the misalignment of the ideal gyro input axes with respect to the spacecraft axis system. These compensated angular changes are calculated by the equation:

\[
\begin{bmatrix}
\Delta \theta_1 \\
\Delta \theta_2 \\
\Delta \theta_3
\end{bmatrix} = B \cdot C \cdot 
\begin{bmatrix}
\Delta \theta_{1m} + W_{Dx} \cdot \Delta t \\
\Delta \theta_{1m} + W_{Dy} \cdot \Delta t \\
\Delta \theta_{3m} + W_{Dz} \cdot \Delta t
\end{bmatrix}
\]
where

\[ B = \text{gyro misalignment transformation matrix} \]

\[ C = \text{gyro scale factor and non-orthogonality transition matrix} \]

\[ \Delta \theta_1, \Delta \theta_2, \Delta \theta_3 \quad = \text{compensated gyro data (radians)} \]

\[ \Delta \theta_1 m, \Delta \theta_2 m, \Delta \theta_3 m \quad = \text{input data from gyros (radians)} \]

\[ W_{Dx}, W_{Dy}, W_{Dz} \quad = \text{gyro constant drift rate (rad/sec)} \]

\[ \Delta t = \text{gyro measurement interval (sec)} \]

The matrix \( C \) has the following components:

\[
C = \begin{bmatrix}
  Sx & 0 & 0 \\
  -Sx \cdot \delta_1 & Sy & 0 \\
  -Sx \cdot \delta_2 & -Sy \cdot \delta_3 & Sz
\end{bmatrix}
\]

where

\[ S_x, S_y, S_z \quad = \text{gyro scale factors (radians/count)} \]

\[ \delta_1, \delta_2, \delta_3 \quad = \text{gyro misalignment angles (radians)} \]

The matrix \( B \) has the following components:

\[
B = \begin{bmatrix}
  C \Delta_3 \cdot C \Delta_2 & S \Delta_3 \cdot C \Delta_2 & -S \Delta_2 \\
  S \Delta_1 \cdot S \Delta_2 \cdot S \Delta_3 - C \Delta_1 \cdot S \Delta_3 & S \Delta_1 \cdot S \Delta_2 \cdot S \Delta_3 + C \Delta_1 \cdot C \Delta_3 & C \Delta_2 \cdot S \Delta_1 \\
  C \Delta_1 \cdot S \Delta_2 \cdot C \Delta_3 + S \Delta_1 \cdot S \Delta_3 & C \Delta_1 \cdot S \Delta_2 \cdot S \Delta_3 - S \Delta_1 \cdot C \Delta_3 & C \Delta_2 \cdot C \Delta_1
\end{bmatrix}
\]

where the abbreviations \( C \) and \( S \) are used for cosine and sine, respectively, and

\[ \Delta_3, \Delta_2, \Delta_1 \quad \text{are Euler angles representing rotations about the } z \text{ axis, } y \text{ axis, and } x \text{ axis, in that order.} \]

The compensated gyro data is then used to compute the differential of the quaternion which is used to propagate the quaternion. The quaternion is used to update the attitude transition matrix. The net result of this module being an update in the SS attitude estimate and a new attitude transition matrix.
GYRO PROCESSING MODULE (GYRO)

- Compensate for Gyro drift (KATT)
- Compensate for Gyro nonorthogonality (KATT)
- Compensate for Gyro coordinate misalignment (KATT)
- Update quaternion with angular changes (KATT)
- Compute state partials, attitude parameter partials, and attitude state transition submatrices (PDAIT)
- Load new attitude transition matrix (PATT)

Figure A-18
SUBROUTINE GYRO

INCLUDE 'DEBUG.COM'

COMMON /DEBUG/ IENTER, IDDEBUG

USER CONTROLLED PARAMETERS TO VARY DEBUG PRINT LEVEL

IER ENTER IF 1, PRINTS WHEN ENTERS MOST SUBROUTINES
IDEBUG 0-10, HIGHER NUMBER MEANS MORE PRINT

#include 'ROTAT.COM'

COMMON /ROTAT/ DTHR(3), DTHEM(3), DTHE

REAL 8 DTHR, DTHEM, DTHE

GYRO ATTITUDE PARAMETERS

DTHR REAL WORLD GYRO DATA (RAD)
DTHEM FILTER WORLD GYRO DATA (RAD)
DTHE FILTER WORLD COMPENSATED GYRO DATA (RAD)

#include 'ASTATE.COM'

COMMON /ASTATE/ DE(4), E(4), WD(3), SF(3), D(3), DO(3)

REAL 8 DE, E, WD, SF, D, DO

ATTITUDE STATE AND CONSIDERED PARAMETERS

D DIFFERENTIAL OF QUATERNIONS
E QUATERNIONS
WD GYRO DRIFT RATE (RAD/SEC)
SF GYRO SCALE FACTOR
D GYRO NON-ORTHOGONALITY (RAD)
DO GYRO RELATIVE ORIENTATION (RAD)

#include 'PHI extortion.COM'

COMMON /PHIA/, PA(4,4), TA(4,12), PDA(4,16), PHIA(16,16),

real 8 PA, TA, PDA, PHIA, COVA, QMAX

THESE ARE THE ATTITUDE TRANSITION AND COVARIANCE MATRICES

PA ATTITUDE STATE TRANSITION MATRIX
TA PARAMETER TRANSITION MATRIX
PDA DERIVATIVE OF TRANSITION MATRICES
PHIA AGGREGATE TRANSITION MATRIX
COVA NEW COVARIANCE MATRIX
PDA PREVIOUS COVARIANCE MATRIX
QMAX COVARIANCE NORM MAX

IF (IENTER.EQ.1) WRITE(6,999)

999 FORMAT('ENTERING GYRO')

IF (IENTER.EQ.2) WRITE(6,100)

100 FORMAT('CALLING KATT FOR FILTER WORLD')
**PROGRAM SECTIONS**

<table>
<thead>
<tr>
<th>Name</th>
<th>Bytes</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>$CODE</td>
<td>94</td>
<td>PIC CON REL LCL SHR ERE RD NOWRT LONG</td>
</tr>
<tr>
<td>$PDATA</td>
<td>52</td>
<td>PIC CON REL LCL SHR NOEKE RD NOWRT LONG</td>
</tr>
<tr>
<td>$LOCAL</td>
<td>48</td>
<td>PIC CON REL LCL NOSHR NOEKE RD WRT LONG</td>
</tr>
<tr>
<td>$DEBUG</td>
<td>150</td>
<td>PIC OVR REL GBL SHR NOEKE RD WRT LONG</td>
</tr>
<tr>
<td>$QSTATE</td>
<td>7176</td>
<td>PIC OVR REL GBL SHR NOEKE RD WRT LONG</td>
</tr>
</tbody>
</table>

**ENTRY POINTS**

<table>
<thead>
<tr>
<th>Address</th>
<th>Type</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-00000000</td>
<td></td>
<td>GYRO</td>
</tr>
</tbody>
</table>

**VARIABLES**

<table>
<thead>
<tr>
<th>Address</th>
<th>Type</th>
<th>Name</th>
<th>Address</th>
<th>Type</th>
<th>Name</th>
<th>Address</th>
<th>Type</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>3-60000004</td>
<td>I*4</td>
<td>IDEBUG</td>
<td>3-60000000</td>
<td>I*4</td>
<td>IENTER</td>
<td>6-00001C00</td>
<td>R+B</td>
<td>QMAX</td>
</tr>
</tbody>
</table>

**ARRAYS**

<table>
<thead>
<tr>
<th>Address</th>
<th>Type</th>
<th>Name</th>
<th>Bytes</th>
<th>Dimensions</th>
</tr>
</thead>
<tbody>
<tr>
<td>6-00000000</td>
<td>R+B</td>
<td>COVA</td>
<td>2048</td>
<td>(16, 16)</td>
</tr>
<tr>
<td>5-00000070</td>
<td>R+B</td>
<td>D</td>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>5-00000088</td>
<td>R+B</td>
<td>DD</td>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>5-00000000</td>
<td>R+B</td>
<td>DE</td>
<td>32</td>
<td>(4)</td>
</tr>
<tr>
<td>4-00000030</td>
<td>R+B</td>
<td>DTHE</td>
<td>21</td>
<td>(3)</td>
</tr>
</tbody>
</table>
GYRO

<table>
<thead>
<tr>
<th>Address</th>
<th>Label</th>
<th>Address</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-00000012</td>
<td>100'</td>
<td>1-00000000</td>
<td>999'</td>
</tr>
</tbody>
</table>

FUNCTIONS AND SUBROUTINES REFERENCED

TOTAL Source Allocated = 7810 Bytes

COMMAND QUALIFIERS

FORTRAN /LIST GCP,INCATA,MATAB.OUTDATA,RUNG,DNAV,EPHEM,TRUEA.SPRESS.OCCULT.GPERT.GCPSEQ.VISIBLE,GENENV,TREG,GYROUT,RATE.BMAT.CMA

/DEBUG=(NOBOUND,OVERFLOW)
/DEBUG=(NOSYMBOLS,TRACEBACK)
/777 /NOG_FLOATING /14 /OPTIMIZE /WARNINGS /NO_GATEINES /NON_MACHINE_CODE /CONTINUATIONS=19

COMPILATION STATISTICS

Run Time: 1.04 seconds
Elapsed Time: 12.51 seconds
Page Faults: 326
Dynamic Memory: 160 pages
2.4.1 **PROPAGATE ATTITUDE (KATT)**

This module finds the differential of the quaternion for the last gyro period and uses it to update the quaternion (i.e., propagate the attitude estimate).

The differential of the quaternion is calculated by multiplying the compensated gyro data in sextant coordinates by the sextant to quaternion transition matrix. This transition matrix is:

\[
\mathbf{EM} = \frac{1}{2} \begin{bmatrix}
-e_1 & -e_2 & -e_3 \\
e_0 & -e_3 & e_2 \\
e_3 & e_0 & -e_1 \\
-e_2 & e_1 & e_0
\end{bmatrix}
\]

where \( e_i \) is the \( i \)th element of the attitude quaternion.

This differential is then added to the quaternion to determine the new attitude.
ATTITUDE PROPAGATION (KATT)

<table>
<thead>
<tr>
<th><strong>COMPUTE COMPENSATION OF GYRO OUTPUT FOR</strong>&lt;br&gt;<strong>RELATIVE ORIENTATION (BMAT)</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>COMPUTE COMPENSATION OF GYRO OUTPUT FOR</strong>&lt;br&gt;<strong>NONORTHOGONALITY (CMAT)</strong></td>
</tr>
</tbody>
</table>
| **COMPUTE TRANSFORMATOR FROM**<br>**INDIVIDUAL GYRO TO S/C**<br>**COORDINATES**
  | **TEMPI** = BMAT * CMAT |
| **COMPENSATE FOR GYRO DRIFT DTH - DRIFT** |
| **TRANSFORM GYRO OUTPUT INTO BODY COORDINATES**
  | DTHUP = TEMP * TI |
| **COMPUTE CHANGE IN QUATERNION** |
| **UPDATE QUATERNION (QMULT)** |
| **UNITIZE QUATERNION (UNIT)** |

Figure A-19
SUBROUTINE KATTIDT..(3,FT,RELOP,SCALEF,NONOR,DIMUP,QUAT)

C=================================================================================================
C THIS SUBROUTINE UPDATES THE REAL AND FILTER WORLD QUATERNIONS ABOUT
C THE BODY AXIS
C=================================================================================================

C-----------------------------------------------------------------------------------------------
C INCLUDE 'DEBUG.COM'
C-----------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------
C COMMON /DEBUG/ INIT,IFDEFB
C-----------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------
C INCLUDE 'NOISE.COM'
C-----------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------
C COMMON /NOISE/ BWD(3),SWD(3),BSF(3),SSF(3),BD(3),SD(3)
C-----------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------
C REAL*8 BWD,SWD,BSF,SSF,BD,SD,BDD,SRM,BRE,SRE
C-----------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------
C REAL WORLD GYRO MEASUREMENT ERRORS
C-----------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------
C COMMON /ARRAYS/ T1(3),T2(3),T3(3),T4(4),T11(3,3),T33(3,3)
C-----------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------
C REAL*8 T1,T2,T3,T4,T11,T33,T44,T66,T17,T5,T6,T1
C-----------------------------------------------------------------------------------------------

IHSE ARE TEMORARY STORAGE ARRAYS FOR USE BY ALL MODULES

C-----------------------------------------------------------------------------------------------
C INCLUDE 'TIME.COM'
C-----------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------
C COMMON /TIME/ TIME,NEXT,TSTOP,TIA,DEL,TIN,DIN,DATED,TZERO
C-----------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------
C REAL*8 TIME,NEXT,TSTOP,TIA,DEL,TIN,DIN,DATED,TMEAS,TRACK,TIS,TZERO,TDM,TZERO,TFR,TFR
C-----------------------------------------------------------------------------------------------

THESE ARE THE TIME REFERENCE FRAMES

C-----------------------------------------------------------------------------------------------
C TIME ATOMIC TIME SINCE INITIALIZED (SEC)
C-----------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------
C TMEAS TIME FOR NEXT POSITION INTEGRATION (SEC)
C-----------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------
C TSTOP RUN TERMINATION TIME (SEC)
C-----------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------
C TIA ATTITUDE INTEGRATION TIME (SEC)
C-----------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------
C D L STEP SIZE (SEC)
C-----------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------
C TIN ATTITUDE INTEGRATION TIME (SEC)
C-----------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------
C DTM STEP SIZE (SEC)
C-----------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------
C DATED DATE OF FLIGHT EPOCH (JD)
C-----------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------
C DATER DATE OF 1950 EPOCH (JD)
C-----------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------
C TZERO START TIME IN SECS. SINCE DATED
C-----------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------
C TSLEW TIME NEEDED TO SLEW AND ACQUIRE (SEC)
C-----------------------------------------------------------------------------------------------
DIMENSION DTH(3), DRIFT(3), RELOR(3), SCALEF(3), NONOR(3), DTHUP(3),
QUAT(4), QTEMP(4), TEMP(3, 3),
REAL*8 DRIFT, DTH, DTHUP, QTEMP, QUAT, RELOR, SCALEF, TEMP, VDOT

IF (IENTER.GT.1) WRITE(6,999)
999 FORMAT(' ENTERING KATT '
C COMPENSATE FOR RELATIVE ORIENTATION AND NONORTHOGONALITY
C CALL BMAT(RELOR, T11)
C CALL CMAT(SCALEF, NONOR, T33)
IF (IENTER.EQ.2)
WRITE(6,50) ((T11(I,J),J=1,3), I=1,3), ((T33(I,J),J=1,3), I=1,3)
50 FORMAT(' ' I5',/,'3(3(I16.9),/)
C COMPUTE TRANSFORMATION FROM GYRO TO S/C COORDINATES
C CALL MATAB(T11, T33, TEMP, 3, 3, 3)
C COMPENSATE FOR DRIFT BY SUBTRACTING ESTIMATED VALUE
C DO 100 I = 1, 3
100 T11(I) = DTH(I) - DRIFT(I) * DTA
C TRANSFORM GYRO RATES TO BODY RATES
C CALL MATAB(TEMP, 3, 3, DTHUP, 3, 3, 1)
C UPDATE QUATERNION
C QTEMP(1) = 1.0 - VDOT(DTHUP, DTHUP, 3)/8.
C DO 200 I = 2, 4
200 QTEMP(I) = .5 * DTHUP(I-1)
C CALL QMULT(QUAT, QTEMP, QUAT)
IF (IENTER.EQ.2) WRITE(6,250) DTHUP, QUAT
250 FORMAT(' ' DTHUP =', '3(1X,16.9),/,' QUAT =', '4(1X,16.9)
C UNITIZE QUATERNION
C CALL UNIT(QUAT, QUAT, 4)
RETURN
END
### PROGRAM SECTIONS

<table>
<thead>
<tr>
<th>Name</th>
<th>Bytes</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>$CODE</td>
<td>475</td>
<td>PIC CON REL LCL SHR EXE RD NOWRT LONG</td>
</tr>
<tr>
<td>$DATA</td>
<td>118</td>
<td>PIC CON REL LCL SHR NOEXE RD NOWRT LONG</td>
</tr>
<tr>
<td>LOCAL</td>
<td>384</td>
<td>PIC CON REL LCL NOWR EXE RD WRT QUAD</td>
</tr>
<tr>
<td>OBUF</td>
<td>8</td>
<td>PIC OVR REL GBL SHR NOEXE RD WRT LONG</td>
</tr>
<tr>
<td>NOISE</td>
<td>216</td>
<td>PIC OVR REL GBL SHR NOEXE RD WRT LONG</td>
</tr>
<tr>
<td>ARRAYS</td>
<td>1096</td>
<td>PIC OVR REL GBL SHR NOEXE RD WRT LONG</td>
</tr>
<tr>
<td>TIME</td>
<td>136</td>
<td>PIC OVR REL GBL SHR NOEXE RD WRT LONG</td>
</tr>
</tbody>
</table>

### ENTRY POINTS

<table>
<thead>
<tr>
<th>Address</th>
<th>Type</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>00000000</td>
<td>KATT</td>
</tr>
</tbody>
</table>

### VARIABLES

<table>
<thead>
<tr>
<th>Address</th>
<th>Type</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>4-00000000</td>
<td>R/E</td>
<td>BRED</td>
</tr>
<tr>
<td>6-00000000</td>
<td>R/E</td>
<td>DTA</td>
</tr>
<tr>
<td>3-00000000</td>
<td>R/S</td>
<td>IDEBUG</td>
</tr>
<tr>
<td>4-00000000</td>
<td>R/S</td>
<td>SWHM</td>
</tr>
<tr>
<td>6-00000000</td>
<td>R/S</td>
<td>TIS</td>
</tr>
<tr>
<td>6-00000000</td>
<td>R/S</td>
<td>PRINT</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Address</th>
<th>Type</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>6-00000000</td>
<td>R/B</td>
<td>DATEO</td>
</tr>
<tr>
<td>6-00000000</td>
<td>R/B</td>
<td>DTOM</td>
</tr>
<tr>
<td>4-00000000</td>
<td>R/B</td>
<td>TIA</td>
</tr>
<tr>
<td>6-00000000</td>
<td>R/B</td>
<td>TISM</td>
</tr>
<tr>
<td>6-00000000</td>
<td>R/B</td>
<td>TRACK</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Address</th>
<th>Type</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>6-00000070</td>
<td>R/E</td>
<td>DATER</td>
</tr>
<tr>
<td>6-00000080</td>
<td>R/E</td>
<td>DTRI</td>
</tr>
<tr>
<td>6-00000090</td>
<td>R/E</td>
<td>TIME</td>
</tr>
<tr>
<td>6-00000010</td>
<td>R/E</td>
<td>TSTOP</td>
</tr>
</tbody>
</table>

### ARRAYS

<table>
<thead>
<tr>
<th>Address</th>
<th>Type</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>4-00000000</td>
<td>R/E</td>
<td>BO</td>
</tr>
<tr>
<td>4-00000000</td>
<td>R/E</td>
<td>BOD</td>
</tr>
<tr>
<td>4-00000000</td>
<td>R/E</td>
<td>BSR</td>
</tr>
<tr>
<td>4-00000000</td>
<td>R/E</td>
<td>BWD</td>
</tr>
<tr>
<td>AP-00000000</td>
<td>R/E</td>
<td>DRIFT</td>
</tr>
<tr>
<td>AP-00000000</td>
<td>R/E</td>
<td>DTH</td>
</tr>
<tr>
<td>AP-00000000</td>
<td>R/E</td>
<td>DHU</td>
</tr>
<tr>
<td>AP-00000000</td>
<td>R/E</td>
<td>HDR</td>
</tr>
<tr>
<td>2-00000000</td>
<td>R/E</td>
<td>OTEMPE</td>
</tr>
<tr>
<td>AP-00000000</td>
<td>R/E</td>
<td>QUIT</td>
</tr>
<tr>
<td>AP-00000000</td>
<td>R/E</td>
<td>RELOR</td>
</tr>
<tr>
<td>AP-00000000</td>
<td>R/E</td>
<td>SCALEF</td>
</tr>
<tr>
<td>4-00000000</td>
<td>R/E</td>
<td>SD</td>
</tr>
<tr>
<td>4-00000000</td>
<td>R/E</td>
<td>SDD</td>
</tr>
<tr>
<td>4-00000000</td>
<td>R/E</td>
<td>SSS</td>
</tr>
<tr>
<td>4-00000000</td>
<td>R/E</td>
<td>SSW</td>
</tr>
<tr>
<td>5-00000000</td>
<td>R/E</td>
<td>T1</td>
</tr>
<tr>
<td>5-00000000</td>
<td>R/E</td>
<td>T2</td>
</tr>
<tr>
<td>5-00000000</td>
<td>R/E</td>
<td>T3</td>
</tr>
<tr>
<td>5-00000000</td>
<td>R/E</td>
<td>T33</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bytes</th>
<th>Dimensions</th>
</tr>
</thead>
<tbody>
<tr>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>72</td>
<td>(3, 3)</td>
</tr>
<tr>
<td>72</td>
<td>(3, 3)</td>
</tr>
</tbody>
</table>
### LABELS

<table>
<thead>
<tr>
<th>Address</th>
<th>Label</th>
<th>Address</th>
<th>Label</th>
<th>Address</th>
<th>Label</th>
<th>Address</th>
<th>Label</th>
<th>Address</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-00000000</td>
<td>50'</td>
<td>1-00000001</td>
<td>00</td>
<td>2-00000001</td>
<td>00</td>
<td>1-00000004</td>
<td>250'</td>
<td>1-00000004</td>
<td>999'</td>
</tr>
</tbody>
</table>

### FUNCTIONS AND SUBROUTINES REFERENCED

- BMAT
- CMAT
- MATAB
- QMUL
- UNIT
- VDCT

#### Total Space Allocated = 2433 Bytes

### COMMAND QUALIFIERS

- FORTRAN / LIST GCP, INDATA, MATAB, OUTDATA, RUNG, DNAV, EPHM, TRUEA, SPRESS, OCCULT, GPERT, GCPEQ, VISIBLE, GENENV, TREG, GYROUT, RATE, BMAT, CMA

### COMPILATION STATISTICS

- Run Time: 1.83 seconds
- Elapsed Time: 19.48 seconds
- Page Faults: 390
- Dynamic Memory: 160 pages
2.4.1.1 Matrix Multiply (MATAB)

Subroutine MATAB is a utility routine that performs a matrix multiply of the form

\[ A(L,M) \times B(M,N) = C(L,N) \]

Algorithm

\[ C(I,J) = \sum_{K=1}^{M} A(I,K) \times B(K,J) \]

Input Variables
- \( A \) = First matrix to be multiplied dimensioned \( L \times M \)
- \( B \) = Second matrix to be multiplied dimensioned \( M \times N \)
- \( L \) = Row dimension of \( A \) and \( C \)
- \( M \) = Column dimension of \( A \) and row dimension of \( B \)
- \( N \) = Column dimension of \( B \) and \( C \)

Output Variables
- \( C \) = Resultant matrix dimensioned \( L \times N \)

Subroutines and functions called - None

Calling Subroutines - Utility
MATLAB VGLR

\[
\begin{align*}
\text{SUM} &= 0 \\
\text{SUM} &= \text{SUM} + A(i,k) \cdot B(k,j) \\
\text{DO UNTIL } K &= M \\
C(i,j) &= \text{SUM} \\
\text{DO UNTIL } J &= N \\
\text{DO UNTIL } J &= L
\end{align*}
\]

Figure A-20
0001 SUBROUTINE MATA8(A,B,C,L,M,N)

0002 INCLUDE 'DEBUG.COM'

0003 COMMON /DEBUG/ IENTER, ID8UG

0004 * C USER CONTROLLED PARAMETERS TO VARY DEBUG PRINT LEVEL

0005 * C INTER IF 1, PRINTS WHEN ENTERS MOST SUBROUTINES

0006 * C ID8UG 0-10, HIGHER NUMBER MEANS MORE PRINT

0007 * C DIMENSION A(L,M), B(M,N), C(L,N)

0008 REAL*8 SUM, A, B, C

0009 DO 20 I = 1, L

0010 DO 10 K = 1, M

0011 C(I,J) = SUM

0012 RETURN

0013 END

PROGRAM SECTIONS

Name Byte Attributes
0 $CODE 216 PIC CON REL LCL SHR EXE RD N3WRT LONG
2 $LOCAL 184 PIC CON REL LCL NOSHR NOEXE RD WRT OAM
3 DEBUG 8 PIC OFR REL GBL SHR NOEXE RD WRT LONG

ENTRY POINTS

Address Type Name
0 00000000 MATA8

VARIABLES

Address Type Name Address Type Name Address Type Name
2 00000008 I=4 I 3 00000004 I=4 DEBUG 3 00000000 I=4 IENTER
2 00000010 I=4 K AP 00000010* I=4 L AP 00000014* I=4 M
2 00000000 R=8 SUM AP 00000018* I=4 N
MATAB

ARAYS

Address Type Name Bytes Dimensions
AP-00000004# R=8 A ++ (*, *)
AP-00000008# R=8 B ++ (*, *)
AP-0000000C# R=8 C ++ (*, *)

LABELS

Address Label Address Label
++ 10 ++ 20

Total Space Allocated = 400 Bytes

COMMAND QUALIFIERS

FORTRAN /LIST GCP,INDATA,MATAB,OUTDATA,RUNG,DNAV,EPHEM,TRUEA,SPRESS,Occult,GPERT,GCPSEQ,VISIBLE,GENENV,TREG,GYROUT,RATE,BMAT,CMA

A-123

## COMPIILATION STATISTICS

Run Time: 0.91 seconds
Elapsed Time: 11.50 seconds
Page Faults: 283
Dynamic Memory: 160 pages
2.4.2 Compute State Partials (PDATT)

This module computes the state partials and the \( \phi \) and \( \theta \) submatrixes of the attitude transition matrix. These modules are loaded in the Load Transition Matrix module to form a new attitude transition matrix.

Two parts of the augmented state transition matrix \( {\phi}_A \) are updated each time the gyro data becomes available. The matrix \( {\phi}_A \) contains 1) a submatrix which represents the partial derivative of the quaternion with respect to the value of the quaternion at the time the last measurement is processed, and 2) a submatrix which represents the partial derivative of the quaternion with respect to gyro model parameters (drift, non-orthogonality, scale factor and misalignment). The mathematical equations are more easily read if the matrix \( {\phi}_A \) is partitioned

\[
{\phi}_A = \begin{bmatrix}
\phi & \theta \\
\cdots & \\
0 & I
\end{bmatrix}
\]

where

\( \phi \) contains partial derivative of attitude quaternion with respect to the value of the quaternion at the last measurement time,

\( \theta \) contains partial derivative of attitude quaternion with respect to gyro model parameters, and

I is an identity submatrix
The submatrix $\phi$ is updated each time the gyro data becomes available with the following equation:

$$\phi(t_i + 1, t_0) = F' \phi(t_i, t_0)$$

where $F$ is the state gradient matrix for attitude and

$$F' = \frac{1}{2}
\begin{bmatrix}
0 & -\Delta \Theta_1 & -\Delta \Theta_2 & -\Delta \Theta_3 \\
\Delta \Theta_1 & 0 & \Delta \Theta_3 & -\Delta \Theta_2 \\
\Delta \Theta_2 & -\Delta \Theta_3 & 0 & \Delta \Theta_1 \\
\Delta \Theta_3 & +\Delta \Theta_2 & -\Delta \Theta_1 & 0
\end{bmatrix}$$

$\phi(t_i + 1, t_0)$ = state transition submatrix between times $t_0$ and $t_i + 1$

$\Theta_i$ = compensated angular changes sensed by the strapdown gyros.

The submatrix $\Theta$ is updated each time the gyro data becomes available using the equation

$$\Theta(t_i + 1, t_0) = F' \Theta(t_i, t_0) + E$$

where $E$ is the parameter gradient matrix for attitude

$$E = \frac{1}{2}
\begin{bmatrix}
-e_1 & -e_2 & -e_3 \\
e_0 & -e_3 & e_2 \\
e_3 & e_0 & -e_1 \\
-e_2 & e_1 & e_0
\end{bmatrix}
\begin{bmatrix}
M_1 \\
M_2 \\
M_3 \\
M_4
\end{bmatrix}$$

and the parts of the partitioned matrix are given by

$$M_1 =
\begin{bmatrix}
\Delta t & 0 & 0 \\
0 & \Delta t & 0 \\
0 & 0 & \Delta t
\end{bmatrix}$$

(drift)
\[ M_2 = \begin{bmatrix} 0 & 0 & 0 \\ -\Delta \theta_{1m} & 0 & 0 \\ 0 & -\Delta \theta_{1m} & -\Delta \theta_{2m} \end{bmatrix} \]  
(non-orthogonality)

\[ M_3 = \begin{bmatrix} \Delta \theta_{1m} & 0 & 0 \\ 0 & \Delta \theta_{2m} & 0 \\ 0 & 0 & \Delta \theta_{3m} \end{bmatrix} \]  
(scale factor)

\[ M_4 = \begin{bmatrix} 0 & -\Delta \theta_{m3} & \Delta \theta_{m2} \\ \Delta \theta_{m3} & 0 & -\Delta \theta_{m1} \\ -\Delta \theta_{m2} & \Delta \theta_{m1} & 0 \end{bmatrix} \]  
(misalignment)

where \( \Delta \theta_{m1}, \Delta \theta_{m2}, \Delta \theta_{m3} \) are most recent gyro data.

The net results of this module are updated \( \phi \) and \( \theta \) submatrices.

2.4.3 LOAD ATTITUDE TRANSITION MATRIX (PATT)

This module uses the attitude state transition submatrix (\( \phi \)) and the attitude parameter transition submatrix (\( \theta \)) which were calculated in the Compute State Partial module as inputs to update the aggregated attitude transition matrix each time gyro data is received.
### COMPUTE ATTITUDE STATE PARTIALS (PDATT)

| COMPUTE F' MATRIX (STATE GRADIENT MATRIX) |
| COMPUTE E (PARAMETER GRADIENT MATRIX) |
| COMPUTE SCALE FACTOR PORTION OF M |
| $E' = E' \ast (I/I/M3/I)$ |
| COMPUTE NONORTHOGONALITY PORTION OF M |
| $E' = E' \ast (I/M2/I/I)$ |
| COMPUTE DRIFT PORTION OF M |
| $E' = E' \ast (M1/I/I/I)$ |
| COMPUTE MISALIGNMENT PORTION OF M |
| $E' = E' \ast (I/I/I/M4)$ |

$$PDA = \begin{bmatrix} \phi(t_1 + 1, t_0) & 0 \\ - & - & - & - & - & - & - & - \\ 0 & 0 & I & I & I & I & I & I \\ \end{bmatrix} = \begin{bmatrix} F' \ast \phi(t_1, t_0) & 0 \\ - & - & - & - & - & - & - & - \\ 0 & 0 & I & I & I & I & I & I \\ \end{bmatrix}$$

$$PDA = \begin{bmatrix} \phi & F' \theta' \\ - & - & - & - & - & - & - & - \\ 0 & T & I & I & I & I & I & I \\ \end{bmatrix}$$

$$PDA = \begin{bmatrix} \phi & F' \theta' + E \\ - & - & - & - & - & - & - & - \\ 0 & I & I & I & I & I & I & I \\ \end{bmatrix} = \begin{bmatrix} \phi & \theta \\ - & - & - & - & - & - & - & - \\ 0 & I & I & I & I & I & I & I \\ \end{bmatrix}$$

**ATTITUDE PARAMETER TRANSITION MATRIX** $TA = PDA$

Figure A-21
SUBROUTINE PDATT

COMPUTES STATE PARTIALS AND FORMS PHI AND THETA SUBMATRIXES

OF THE ATTITUDE TRANSITION MATRIX WHERE

1) PHI CONTAINS PARTIAL DERIVATIVES OF THE ATTITUDE QUATERNION
   WITH RESPECT TO LAST MEASUREMENT TIME, AND

2) THETA CONTAINS PARTIAL DERIVATIVES OF ATTITUDE QUATERNION
   WITH RESPECT TOGYRO MODEL PARAMETERS.

INCLUDE 'DEBUG.COM'
COMMON /DEBUG/ IENTER.IDEBUG

USER CONTROLLED PARAMETERS TO VARY DEBUG PRINT LEVEL

INTER IF 1, PRINTS WHEN ENTERS MOST SUBROUTINES
IDEBUG 0-10, HIGHER NUMBER MEANS MORE PRINT

INCLUDE 'ASTATE.COM'
COMMON /ASTATE/ DE(4),E(4),WD(3),SF(3),D(3),DD(3)

REAL*6 E,E,WD,SF,D,DD

ATTITUDE STATE AND CONSIDERED PARAMETERS

D DIFFERENTIAL OF QUATERNIIONS
E QUATERNIONS
WD GYRO DRIFT RATE (RAD/SEC)
SF GYRO SCALE FACTOR
D GYRO NON-ORTHOGONALITY (RAD)
DD GYRO RELATIVE ORIENTATION (RAD)

INCLUDE 'GFPART.COM'
COMMON/GFPART/ FA(4,4),EA(4,12),FN(6,6)
REAL*8 FA,EA,FN

MEASUREMENT AND STATE PARTIALS

FA ATTITUDE STATE PARTIALS
EA CONSIDERED PARAMETERS PARTIALS
FN STATE PARTIALS

INCLUDE 'PHIA.COM'
COMMON /PHIA/ PA(4,4),TA(4,12),PDA(4,16),PHIA(16,16),
       COVA(16,16),POA(16,16),QMAX
       REAL*8 PA,TA,PDA,PHIA,COVA,POA,QMAX

 THESE ARE THE ATTITUDE TRANSITION AND COVARIANCE MATRICES

PA ATTITUDE STATE TRANSITION MATRIX
TA PARAMETER TRANSITION MATRIX
PDA DERIVATIVE OF TRANSITION MATRICES
PHIA AGGREGATE TRANSITION MATRIX
COVA NEW COVARIANCE MATRIX
POA PREVIOUS COVARIANCE MATRIX
QMAX COVARIANCE NORM MAX

INCLUDE 'ARRAYS.COM'
COMMON /ARRAYS/ T1(3),T2(3),T3(3),T4(10),T11(3,3),T33(3,3)

REAL*8 T1,T2,T3,T4,T11,T33

T1 - T4 SINGLE DIMENSION ARRAYS
T11 - T77 DUAL DIMENSIONED ARRAYS
T11 DUAL ARRAY: OFF DIAGONAL SET TO ZERO

INCLUDE 'TMAT.COM'

COMMON /TMAT/ A(3,3),B(3,3),C(3,3),EM(4,3)

REAL*8 A,B,C,EM

TRANSFORMATION MATRICES

A INERTIAL TO BODY AXES
B GYRO TO BODY AXES
C GYRO NON-ORTHOGONAL TO GYRO AXES
EM BODY TO QUATERNIAN AXES

INCLUDE 'ROTAT.COM'

COMMON /ROTAT/ DTHR(3),DTHEM(3),DTHE(3)

REAL*8 DTHR,DTHEM,DTHE

GYRO ATTITUDE PARAMETERS

DTHR REAL WORLD GYRO DATA (RAD)
DTHEM FILTER WORLD GYRO DATA (RAD)
DTHE FILTER WORLD COMPENSATED GYRO DATA (RAD)

INCLUDE 'TIME.COM'

COMMON /TIME/ TIME,TNEXT,TSTOP,TIA,DEL,TIN,DTN,DTA,DTHR,TZERO,TMEAS,TRACK,TIS,TISN,DTA,TMEAS,TRACK,TIS,TISN,DTA,TMEAS,TRACK

REAL*8 TIME,TNEXT,TSTOP,TIA,DEL,TIN,DTN,DTA,TMEAS,TRACK,TIS,TISN,DTA,TMEAS,TRACK

THESE ARE THE TIME REFERENCE FRAMES

TIME ATOMIC TIME SINCE INITIALIZATION (SEC)
TNEXT TIME FOR NEXT POSITION INTEGRATION (SEC)
TSTOP RUN TERMINATION TIME (SEC)
TIA ATTITUDE INTEGRATION TIME (SEC)
DEL STEP SIZE (SEC)
TIN POSITION INTEGRATION TIME (SEC)
DTN STEP SIZE (SEC)
DTA DATE OF FLIGHT EPOCH (JD)
DTHR DATE OF 1950 EPOCH (JD)
TZERO START TIME IN SECS. SINCE DATED
TSLW TIME NEEDED TO SLEW AND ACQUIRE (SEC)
TIS REAL WORLD REFERENCE TIME (SEC)
TISN TIME FOR NEXT RW POSITION INTEGRATION (SEC)
DTA USUALLY DEL BUT + TSLW - TIA WHEN DEL

TPRINT TIME FOR PRINT (SEC)
* C
02500 DTPRINT INCREMENT ON TPRINT (SEC)
02500 * C
02600 REAL*8 DT1, DT2, DT3, E0, E1, E2, E3, TSLEW
01900 0025 IF (IENTER.GT.1) WRITE(6, 999)
02000 0026 IF (IDEBUG.GT.3) WRITE(6, 998) TSLEW
02100 0027 FORMAT(' ENTERING PDATT ')
02200 0028 FORMAT(4X, 'PDATT, TSLEW' ', F20, 6)
02300
02400 C COMPUTE F' STATE GRADIENT MATRIX
02500 C
02600 0030 DT1 = .5*DTHE(1)
02700 0031 DT2 = .5*DTHE(2)
02800 0032 DT3 = .5*DTHE(3)
02900 0033 FA(1, 2) = -DT1
03000 0034 FA(1, 3) = -DT2
03100 0035 FA(1, 4) = -DT3
03200 0036 FA(2, 1) = DT1
03300 0037 FA(2, 3) = DT2
03400 0038 FA(2, 4) = DT3
03500 0039 FA(3, 1) = DT1
03600 0040 FA(3, 2) = DT2
03700 0041 FA(3, 3) = DT3
03800 0042 FA(4, 1) = DT1
03900 0043 FA(4, 2) = DT2
04000 0044 FA(4, 3) = DT3

C COMPUTE BODY TO QUATERNION AXES TRANSFORMATION
C
C
04400 0045 C
04500 0046 E0 = .5*E(1)
04600 0047 E1 = .5*E(2)
04700 0048 E2 = .5*E(3)
04800 0049 E3 = .5*E(4)
04900 0049 EM(1, 1) = -E1
05000 0050 EM(1, 2) = -E2
05100 0051 EM(1, 3) = -E3
05200 0052 EM(2, 1) = E0
05300 0053 EM(2, 2) = -E3
05400 0054 EM(2, 3) = E2
05500 0055 EM(3, 1) = E3
05600 0056 EM(3, 2) = E0
05700 0057 EM(3, 3) = E1
05800 0058 EM(4, 1) = -E2
05900 0059 EM(4, 2) = E1
06000 0060 EM(4, 3) = E0

C COMPUTE SCALE FACTOR PORTION OF M, M3
C
06100 C
06200 DO 20 I=1, 3  
06300 DO 10 J=1, 3  
06400 10 T33(1, J) = 0.  
06500 CALL MATAB(EM, T11, EA(1, 1, 4, 1, 4, 3), 1 E*[0/0/M3])  
06600 20 T33(1, J) = DTHEM(I)  
06700 20 T33(2, 1) = -DTH M(1)
06800 T33(3, 2) = -DTH M(1)

C COMPUTE NONORTHOGONAL PORTION OF M, M2
C
07100 C
07200 C
CALL MATA8(EM,T33,EA(1,7),4,3,3)  I E*[0/M2/M3]

C COMPUTE DRIFT PORTION OF M

DO 30 I=1,3

30 Ti1(I,1) = - DTA

CALL MATA8(EM,T11,EA,4,3,3)  I E*[M1/M2/M3]

C COMPUTE MISALIGNMENT PORTION OF M

CALL MATA8(EM,T33,EA(1,10),4,3,3)  I E*[M1/M2/M3/M4]

CALL MATA8(FA,PA,PDA(I,5),4,4,12)  I F = THETA

DO 40 J=1,4

DC 40 J=5,16

PDA(I,J) = EA(I,J-4)+PDA(I,J)  I F = THETA + E

CONTINUE

DO 50 I=1,4

DO 50 J=1,4

PA(I,J) = PA(I,J)+PDA(I,J)  I [PHI, THETA]

DO 60 I=1,4

DO 60 J=1,4

TA(I,J) = TA(I,J)+PDA(I,J+4)

RETURN

END
**PROGRAM SECTIONS**

<table>
<thead>
<tr>
<th>Name</th>
<th>Bytes</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCODE</td>
<td>606</td>
<td>PIC CON REL LCL SHR EXE RD NOWRT LONG</td>
</tr>
<tr>
<td>SRCATA</td>
<td>58</td>
<td>PIC CON REL LCL SHR NOEXE RD NOWRT LONG</td>
</tr>
<tr>
<td>LOCAL</td>
<td>248</td>
<td>PIC CON REL LCL NOSHR NOEXE RD WRT QUAD</td>
</tr>
<tr>
<td>DEBUG</td>
<td>4</td>
<td>PIC OVR REL GBL SHR NOEXE RD WRT LONG</td>
</tr>
<tr>
<td>ASTATE</td>
<td>160</td>
<td>PIC OVR REL GBL SHR NOEXE RD WRT LONG</td>
</tr>
<tr>
<td>GFPART</td>
<td>800</td>
<td>PIC OVR REL GBL SHR NOEXE RD WRT LONG</td>
</tr>
<tr>
<td>PHIA</td>
<td>7176</td>
<td>PIC OVR REL GBL SHR NOEXE RD WRT LONG</td>
</tr>
<tr>
<td>ARRAYS</td>
<td>1036</td>
<td>PIC OVR REL GBL SHR NOEXE RD WRT LONG</td>
</tr>
<tr>
<td>TIMAT</td>
<td>312</td>
<td>PIC OVR REL GBL SHR NOEXE RD WRT LONG</td>
</tr>
<tr>
<td>ROTAT</td>
<td>72</td>
<td>PIC OVR REL GBL SHR NOEXE RD WRT LONG</td>
</tr>
<tr>
<td>TIME</td>
<td>136</td>
<td>PIC OVR REL GBL SHR NOEXE RD WRT LONG</td>
</tr>
</tbody>
</table>

**ENTRY POINTS**

<table>
<thead>
<tr>
<th>Address</th>
<th>Type</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-00000000</td>
<td></td>
<td>PDATT</td>
</tr>
</tbody>
</table>

**VARIABLES**

<table>
<thead>
<tr>
<th>Address</th>
<th>Type</th>
<th>Name</th>
<th>Address</th>
<th>Type</th>
<th>Name</th>
<th>Address</th>
<th>Type</th>
<th>Name</th>
<th>Address</th>
<th>Type</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>10-00000000</td>
<td>R=B</td>
<td>DATE0</td>
<td>10-00000070</td>
<td>R=B</td>
<td>DATE0</td>
<td>10-00000020</td>
<td>R=B</td>
<td>DEL</td>
<td>2-00000000</td>
<td>R=B</td>
<td>DT1</td>
</tr>
<tr>
<td>2-00000000</td>
<td>R=B</td>
<td>DT2</td>
<td>2-00000010</td>
<td>R=B</td>
<td>DT3</td>
<td>10-00000068</td>
<td>R=B</td>
<td>GTA</td>
<td>10-00000030</td>
<td>R=B</td>
<td>DTN</td>
</tr>
<tr>
<td>10-00000000</td>
<td>R=B</td>
<td>DPRINT</td>
<td>2-00000008</td>
<td>R=B</td>
<td>E0</td>
<td>2-00000020</td>
<td>R=B</td>
<td>E1</td>
<td>2-00000028</td>
<td>R=B</td>
<td>E2</td>
</tr>
<tr>
<td>2-00000000</td>
<td>R=B</td>
<td>E3</td>
<td>2-00000040</td>
<td>I=4</td>
<td>I</td>
<td>3-00000004</td>
<td>I=4</td>
<td>IDEBUG</td>
<td>3-00000000</td>
<td>I=4</td>
<td>ENTER</td>
</tr>
<tr>
<td>2-00000000</td>
<td>I=4</td>
<td>J</td>
<td>6-00000100</td>
<td>R=B</td>
<td>QMAX</td>
<td>10-00000018</td>
<td>R=B</td>
<td>TIA</td>
<td>10-00000000</td>
<td>R=B</td>
<td>TIME</td>
</tr>
<tr>
<td>10-00000000</td>
<td>R=B</td>
<td>TIN</td>
<td>10-00000058</td>
<td>R=B</td>
<td>TIS</td>
<td>10-00000060</td>
<td>R=B</td>
<td>TIMAS</td>
<td>10-00000000</td>
<td>R=B</td>
<td>TIME</td>
</tr>
<tr>
<td>10-00000000</td>
<td>R=B</td>
<td>TNEXT</td>
<td>10-00000078</td>
<td>R=B</td>
<td>TRPRINT</td>
<td>10-00000050</td>
<td>R=B</td>
<td>TRACK</td>
<td>2-00000038</td>
<td>R=B</td>
<td>TSLEW</td>
</tr>
<tr>
<td>10-00000010</td>
<td>R=B</td>
<td>TSTOP</td>
<td>10-00000040</td>
<td>R=B</td>
<td>TZERO</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**ARRAYS**

<table>
<thead>
<tr>
<th>Address</th>
<th>Type</th>
<th>Name</th>
<th>Bytes</th>
<th>Dimensions</th>
</tr>
</thead>
<tbody>
<tr>
<td>8-00000000</td>
<td>R=B</td>
<td>A</td>
<td>72</td>
<td>(3, 3)</td>
</tr>
<tr>
<td>8-00000048</td>
<td>R=B</td>
<td>B</td>
<td>72</td>
<td>(3, 3)</td>
</tr>
<tr>
<td>8-00000090</td>
<td>R=B</td>
<td>C</td>
<td>72</td>
<td>(3, 3)</td>
</tr>
<tr>
<td>6-00000000</td>
<td>R=B</td>
<td>COVA</td>
<td>2048</td>
<td>(16, 16)</td>
</tr>
<tr>
<td>4-00000000</td>
<td>R=B</td>
<td>D</td>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>4-00000008</td>
<td>R=B</td>
<td>DD</td>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>4-00000000</td>
<td>R=B</td>
<td>DE</td>
<td>32</td>
<td>(4)</td>
</tr>
<tr>
<td>9-00000030</td>
<td>R=B</td>
<td>DTME</td>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>9-00000018</td>
<td>R=B</td>
<td>DTMEH</td>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>9-00000000</td>
<td>R=B</td>
<td>DTHR</td>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>4-00000020</td>
<td>R=B</td>
<td>E</td>
<td>32</td>
<td>(4)</td>
</tr>
<tr>
<td>5-00000000</td>
<td>R=B</td>
<td>EA</td>
<td>384</td>
<td>(4, 12)</td>
</tr>
<tr>
<td>8-00000000</td>
<td>R=B</td>
<td>EM</td>
<td>96</td>
<td>(4, 3)</td>
</tr>
<tr>
<td>5-00000000</td>
<td>R=B</td>
<td>FA</td>
<td>128</td>
<td>(4, 4)</td>
</tr>
<tr>
<td>5-00000020</td>
<td>R=B</td>
<td>FN</td>
<td>288</td>
<td>(6, 6)</td>
</tr>
</tbody>
</table>
```
<table>
<thead>
<tr>
<th>Address</th>
<th>Label</th>
<th>Address</th>
<th>Label</th>
<th>Address</th>
<th>Label</th>
<th>Address</th>
<th>Label</th>
<th>Address</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>6-00000000</td>
<td>R=B PA</td>
<td>128 (4, 4)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6-00000200</td>
<td>R=B POA</td>
<td>512 (4, 16)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6-00000400</td>
<td>R=B PHOA</td>
<td>2048 (16, 16)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6-00000600</td>
<td>R=B POA</td>
<td>2048 (16, 16)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4-00000058</td>
<td>R=B SF</td>
<td>24 (3)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7-00000000</td>
<td>R=B T1</td>
<td>24 (3)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7-00000009</td>
<td>R=B T11</td>
<td>72 (3, 3)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7-00000018</td>
<td>R=B T2</td>
<td>24 (3)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7-00000030</td>
<td>R=B T3</td>
<td>24 (3)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7-00000060</td>
<td>R=B T33</td>
<td>72 (3, 3)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7-00000098</td>
<td>R=B T4</td>
<td>80 (10)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7-00000128</td>
<td>R=B T44</td>
<td>128 (4, 4)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7-00000138</td>
<td>R=B T5</td>
<td>32 (4)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7-00000308</td>
<td>R=B T6</td>
<td>32 (4)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7-00000408</td>
<td>R=B T66</td>
<td>268 (6, 6)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7-00000428</td>
<td>R=B T7</td>
<td>32 (4)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7-000002CB</td>
<td>R=B T77</td>
<td>288 (6, 6)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8-00000080</td>
<td>R=B TA</td>
<td>384 (4, 12)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8-00000040</td>
<td>R=B WD</td>
<td>24 (3)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

FUNCTIONS AND SUBROUTINES REFERENCED
MATAB

Total Space Allocated = 10666 Bytes

COMMAND QUALIFIERS

FORTRAN /LIST GCP,INDATA,MATAB,OUTDATA,RUN,DRN,S,EPHEM,TRUEA,SPRESS,DCS3,DCS3,GPOA,GCPSEQ,VISIBLE,GENENV,TREG,GRYOUT,RATE,MMAT,CMA

/CHECK=NO_SYMBOLS,OVERFLOW
/DEBUG=NO_SYMBOLS,TRACEBACK
/F77 /NOG_FLOATING /14 /OPTIMIZE /WARNINGS /NOO_LINES /MACHINE_CODE /CONTINUATIONS=19

COMPIILATION STATISTICS

Run Time: 3.46 seconds
Elapsed Time: 62.49 seconds
Page Faults: 370
Dynamic Memory: 160 pages
```
2.4.3 LOAD ATTITUDE TRANSITION MATRIX (PATT)

This module uses the attitude state transition submatrix (φ) and the attitude parameter transition submatrix (θ) which were calculated in the Compute State Partial module as inputs to update the aggregated attitude transition matrix each time gyro data is received.

ALGORITHM AND PROCESS

The state transition matrix \( \Phi_A \) is updated each time the attitude quaternion is updated, i.e., each time gyro data becomes available. The matrix \( \Phi_A \) contains 1) a submatrix which represents the partial derivative of the quaternion with respect to the value of the quaternion at the time the last measurement is processed, and 2) a submatrix which represents the partial derivative of the quaternion with respect to gyro model parameters (drift, non-orthogonality, scale factor and misalignment). The \( \Phi_A \) is partitioned as follows:

\[
\Phi_A = \begin{bmatrix}
\phi & \theta \\
\cdots & \cdots \\
0 & I
\end{bmatrix}
\]

where \( \phi \) contains partial derivative of attitude quaternion with respect to the value of the quaternion at the last measurement time

\( \theta \) contains partial derivative of attitude quaternion with respect to gyro model parameters, and

I is an identity submatrix

\( \phi \) and \( \theta \) were calculated in the Computer State Partial module. They, along with I, are loaded together in this module to form the updated state transition matrix, \( \Phi_A \).
LOAD ATTITUDE TRANSITION MATRIX (PATT)

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Initialize total transition matrix to 0.</td>
</tr>
<tr>
<td>2</td>
<td>Load attitude state transition matrix ($\Theta$) into aggregated transition matrix.</td>
</tr>
<tr>
<td>3</td>
<td>Load attitude parameter transition matrix ($\Theta$) into aggregated transition matrix.</td>
</tr>
<tr>
<td>4</td>
<td>Set remaining diagonal elements to 1.</td>
</tr>
</tbody>
</table>

Figure A-22
SUBROUTINE PATT(PHIA, PA, TA)

DIMENSION PHIA(16,16), PA(4,4), TA(4,12)
REAL PA, PHIA, TA

DO 10 I=1,16
  DO 10 J=1,4
    PHIA(I,J) = 0.
  10 CONTINUE

DO 20 I=1,4
  DO 20 J=1,16
    PHIA(I,J) = PA(I,J)
  20 CONTINUE

DO 30 I=1,4
  DO 30 J=5,16
    PHIA(I,J) = TA(I,J-4)
  30 CONTINUE

RETURN
END

0001
0002
0003
0004
0005
0006
0007
0008
0009
0010
0011
0012
0013
0014
0015
0016
0017
0018

PROGRAM SECTIONS

Name Bytes Attributes
--- --- ----------------------
 0 $CODE 175 PIC CON REL LCL SHR EXE RD NOWRT LONG
 2 $LOCAL 68 PIC CON REL LCL NOSHR NOEXE RD WRT LONG
 3 DEBUG 8 PIC OVR REL GBL SHR NOEXE RD WRT LONG

ENTRY POINTS

Address Type Name
--- --- ----
0-00000000 PATT

VARIABLES

Address Type Name Address Type Name Address Type Name Address Type Name
--- --- ---- --- --- ---- --- --- ---- --- --- ---- --- --- ---- --- --- ---- 2-00000004 I=4 I 3-00000004 I=4 IDEBUG 3-00000000 I=4 IENTER 2-00000004 I=4 J
ARRAYS

<table>
<thead>
<tr>
<th>Address</th>
<th>Type</th>
<th>Name</th>
<th>Bytes</th>
<th>Dimensions</th>
</tr>
</thead>
<tbody>
<tr>
<td>AP-0000000000</td>
<td>R=8</td>
<td>PA</td>
<td>128</td>
<td>(4, 4)</td>
</tr>
<tr>
<td>AP-0000000000</td>
<td>R=8</td>
<td>PHIA</td>
<td>2048</td>
<td>(16, 16)</td>
</tr>
<tr>
<td>AP-0000000000</td>
<td>R=8</td>
<td>TA</td>
<td>384</td>
<td>(4, 12)</td>
</tr>
</tbody>
</table>

LABELS

<table>
<thead>
<tr>
<th>Address</th>
<th>Label</th>
<th>Address</th>
<th>Label</th>
<th>Address</th>
<th>Label</th>
<th>Address</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>**</td>
<td>10</td>
<td>**</td>
<td>20</td>
<td>**</td>
<td>30</td>
<td>**</td>
<td>40</td>
</tr>
</tbody>
</table>

Total Space Allocated = 251 Bytes

COMMAND QUALIFIERS

FORTRAN /LIST GCPI,INDATA,MATAB,OUTDATA,RUNG,DN5EPHEM,TRUEA,SPRESS,OC6CULT,GPERJ,GCPSEQ,VISEBLE,GENENV,TRG,GRYOUT,RATE,DMAT,CM

/CHECK=(NOBOUNDS,OVERFLOW)
/DEBUG=(NOSYMBOLS,TRACEBACK)
/F77 /NOG_FLOATING /4 /OPTIMIZE /WARNINGS /NOQ_LINES /NONMACHINE_CODE /CONTINUATIONS=19

COMPILATION STATISTICS

Run Time: 0.92 seconds
Elapsed Time: 11.55 seconds
Page Faults: 307
Dynamic Memory: 160 pages
2.5 INTEGRATE POSITION STATE (INTG)

The Navigation State Time Update module shall numerically integrate the differential equations of motion of the spacecraft and the linear differential equation for the state transition matrix. The fourth order Runge-Kutta-Gill method shall be used to perform the numerical integration.

The differential equations of motion require that the inertial accelerations of the spacecraft be calculated. These accelerations are due to

1. earth gravity
2. harmonics of earth gravity
3. solar gravity, solar pressure and lunar gravity

The ephemeris of the moon and sun must be determined to evaluate the lunar gravitational effects and solar perturbations.

The state transition matrix for the navigation state shall be integrated simultaneously with the equations of motion.

This is a procedural module in that it only orders events and does not perform calculations. The position state is integrated to the next specified time needed to restrict integration errors or the measurement time, whichever comes sooner. The maximum next integration step size is computed after each integration.
FILTER WORLD STATE INTEGRATOR (INTG)

- Compute integration endpoint using a fixed step size (TNEXT)
- Integration interval is the minimum of TNEXT and TMEAS minus current time
- Integrate over interval (RKG)

Figure A-23
SUBROUTINE INTG
0002 INCLUDE 'DEBUG.COM'
0010 COMMON /DEBUG/ CENTER, DEBUG
0020 COMMON /DEBUG/ DEBUG, IODEBUG 0-10, HIGHER NUMBER YAYS MORE PRINT
0040 COMMON /DEBUG/ USER CONTROLLED PARAMETERS TO VARY DEBUG PRINT LEVEL
0050 COMMON /DEBUG/ INCLUDE 'TIME.COM'
0060 COMMON /TIME/ TIME, TNEXT, TSTGP, TIA, DEL, TIN, DTN, DATEO, TZERO
0080 COMMON /TIME/ TMEAS, TRACK, TIS, TSN, DTA, DATE, TPRINT, DPRINT
0100 REAL*8 TIME, TNEXT, TSTGP, TIA, DEL, TIN, DTN, DATEO, TMEAS, TRACK, TIS,
0120 TSN, DTA, TZERO, DATE, TPRINT, DPRINT
0140 COMMON /TIME/ THESE ARE THE TIME REFERENCE FRAMES
0160 COMMON /TIME/ TIME ATOMIC TIME SINCE INITIALIZATION (SEC)
0180 COMMON /TIME/ TNEXT TIME FOR NEXT POSITION INTEGRATION (SEC)
0200 COMMON /TIME/ TSTGP RUN TERMINATION TIME (SEC)
0220 COMMON /TIME/ TIA ATTITUDE INTEGRATION TIME (SEC)
0240 COMMON /TIME/ DEL STEP SIZE (SEC)
0260 COMMON /TIME/ TIN POSITION INTEGRATION TIME (SEC)
0280 COMMON /TIME/ DTN STEP SIZE (SEC)
0300 COMMON /TIME/ DATEO DATE OF FLIGHT EPOCH (JD)
0320 COMMON /TIME/ DATEP DATE OF 1950 EPOCH (JD)
0340 COMMON /TIME/ TZERO START TIME IN SECS. SINCE DATEO
0360 COMMON /TIME/ TSTLEW TIME NEEDED TO SLEW AND ACQUIRE (SEC)
0380 COMMON /TIME/ TIS REAL WORLD REFERENCE TIME (SEC)
0400 COMMON /TIME/ TSN TIME FOR NEXT RW POSITION INTEGRATION (SEC)
0420 COMMON /TIME/ DTA USUALLY + DEL BUT + TSTLEW - TIA WHEN DEL
0440 COMMON /TIME/ TOO LARGE AT MEASUREMENT TIME
0460 COMMON /TIME/ TPRINT TIME FOR PRINT (SEC)
0480 COMMON /TIME/ DTPRINT INCREMENT ON TPRINT (SEC)
0500 COMMON /NSTATE/ THESE ARE THE POSITION STATE AND CONSIDERED PARAMETERS
0520 COMMON /NSTATE/ XD, X, RADM, RADE
0540 COMMON /NSTATE/ REAL*8 XD, X, RADM, RADE
0560 COMMON /NSTATE/ POSITION STATE DERIVATIVES (KM/SEC AND KM/SEC/SEC)
0580 COMMON /NSTATE/ STATE POSITION PARAMETERS (KM AND KM/SEC)
0600 COMMON /NSTATE/ RADM RADIUS OF THE MOON (KM)
0620 COMMON /NSTATE/ RADE EARTH DETECTABLE RADIUS (KM)
0640 COMMON /NSTATE/ DATA INIT/-1.-/"
### Program Sections

<table>
<thead>
<tr>
<th>Name</th>
<th>Bytes</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 $CODE</td>
<td>67</td>
<td>PIC CON REL LCL SHR EXE RD NOWRT LONG</td>
</tr>
<tr>
<td>2 $LOCAL</td>
<td>28</td>
<td>PIC CON REL LCL NOSHR NOEXE RD WRT LONG</td>
</tr>
<tr>
<td>3 DEBUG</td>
<td>8</td>
<td>PIC OVR REL GBL SHR NOEXE RD WRT LONG</td>
</tr>
<tr>
<td>4 TIME</td>
<td>136</td>
<td>PIC OVR REL GBL SHR NOEXE RD WRT LONG</td>
</tr>
<tr>
<td>5 NSTATE</td>
<td>112</td>
<td>PIC OVR REL GBL SHR NOEXE RD WRT LONG</td>
</tr>
</tbody>
</table>

### Entry Points

<table>
<thead>
<tr>
<th>Address</th>
<th>Type</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-00000000</td>
<td>INTG</td>
<td></td>
</tr>
</tbody>
</table>

### Variables

<table>
<thead>
<tr>
<th>Address</th>
<th>Type</th>
<th>Name</th>
<th>Bytes</th>
<th>Dimensions</th>
</tr>
</thead>
<tbody>
<tr>
<td>4-00000030</td>
<td>R8</td>
<td>DATE</td>
<td>48 (6)</td>
<td></td>
</tr>
<tr>
<td>4-00000030</td>
<td>R8</td>
<td>DATER</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4-00000040</td>
<td>R8</td>
<td>DTPRINT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2-00000000</td>
<td>I*4</td>
<td>TIME</td>
<td>48 (6)</td>
<td></td>
</tr>
<tr>
<td>4-00000000</td>
<td>R8</td>
<td>TSTOP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4-00000008</td>
<td>R8</td>
<td>TMEAS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4-00000010</td>
<td>R8</td>
<td>TSTOP</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Arrays

<table>
<thead>
<tr>
<th>Address</th>
<th>Type</th>
<th>Name</th>
<th>Bytes</th>
<th>Dimensions</th>
</tr>
</thead>
<tbody>
<tr>
<td>5-00000030</td>
<td>R8</td>
<td>X</td>
<td>48 (6)</td>
<td></td>
</tr>
<tr>
<td>5-00000000</td>
<td>R8</td>
<td>XD</td>
<td>48 (6)</td>
<td></td>
</tr>
</tbody>
</table>

### Functions and Subroutines Referenced

- RKG
- TREG

Total Space Allocated = 351 Bytes

### Command Qualifiers

- FORTRAN /LIST GCP,INDATA,MATAB,OUTDATA,RUNG,DNAV,EPHEM,TRUEA,SPRESS,OCULT,GPERT,GCPSEQ,VISIBLE,GENENV,T:EG,GYROUT,RATE,EMAT,CMR
- /CHECK=(NOSOUNDS,OVERFLOW)
- /DEBUG=(NOSYMBOLS,TRACEBACK)
- /F77 /NOS_FLOATING /14 /OPTIMIZE /WARNINGS /NOD_LINES /NOMACHINE_CODE /CONTINUATIONS=19
COMPILED STATISTICS

Run Time: 0.74 seconds
Elapsed Time: 7.34 seconds
Page Faults: 329
Dynamic Memory: 160 pages
2.5.1 INTEGRATE POSITION PARAMETERS MODULE (RKG)

The Runge Kutta Gill (RKG) numerical integration method is used to integrate the position state and the position transition matrix simultaneously. It is a fourth order technique that adequately accounts for the nonlinear equations of motion for orbital flight and the higher order terms in the transition matrix.

The purpose of the GCFSIM navigation software is to incorporate filtered measurement data as artificial corrections to the state position vector. This requirement disqualifies multi-step techniques during the processing of measurement data since artificial corrections would violate the continuity of the inherent curve fits coefficients of these techniques.

A second consideration for measurement processing is provisions for variable time steps. Frequent navigation measurements will be needed during convergence followed by infrequent measurements needed to maintain a specified error tolerance. Changing the integration step size is simple and direct with single step methods, but difficult with multi-step methods. Variations in step size for multi-step methods is accomplished by interpolations and extrapolations for the Variable-Step/Variable-Order Methods and by intervening with a single step method to define the initial values at the new time increment for standard multi-step methods.

A study showed that the Runge Kutta Gill (RKG) 4th order numerical integration method is optimal for this application. It is self-starting, handles variable step sizes, and sufficiently accurate.

The Runge Kutta Gill (RKG) method for numerically integrating differential equations is described here.

The change in the value of the function during the computing interval is calculated by

\[ \Delta y = \frac{1}{6} \left( k_1 + 2(1 - k)k_2 + 2(1 + k)k_3 + k_4 \right) \]

where

- \( k_1 = h \cdot f(t_n, y_n) \)
- \( k_2 = h \cdot f(t_n + \frac{1}{2}h, y_n + \frac{1}{2}k_1) \)
- \( k_3 = h \cdot f(t_n + \frac{1}{2}h, y_n + (-\frac{1}{2} + k)k_1 + (1 - k)k_2) \)
- \( k_4 = h \cdot f(t_n + h, y_n - kK_2 + (1 + k)k_3) \)
- \( h \) = computing interval (seconds)
- \( t_n \) = time of beginning of computing interval (seconds)
- \( y_n \) = value of function at beginning of computing interval

The derivative function \( f \) shall be evaluated four times to calculate the change in the function being integrated during the computing interval.
INTEGRATE POSITION DYNAMICS MODULE (RKG)

Initial Entry

NULL Q Matrix | NULL

Compute Position Dynamics (5.2.1)

Compute Position Transition Matrix Derivatives (5.2.2)

\( K = 1 \) or \( 3 \)

\( T = T + DT/2 \) | NULL

\( TP = AA(K) \times (ZD(I) - BB(K) \times Q(I)) \)

\( W = Z(I) \)

\( Z(I) = Z(I) + TP \times DT \)

\( TP = (Z(I) - W)/DT \)

\( Q(I) = Q(I) + 3 \times TP - CC(K) \times ZD(I) \)

Where \( AA = 0.5, 1 - \sqrt{2}/2, 1 + \sqrt{2}/2, 0.5/3 \) \( BB = 2, 1, 1, 2 \) \( CC = 0.5, 1 - \sqrt{2}/2, 1 + \sqrt{2}/2, 0.5 \)

Do until \( I = 6 \)

Do until \( K = 4 \)

Figure A-24
SUBROUTINE RKG(INIT,DT)
INCLUDE 'DEBUG.COM'
COMMON /DEBUG/ IENTER, IDDEBUG

USER CONTROLLED PARAMETERS TO VARY DEBUG PRINT LEVEL

INTER IF 1, PRINTS WHEN ENTERS MOST SUBROUTINES
IDDEBUG 0-10, HIGHER NUMBER MEANS MORE PRINT

INCLUDE 'MSTATE.COM'
COMMON /MSTATE/ XD(6), X(6), RADM, RAE
REAL*8 XD, X, RADM, RAE

POSITION STATE AND CONSIDERED PARAMETERS

XD STATE DERIVATIVES (KM/SEC AND KM/SEC/SEC)
X STATE POSITION PARAMETERS (KM AND KM/SEC)
RADM RADIUS OF THE MOON (KM)
RAE EARTH DETECTABLE RADIUS (KM)

INCLUDE 'PHIN.COM'
COMMON /PHIN/ PN(6,6), PDN(6,6), PHIN(5,6), COVN(6,6),
PON(6,6)
REAL*8 PN, PDN, PHIN, COVN, PON

这些是过渡和协方差数组

PN POSITION STATE TRANSITION MATRIX
PDN DERIVATIVE OF TRANSITION MATRIX
PHIN AGGREGATE TRANSITION MATRIX
COVN NEW COVARIANCE MATRIX
PON PREVIOUS COVARIANCE MATRIX

INCLUDE 'TIME.COM'
COMMON /TIME/ TIME, TNEXT, TSTOP, TIA, DEL, TIN, DTN, DATEO, TZERO
TIME, TRACK, TIS, TISN, TIA, DATEO, TZERO, TPRINT, DTPRINT

这些是时间参考系

TIME ATOMIC TIME SINCE INITIALIZATION (SEC)
TNEXT TIME FOR NEXT POSITION INTEGRATION (SEC)
TSTOP RUN TERMINATION TIME (SEC)
TIA ATTITUDE INTEGRATION TIME (SEC)
DEL " " STEP SIZE (SEC)
TIN POSITION INTEGRATION TIME (SEC)
DTN " " STEP SIZE (SEC)
DATEO " " DATE OF FLIGHT EPOCH (JD)
DATE " " DATE OF 1950 EPOCH (JD)
TZERO START TIME IN SECS. SINCE DATEO
TISN " " TIME TO SLEW AND ACQUIRE (SEC)
TIS REAL WORLD REFERENCE TIME (SEC)
TISN " " TIME FOR NEXT RW POSITION INTEGRATION (SEC)
COMMON /UPDT/,QA,QA(Q),QDOT(Q)
STATE STIMATION PARAMETERS

STATE NAV. DYN. NOISE COVARIANCE DIAGONAL
STATE NAV. DYN. COVARIANCE COEFFICIENTS
STATE CONTRIBUTION TO NAV. COV. FOR DYN. NOISE

COMMON /UPDT/ ON(6),OA(O,OA),QOO(6,6)
COMMON /PAAM/ RTS(N),RTS(N),A(4),B(4),C(4),ANDOF1(6,6),RNDOF3(6,6)

DATA AA /0.5 0.292893.1.707107,.1668687/
DATA BB /2.1.,.2./
DATA CC /5.292893.1.707107,.5/

IF(INIT.EQ.0) GO TO 20
DOB T=INIT.

DO 10 I=1,6
10 RNDOF1(I) = 0.
DO 10 J=1,6
RNDOF2(I,J) = 0.
10 DO 20 J=1,6
20 RNDOF3(I,J) = 0.
20 INIT = 0

DO 30 K = 1,4
30 CALL DNAV(T,N,K,XD,K)
DO 40 I=1,6
40 T = AA(I,K)*CC(I,J)*BB(K)*RNDOF1(I,J)

T = X(I)+T+DT
X(I) = X(I)+T+DT

DO 50 I = 1,6
50 CALL PDNAV(IPN,P)

T = X(I)+T+DT
X(I) = X(I)+T+DT

RETURN
END
<table>
<thead>
<tr>
<th>Name</th>
<th>Bytes</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 $CODE</td>
<td>455</td>
<td>PIC CON REL LCL SHR EKE RD NWRT LONG</td>
</tr>
<tr>
<td>2 $LOCAL</td>
<td>760</td>
<td>PIC CON REL LCL NOHDR NQEME RD WRT QUAD</td>
</tr>
<tr>
<td>3 DELEG</td>
<td>8</td>
<td>PIC OVR REL GBL SHR NQEME RD WRT LONG</td>
</tr>
<tr>
<td>4 NSTATE</td>
<td>112</td>
<td>PIC OVR REL GBL SHR NQEME RD WRT LONG</td>
</tr>
<tr>
<td>5 PHIN</td>
<td>1440</td>
<td>PIC OVR REL GBL SHR NQEME RD WRT LONG</td>
</tr>
<tr>
<td>6 TIME</td>
<td>136</td>
<td>PIC OVR REL GBL SHR NQEME RD WRT LONG</td>
</tr>
<tr>
<td>7 UDPT</td>
<td>752</td>
<td>PIC OVR REL GBL SHR NQEME RD WRT LONG</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Address</th>
<th>Type</th>
<th>Name</th>
<th>Address</th>
<th>Type</th>
<th>Name</th>
<th>Address</th>
<th>Type</th>
<th>Name</th>
<th>Address</th>
<th>Type</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-00000000</td>
<td>R&gt;B</td>
<td>DATES</td>
<td>6-00000070</td>
<td>R&gt;B</td>
<td>DATER</td>
<td>6-00000020</td>
<td>R&gt;B</td>
<td>DEL</td>
<td>AP-0000000680</td>
<td>R&gt;B</td>
<td>DT</td>
</tr>
<tr>
<td>6-00000008</td>
<td>R&gt;B</td>
<td>DTA</td>
<td>6-00000030</td>
<td>R&gt;B</td>
<td>DIFN</td>
<td>6-00000080</td>
<td>R&gt;B</td>
<td>DPRINT</td>
<td>2-00000220</td>
<td>I+4</td>
<td>I</td>
</tr>
<tr>
<td>3-00000004</td>
<td>I+4</td>
<td>IDEBUG</td>
<td>3-00000000</td>
<td>I+4</td>
<td>JENTER</td>
<td>AP-000000040</td>
<td>I+4</td>
<td>INIT</td>
<td>2-000002E4</td>
<td>I+4</td>
<td>J</td>
</tr>
<tr>
<td>2-00000260</td>
<td>I+4</td>
<td>K</td>
<td>4-00000008</td>
<td>R&gt;B</td>
<td>RAEI</td>
<td>4-00000060</td>
<td>R&gt;B</td>
<td>RADM</td>
<td>2-000002E0</td>
<td>R&gt;B</td>
<td>T</td>
</tr>
<tr>
<td>6-00000000</td>
<td>R&gt;B</td>
<td>PIA</td>
<td>6-00000000</td>
<td>R&gt;B</td>
<td>PIA</td>
<td>6-00000018</td>
<td>R&gt;B</td>
<td>PAI1</td>
<td>6-00000058</td>
<td>R&gt;B</td>
<td>TIPS</td>
</tr>
<tr>
<td>6-00000000</td>
<td>R&gt;B</td>
<td>TISN</td>
<td>6-00000010</td>
<td>R&gt;B</td>
<td>TMEAS</td>
<td>6-00000000</td>
<td>R&gt;B</td>
<td>R1EXT</td>
<td>6-00000078</td>
<td>R&gt;B</td>
<td>TPRINT</td>
</tr>
<tr>
<td>6-00000050</td>
<td>R&gt;B</td>
<td>TRACK</td>
<td>6-00000010</td>
<td>R&gt;B</td>
<td>TSTOP</td>
<td>6-00000050</td>
<td>R&gt;B</td>
<td>TZERO</td>
<td>2-00000250</td>
<td>R&gt;B</td>
<td>W</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Address</th>
<th>Type</th>
<th>Name</th>
<th>Bytes</th>
<th>Dimensions</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-00000000</td>
<td>R&gt;B</td>
<td>AA</td>
<td>32</td>
<td>(4)</td>
</tr>
<tr>
<td>2-00000020</td>
<td>R&gt;B</td>
<td>BB</td>
<td>32</td>
<td>(4)</td>
</tr>
<tr>
<td>2-00000040</td>
<td>R&gt;B</td>
<td>CC</td>
<td>32</td>
<td>(4)</td>
</tr>
<tr>
<td>5-00000050</td>
<td>R&gt;B</td>
<td>COV</td>
<td>288</td>
<td>(6, 6)</td>
</tr>
<tr>
<td>5-00000090</td>
<td>R&gt;B</td>
<td>PCHN</td>
<td>288</td>
<td>(6, 6)</td>
</tr>
<tr>
<td>5-00000000</td>
<td>R&gt;B</td>
<td>PHIN</td>
<td>288</td>
<td>(6, 6)</td>
</tr>
<tr>
<td>5-00000000</td>
<td>R&gt;B</td>
<td>PM</td>
<td>288</td>
<td>(6, 6)</td>
</tr>
<tr>
<td>5-00000040</td>
<td>R&gt;B</td>
<td>PON</td>
<td>288</td>
<td>(6, 6)</td>
</tr>
<tr>
<td>7-00000000</td>
<td>R&gt;B</td>
<td>Q</td>
<td>288</td>
<td>(6, 6)</td>
</tr>
<tr>
<td>7-00000030</td>
<td>R&gt;B</td>
<td>QA</td>
<td>128</td>
<td>(16)</td>
</tr>
<tr>
<td>7-00000010</td>
<td>R&gt;B</td>
<td>QDTS</td>
<td>288</td>
<td>(6, 6)</td>
</tr>
<tr>
<td>7-00000000</td>
<td>R&gt;B</td>
<td>QN</td>
<td>48</td>
<td>(6)</td>
</tr>
<tr>
<td>2-00000050</td>
<td>R&gt;B</td>
<td>RNOFF</td>
<td>48</td>
<td>(6)</td>
</tr>
<tr>
<td>2-00000090</td>
<td>R&gt;B</td>
<td>RNOFF</td>
<td>288</td>
<td>(6, 6)</td>
</tr>
<tr>
<td>2-00000180</td>
<td>R&gt;B</td>
<td>RNOFF</td>
<td>288</td>
<td>(6, 6)</td>
</tr>
<tr>
<td>4-00000030</td>
<td>R&gt;B</td>
<td>X</td>
<td>48</td>
<td>(6)</td>
</tr>
<tr>
<td>4-00000000</td>
<td>R&gt;B</td>
<td>XD</td>
<td>48</td>
<td>(6)</td>
</tr>
</tbody>
</table>
FUNCTIONS AND SUBROUTINES REFERENCED

DNAV  PDNAV

Total Space Allocated = 3683 Bytes

COMMAND QUALIFIERS

FORTRAN /LIST GCP,INDATA,MATAB,OUTDATA,RUNG,DNAV,EPHE4,TRUEA,SPRESS,OCULGT,GPERT,GCPSEQ,VISIBLE,GENENV,TREG,GYROUT,RATE,SMAT,CMA

/NOBORDS,OVERFLOW
/DEBUG,INOSYMBOLS,TRACEBACK
/ICONV /NOG_FLOATING /I8 /OPTIMIZE /WARNINGS /MOD_LINES /NOMACHINE_CODE /CONTINUATIONS=19

COMPILATION STATISTICS

Run Time: 2.40 seconds
Elapsed Time: 28.11 seconds
Page faults: 301
Dynamic Memory: 160 pages
2.5.1.1 COMPUTE SOLAR/LUNAR EPHEMERIDES (EPHEM AND TRUEA)

The moon and sun positions are needed to compute gravitational perturbations, and star and lunar limb occultations. This module is used for both actual and filter calculations and for a given time in Julian days computes both positions in ECI coordinates to within 1 degree of arc.

The ephemerides of the sun and the moon are based upon the fundamental orbital elements for these bodies given in the Explanatory Supplement to the Nautical Almanac, pp. 98-107. These elements are expressed as second and third degree polynomials in time where the independent variable $T$ is measured in Julian centuries of 36525 days. The epoch for both orbits is January 0.5 E.T. 1900.

Mean Orbital Elements of the Sun and Moon

Mean Orbital Elements of the Moon:

$\lambda$ - The mean longitude of the moon, measured in the ecliptic from the mean equinox of date to the mean ascending node of the lunar orbit, and then along the orbit;

$\lambda' = \Gamma$ - The mean longitude of the lunar perigee, measured in the ecliptic from the mean equinox of date to the mean ascending node of the lunar orbit, and then along the orbit;

$\Omega$ - The longitude of the mean ascending node of the lunar orbit on the ecliptic measured from the mean equinox of date;
\[ e = \text{The constant of eccentricity}; \]

\[ \gamma = \sin i/2, \text{ where } i \text{ is the constant of inclination}; \]

\[ \Pi(\gamma) = \text{The equatorial horizontal parallax at } 60.2665 \text{ equatorial radii.} \]

\[ (\gamma' = 270.026.02.02.02 + 13.36.030.032.059.031T - 4.08T^2 + 0.0068T^3 \]

\[ \eta' = 334.019.046.040 + 11.010.002.02 + 52T - 37.017T^2 - 0.045T^3 \]

\[ \Pi' = 259.010.059.079 - 5.134.008.031T + 2.48T^2 + 0.008T^3 \]

\[ e = 0.054900489 \]

\[ r = 0.044886967 (i = 50145396366) \]

\[ \Pi (\gamma) = 57.02.02.70 (\Pi = 0.0165932496 \text{ radians}) \]

**Mean Orbital Elements of the Sun:**

\[ L = \text{The geometric mean longitude of the sun measured in the ecliptic from the mean equinox of date} \]

\[ \Gamma = \text{The mean longitude of perigee measured in the ecliptic frame from the mean equinox of date} \]

\[ e = \text{The mean eccentricity} \]

\[ c = \text{The mean obliquity of the ecliptic} \]

\[ \Pi_0 = \text{Horizontal parallax} \]

\[ L = 279.041.48.04 + 129.602.768T + 1.089T^2 \]

\[ \Gamma = 28.013.15.00 + 618.9T + 1.63T^2 + 0.0126T^2 \]

\[ e = 0.01675104 - 0.00004180T - 0.00000126T^2 \]

\[ c = 23.027.08.26 - 46.845T^2 - 0.0059T^2 + 0.00181T^3 \]

\[ \Pi_0 = 8.0180 (\Pi_0 = 4.26624E - 5 \text{ radians}) \]

The position vectors of the sun and moon in the ECI system are computed through the following chain of orthogonal transformations:

\[
\begin{bmatrix}
\bar{R}_M \\
\bar{R}_E
\end{bmatrix} =
\begin{bmatrix}
\bar{e} - X \\
\bar{e} - X
\end{bmatrix}
\begin{bmatrix}
\bar{u} - Z \\
\bar{u} - Z
\end{bmatrix}
\begin{bmatrix}
\bar{v} + \bar{w} \\
\bar{v} + \bar{w}
\end{bmatrix}
\begin{bmatrix}
\bar{R}_E \\
\bar{R}_E
\end{bmatrix}
\]
where

\[ \nu(\ell) = \text{The true anomaly of the moon computed from a literal expansion in terms of the sine function of the mean anomaly, } M \text{ and the eccentricity to the seventh degree;} \]

\[ \omega(\ell) = \text{The argument of perigee of the moon's orbit;} \]

and similarly for \( \nu_0 \) and \( \omega_0 \).

These elements are determined by the following process:

\[ M_{\ell} = C - \Gamma' \]

\[ M_0 = L - \Gamma \]

Thence, for each respective orbit,

\[ \nu = M + \left(2e - \frac{1}{4}e^3 + \frac{5}{96}e^5 + \frac{107}{4608}e^7 \right) \sin(M) \]

\[ + \left(\frac{5}{4}e^2 - \frac{11}{24}e^4 + \frac{17}{192}e^6 \right) \sin(2M) \]

\[ + \left(\frac{13}{12}e^3 - \frac{43}{64}e^5 + \frac{95}{512}e^7 \right) \sin(3M) \]

\[ + \left(\frac{103}{96}e^4 - \frac{451}{480}e^6 \right) \sin(4M) \]

Furthermore, the arguments of perigee are determined,

\[ \omega(\ell) = \Gamma' - \Omega \]

\[ \omega_0 = \Gamma' \]

and finally the radius vectors for the sun and the moon are computed,

\[ r = \frac{a(1 - e^2)}{1 + e \cos \nu} \]

where the orbit mean distances, \( a \), are obtained from the respective horizontal parallaxes,

\[ a(\ell) = \frac{R_E}{\pi(\ell)} \]

\[ a_0 = \frac{R_E}{\pi \theta} \]

\[ a(\ell) = 384421.87 \text{ km} \]

\[ a_0 = 149605590. \text{ km} \]
COMPUTE SOLAR/LUNAR Ephemeris (Ephem)

Calculate Lunar Ephemeris

Calculate Solar Ephemeris

Calculate Ellipses Properties

Transform Axes

Figure A-25
0001 SUBROUTINE EPHEM (T, RM, RC)
    C---------------------------------------------- LUNAR SOLAR EPHEMERIS DATA ---------------
    C
    C INPUT PARAMETER
    C T = TIME IN JULIAN DAYS SINCE JD 0.0
    C
    C OUTPUT PARAMETERS
    C RO = RADIUS VECTOR (EARTH CENTER TO SUN) -- ECI COORDINATES (KM)
    C RM = RADIUS VECTOR (EARTH CENTER TO MOON) -- ECI COORDINATES (KM)
    C
    C
    C INCLUDE 'DEBUG.COM'

0003 COMMON /DEBUG/ ENTER, IDEBUG

0006 DATA AM, AO, EM, EO, GM, GO, LM, LO, MM, MO, OM, RMS

0007 DATA AM, AO, EM, EO, GM, GO, LM, LO, MM, MO, OM, RMS

0008 D = 1.0 E-4 + (T-2451502.0)

0009 C MOD EPHEM RIS

0010 LM = 4.7199657165 + D*(1.48329856E-6 + D*-8.06784E-10))

0011 GM = 5.8351515389 + D*(1.9.44368 - D*1.3507103 E-5 + D*4.537656E-9))

0012 OM = 4.5236015148 + D*(2.71747764E-6 + D*8.72665E-10)

0013 C SUN EPHEMERIS

0014 ED = 4.90822946103 + D*(6.2149893664E-3 + D*5.9166662E-7 + D*1.22173E-9))

0015 EP = 0.01615104 + D*(4.1444E-5 + D*9.4E-9)

0016 C ELLIPSES PROPERTIES

0017 MM = LM + GM

0018 RMS = AM + (1,-EM)/1.+EMcos(VM)

0019 MO = LO + GO

0020 C CALL TRUEA(MO, EO, VO)

0021 C CALL TRUEA(MM, EM, VM)

0022 VWM = GM + GM

0023 CVM = COS(VVM)

0024 SVM = SIN(VVM)

0025 CO = COS(MO)

0026 SO = SIN(MO)

0027 CE = COS(EP)

0028 SE = SIN(EP)

0029 C TRANSFORM FROM LOCAL TO ECI COORDINATES

0030 C MOON

0031 RM(1) = RMS*(CVM*CC-CIM+SVM*SO)

0032 TI = -RMS*(CVM*SO+CIM+SVM*CD)
EPHEM

0031  T2 = RMS+SIM+SVW
0032  RM(2) = T1+CE+T2+SE
0033  RM(3) = T2+CE-T1+SE

C **** SUN
0034  VWS = (VO+G0)
0035  CVW = COS(VWS)
0036  SVW = SIN(VWS)
0037  RO(1) = ROS-CVW
0038  RO(2) = -ROS+CE+SVW
0039  RO(3) = ROS+SE+SVW
0040  RETURN
0041  END

PROGRAM SECTIONS

Name                   Bytes      Attributes
0 $CODE                628        PIC CON REL LCL SHR EXE RD NDWRT LONG
2 $LOCAL               304        PIC CON REL LCL NOSHR NOEXE RD WRT QUAD
3 $DEBUG               8          PIC OVR REL GBL SHR NOEXE RD WRT LONG

ENTRY POINTS

Address  Type  Name
0-00000000  EPHEM

VARIABLES

Address  Type  Name       Address  Type  Name       Address  Type  Name       Address  Type  Name
2-00000000 R+B  AM       2-00000008 R+B  AO       2-00000010 R+B  CE
2-00000020 R+B  CD       2-00000028 R+B  CVW      2-00000030 R+B  D
2-00000040 R+B  ED       2-00000048 R+B  EP       2-00000050 R+B  GM
2-00000004 I+4  IDEBUG    3-00000000 I+4  IENTER  2-00000060 R+B  LM
2-00000070 R+B  MM       2-00000078 R+B  MO       2-00000080 R+B  GM
2-00000090 R+B  ROS      2-00000098 R+B  SE       2-00000090 R+B  SIM
2-00000080 R+B  SVW      AP-00000000 R+B  T
2-000000C8 R+B  VM

ARRAYS

Address  Type  Name       Bytes  Dimensions
AP-00000000 R+B  RM       24   (3)
AP-000000C0 R+B  RD       24   (3)
EPHEM

FUNCTIONS AND SUBROUTINES REFERENCED
MTHSDCOS  MTHSINDSIN  TRUEA

Total Space Allocated = 1140 Bytes

COMMAND QUALIFIERS
FORTRAN /LIST GCP,INDATA,MATAB,OUTDATA,RUNG,NAV,EPHEM,TRUEA,SPRESS,OCULT,GPRT,GPSEQ,VISIBLE,GENENV,TREG,GYROUT,RATE,BMAT,CMA

/CHECK=NOBOUNDS,OVERFLOW)
/DEBUG=NOSYMBOls,TRACEBACK)
/F77 /NOG FLOATING /14 /OPTIMIZE /WARNINGS /NOD_LINES /NOMACHINE_CODE /CONTINUATIONS=19

COMPILATION STATISTICS
Run Time:  2.07 seconds
Elapsed Time:  16.69 seconds
Page Faults:  363
Dynamic Memory:  160 pages
SUBROUTINE TRUEA(M,E,V)
C******************************************************************************
C THIS ROUTINE COMPUTES THE TRUE ANOMALY
C******************************************************************************
REAL*8 E,E2,E3,E4,E5,E6,E7,M,S2M,S3M,S4M,S5M,V
0002
0003 E2=E*E
0004 E3=E*E2
0005 E4=E*E3
0006 E5=E*E4
0007 E6=E*E5
0008 E7=E*E6
0009 S5M=SIN(M)
0010 S2M=SIN(2.*M)
0011 S3M=SIN(3.*M)
0012 S4M=SIN(4.*M)
0013 V=M+S5M*(2.0E-25+E3*0.05298333*E5+0.023220486*E7)+S2M*(1.25+E2
1-0.456333333*E4+0.088541667*E6)+S3M*(1.083333333*E3-0.671875*E5+
2.085546875*E7)+S4M*(1.072916667*E4-0.939583333*E6)
0014 RETURN
0015 END

PROGRAM SECTIONS

<table>
<thead>
<tr>
<th>Name</th>
<th>Bytes</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 $CODE</td>
<td>311</td>
<td>PIC CON REL LCL SHR EXE NOWRT LONG</td>
</tr>
<tr>
<td>2 $LOCAL</td>
<td>80</td>
<td>PIC CON REL LCL NOSHR NOEAE RD WRT QUAD</td>
</tr>
</tbody>
</table>

ENTRY POINTS

<table>
<thead>
<tr>
<th>Address Type Name</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0-00000000</td>
<td>TRUEA</td>
</tr>
</tbody>
</table>

VARIABLES

<table>
<thead>
<tr>
<th>Address Type Name</th>
<th>Address Type Name</th>
<th>Address Type Name</th>
<th>Address Type Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>AP-000000008&amp;R=E</td>
<td>2-00000000 R=B E2</td>
<td>2-00000008 R=B E3</td>
<td>2-00000010 R=B E4</td>
</tr>
<tr>
<td>2-00000018 R=B E5</td>
<td>2-00000020 R=B E6</td>
<td>2-00000028 R=B E7</td>
<td>AP-00000040 R=B M</td>
</tr>
<tr>
<td>2-00000030 R=B S2M</td>
<td>2-00000038 R=B S3M</td>
<td>2-00000040 R=B S4M</td>
<td>2-00000048 R=B S5M</td>
</tr>
<tr>
<td>AP-000000000C&amp;R=B V</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

FUNCTIONS AND SUBROUTINES REFERENCED

MTHSOSIN

Total Space Allocated = 391 Bytes
COMMAND QUALIFIERS

FORTRAN /LIST GCP,INDATA,MATAI,OUTDATA,RUNG,DNAV,EPHEM,TRUEA,SPRESS,OCUCLT,GPERT,GCPSEQ,VISIBLE,GENENV,TREG,GYROUT,RATE,BMAT,CMA

/CHECK=(NOBOUNDS,OVERFLOW)
/DEBUG=(NOSYMBOLO,TRACEBACK)
/F77 /NOG_FLOATING /14 /OPTIMIZE /WARNINGS /NOD_LINES /NOMACHINE_CODE /CONTINUATIONS=19

COMPILATION STATISTICS

Run Time: 0.94 seconds
Elapsed Time: 7.78 seconds
Page Faults: 297
Dynamic Memory: 160 pages
2.5.1.2 COMPUTE SOLAR PRESSURE MODULE (SPRESS)

Solar radiation pressure is an external force acting on the spacecraft and must be included in the equations of motions. This force is calculated with the assumption that the satellite is symmetric, the photon flux is constant, and the sun behaves as a point source of photons. If the sun is occulted by either the moon or the earth, the acceleration is zero.

The acceleration of the spacecraft caused by solar radiation impinging on the exposed vehicle body is equivalent to the change in momentum of the incident radiation, divided by the mass of the spacecraft. The change in momentum is equal to the product of the momentum per photon $P_o$, the photon rate per second per unit area $K$, the exposed area projected to a flat surface in the direction of the radiation $A$, the reflectivity of the surface $K_1$, and the unit vector from the spacecraft to the sun $R_s$. The acceleration is the change in momentum divided by the mass of the spacecraft, $m$.

$$a_r = \frac{P_o K \cdot A \cdot R_s}{m}$$

Assuming that the spacecraft operates at a constant distance from the sun the value of $K$ is constant and may be combined with $P_o$.

$$P = P_o \cdot K$$

For a spherical or near spherical spacecraft the factor $(A/m)$ may be considered to be constant; furthermore a single value for $K_1$ may be used for most vehicles.

The vehicle will not experience radiation pressure from the sun if it is shaded by the earth or moon. No shading will occur if either 1) the occulting body and the sun are on opposite hemispheres from the spacecraft, i.e., the dot products of the unit vectors from the spacecraft to these bodies is negative; or 2) the radius of the occulting body is less than the perpendicular from the spacecraft-sun direction to the center of the body.

The procedure used in this module is shown in Figure A-26. Compute the unit vectors from the spacecraft to the sun, $R_s$, and to the earth, $R_e$, and moon, $R_m$ (see algorithms SE-1 and ST-4). Do not compute radiation pressure if
\[ R_b \cdot R_e < 0 \quad (b = \text{earth or moon}) \]

or if

\[ r_b < |R_b| \sin \beta \]

where

\[ \beta = \cos^{-1} |R_b \cdot R_s| \]

If neither of the above conditions is met the spacecraft is irradiated by the sun and radiation pressure should be computed

\[ \ddot{x} = PK_1 \cdot \frac{A}{mv} \cdot Rs \quad \text{Km/sec}^2 \]

where

\[ P = 4.59 \times 10^{-9} \quad (\text{Newtons/m}) \quad (\text{Km/m}) \]

\[ K_1 = 1.7 \]

\( A/mv = \text{exposed vehicle area in m}^2/\text{vehicle mass in kg} \)
COMPUTE SOLAR RADIATION PRESSURE

COMPUTE SOLAR RADIATION PRESSURE CONSTANT

CHECK FOR EARTH OCCULTATION OF THE SUN

CHECK FOR LUNAR OCCULTATION OF THE SUN

SUN IS OCCULTED

T

F

NULL ACCELERATION VECTOR

COMPUTE ACCELERATION VECTOR ALONG S/C - SUN VECTOR

Figure A-26
SUBROUTINE SPRESS(ACT)

INCLUDE 'DEBUG.COM'

COMMON /DEBUG/ IENTER, IDEBUG

C USER CONTROLLED PARAMETERS TO VARY DEBUG PRINT LEVEL

I ENTER     IF 1, PRINITS WHEN ENTERS MOST SUBROUTINES
IDEBUG     0-10, HIGHER NUMBER MEANS MORE PRINT

INCLUDE 'RVEC.COM'

COMMON /RVEC/ R(3), RM(3), RO(3), RSM(3), RSO(3), SB(3)

REAL R, RM, RO, RSM, RSO, SB, RA, R2, RSMA, RTG

C THESE ARE RADIUS VECTORS IN ECI AND BODY COORDINATES
R    EARTH CENTER TO S/C  -  ECI (KM)
RM   "  MOON  -  ECI (KM)
RO   "  SUN  -  ECI (KM)
RSM  "  SPACECRAFT TO MOON  -  ECI (KM)
RSO  "  SPACECRAFT TO SUN  -  ECI (KM)
R55  EARTH CENTER TO STAR  -  ECI
RA   ABSOLUTE OF VECTOR R (KM)
R2   SQUARE OF RA (KM 2)
R3   CUBE OF RA (KM 3)
RSMA  ABSOLUTE OF RSM (KM)

INCLUDE 'CONST.COM'

COMMON /CONST/ ATM, RBW, RB2, RBE, RBQ, AM2, UM, US, U2, U3, U4, DTU, PKI

REAL ATM, RBW, RB2, RBE, RBQ, AM2, UM, US, U2, U3, U4, DTU, PKI

C PROGRAM CONSTANTS
ATM   'C AREA TO MASS RATIO (METERS/KG)
RBW  'C OBSTRUCTION RADIUS OF THE MOON (KM)
RBE   "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  "  

DIMENSION ACC(3)

REAL AM, FSP, ACC, VMAG

C SOLAR RADIATION PRESSURE COMPUTATIONS

LOGICAL L
FSP = PKI*ATM
CALL OCCULT(R, RSO, RBE, L)
IF (L, MOT. L) CALL OCCULT(RSM, RSO, RBE, L)
DO 10 I = 1, 3
10 ACC(I) = 0.0
PROGRAM SECTIONS

Name | Bytes | Attributes
--- | --- | ---
$CODE | 112 | PIC CON REL LCL SHR EXE RD NOWRT LONG
$DATA | 4 | PIC CON REL LCL SHR NOEXE RD NOWRT长长
LOCAL | 96 | PIC CON REL LCL NOSHR NOEXE LD WRT QUAD
DEBUG | 8 | PIC OVR REL GBL SHR NOEXE LD WRT LONG
$VEC | 224 | PIC OVR REL GBL SHR NOEXE RD WRT LONG
CONST | 112 | PIC OVR REL GBL SHR NOEXE RD WRT LONG

ENTRY POINTS

Address Type | Name
--- | ---
0-00000000 | SPRESS

VARIABLES

Address Type | Name | Address Type | Name | Address Type | Name | Address Type | Name
--- | --- | --- | --- | --- | --- | --- | ---
11-00000000 | R=8 ATM | 5-00000000 | R=8 ATM | 5-00000060 | R=8 DIU | 2-00000000 | R=8 FSP
11-00000014 | I=4 J | 3-00000004 | I=4 IDBUG | 3-00000000 | I=4 ENTER | 5-00000004 | R=8 J2
5-00000050 | R=8 J3 | 5-00000058 | R=8 J4 | 2-00 00010 L=4 L | 5-00000006 | R=8 PK1
5-00000008 | R=8 R2 | 4-00000088 | R=8 R3 | 4-000000A8 | R=8 RA | 5-00000010 | R=8 R8E
5-00000008 | R=8 R8M | 5-00000018 | R=8 R8O | 5-00000020 | R=8 L2 | 5-00000028 | R=8 RM2
5-00000000 | R=8 R5M | 5-000000040 | R=8 VE | 5-00000030 | R=8 U5 | 5-00000038 | R=8 US

ARRAYS

Address Type | Name | Bytes | Dimensions
--- | --- | --- | ---
AP-0000000048 | R=8 ACC | 24 | (3)
4-00000000 | R=8 R | 24 | (3)
4-000000018 | R=8 RM | 24 | (3)
4-000000030 | R=8 RO | 24 | (3)
4-000000048 | R=8 RMS | 24 | (3)
4-000000060 | R=8 R5O | 24 | (3)
4-00000007 | R=8 RSS | 24 | (3)
4-0000000C8 | R=8 RTG | 24 | (3)
FUNCTIONS AND SUBROUTINES REFERENCED

OF: ULT
VMAG

Total Space Allocated = 556 Bytes

COMMAND QUALIFIERS

FORTRAN /LIST GCP,INDATA,MATAB,OUTDATA,SONG,NAV,EPHEM,TRUEA,SPRESS,OCULT,GPERT,GCPSEQ,VISIBLE,GENENV,TREG,GYROUT,RATE,DMAT,CNA

/CHECK=(NOUNBOUND,OVERFLOW)
/DEBUG=(NOSYMBOLS,TRACEBACK)
/F77 /NOF FLOATING /14 /OPTIMIZE /WARNINGS /NOD_LINES /NOMACHINE_CODE /CONTINUATIONS=19

COMPILATION STATISTICS

Run Time: 0.03 seconds
Elapsed Time: 14.83 seconds
Page Faults: 348
Dynamic Memory: 160 pages
2.5.1.3 COMPUTE SOLAR/LUNAR GRAVITATIONAL PERTURBATIONS (GPERT AND FQ)

The gravitational accelerations from the moon and sun must be considered if the equations of motion are to be accurate, especially for high altitude orbits.

This algorithm computes the accelerations due to sun and moon in rectangular coordinates to be added directly to the accelerations caused by other forces. The sun and moon are considered to be point masses, and their positions as a function of time are computed in the module EPHEM.

Classically, the accelerations due to the sun and moon acting on the spacecraft in a geocentric frame of reference are expressed as

\[ \ddot{a} = \mu_s \left[ \frac{\vec{R}_s}{|\vec{R}_s|^3} - \frac{\vec{R}_s}{|\vec{R}_s|^3} \right] \]

\[ \ddot{a}_m = \mu_m \left[ \frac{\vec{R}_m}{|\vec{R}_m|^3} - \frac{\vec{R}_m}{|\vec{R}_m|^3} \right] \]

However, owing to the numerical difficulties that may arise in the differencing of large numbers such as \( R_s \) and \( R_s \), a somewhat different technique is used to compute \( \ddot{a}_s \) and \( \ddot{a}_m \). This technique is described in the following procedure.

**Procedure:**

Compute solar perturbations,

\[ \ddot{a}_s = -\frac{\mu_s}{|\vec{R}_s|^3} \left[ \vec{R} - f(qs) \cdot \vec{R}_s \right] \]

Compute lunar perturbations,

\[ \ddot{a}_m = -\frac{\mu_m}{|\vec{R}_m|^3} \left[ \vec{R} - f(qm) \cdot \vec{R}_m \right] \]

where

\[ f(qs) = -qs \frac{(3 + 3qs + q^2s^2)}{1 + (1 + qs)^3} \]

\[ q^2 = \frac{|\vec{R}|^2}{|\vec{R}_s|^2} - 2 \frac{|\vec{R}|}{|\vec{R}_s|} \cdot \cos \alpha \]

and similarly for \( f(qm) \), \( q_m \), and \( \cos q_m \).
COMPUTE SOLAR/LUNAR PERTURBATION

1. Do while all vector components

   \[ \text{ACC(SUN)} = -\frac{US}{(RS^2)} \times (R - FQS \times RS) \]

2. Do while all vector components

   \[ \text{ACC(MOON)} = \frac{UM}{(RM^2)} \times (R - RQM \times RM) \]

   Where \( FQ = -\frac{Q(3 + 3Q + Q^2)}{(1 + (1 + Q)^2)} \)

   And, \( Q = \frac{R^2}{RS^2} - 2 \text{ABS}(R/RS) \times \text{COS(\text{ALPHA})} \)

   \[ \text{ALPHA} = \frac{R \times \text{DOT} \times RS}{(\text{ABS}(R) \times \text{ABS}(RS))} \]

Figure A-27
SUBROUTINE GPERTE(X, ACC)

DIMENSION X(6), ACC(3)
REAL X, ACC, T1, T2, VMAG

INCLUDE 'DEBUG.COM'

COMMON /DEBUG/, IENTER, IDBUG

USER CONTROLLED PARAMETERS TO VARY DEBUG PRINT LEVEL

! I ENTER IF 1, PRINTS WHEN ENTERS MOST SUBROUTINES
! I DEBUG 0-10, HIGHER NUMBER MEANS MORE PRINT

INCLUDE 'RVEC.COM'

COMMON /RVEC/, R(3), RM(3), RO(3), RSM(3), RSO(3), RSS(3), SB(3)

REAL RM, RO, RSM, RSO, RB(3), RA, R2, R3, RSMA, RTG

REAL B, R, RM, RO, RSM, RSO, SB, RA, R2, R3, RSMA, RTG

00000 0006

INCLUDE 'RVEC.COM'

00000 0007

COMMON /RVEC/, R(3), RM(3), RO(3), RSM(3), RSO(3), RSS(3), SB(3)

00000 0008

REAL B, R, RM, RO, RSM, RSO, SB, RA, R2, R3, RSMA, RTG

00000 0009

INCLUDE 'CONST.COM'

00000 0010

COMMON /CONST/, ATM, RBE, RBM, RB0, RE2, RM2, UM, US, UE, J2, J3, J4, DTU, PK1

00000 0011

REAL B, ATM, RBE, RBM, RB0, RE2, RM2, UM, US, UE, J2, J3, J4, DTU, PK1

00000 0000

PROGRAM CONSTANTS

ATM 5/C AREA TO MASS RATIO (METERS/KG)

RBM OBSTRUCTION RADIUS OF THE MOON (KM)

RBE EARTH (KM)

RRO SUN (KM)

R2 SQUARE OF THE EARTHS RADIUS (KM 2)

RM2 LUNAR RADIUS (KM 2)

UM LUNAR GRAVITATION CONSTANT (KM 3/SEC 2)

US SOLAR " " " "

UE EARTH " " " "

J2, J3, J4 ZONAL GRAVITATIONAL HARMONIC TERMS

DTU REGULARIZED TIME STEP SIZE (SEC)

PK1 SOLAR PRESSURE CONSTANT

00000 0000

GRAVITY PERTURBATIONS FOR HANSS

EQ=15

ACC - PERTURBATION ACCELERATION VECTOR (ECI)

T1 = -US/VMAG(RSO, 3)**3

T2 = -T1*FQ(X, RO)

DO 10 I=1, 3
### FUNCTIONS AND SUBROUTINES REFERENCED

- **FO**
- **VMAG**

### Total Space Allocated

Total Space Allocated = 669 Bytes

### COMMAND QUALIFIERS

- **FORTRAN /LIST GCP,INDATA,MATAB,OUTDATA,RUNG,DNAV,EPHEM,TRUEA,SPRESS,OCCLUD,GPERT,GCPSEQ,VISIBLE,GENENV,TREG,GRYOUT,RAI:EBMAT,CMA**
- **/CHECK=(NOBOUNDS,OVERFLOW)**
- **/DEBUG=(NOSYMBOLS,TRACEBACK)**
- **/F77 /NOG_FLOATING /14 /OPTIMIZE /WARNINGS /NOD_LINES /NOMACHINE_CODE /CONTINUATIONS=19**

### COMPIATION STATISTICS

- **Run Time:** 1.17 seconds
- **Elapsed Time:** 7.89 seconds
- **Page Faults:** 348
- **Dynamic Memory:** 160 pages
FUNCTION FQ(R,RS)
C***************************************************************
C OS FUNCTION FOR MODY-SUN GRAVITY PERTURBATIONS
C***************************************************************
00500 0002
00600 0003
REAL*8 CA, Q, RM, RSM, R, RS, FQ, VMAG, VDOT
00700 0004
RM = VMAG(R,3)
00800 0005
RSM = VMAG(RS,3)
00900 0006
CA = VDOT(R,RS,3)/(RM*RSM)
01000 0007
Q = RM/RSM*(RM/RSM - 2.0*CA)
01100 0008
FQ = -Q*(3.+Q*(3.+Q))/(1.+(1.+Q)**1.5)
01200 0009
RETURN
01300 0010
END

PROGRAM SECTIONS
Name              Bytes  Attributes
0 $CODE           158     PIC CON REL LCL  SHR  EXE   RD NOWRT LONG
1 $DATA           4       PIC CON REL LCL  SHR NOEXE  RD NOWRT LONG
2 $LOCAL          116     PIC CON REL LCL NOSHR NOEXE  RD WRT QUAD

ENTRY POINTS
Address  Type  Name
0-00000000  R*8  FQ

VARIABLES
Address  Type  Name          Address  Type  Name          Address  Type  Name
2-00000008  R*8  CA          2-00000010  R*8  Q          2-00000018  R*8  RM

ARRAYS
Address  Type  Name          Bytes  Dimensions
AP-00000004@  R*8  R          24   (3)
AP-00000008@  R*8  RS         24   (3)

FUNCTIONS AND SUBROUTINES REFERENCED
VDOT     VMAG

Total Space Allocated = 270 Bytes
FORTRAN /LIST GCP, Indata, Matab, Outdata, Rung, DNav, Ephem, Truea, Spres, Occult, Gpert, GCPseq, Visible, Genenv, Treg, Gyrot, Rate, Bmat, Cma

/Check=(NoBounds, OverFlow)
/Debug=(NoSuchSym, Traceback)
/F77 /NoG_Floating /14 /Optimize /Warnings /NoO_Lines /NoMACHINE_CODE /Continuations=19

Compilation Statistics:
Run Time: 0.70 seconds
Elapsed Time: 11.55 seconds
Page Faults: 332
Dynamic Memory: 160 pages
2.5.1.4 COMPUTE NAVIGATION STATE PARTIALS (FDNAV)

The Position State Transition Matrix is a $6 \times 6$ matrix of partial derivatives which relate small changes in the position and velocity state at one time to changes in the position and velocity state at a later time.

The state transition matrix shall be calculated by integrating the linear differential equation

$$\phi_N = F \phi_N$$

with the initial condition $\phi_N(t_0, t_0) = \text{Identity (6 x 6)}$ where the non-zero elements of $F$ are given by

$$F_{u1} = -\frac{\mu}{R^3} \left[ 1 - 3(x_1)^2 \right] - 2 \frac{\mu}{R^3} \left[ 35(x_1 \cdot x_3)^2 - 5(x_1^2 + x_3^2) + 1 \right]$$

$$F_{u2} = \frac{\mu}{R^3} \left[ 3x_1 \cdot x_2 \right] - 2 \frac{\mu}{R^3} \left[ 3J_2(Re)^2 \cdot \frac{5x_1 \cdot x_2}{R^2} \right]$$

$$F_{u3} = \frac{\mu}{R^3} \left[ 3x_1 \cdot x_3 \right] - 2 \frac{\mu}{R^3} \left[ 3J_2(Re)^2 \cdot \frac{5x_1 \cdot x_2}{R^2} \right]$$

$$F_{s1} = F_{u1}$$

$$F_{s2} = -\frac{\mu}{R^3} \left[ 1 - 3(x_2)^2 \right] - 2 \frac{\mu}{R^3} \left[ 35(x_2 \cdot x_3)^2 - 5(x_2^2 + x_3^2) + 1 \right]$$

$$F_{s3} = \frac{\mu}{R^3} \left[ 3x_2 \cdot x_3 \right] - 2 \frac{\mu}{R^3} \left[ 3J_2(Re)^2 \cdot \frac{5x_1 \cdot x_2}{R^2} \right]$$

$$F_{s1} = F_{u3}$$

$$F_{s2} = F_{s3}$$

$$F_{s3} = -\frac{\mu}{R^3} \left[ 1 - 3(x_3)^2 \right] - 2 \frac{\mu}{R^3} \left[ 35(x_3)^2 - 30(x_3)^2 + 3 \right]$$

where

$$R_e = \text{equatorial radius of earth}$$
<table>
<thead>
<tr>
<th>COMPUTE NAVIGATION STATE PARTIALS</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMPUTE STATE PARTIALS - 2 ZONAL HARMONICS</td>
</tr>
<tr>
<td>COMPUTE THE TRANSITION MATRIX DERIVATIVES</td>
</tr>
<tr>
<td>BY MULTIPLYING TRANSITION MATRIX BY STATE PARTIALS MATRIX</td>
</tr>
</tbody>
</table>

Figure A-28
SUBROUTINE PNAV(PN,PDN)

DIMENSION PN(6,6),PDN(6,6)

REAL*8 F1,F2,X2R,X2R,X2R,Y2R,Y2R,Z2R,PN,PDN

INCLUDE 'DEBUG.COM'

COMMON /OESUGI IENTER,IDEBUG

C     C     C     C
C     C     C     C
C     C     C     C
C     C     C     C
C     C     C     C
C     C     C     C
C     C     C     C
C     C     C     C
C     C     C     C
C     C     C     C
C     C     C     C
C     C     C     C
C     C     C     C
C     C     C     C
C     C     C     C
C     C     C     C
C     C     C     C
C     C     C     C
C     C     C     C
C     C     C     C
C     C     C     C
C     C     C     C
C     C     C     C
C     C     C     C
C     C     C     C
C     C     C     C
C     C     C     C
C     C     C     C
C     C     C     C
C     C     C     C
C     C     C     C
C     C     C     C
C     C     C     C
C     C     C     C
C     C     C     C
C

USER CONTROLLED PARAMETERS TO VARY DEBUG PRINT LEVEL

ENTER IF 1. PRINTS WHEN ENTERS MOST SUBROUTINES
IDEBUG 0-10, HIGHER NUMBER MEANS MORE PRINT

INCLUDE 'CONST.COM'

COMMON /CONST/ ATM,RBV,RBV,RE0,RE2,UM,US,UE,DTU,PK1

REAL*8 ATM,RBV,RBV,RE0,RE2,UM,US,UE,DTU,PK1

PROGRAM CONSTANTS

ATM  S/C AREA TO MASS RATIO (METERS/KG)
RBV  OBSTRUCTION RADII OF THE MOON (KM)
RBV  EARTH (KM)
RBV  SUN (KM)
R 2  SQUARE OF THE EARTH'S RADIUS (KM 2)
RM2  LUNAR RADIUS (KM 2)
UM  LUNAR GRAVITATION CONSTANT (KM 3/SEC 2)
US  SOLAR (KM)
U  EARTH (KM)
J2, J3, J4  ZONAL GRAVITATIONAL HARMONIC TERMS
DTU  REGULARIZED TIME STEP SIZE (SEC)
PK1  SOLAR PRESSURE CONSTANT

INCLUDE 'RVEC.COM'

COMMON /RVEC/ R(3),RM(3),RD(3),RSM(3),RSO(3),RSS(3),SB(3)

REAL*8 R, RM, RD, RSM, RSS, SB, RA, R2, R3, RSMA, RTG

THESE ARE RADIUS VECTORS IN ECI AND BODY COORDINATES

R  EARTH CENTER TO S/C - ECI (KM)
RM  EARTH CENTER TO MOON - ECI (KM)
RD  EARTH CENTER TO SUN - ECI (KM)
RSM  SPACECRAFT TO MOON - ECI (KM)
RSD  SPACECRAFT TO SUN - ECI (KM)
RSO  EARTH CENTER TO MOON - ECI (KM)
RSM  EARTH CENTER TO STAR - ECI
RA  ABSOLUTE OF VECTOR R (KM)
R2  SQUARE OF RA (KM 2)
R3  CUBE OF RA (KM 3)
RSMA  ABSOLUTE OF RSM (KM)

INCLUDE 'GFPART.COM'

COMMON/GFPART/ FA(4,4),EA(4,12),FN(6,6)

REAL*8 FA,EA,FN

MEASUREMENT AND STATE PARTIALS

FA  ATTITUDE STATE PARTIALS
EA  CONSIDERED PARAMETERS PARTIALS
FN  STATE PARTIALS
INCLUDE 'NSTATE.COM'

COMMON /NSTATE/ XD(5),X(6),RADM,RADE

REAL*8 XD,X,RADM,RADE

POSITION STATE AND CONSIDERED PARAMETERS

* XD  STATE DERIVATIVES (KM/SEC AND KM/SEC/SEC)
* X   STATE POSITION PARAMETERS (KM AND KM/SEC)
* RAD  RADIUS OF THE MOON (KM)
* RADE EARTH DETECTABLE RADIUS (KM)

INCLUDE 'ARRAYS.COM'

COMMON /ARRAYS/ T1(3),T2(3),T3(3),T4(10),T11(3,3),T33(3,3)

REAL*8 T1,T2,T3,T4,T11,T33,T44,166,177,15,16,17

T1 - T4  SINGLE DIMENSION ARRAYS
T11 - T77 DUAL DIMENSIONED ARRAYS
T1  DUAL ARRAY; OFF DIAGONAL SET TO ZERO

INCLUDE 'UPDT.COM'

COMMON /UPDT/ QN(6),QA(16),Q(6,6),QDOT(6,6)

REAL*8 QN,QA,Q,QDOT

STATE STIMATION PARAMETERS

QN   NAV. DYN. NOISE COVARIANCE DIAGONAL
QA   MIN. VALUES FOR ATT. COVARIANCE DIAGONAL
Q   CONTRIBUTION TO NAV. COV. FOR DYN. NOISE
QDOT DIFFERENTIAL OF Q

F1 = UE/R3
F2 = 1.5+F1*J2+RE2/R2
X2R = X(1)*X(1)/R2
Y2R = X(2)*X(2)/R2
Z2R = X(3)*X(3)/R2
X2R = X(1)*X(1)/R2
Y2R = X(2)*X(2)/R2
Z2R = X(3)*X(3)/R2
FN(4,1) = F1*(3. +X2R-1.) +F2*(5. +Z2R+X2R-7.+X2R+Z2R-1.)
FN(5,2) = F1*(3. +X2R-1.) +F2*(5. +Z2R+X2R-7.+X2R+Z2R-1.)
FN(6,3) = F1*(3. +Z2R-1.) +F2*(5. +6.*Z2R -7.*Z2R+Z2R-3.)
FN(4,2) = 3.F1*X2R+5.*F2*X2R*(1.-7.*Z2R)
FN(4,3) = 3.*F1*X2R+5.*F2*X2R*(3.-7.*Z2R)
FN(5,3) = 3.*F1*X2R+5.*F2*X2R*(3.-7.*Z2R)
FN(5,1) = FN(4,2)
FN(6,1) = FN(4,3)
FN(6,2) = FN(5,3)

CALL MATA8(FN,PN,PON,6,6,6)
COMPUTE THE DIFFERENTIAL FOR CONTRIBUTION TO COVn DUE TO DYNAMIC NOISE

\[
C = F_{n+Q} + Q_{n}(F_n^{TRANSPOSE}) + Q_n
\]

CALL MATAB(FN,Q,QDOT,6,6)

DO 200 I = 1,6

DO 200 J = 1,6

QDOT(I,J) = QDOT(I,J) + QDOT(J,I)

IF (I.EQ.J) GO TO 205

GO TO 200

QDOT(I,J) = CQDOT(I,J) + QN(I)

GO TO 200

QDOT(I,J) = QDOT(I,J)

CONTINUE

IF (IDEBUG.GT.11) WRITE(6,900) F1,F2,((FN(I,J),J=1,6),I=1,6),

((PDQ(I,J),J=1,6),I=1,6),((QDOT(I,J),J=1,6),I=1,6)

FORMAT(4X,'IN PDNAV:',I,1ox,'F0:',EI7.10,,'PON:',EI7.10),

RETURN

END
**ARRAYS**

<table>
<thead>
<tr>
<th>Address</th>
<th>Type</th>
<th>Name</th>
<th>Bytes</th>
<th>Dimensions</th>
</tr>
</thead>
<tbody>
<tr>
<td>5-00000000</td>
<td>R-B</td>
<td>EA</td>
<td>384</td>
<td>(4, 12)</td>
</tr>
<tr>
<td>6-00000000</td>
<td>R-B</td>
<td>FA</td>
<td>120</td>
<td>(4, 4)</td>
</tr>
<tr>
<td>6-00000200</td>
<td>R-B</td>
<td>FN</td>
<td>288</td>
<td>(6, 6)</td>
</tr>
<tr>
<td>AP-00000008</td>
<td>R-B</td>
<td>PDN</td>
<td>288</td>
<td>(6, 6)</td>
</tr>
<tr>
<td>AP-00000040</td>
<td>R-B</td>
<td>PN</td>
<td>288</td>
<td>(6, 6)</td>
</tr>
<tr>
<td>9-00000000</td>
<td>R-B</td>
<td>Q</td>
<td>288</td>
<td>(6, 16)</td>
</tr>
<tr>
<td>9-00000030</td>
<td>R-B</td>
<td>QA</td>
<td>128</td>
<td>(6, 6)</td>
</tr>
<tr>
<td>9-000000100</td>
<td>R-B</td>
<td>QDOT</td>
<td>288</td>
<td>(6, 6)</td>
</tr>
<tr>
<td>9-00000000</td>
<td>R-B</td>
<td>QV</td>
<td>40</td>
<td>(6)</td>
</tr>
<tr>
<td>5-00000000</td>
<td>R-B</td>
<td>R</td>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>5-00000018</td>
<td>R-B</td>
<td>RM</td>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>5-00000030</td>
<td>R-B</td>
<td>RO</td>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>5 J0000008</td>
<td>R-B</td>
<td>RSM</td>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>5-00000000</td>
<td>R-B</td>
<td>RSD</td>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>5-00000070</td>
<td>R-B</td>
<td>RSS</td>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>5-00000006</td>
<td>R-B</td>
<td>RTG</td>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>5-00000000</td>
<td>R-B</td>
<td>SB</td>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>8-00000000</td>
<td>R-B</td>
<td>TI</td>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>8-00000098</td>
<td>R-B</td>
<td>T11</td>
<td>72</td>
<td>(3, 3)</td>
</tr>
<tr>
<td>8-00000018</td>
<td>R-B</td>
<td>T2</td>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>8-00000030</td>
<td>R-B</td>
<td>T3</td>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>8-00000060</td>
<td>R-B</td>
<td>T33</td>
<td>72</td>
<td>(3, 3)</td>
</tr>
<tr>
<td>8-00000048</td>
<td>R-B</td>
<td>T4</td>
<td>80</td>
<td>(10)</td>
</tr>
<tr>
<td>8-00000128</td>
<td>R-B</td>
<td>T44</td>
<td>128</td>
<td>(4, 4)</td>
</tr>
<tr>
<td>8-00000368</td>
<td>R-B</td>
<td>T5</td>
<td>32</td>
<td>(4)</td>
</tr>
<tr>
<td>8-00000048</td>
<td>R-B</td>
<td>T5</td>
<td>32</td>
<td>(4)</td>
</tr>
<tr>
<td>8-00000148</td>
<td>R-B</td>
<td>T6</td>
<td>288</td>
<td>(6, 6)</td>
</tr>
<tr>
<td>8-00000428</td>
<td>R-B</td>
<td>T7</td>
<td>32</td>
<td>(4)</td>
</tr>
<tr>
<td>8-00000028</td>
<td>R-B</td>
<td>T77</td>
<td>288</td>
<td>(6, 6)</td>
</tr>
<tr>
<td>7-00000030</td>
<td>R-B</td>
<td>X</td>
<td>48</td>
<td>(6)</td>
</tr>
<tr>
<td>7-00000000</td>
<td>R-B</td>
<td>XD</td>
<td>48</td>
<td>(6)</td>
</tr>
</tbody>
</table>

**LABELS**

<table>
<thead>
<tr>
<th>Address</th>
<th>Label</th>
<th>Address</th>
<th>Label</th>
<th>Address</th>
<th>Label</th>
<th>Address</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-000001FD</td>
<td>200</td>
<td>0-000001EA</td>
<td>205</td>
<td>0-000001F4</td>
<td>210</td>
<td>1-00000000</td>
<td>900</td>
</tr>
</tbody>
</table>
FUNCTIONS AND SUBROUTINES REFERENCED
MATAB

Total Space Allocated = 4096 Bytes

COMMAND QUALIFIERS
FORTRAN /LIST GCP,INDATA,MATAB,OUTDATA,RUNG,NAV,EPHEM,TRUEA,SPRESS,OCUCLT,GPENT,GCPSEQ,VISIBLE,GENENV,TREG,GYROUT,RATE,DRAT,CMA

/DEBUG=(NOSYMBOLS,TRACEBACK)
/F77 /NOG_FLOATING /14 /OPTIMIZE /WARNINGS /MOD_LINES /NOMACHINE_CODE /CONTINUATIONS=19

COMPILATION STATISTICS
Run Time: 3.16 seconds
Elapsed Time: 44.34 seconds
Page Faults: 387
Dynamic Memory: 160 pages
2.6 Generate Measurement (MEASURE)

MEASURE acts as an executive directing the program flow to the module appropriate for the simulation of the type of measurement to be made. The branch to the specific measurement simulation subroutine is based on an event code that may range from 1 through 15. Those significant to MEASURE are 1 through 5. The mathematics relating to specific simulations will be spoken to under that routine name. The measurement simulation performed under specific event codes are listed below.

<table>
<thead>
<tr>
<th>Event Code</th>
<th>Measurement Simulation Selected</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Landmark Tracker Measurement (GCP) with alternate actions</td>
</tr>
<tr>
<td>2</td>
<td>Global Positioning System (GPS) Measurement</td>
</tr>
<tr>
<td>3</td>
<td>Star Tracker Measurement with ST1</td>
</tr>
<tr>
<td>4</td>
<td>Star Tracker Measurement with ST2</td>
</tr>
<tr>
<td>5</td>
<td>No Measurements</td>
</tr>
</tbody>
</table>

Alternate actions are made available under event code 1 if cloud cover is found to be in excess of 40%. These actions are to make star tracker measurements with priority ST1, ST2 unless ST1 or both fixed trackers are occulted by a major body—earth, sun, or moon.

Should the cloud cover be determined to be less than 40% but greater than 10%, the accuracy of the landmark measurement is degraded from an uncertainty of 3 meters 1σ to 30 meters 1σ.

If the event code is one, indicating a GCP sighting, the cloud cover is determined by table hook-up with linear interpolation driven by a random number from a random number generator.

If the cloud coverage is greater than 40% the need for alternate action is tested by examining the norm of the body attitude quaternion covariance. If this exceeds a specified bound an attitude update will be made using a star tracker if not occulted by a major body.

After determination of cloud coverage, or if the original event code was not 1, the program branches to the appropriate subroutine responsible for generating the measurement.

Figure A-29 illustrates the process of MEASURE.
SUBROUTINE MEASURE

SUBROUTINE MEASURE ACTS AS THE EXECUTIVE FOR ALL SPACECRAFT MEASUREMENTS MADE USING AN EXTERNAL STIMULUS.

INPUT VARIABLES

MCODE = MEASUREMENT CODE = THOSE SIGNIFICANT TO THIS MODULE ARE 1, 2, 3, AND 4
1 = LANDMARK TRACKER MEASUREMENT
2 = GPS MEASUREMENT
3 = STARTRACKER 1 MEASUREMENT
4 = STARTRACKER 2 MEASUREMENT

OUTPUT VARIABLES

THE OUTPUT VARIABLES ARE EXPLICIT IN THE SUBROUTINES CALLED

PROGRAMMED BY JACK MYERS 13 JUNE 1980

REAL * QNORM, PNT, VECT1, VECT2

COMMON BLOCKS

INCLUDE 'TARG TS.COM'

COMMON /TARGETS/ MYTYPE, IS, NS, JFLAG, MCODE, PI, TPI

LOGICAL JFLAG

REAL * PI, TPI

MEASUREMENT SPECIFICATIONS

MTYPE = MEASUREMENT TYPE
UFLAG = SET FOR STAR OBSTRUCTION
"MEASURE" = MEASUREMENT PROCESSING
PI = PI
TPI = 2 * PI

REAL * PI, TPI

INCLUDE 'DEBUG.COM'

COMMON /DEBUG/ IFINTER, IODEBUG

USER CONTROLLED PARAMETERS TO VARY DEBUG PRINT LEVEL

INTER = IF 1, PRINTS WHEN ENTERS MOST SUBROUTINES

IODEBUG = 0-10, HIGHER NUMBER MEANS MORE PRINT

REAL * AL, TBNL, TNL, BL, SKL, TIEO, SIGGCP, LAT, LON,

LANDMARK TRACKER PARAMETERS

AL = ALTITUDE OF LANDMARK (KM)
LON = LONGITUDE OF LANDMARK (DEG)
LAT = LATITUDE OF LANDMARK (DEG)
TBNL = ORIENTATION ARRAY FOR LANDMARK TRACKER
MEASURE

01200  * C  TNL  = MISALIGNMENT ARRAY - ACTUAL
01300  * C  TRACKER TO NOMINAL
01400  * C  BL  = BIAS - ACTUAL (RAD)
01500  * C  SL  = NOISE STANDARD DEVIATION - ACTUAL (RAD)
01600  * C  BKL  = BIAS - KNOWLEDGE (RAD)
01700  * C  TET  = LOOK ANGLE (RAD)
01800  * C  SKL  = NOISE STANDARD DEVIATION - KNOWLEDGE (RAD)
01900  * C  TIEO  = INITIAL EARTH FIXED TO INERTIAL
02000  * C  T10  = TRANSFORMATION
02100  * C  TNLK  = MISALIGNMENT ARRAY KNOWLEDGE
02200  * C  TRACKER TO NOMINAL
02300  * C  SIGGCP  = POSITION UNCERTAINTY DUE TO CLOUDS
02400  * C
02500  * C
02600  * C
02700  * C
02800  0012  * C
02900  0013  * C
03000  0014  * C
03100  * C
03200  * C
03300  * C
03400  * C
03500  * C
03600  * C
03700  * C
03800  * C
03900  * C
04000  * C
04100  * C
04200  * C
04300  * C
04400  * C
04500  * C
04600  * C
04700  * C
04800  * C
04900  * C
05000  * C
05100  * C
05200  * C
05300  * C
05400  * C
05500  * C
05600  * C
05700  * C
05800  * C
05900  * C
06000  * C
06100  * C
06200  * C
06300  * C
06400  * C
06500  * C
06600  * C
06700  * C
06800  * C
06900  * C
07000  * C
07100  * C
07200  * C
07300  * C
07400  * C
07500  * C
07600  * C
07700  * C
07800  * C
07900  * C
08000  * C
08100  * C
08200  * C
08300  * C
08400  * C
08500  * C
08600  * C
08700  * C
08800  * C
08900  * C
09000  * C
09100  * C
09200  * C
09300  * C
09400  * C
09500  * C
09600  * C
09700  * C
09800  * C
09900  * C
10000  * C
10100  * C
10200  * C
10300  * C
10400  * C
10500  * C
10600  * C
10700  * C
10800  * C
10900  * C
11000  * C
11100  * C
11200  * C
11300  * C
11400  * C
11500  * C
11600  * C
11700  * C
11800  * C
11900  * C
12000  * C
12100  * C
12200  * C
12300  * C
12400  * C
12500  * C
12600  * C
12700  * C
12800  * C
12900  * C
13000  * C
13100  * C
13200  * C
13300  * C
13400  * C
13500  * C
13600  * C
13700  * C
13800  * C
13900  * C
14000  * C
14100  * C
14200  * C
14300  * C
14400  * C
14500  * C
14600  * C
14700  * C
14800  * C
14900  * C
15000  * C
15100  * C
15200  * C
15300  * C
15400  * C
15500  * C
15600  * C
15700  * C
15800  * C
15900  * C
16000  * C
16100  * C
16200  * C
16300  * C
16400  * C
16500  * C
16600  * C
16700  * C
16800  * C
16900  * C
17000  * C
17100  * C
17200  * C
17300  * C
17400  * C
17500  * C
17600  * C
17700  * C
17800  * C
17900  * C
18000  * C
18100  * C
18200  * C
18300  * C
18400  * C
18500  * C
18600  * C
18700  * C
18800  * C
18900  * C
19000  * C
19100  * C
19200  * C
19300  * C
19400  * C
19500  * C
19600  * C
19700  * C
19800  * C
19900  * C
20000  * C
20100  * C
20200  * C
20300  * C
20400  * C
20500  * C
20600  * C
20700  * C
20800  * C
20900  * C
21000  * C
21100  * C
21200  * C
21300  * C
21400  * C
21500  * C
21600  * C
21700  * C
21800  * C
21900  * C
22000  * C
22100  * C
22200  * C
22300  * C
22400  * C
22500  * C
22600  * C
22700  * C
22800  * C
22900  0015  * C
23000  0016  * C
23100  0017  * C
23200  0018  * C
23300  * C
23400  * C
23500  * C
23600  * C
23700  * C
23800  * C
23900  * C
24000  * C
24100  * C
24200  * C
24300  * C
24400  * C
24500  * C
24600  * C
24700  * C
24800  * C
24900  * C
25000  * C
25100  * C
25200  * C
25300  * C
25400  * C
25500  * C
25600  * C
25700  * C
25800  * C
25900  * C
26000  * C
26100  * C
26200  * C
26300  * C
26400  * C
26500  * C
26600  * C
26700  * C
26800  * C
26900  * C
27000  * C
27100  * C
27200  * C
27300  * C
27400  * C
27500  * C
27600  * C
27700  * C
27800  * C
27900  * C
28000  * C
28100  * C
28200  * C
28300  * C
28400  * C
28500  * C
28600  * C
28700  * C
28800  * C
28900  * C
29000  * C
29100  * C
29200  * C
29300  * C
29400  * C
29500  * C
29600  * C
29700  * C
29800  * C
29900  * C
30000  0018  * C

MEASURE

00100 0019  COMMON /MODE/ MODE(10)
00200  C  MODE(1) = LANDMARK TRACKER SWEEP MODE
00300  C  0 = RANDOM
00400  C  1 = FIXED AT INPUT THET
00500  C  2 = NO DEFAULT TO STAR TRACKER
00600  C  MODE(2) = CLOUD SELECTION MODE
00700  C  0 = RANDOM CLOUD DENSITIES BASED
00800  C  ON INPUT TABLES CLOTBL
00900  C  1 = FIXED DENSITY AT NO CLOUDS
01000  C  2 = NO CLOUDS WITH 100% CLOUD
01100  C  COVER FOR A SPECIFIED
01200  C  PERIOD (CLOTBL(11,12))
01300  C  MODE(3-10) NOT SPECIFIED AT PRESENT

01400  C
01500
03100  0020  DATA 11,12/54321,123-15/
03200  0021  IF(MCODE .NE. 1) GO TO 1000
03300  C  CALL CALCULATE PERCENT CLOUD COVER
03400  C **************************************************************************
03500  C
03600  0022  CALL RANDU(11,12,X)
03700  0023  CALL CLOTBL(X,PCNT)
03800  0024  IF(PCNT .GT. 40.) GO TO 100
03900  0025  SIGGCP=3.
04000  0026  IF(PCNT .GT. 10.) SIGGCP=30.
04100  C **************************************************************************
04200  C  CALL COMPUTE OFFSET ANGLE
04300  C **************************************************************************
04400  C  CALL RDGCP
04500  0027  GO TO 1000
04600  C **************************************************************************
04700  C  CALL COMPUTE NORM OF ATTITUDE QUATERNIAN COVARIANCE
04800  C  (AN APPROXIMATION TO VARIANCE)
04900  C  ************************************************************************
05000  0029  COMMON /CHNRM/.CHNRM=0.
05100  0030  DO 105 I=1,4
05200  0031  105  QNORM=QNORM+COVA(I,1)
05300  0032  QNORM=SQRT(QNORM)
05400  0033  IF(QNORM .LE. 2.0MAX) GO TO 500
05500  0034  IF(MODE(I) .EQ. 2) GO TO 500
05600  C  ************************************************************************
05700  C  CALL BVECT(VECT1.1)
05800  C  TO ST1
05900  C  ************************************************************************
06000  0035  CALL BVECT(VECT2.1)
06100  0036  CALL BVECT(VECT2.5,L)
06200  0037  IF(.NOT. L) GO TO 300
06300  C  ************************************************************************
06400  C  CALL BVECT(VECT2.2)
06500  C  TO ST2
06600  C  ************************************************************************
06700  0038  CALL BVECT(VECT2.3)
06800  0039  CALL BVECT(VECT2.5,L)
06900  0040  IF(.NOT. L) GO TO 200
07000  0041  500  MCODE=5
07100  0042  GO TO 1000
07200  0043  200  MCODE=4

10-Apr-1981 09:13:05  VAX-11 FORTRAN V2.0-3
GO TO 1000

07400 0045 300 MCODE=3

07500 0046 1000 CONTINUE

07600 C******************************************************************************
07700 C BRANCH TO THE APPROPRIATE SUBROUTINE ON MCODE
07800 C******************************************************************************

07900 0047 GO TO (10,20,30,40),MCODE
08000 0048 RETURN

08100 C******************************************************************************
08200 C MAKE LANDMARK TRACKER MEASUREMENTS
08300 C******************************************************************************

08400 0049 10 CALL LAMKT
08500 0050 RETURN

08600 C******************************************************************************
08700 C MAKE GPS MEASUREMENTS
08800 C******************************************************************************

08900 0051 20 CALL GPS
09000 0052 RETURN

09100 C******************************************************************************
09200 C MAKE MEASUREMENTS WITH STARTRACKER 1
09300 C******************************************************************************

09400 0053 30 CALL START(1)
09500 0054 RETURN

09600 C******************************************************************************
09700 C MAKE MEASUREMENTS WITH STARTRACKER 2
09800 C******************************************************************************

09900 0055 40 CALL START(2)
10000 0056 RETURN
10100 0057 END
PROGRAM SECTIONS

<table>
<thead>
<tr>
<th>Name</th>
<th>Bytes</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 $CODE</td>
<td>248</td>
<td>PIC CON REL LCL, SHR, EXE, RD, NOWRT, LONG</td>
</tr>
<tr>
<td>1 $DATA</td>
<td>12</td>
<td>PIC CON REL LCL, SHR, NOEXE, RD, NOWRT, LONG</td>
</tr>
<tr>
<td>2 $LOCAL</td>
<td>152</td>
<td>PIC CON REL LCL, NO, SHR, NOEXE, RD, WRT, QUAD</td>
</tr>
<tr>
<td>3 $TARGETS</td>
<td>36</td>
<td>PIC OVR REL GBL, SHR, NOEXE, RD, WRT, LONG</td>
</tr>
<tr>
<td>4 $DEBUG</td>
<td>8</td>
<td>PIC OVR REL GBL, SHR, NOEXE, RD, WRT, LONG</td>
</tr>
<tr>
<td>5 $LITPAR</td>
<td>302</td>
<td>PIC OVR REL GBL, SHR, NOEXE, RD, WRT, LONG</td>
</tr>
<tr>
<td>6 $TIME</td>
<td>136</td>
<td>PIC OVR REL GBL, SHR, NOEXE, RD, WRT, LONG</td>
</tr>
<tr>
<td>7 $PHIA</td>
<td>7176</td>
<td>PIC OVR REL GBL, SHR, NOEXE, RD, WRT, LONG</td>
</tr>
<tr>
<td>8 $MODE</td>
<td>40</td>
<td>PIC OVR REL GBL, SHR, NOEXE, RD, WRT, LONG</td>
</tr>
</tbody>
</table>

ENTRY POINTS

<table>
<thead>
<tr>
<th>Address</th>
<th>Type</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-00000000</td>
<td>MEASURE</td>
<td></td>
</tr>
</tbody>
</table>

VARIABLES

<table>
<thead>
<tr>
<th>Address</th>
<th>Type</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>5-00000000</td>
<td>R8</td>
<td>AL</td>
</tr>
<tr>
<td>6-00000000</td>
<td>R8</td>
<td>DATE</td>
</tr>
<tr>
<td>6-00000030</td>
<td>R8</td>
<td>DTM</td>
</tr>
<tr>
<td>7-00000000</td>
<td>R8</td>
<td>PA</td>
</tr>
<tr>
<td>7-00000008</td>
<td>R8</td>
<td>PCA</td>
</tr>
<tr>
<td>8-00000000</td>
<td>R8</td>
<td>SDA</td>
</tr>
<tr>
<td>9-00000000</td>
<td>R8</td>
<td>TNG</td>
</tr>
<tr>
<td>9-00000000</td>
<td>R8</td>
<td>DIF</td>
</tr>
</tbody>
</table>

ARRAYS

<table>
<thead>
<tr>
<th>Address</th>
<th>Type</th>
<th>Name</th>
<th>Bytes</th>
<th>Dimensions</th>
</tr>
</thead>
<tbody>
<tr>
<td>5-00000000</td>
<td>R8</td>
<td>BKL</td>
<td>16</td>
<td>(2)</td>
</tr>
<tr>
<td>5-00000004</td>
<td>R8</td>
<td>BL</td>
<td>16</td>
<td>(2)</td>
</tr>
<tr>
<td>7-00000000</td>
<td>R4</td>
<td>COVA</td>
<td>128</td>
<td>(4, 4)</td>
</tr>
<tr>
<td>7-00000004</td>
<td>R8</td>
<td>PA</td>
<td>512</td>
<td>(4, 16)</td>
</tr>
<tr>
<td>7-00000008</td>
<td>R8</td>
<td>PDA</td>
<td>2048</td>
<td>(16, 16)</td>
</tr>
<tr>
<td>7-0000000c</td>
<td>R8</td>
<td>PDA</td>
<td>2048</td>
<td>(16, 16)</td>
</tr>
<tr>
<td>5-00000000</td>
<td>R8</td>
<td>SKL</td>
<td>16</td>
<td>(2)</td>
</tr>
<tr>
<td>5-00000008</td>
<td>R8</td>
<td>SL</td>
<td>16</td>
<td>(2)</td>
</tr>
<tr>
<td>7-00000000</td>
<td>R8</td>
<td>TA</td>
<td>128</td>
<td>(4, 4)</td>
</tr>
<tr>
<td>5-00000100</td>
<td>R8</td>
<td>TIEO</td>
<td>72</td>
<td>(3, 3)</td>
</tr>
<tr>
<td>5-00000130</td>
<td>R8</td>
<td>TIEO</td>
<td>72</td>
<td>(3, 3)</td>
</tr>
<tr>
<td>5-00000060</td>
<td>R8</td>
<td>TNL</td>
<td>72</td>
<td>(3, 3)</td>
</tr>
</tbody>
</table>
MEASURE

10-Apr-1981 09:08:14  _DBA0:[DI1R.GCP]MEASURE.FOR;

5-000000EB  R=E  TNLK  72 (3, 3)

LABELS

<table>
<thead>
<tr>
<th>Address</th>
<th>Label</th>
<th>Address</th>
<th>Label</th>
<th>Address</th>
<th>Label</th>
<th>Address</th>
<th>Label</th>
<th>Address</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-000000D4</td>
<td>10</td>
<td>0-0000000C</td>
<td>20</td>
<td>0-000000E4</td>
<td>30</td>
<td>0-000000EE</td>
<td>40</td>
<td>0-00000048</td>
<td>100</td>
</tr>
<tr>
<td>0-000000B3</td>
<td>200</td>
<td>0-000000BC</td>
<td>300</td>
<td>0-000000AA</td>
<td>500</td>
<td>0-000000C3</td>
<td>1000</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

FUNCTIONS AND SUBROUTINES REFERENCED

BVECT  CLDTBL  FOR$RANOU  GPS  LAMKT  MTH$DSQRT  RDGCP  START

Visible

Total Space Allocated = 8200 Bytes

COMMAND QUALIFIERS

FORTRAN /L MEASURE

/CHECK=(NOBOUNDS,OVERRIDE)
/DEBUG=(NOSYMBOLE,TRACEBACK)
/F77 /NOG_FLOATING /14 /OPTIMIZE /WARNINGS /NOG_LINES /NOMACHINE_CODE /CONTINUATIONS=19

COMPILATION STATISTICS

Run Time: 1.99 seconds
Elapsed Time: 20.71 seconds
Page Faults: 171
Dynamic Memory: 47 pages
2.6.1 GPS Model (GPS)

Subroutine GPS simulates a Global Positioning System measurement by corrupting the true vehicle position-velocity vector with bias and random noise. Compensation is then made for any biases known to be in the measurement. On the first pass through the subroutine, the measurement noise covariance is computed. All constraints and variables are communicated through common.

Simulation Equations

Measurements

\[
\begin{align*}
\mathbf{MP} &= \mathbf{P} + \mathbf{PB} + \mathbf{N}(0,\text{PS}) - \mathbf{PBK} \\
\mathbf{MV} &= \mathbf{V} + \mathbf{VB} + \mathbf{N}(0,\text{VS}) - \mathbf{VBK}
\end{align*}
\]

where \( \mathbf{N}(\mu, \sigma) \) = normal random variable with mean, \( \mu \), and standard deviation, \( \sigma \).

Covariance

\[
[R_{GPS}] = \begin{bmatrix}
PSK(1)^2 \\
PSK(2)^2 \\
PSK(3)^2 \\
VSK(1)^2 \\
VSK(2)^2 \\
VSK(3)^2
\end{bmatrix}
\]
<table>
<thead>
<tr>
<th>C</th>
<th>UPT POSITION-VELOCITY VECTOR WITH BIAS AND NOISE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>COMPENSATE FOR BIAS KNOWLEDGE</td>
</tr>
<tr>
<td>T</td>
<td>FIRST PASS</td>
</tr>
<tr>
<td>F</td>
<td>GENERATE MEASUREMENT NOISE COVARIANCE</td>
</tr>
<tr>
<td></td>
<td>NULL</td>
</tr>
</tbody>
</table>

Figure A-30
SUBROUTINE GPS

GPS receiver by simply adding bias and noise to the actual vehicle position and velocity state vector.

Once the measurement is made, compensation for the bias to the extent of bias knowledge is subtracted from the measurement position and velocity vector.

INPUT PARAMETERS

P = actual position vector (km)
V = velocity (km/s)
PB = position bias vector (km)
VB = velocity bias (km/s)
PS = position noise standard deviation
VS = velocity (km/s)

bias vector (km/s)
noise standard deviation
bias knowledge
velocity (km/s)
position standard deviation knowledge
velocity (km/s)

OUTPUT PARAMETERS

MP = measurement position vector (km)
MV = velocity (km/s)
RGPS = covariance of GPS measurement noise (km)**2

Programmed by Jack Myers ------ Extension 4443
4 June 1980

COMMON BLOCKS

include 'envir.com'
COMMON /ENVIR/ STATE(10), PROFILE(10,4), INIT

include 'DEBUG.COM'
COMMON /DEBUG/ IENTER, IDEBUG

user controlled parameters to vary debug print level
include 'gpspar.com'
common /gpspar/ pb(3),vb(3),ps(3),vs(3),pbk(3),vbk(3),psk(3),
                  vsk(3)
real*8 pb,vb,ps,vs,pbk,vbk,psk,vsk

**GPS Parameters**

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pb</td>
<td>Position Bias - Actual</td>
</tr>
<tr>
<td>vb</td>
<td>Velocity Bias - Actual</td>
</tr>
<tr>
<td>ps</td>
<td>Position Noise Standard Deviation - Actual</td>
</tr>
<tr>
<td>vs</td>
<td>Velocity Noise Standard Deviation - Actual</td>
</tr>
<tr>
<td>pbk</td>
<td>Position Bias - Knowledge</td>
</tr>
<tr>
<td>vbk</td>
<td>Velocity Bias - Knowledge</td>
</tr>
<tr>
<td>psk</td>
<td>Position Noise Standard Deviation - Knowledge</td>
</tr>
<tr>
<td>vsk</td>
<td>Velocity Noise Standard Deviation - Knowledge</td>
</tr>
</tbody>
</table>

include 'measout.com'
common /measout/ mx(6),rgps(6,6),dchs(2),dvcs(2),ms(3,2),
                      rs(2,2,2),dchcl,dcvl,lmu(3),rl(2,2),emxg(6),
                      edhs(2),edvs(2),edhl,edvl
real*8 mx,rgps,dcvs,ms,rs,dchcl,dcvl,lmu,rl,emxg,
       edhs,edvs,edhl,edvl

**Measurement Output Parameters**

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mx</td>
<td>Position/Velocity State Measurement - GPS</td>
</tr>
<tr>
<td>emg</td>
<td>Estimated Position/Velocity State Measurement - Hgps</td>
</tr>
<tr>
<td>rgps</td>
<td>State Measurement Noise Covariancs (Knowledge) - GPS (Km, Km/sec**2)</td>
</tr>
<tr>
<td>dchs</td>
<td>Star Measurement Horizontal Deviation from Boresight - Start (Rad)</td>
</tr>
<tr>
<td>dvcs</td>
<td>Star Measurement Vertical Deviation from Boresight - Start (Rad)</td>
</tr>
<tr>
<td>edhs</td>
<td>Estimated Star Measurement Horizontal Deviation from Boresight (Rad)</td>
</tr>
<tr>
<td>edvs</td>
<td>Estimated Star Measurement Vertical Deviation from Boresight (Rad)</td>
</tr>
<tr>
<td>ms</td>
<td>Star Measurement Unit Vector (Second Subscript Refers To Tracker) - Start</td>
</tr>
<tr>
<td>rs</td>
<td>Star Measurement Noise Covariance (Knowledge) - Start (Rad**2)</td>
</tr>
<tr>
<td>dhcl</td>
<td>Landmark Measurement Horizontal Deviation from Boresight - Lamkt (Rad)</td>
</tr>
<tr>
<td>dvcl</td>
<td>Landmark Measurement Vertical Deviation from Boresight - Lamkt (Rad)</td>
</tr>
<tr>
<td>edml</td>
<td>Estimated Landmark Measurement Horizontal Deviation from Boresight (Rad)</td>
</tr>
<tr>
<td>edvl</td>
<td>Estimated Landmark Measurement Vertical Deviation from Boresight (Rad)</td>
</tr>
<tr>
<td>lmu</td>
<td>Landmark Measurement Unit Vector - Lamkt</td>
</tr>
<tr>
<td>rl</td>
<td>Landmark Measurement Noise Covariance (Knowledge) - Lamkt (Rad**2)</td>
</tr>
</tbody>
</table>
EQUIVALENCE  
EQUIVALENCE (MX,MP),(MX(4),MV),(P,STATE),(V,STATE(4))

DIMENSION  
DIMENSION MP(3),MV(3),P(3),V(3)

REAL*8 MP,MV,P,Gauss

DATA I FLAG /0/

C*-----------------------------------------------------------------------------------------

ADD BIAS AND NOISE TO THE POSITION VELOCITY VECTOR

THEN SUBTRACT BIAS KNOWLEDGE

C*-----------------------------------------------------------------------------------------

DO 10 I=1,3
MP(I)+=P(I)+GAUSS(0.,PS(I))-PBK(I)

MV(I)+=V(I)+VBK(I)

IF (I FLAG .EQ. 1) RETURN

C*-----------------------------------------------------------------------------------------

SET UP NOISE COVARIANCE - FIRST PASS ONLY

C*-----------------------------------------------------------------------------------------

IFLAG=1

 gọi 15 1=1,6
PC 15 J=1,6

DO 20 I=1,3
RGPS(I,J)=0.

RGPS(I,I)+=PSK(I)+PSK(I)

RGPS(I+3,J)+=VSK(I)+VSK(I)

RETURN

END
### PROGR AM SECTIONS

<table>
<thead>
<tr>
<th>Name</th>
<th>Bytes</th>
<th>Attributes</th>
<th>Name</th>
<th>Bytes</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 CODE</td>
<td>210</td>
<td>PIC CON REL LCL SHR EXE RD NO#R WT LONG</td>
<td>7 LOCAL</td>
<td>24</td>
<td>PIC CON REL LCL NO#R NO#E RD WT LONG</td>
</tr>
<tr>
<td>1 SPRATA</td>
<td>2</td>
<td>PIC CON REL LCL NO#R NO#E RD WT LONG</td>
<td>5 ENVIR</td>
<td>404</td>
<td>PIC OVR REL GBL SHR NO#E RD WT LONG</td>
</tr>
<tr>
<td>2 LOCAL</td>
<td>2</td>
<td>PIC CON REL LCL NO#R NO#E RD WT LONG</td>
<td>4 DEBUI G</td>
<td>8</td>
<td>PIC OVR REL GBL SHR NO#E RD WT LONG</td>
</tr>
<tr>
<td>3 ENVIR</td>
<td>2</td>
<td>PIC CON REL LCL NO#R NO#E RD WT LONG</td>
<td>5 GPSPAR</td>
<td>192</td>
<td>PIC CON REL GBL SHR NO#E RD WT LONG</td>
</tr>
<tr>
<td>4 DEBUI G</td>
<td>2</td>
<td>PIC CON REL LCL NO#R NO#E RD WT LONG</td>
<td>6 MEASOUT</td>
<td>648</td>
<td>PIC OVR REL GBL SHR NO#E RD WT LONG</td>
</tr>
</tbody>
</table>

### ENTRY POINTS

<table>
<thead>
<tr>
<th>Address</th>
<th>Type</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-00000000</td>
<td></td>
<td>GPS</td>
</tr>
</tbody>
</table>

### VARIABLES

<table>
<thead>
<tr>
<th>Address</th>
<th>Type</th>
<th>Name</th>
<th>Address</th>
<th>Type</th>
<th>Name</th>
<th>Address</th>
<th>Type</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>6-00000160</td>
<td>R=8</td>
<td>DMCL</td>
<td>6-00000110</td>
<td>R=8</td>
<td>DVCL</td>
<td>6-00000278</td>
<td>R=8</td>
<td>EDMCL</td>
</tr>
<tr>
<td>2-00000004</td>
<td>I=4</td>
<td>I</td>
<td>4-00000004</td>
<td>I=4</td>
<td>IDEBUG</td>
<td>4-00000000</td>
<td>I=4</td>
<td>IENTER</td>
</tr>
<tr>
<td>3-00000190</td>
<td>I=4</td>
<td>INIT</td>
<td>2-00000000</td>
<td>I=4</td>
<td>J</td>
<td>2-00000000</td>
<td>I=4</td>
<td>IFLAG</td>
</tr>
</tbody>
</table>

### ARRAYS

<table>
<thead>
<tr>
<th>Address</th>
<th>Type</th>
<th>Name</th>
<th>Bytes</th>
<th>Dimensions</th>
</tr>
</thead>
<tbody>
<tr>
<td>6-00000150</td>
<td>R=8</td>
<td>DMCS</td>
<td>16</td>
<td>(2)</td>
</tr>
<tr>
<td>6-00000160</td>
<td>R=8</td>
<td>DVC5</td>
<td>16</td>
<td>(2)</td>
</tr>
<tr>
<td>6-00000258</td>
<td>R=8</td>
<td>EWS</td>
<td>16</td>
<td>(2)</td>
</tr>
<tr>
<td>6-00000268</td>
<td>R=8</td>
<td>EDYS</td>
<td>16</td>
<td>(2)</td>
</tr>
<tr>
<td>6-00000278</td>
<td>R=8</td>
<td>EMXG</td>
<td>48</td>
<td>(6)</td>
</tr>
<tr>
<td>6-00000170</td>
<td>R=8</td>
<td>MP</td>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>6-00000170</td>
<td>R=8</td>
<td>MS</td>
<td>48</td>
<td>(3, 2)</td>
</tr>
<tr>
<td>6-00000004</td>
<td>R=8</td>
<td>MV</td>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>6-00000004</td>
<td>R=8</td>
<td>MX</td>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>3-00000000</td>
<td>R=8</td>
<td>P</td>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>5-00000000</td>
<td>R=8</td>
<td>PB</td>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>5-00000060</td>
<td>R=8</td>
<td>PBK</td>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>1-00000000</td>
<td>R=8</td>
<td>PROFILE</td>
<td>520</td>
<td>(10, 4)</td>
</tr>
<tr>
<td>5-00000030</td>
<td>R=8</td>
<td>PS</td>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>5-00000000</td>
<td>R=8</td>
<td>PSK</td>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>6-00000090</td>
<td>R=8</td>
<td>GPS</td>
<td>268</td>
<td>(6, 6)</td>
</tr>
<tr>
<td>6-00000208</td>
<td>R=8</td>
<td>RL</td>
<td>32</td>
<td>(2, 2)</td>
</tr>
<tr>
<td>6-00000140</td>
<td>R=8</td>
<td>RS</td>
<td>64</td>
<td>(2, 2, 2)</td>
</tr>
<tr>
<td>3-00000000</td>
<td>R=8</td>
<td>STATE</td>
<td>60</td>
<td>(10)</td>
</tr>
<tr>
<td>3-00000018</td>
<td>R=8</td>
<td>V</td>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>5-00000010</td>
<td>R=8</td>
<td>VB</td>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>5-00000078</td>
<td>R=8</td>
<td>VDK</td>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>5-00000045</td>
<td>R=8</td>
<td>VS</td>
<td>24</td>
<td>(3)</td>
</tr>
</tbody>
</table>
GPS

5-000000AB R=8 VSK

24 (3)

LABELS

<table>
<thead>
<tr>
<th>Address</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>** 10</td>
<td>** 15</td>
</tr>
<tr>
<td>** 20</td>
<td></td>
</tr>
</tbody>
</table>

FUNCTIONS AND SUBROUTINES REFERENCED

GAUSS

Total Space Allocated = 1490 Bytes

COMMAND QUALIFIERS

FORTRAN /LIST GCP,INDATA,MATAB,OUTDATA,RUNG,DNAV,EPHEM,TRUEA,SPRESS,OCCLUD,GPSEQ,VISIBLE,GENENV,TREG,GYROUT,RATE,SMART,CMA

/ICHECK=(NOBOUNDS,OVERFLOW)
/DEBUG=(NOSYMBOLS,TRACEBACK)
/F77 /NOG_FLOATING /14 /OPTIMIZE /WARNINGS /NOD_LINES /NOMACHINE_CODE /CONTINUATIONS=19

COMPILATION STATISTICS

Run Time: 1.42 seconds
Elapsed Time: 21.19 seconds
Page Faults: 358
Dynamic Memory: 160 pages
2.6.2 Star Tracker Model (START)

Subroutine START simulates a star tracker measurement made by the Kth tracker, where K is passed through the subroutine calling sequence. For purposes of simulation the star of interest is assumed to be along the Kth tracker boresight. A unit vector along the Kth tracker boresight is established and corrupted with bias and noise. Compensation for known bias is made by subtracting this knowledge from the unit vector. After the corrupted vector is unitized, the data is passed out through common. The measurement noise covariance is reestablished on each pass. This information is also passed out through common.

GAUSS is required to generate normally distributed random measurement noise.

Mathematical Specification -

Establish a unit vector along the tracker boresight.

\[ U = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \]

Establish star measurement vector by adding bias and noise to the unit vector, and compensating for bias knowledge.

\[ MS(K) = U + BS(K) + N(0, SS(K)) - BSK(K) \]

\[ N(\mu, \sigma) = \text{Normal variable with mean } \mu \text{, and standard deviation } \sigma \ . \]

Reconstruct a unit vector

Establish sensor measurement

\[ DVCS = \text{ASIN}(MS(1,K)) \]
\[ DHCS = \text{ASIN}(MS(2,K)/\text{COS}(DVCS(K))) \]

Establish measurement covariance

\[ [RS(K)] = \begin{bmatrix} SS(1,K)^2 & 0 \\ 0 & SS(2,K)^2 \end{bmatrix} \]
STAR TRACKER MODEL (START)

- GENERATE MEASUREMENT UNIT VECTOR
- CORRUPT MEASUREMENT WITH BIAS AND NOISE
- COMPENSATE FOR BIAS KNOWLEDGE
- NORMALIZE MEASUREMENT VECTOR

FIRST PASS

<table>
<thead>
<tr>
<th>T</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>GENERATE MEASUREMENT NOISE COVARIANCE</td>
<td>NULL</td>
</tr>
</tbody>
</table>

Figure A-31
SUBROUTINE START()

---

## SUBROUTINE START SIMULATES A MEASUREMENT MADE BY AN ONBOARD FIXED STAR TRACKER BY ADDING BIAS AND NOISE TO A UNIT VECTOR ALONG THE STAR TRACKER BORESIGHT.

---

### ONCE THE MEASUREMENT IS MADE COMPENSATION FOR THE KNOWN BIAS IS MADE BY SUBTRACTING THAT BIAS FROM THE MEASUREMENT.

---

#### INPUT PARAMETERS

- Q = INERTIAL TO BODY QUATERNIANS (4x1)
- BS = STAR TRACKER ACTUAL BIAS VECTOR (2x1)
- SS = (RAD) NOISE STANDARD DEVIATION (2x1)
- TNS = MISEALIGNMENT ARRAY (3x3)
- TBNS = ORIENTATION ARRAY (3x3)
- BSK = BIAS KNOWLEDGE (3x3)
- SSK = NOISE STANDARD DEVIATION KNOWLEDGE (2x1)
- TNSK = MISEALIGNMENT KNOWLEDGE (3x3)

#### OUTPUT VARIABLES

- DMCS = HORIZONTAL DEVIATION FROM BORESIGHT (RAD)
- DVCS = VERTICAL DEVIATION FROM BORESIGHT (RAD)
- MS = MEASUREMENT STAR VECTOR (3x1)
- RS = STAR NOISE COVARIANCE (2x2)

---

**PROGRAMMED BY J. W. K. 11 JUNE 1980**

**EXT 4443**

---

### COMMON BLOCKS

- INCLUDE 'DEBUG.COM'
- COMMON /DEBUG/ IN/DEBUG

### USER CONTROLLED PARAMETERS TO VARY DEBUG PRINT LEVEL

- IF 1, PRINTS WHEN ENTERS MOST SUBROUTINES
- DEBUG 0-10, HIGHER NUMBER MEANS MORE PRINT

---

### INCLUDE 'STARPAR.COM'

- COMMON /STARPAR/ B5(2,2),SS(2,2),TNS(3,3,2),TBNS(3,3,2)
- BSK(2,2), SSK(2,2), TNSK(3,3,2)

---

### REAL*8 B5,SS,TNS,TBNS,BSK,SSK,TNSK

---

### STAR TRACKER PARAMETERS

- IN EACH CASE THE LAST SUBSCRIPT REFERS TO THE TRACKER USED
INCLUDE 'MEASOUT.CGY'

REAL*8 MX, GPS, DVC, MS, RS, DHCL, LMU, RL, EMXG, EDHS, DVS, EDVL

MEASUREMENT OUTPUT PARAMETERS

MX = POSITION/VELOCITY STATE MEASUREMENT - GPS (KM./HR/SEC)
EMXG = ESTIMATED POSITION/VELOCITY STATE MEASUREMENT - GPS
RGPS = STATE MEASUREMENT NOISE COVARIANCE (KNOWLEDGE) - GPS (KM./SEC**2)
DMCS = STAR MEASUREMENT HORIZONTAL DEVIATION FROM BORESIGHT - START (RAD)
DVCS = STAR MEASUREMENT VERTICAL DEVIATION FROM BORESIGHT - START (RAD)
EDHS = ESTIMATED STAR MEASUREMENT HORIZONTAL DEVIATION FROM BORESIGHT (RAD)
EDVS = ESTIMATED STAR MEASUREMENT VERTICAL DEVIATION FROM BORESIGHT (RAD)
MS = STAR MEASUREMENT UNIT VECTOR (SECOND SUFFIX REFERS TO TRACKER) - START
RS = STAR MEASUREMENT NOISE COVARIANCE (KNOWLEDGE) - START (RAD**2)
DHCL = LANDMARK MEASUREMENT HORIZONTAL DEVIATION FROM BORESIGHT - LAMKT (RAD)
DVCL = LANDMARK MEASUREMENT VERTICAL DEVIATION FROM BORESIGHT - LAMKT (RAD)
EDML = ESTIMATED LANDMARK MEASUREMENT HORIZONTAL DEVIATION FROM BORESIGHT (RAD)
EDVL = ESTIMATED LANDMARK MEASUREMENT VERTICAL DEVIATION FROM BORESIGHT (RAD)
LMU = LANDMARK MEASUREMENT UNIT VECTOR - LAMKT
RL = LANDMARK MEASUREMENT NOISE COVARIANCE (KNOWLEDGE) - LAMKT (RAD**2)

C DIMENSION U(3)
DATA :F.Int(0/0)
C GENERATE MEASUREMENT UNIT VECTOR IN STAR TRACKER COORDINATES
```
C**---------------------------------------------**
C ADD BIAS AND NOISE TO THE IDEAL MEASUREMENT AND
C SUBTRACT OFF BIAS KNOWLEDGE
C**---------------------------------------------**
DO 10 I=1,2
   MS(I,K)=U(I)+BS(I,I)+GAUSS(O,SS(I,K))-BSK(I,K)
C**---------------------------------------------**
C NORMALIZE THE VECTOR
MS(3,K)=SQRT(1.-MS(1,K)**2-MS(2,K)**2)
DVCS(K)=ASIN(MS(1,K))
DMCS(K)=ASIN(MS(2,K)/COS(DVCS(K)))
C GENERATE STAR TRACKER NOISE COVARIANCE
DO 20 I=1,2
   RS(I,I,K)=0.
DO 30 I=1,2
   DC 30 I=1,2
   RS(I,I,K)=SSK(I,K)**2
RETURN
END

\textbf{PROGRAM SECTIONS}

<table>
<thead>
<tr>
<th>Name</th>
<th>Bytes</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>CODE</td>
<td>298</td>
<td>PIC CON REL LCL SHR EXE RD NOWRT LONG</td>
</tr>
<tr>
<td>DATA</td>
<td>4</td>
<td>PIC CON REL LCL SHR NOEXE RD NOWRT LONG</td>
</tr>
<tr>
<td>LOCAL</td>
<td>48</td>
<td>PIC CON REL LCL NOEXE NOEXE RD WRT QUAD</td>
</tr>
<tr>
<td>DEBUG</td>
<td>568</td>
<td>PIC OVR REL GBL NOEXE NOEXE RD WRT QUAD</td>
</tr>
<tr>
<td>STARPAR</td>
<td>648</td>
<td>PIC OVR REL GBL NOEXE NOEXE RD WRT LONG</td>
</tr>
<tr>
<td>MEASOUT</td>
<td>564</td>
<td>PIC OVR REL GBL NOEXE NOEXE RD WRT LONG</td>
</tr>
</tbody>
</table>

\textbf{ENTRY POINTS}

<table>
<thead>
<tr>
<th>Address Type Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-00000000 START</td>
</tr>
</tbody>
</table>

\textbf{VARIABLES}

<table>
<thead>
<tr>
<th>Address Type Name</th>
</tr>
</thead>
</table>
```

<table>
<thead>
<tr>
<th>Address Type Name</th>
<th>Address Type Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>5-000001E0 R+8 DMCL</td>
<td>5-020001EB R+8 DVCL</td>
</tr>
<tr>
<td>2-0000001C I+4 I</td>
<td>3-00000004 I+4 IDEBUG</td>
</tr>
<tr>
<td>2-00000020 I+4 J</td>
<td>AP-00000000 I+4 K</td>
</tr>
<tr>
<td>5-00000278 R+8 EDML</td>
<td>5-00000280 R+8 EDVL</td>
</tr>
<tr>
<td>3-00000000 I+4 IENTER</td>
<td>2-00000018 I+4 IFLAG</td>
</tr>
</tbody>
</table>
## ARRAYS

<table>
<thead>
<tr>
<th>Address</th>
<th>Type</th>
<th>Name</th>
<th>Bytes</th>
<th>Dimensions</th>
</tr>
</thead>
<tbody>
<tr>
<td>4-00000000</td>
<td>R+8</td>
<td>BS</td>
<td>32</td>
<td>(2, 2)</td>
</tr>
<tr>
<td>4-00000160</td>
<td>R+8</td>
<td>BSK</td>
<td>32</td>
<td>(2, 2)</td>
</tr>
<tr>
<td>5-00000150</td>
<td>R+8</td>
<td>DMCS</td>
<td>16</td>
<td>(2)</td>
</tr>
<tr>
<td>5-00000160</td>
<td>R+8</td>
<td>DVC5</td>
<td>16</td>
<td>(2)</td>
</tr>
<tr>
<td>5-00000258</td>
<td>R+8</td>
<td>EDYS</td>
<td>16</td>
<td>(2)</td>
</tr>
<tr>
<td>5-00000268</td>
<td>R+8</td>
<td>EDVS</td>
<td>16</td>
<td>(2)</td>
</tr>
<tr>
<td>5-00000228</td>
<td>R+8</td>
<td>EMKG</td>
<td>48</td>
<td>(6)</td>
</tr>
<tr>
<td>5-000001F0</td>
<td>R+8</td>
<td>LMU</td>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>5-00000170</td>
<td>R+8</td>
<td>MS</td>
<td>48</td>
<td>(3, 2)</td>
</tr>
<tr>
<td>5-00000000</td>
<td>R+8</td>
<td>MX</td>
<td>48</td>
<td>(6)</td>
</tr>
<tr>
<td>5-00000208</td>
<td>R+8</td>
<td>RL</td>
<td>32</td>
<td>(2, 2)</td>
</tr>
<tr>
<td>5-000001A0</td>
<td>R+8</td>
<td>RS</td>
<td>64</td>
<td>(3, 2, 2)</td>
</tr>
<tr>
<td>4-00000200</td>
<td>R+8</td>
<td>SS</td>
<td>32</td>
<td>(2, 2)</td>
</tr>
<tr>
<td>4-00000180</td>
<td>R+8</td>
<td>JSK</td>
<td>32</td>
<td>(2, 2)</td>
</tr>
<tr>
<td>4-00000000</td>
<td>R+8</td>
<td>TBSN</td>
<td>144</td>
<td>(3, 3, 2)</td>
</tr>
<tr>
<td>4-00000040</td>
<td>R+8</td>
<td>TNS</td>
<td>144</td>
<td>(3, 3, 2)</td>
</tr>
<tr>
<td>4-000001A0</td>
<td>R+8</td>
<td>TNSK</td>
<td>144</td>
<td>(3, 3, 2)</td>
</tr>
<tr>
<td>2-60000000</td>
<td>R+8</td>
<td>U</td>
<td>24</td>
<td>(3)</td>
</tr>
</tbody>
</table>

## LABELS

<table>
<thead>
<tr>
<th>Address</th>
<th>Label</th>
<th>Address</th>
<th>Label</th>
<th>Address</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>**</td>
<td>10</td>
<td>**</td>
<td>20</td>
<td>**</td>
<td>30</td>
</tr>
</tbody>
</table>

## FUNCTIONS AND SUBROUTINES REFERENCED

- GAUSS
- MTH$DASIN
- MTH$DCOS
- MTH$DSQRT

Total Space Allocated = 1566 Bytes

## COMMAND QUALIFIERS

- FORTRAN /LIST GCP,INDATA,MATAB,OUTDATA,RUNG,DNAV,EPHEM,TRUEA,SPRESS,occult,GPert,GCPEQ,VISIBLE,GENENV,TREG,GROUT,RATE,BMAT,CMA
- /CHECK=(NOBOUNDS,OVERFLOW)
- /DEBUG=(NOSYMBOLS,TRACEBACK)
- /F?7 ,NOG_FLOATING /14 /OPTIMIZE /WARNINGS /NOD_LINES /NOMACHINE_CODE /CONTINUATIONS=19

## COMPILATION STATISTICS

- Run Time: 1.55 seconds
- Elapsed Time: 15.94 seconds
- Page Faults: 355
- Dynamic Memory: 160 pages
2.6.3 Landmark Tracker Model (LAMKT)

Subroutine LAMKT simulates a measurement made by the landmark tracker. The landmark position vector is established in earth fixed coordinates through knowledge of the landmark’s longitude, latitude, and altitude, passed through common. This position vector is then transformed to inertial coordinates. A line-of-sight vector to the landmark from the spacecraft is established by subtracting from the landmark position vector the spacecraft position vector. The line-of-sight vector is transformed to exact landmark tracker coordinates where it is normalized to a unit vector. The exact DH and DV measurements are computed based on the computed line-of-sight unit vector. These DH and DV measurements are corrupted with bias and noise and compensated by subtraction of bias knowledge. A unit vector is re-established from the corrupt measurement and the noise covariance is established. All data output is through common.

Processing Requirements

The landmark location on the surface of the earth in Local Landmark Coordinates (Figure VII) will be a function of the altitude ($A_L$) above the earth’s mean radius.

$$L_L = \begin{bmatrix} A_L \\ 0 \\ 0 \end{bmatrix}$$

However, in earth fixed coordinates, the landmark will have the earth’s mean radius ($r_E$) added to the altitude. Using the angular transformation from local landmark to earth fixed coordinates produces

$$L_E = e^T_L \cdot L_L = e^T_L \begin{bmatrix} r_E + A_L \\ 0 \\ 0 \end{bmatrix} = (r_E + A_L) \begin{bmatrix} CLC \alpha & -SL \alpha & -CLS \alpha \\ SLC \alpha & CL \alpha & -SLS \alpha \\ SA & 0 & CA \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

As shown in Figure A-32 the position vector of the spacecraft ($P_S/C$), when subtracted from the landmark position in some coordinate frame, will provide the measurement vector ($M$).

$$M_T = (i^T_{E_L E}) - P_S/C$$

Accounting for hardware misalignments, the same measurement vector in landmark tracker coordinates is:

$$M_I = i^T (i^T_{E_L E}) - P_S/C$$

$$= i^T_{E_L E} - T_I P_S/C$$
Figure A-32
Landmark Tracker Geometry - Sighting Plane

Figure A-33
Landmark Tracker Geometry - General
From examination of Figure A-33, the unit measurement vector in landmark tracker coordinates is:

\[
\mathbf{U_L} = \frac{M_L}{|M_L|} = \begin{bmatrix} U_L x \\ U_L y \\ U_L z \end{bmatrix} = \begin{bmatrix} \sin\Delta V \\ \cos\Delta V \sin\Delta H \\ -\cos\Delta V \cos\Delta H \end{bmatrix}
\]

However, the tracker instrument has no sensitivity to projections along its boresight axis. Therefore, the tracker response to the unit vector \( \mathbf{U_L} \) will be:

\[
\mathbf{U'_L} = \begin{bmatrix} U'_L x \\ U'_L y \end{bmatrix} = \begin{bmatrix} \sin\Delta V \\ \cos\Delta V \sin\Delta H \end{bmatrix}
\]

producing \( \Delta H \) and \( \Delta V \) as shown in Figure A-33 as sensor outputs. Since the sensor output will be corrupted by bias and noise, the sensed measurement will be:

\[
\mathbf{Z} = \begin{bmatrix} \Delta H_s \\ \Delta V_s \end{bmatrix} = \begin{bmatrix} \Delta H + b_H + V_H \\ \Delta V + b_V + V_V \end{bmatrix}
\]

Where:

- \( b_H, b_V = \) Component landmark tracker bias.
- \( V_H, V_V = \) Component landmark tracker zero mean random noise, \( N(0, \sigma^2) \).

The component biases and standard deviations \( (\sigma) \) are user selectables.

The landmark tracker measurement may be uncompensated for knowledge of instrument bias. The bias knowledge may be a priori or through estimation. The compensated sensor output will be:

\[
\hat{\mathbf{Z}} = \begin{bmatrix} \Delta H_c \\ \Delta V_c \end{bmatrix} = \begin{bmatrix} \Delta H_s - b_H \\ \Delta V_s - b_V \end{bmatrix}
\]

Where:

- \(LM = \) the landmark being used = \( f(L, \lambda, A_L) \)
$A_L$ = the altitude of the landmark above the mean radius of the earth
$L'$ = longitude of the landmark
$\lambda$ = latitude of the landmark
$L$ = vector position of the landmark relative to the center of the earth
$E_{s/c}$ = vector position of the spacecraft relative to the center of the earth
$A_{s/c}$ = altitude of the spacecraft above the mean radius of the earth
$M$ = measurement vector from the spacecraft to the landmark
$\hat{M}$ = unit vector along $M$
$U$ = the landmark tracker horizontal plane angular deflection from the boresight axis
$V$ = the landmark tracker vertical plane angular deflection from the boresight axis
Landmark Tracker Model (LAMKT)

<table>
<thead>
<tr>
<th>Step</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generate Landmark Position in Earth Fixed Coordinates</td>
</tr>
<tr>
<td>Bring Landmark Position into Inertial Space</td>
</tr>
<tr>
<td>Subtract Spacecraft Position Vector</td>
</tr>
<tr>
<td>Transform To Landmark Tracker Coordinates</td>
</tr>
<tr>
<td>Normalize Line-of-Sight Vector</td>
</tr>
<tr>
<td>Generate DN and DV Measurements</td>
</tr>
<tr>
<td>Corrupt Measurement With Bias and Noise</td>
</tr>
<tr>
<td>Compensate for Bias Knowledge</td>
</tr>
<tr>
<td>Construct Measurement Unit Vector</td>
</tr>
<tr>
<td>Construct Measurement Noise Covariance</td>
</tr>
</tbody>
</table>

Figure A-34
SUBROUTINE LAMKT

C******************************************************************************
C SUBROUTINE LAMKT SIMULATES A MEASUREMENT MADE BY THE
C LANDMARK TRACKER. IT DOES SO BY SUBTRACTING FROM PERFECT
C KNOWLEDGE OF THE LANDMARK POSITION (LONGITUDE, LATITUDE,
C AND ALTITUDE) PERFECT KNOWLEDGE OF THE SPACECRAFT POSITION.
C TO THIS VECTOR BIAS AND NOISE ARE ADDED WHILE IN LANDMARK
C TRACKER COORDINATE. COMPENSATION FOR KNOWN BIAS IS MADE
C BY SUBTRACTING THIS KNOWLEDGE AT THIS POINT. COMPENSATION
C FOR KNOWN LANDMARK TRACKER MISALIGNMENT IS MADE WHEN THE
C RESULTING VECTOR IS TRANSFORMED BACK TO INERTIAL SPACE.
C******************************************************************************

INPUT VARIABLES

C RE = AVERAGE RADIUS OF THE EARTH (KM)
C AL = LANDMARK ALTITUDE (KM)
C LON = LANDMARK LONGITUDE (DEG)
C LAT = LANDMARK LATITUDE (DEG)
C T = TIME (TBD)
C P = SPACECRAFT POSITION VECTOR (3*1) (KM)
C Q = EXACT INERTIAL TO BODY QUATERNIAN (4*1)
C TBML = LANDMARK TRACKER ORIENTATION ARRAY
C (NOMINAL TO BODY) (3*3)
C TNL = LANDMARK TRACKER MISALIGNMENT ARRAY
C (TRACKER TO NOMINAL) (3*3)
C BL = LANDMARK TRACKER BIAS (2*1) (RAD)
C SL = LANDMARK TRACKER NOISE STANDARD DEVIATION
C (2*1) (RAD)
C BKL = LANDMARK TRACKER BIAS KNOWLEDGE (2*1) (RAD)
C SKL = LANDMARK TRACKER NOISE STANDARD
C DEVIATION (2*1) (RAD)
C TNLK = LANDMARK TRACKER MISALIGNMENT KNOWLEDGE
C ARRAY (TRACKER TO NOMINAL) (3,3)
C SIGGCP = POSITION UNCERTAINTY DUE TO CLOUDS

OUTPUT VARIABLES

C DHCL = LANDMARK HORIZONTAL DEFLECTION (RAD)
C DVCL = LANDMARK VERTICAL DEFLECTION (RAD)
C LWU = SIMULATED LANDMARK TRACKER UNIT VECTOR
C TO LANDMARK (3*1)
C RL = LANDMARK TRACKER MEASUREMENT COVARIANCE
C (3*3) (RAD)

PRORAMMED BY JACK MYERS 11JUNE1980
EXT 4443

COMMON BLOCKS

C INCLUDE 'ENVIR.COM'
C COMMON /ENVIR/, STATE(10), PROFILE(10, 4), INIT
C REAL*8 STATE, PROFILE
C REAL WORLD STATE PARAMETERS
C STATE STATE VALUES; X,Y,Z,XD,YD,ZD,E0, E1, E2, E3
INCLUDE 'TIME.COM'

include 'DEBUG.COM'

USER CONTROLLED PARAMETERS TO VARY DEBUG PRINT LEVEL

LANDMARK TRACKER PARAMETERS

AL = ALTITUDE OF LANDMARK (KM)
LON = LONGITUDE OF LANDMARK (DEG)
LAT = LATITUDE OF LANDMARK (DEG)
\"BNL = ORIENTATION ARRAY FOR LANDMARK TRACKER
\"NOMINAL TO BODY
TNL = MISALIGNMENT ARRAY - ACTUAL
TRACKER TO NOMINAL
BL = BIAS - ACTUAL (RAD)
SL = NOISE STANDARD DEVIATION - ACTUAL (RAD)
SKL = BIAS - KNOWLEDGE (RAD)
THET  =  LOOK  ANGLE  (RAD)  
SKL   =  NOISE  STANDARD  DEVIATION  -KNOWLEDGE  (RAD)  
TIEO  =  INITIAL  EARTH  FIXED  TO  INERTIAL  
TNLK  =  MISALIGNMENT  ARRAY  KNOWLEDGE  
TRACKER  TO  NOMINAL  
SIGGCP =  POSITION  UNCERTAINTY  DUE  TO  CLOUDS  

INCL'IDE  'MEASOUT.COM'  
COMMON  /MEASOUT/  MX,RGPS,DHCS,DVCS,MS,RS,DHCL,DVCL,LMU,RL,EMXG,EDHS,EDVS,EDHL,EDVL  
REAL  S  MX,RGPS,DHCS,DVCS,MS,RS,DHCL,DVCL,LMU,RL,EMXG,EDHS,EDVS,EDHL,EDVL  

MEASUREMENT  OUTPUT  PARAMETERS  
MX  =  POSITION/VELOCITY  STATE  MEASUREMENT  -  GPS  
   (KM, KM/SEC)  
EMAG  =  ESTIMATED  POSITION/VELOCITY  STATE  
RGPS  =  STATE  MEASUREMENT  NOISE  COVARIANCE  
(DHCS  =  STAR  MEASUREMENT  HORIZONTAL  DEVIATION  
DVCS  =  STAR  MEASUREMENT  VERTICAL  DEVIATION  
EDHS  =  ESTIMATED  STAR  MEASUREMENT  HORIZONTAL  
EDVS  =  ESTIMATED  STAR  MEASUREMENT  VERTICAL  
MS  =  STAR  MEASUREMENT  UNIT  VECTOR  (SECOND  
SUBSCRIPT  REFERS  TO  TRACKER)  
RS  =  STAR  MEASUREMENT  NOISE  COVARIANCE  
(DHCL  =  LANDMARK  MEASUREMENT  HORIZONTAL  DEVIATION  
DVCL  =  LANDMARK  MEASUREMENT  VERTICAL  DEVIATION  
EDHL  =  ESTIMATED  LANDMARK  MEASUREMENT  HORIZONTAL  
EDVL  =  ESTIMATED  LANDMARK  MEASUREMENT  VERTICAL  
LMU  =  LANDMARK  MEASUREMENT  UNIT  VECTOR  
RL  =  LANDMARK  MEASUREMENT  NOISE  COVARIANCE  

INCLUDE  'NSTATE.COM'  
COMMON  /NSTATE/  XD,X(6),RADM,RADE  
REAL  S  XD,X,RADM,RADE  

POSITION  STATE  AND  CONSIDERED  PARAMETERS  
XD  =  STATE  DERIVATIVES  (KM/SEC  AND  KM/SEC/SEC)  
X  =  STATE  POSITION  PARAMETERS  (KM  AND  KM/SEC)  
RADM  =  RADIUS  OF  THE  MOON  (KM)
**Measurements Specifications**

- **TYPE** MEASUREMENT TYPE
- **JFLAG** SET FOR STAR OBSTRUCTION
- **MCODE** "MEASUREMENT PROCESSING"
- **PI** PI
- **TPI** TPI

**Equivalence**
- \( \text{STATE}(7), (P, \text{STATE}), \text{DELT}, \text{DELTL}, (B, \text{BL}), (S, \text{SL}), (B, \text{BL}), (S, \text{SL}) \)
- \( \text{LAT} \)
- \( \text{TIE} \)
- \( \text{LAT} \)

**Data Processing**
- **RAE** EARTH DETECTABLE RADIUS (KM)

**Conversion**
- \( \text{RADIAN TO DEGREE CONVERSION} \)

**Measurement Generation**
- \( \text{GENERATE THE LANDMARK POSITION IN EARTH FIXED COORDINATES} \)

**Tie Generation**
- \( \text{GENERATE TIE} \)

**Position Transform**
- \( \text{TRANSFORM TO LANDMARK TRACKER COORDINATES} \)

**Normalization**
- \( \text{NORMALIZE THE MEASUREMENT VECTOR} \)

**Unit Generation**
- \( \text{GENERATE DV MEASUREMENTS FROM UNIT MEASUREMENT VECTOR} \)

**Coordinate Transform**
- \( \text{TRANSFORM TO INERTIAL SPACE} \)

**Degree Conversion**
- \( \text{DEGREE TO RADIAN} \)

**Landmark Position**
- \( \text{GENERATE LANDMARK POSITION IN INERTIAL SPACE} \)

**Tie Calculation**
- \( \text{CALCULATE TIE} \)

**Unit Calculation**
- \( \text{CALCULATE UNIT} \)

**Data Processing**
- \( \text{DATA PROCESSING} \)

**Measurement Specifications**
- **TYPE** MEASUREMENT TYPE
- **JFLAG** SET FOR STAR OBSTRUCTION
- **MCODE** "MEASUREMENT PROCESSING"
- **PI** PI
- **TPI** TPI

**Equivalence**
- \( \text{STATE}(7), (P, \text{STATE}), \text{DELT}, \text{DELTL}, (B, \text{BL}), (S, \text{SL}), (B, \text{BL}), (S, \text{SL}) \)
- \( \text{LAT} \)
- \( \text{TIE} \)
- \( \text{LAT} \)

**Data Processing**
- **RAE** EARTH DETECTABLE RADIUS (KM)

**Conversion**
- \( \text{RADIAN TO DEGREE CONVERSION} \)

**Measurement Generation**
- **RAE** EARTH DETECTABLE RADIUS (KM)

**Tie Generation**
- **RAE** EARTH DETECTABLE RADIUS (KM)

**Position Transform**
- **RAE** EARTH DETECTABLE RADIUS (KM)

**Normalization**
- **RAE** EARTH DETECTABLE RADIUS (KM)

**Unit Generation**
- **RAE** EARTH DETECTABLE RADIUS (KM)

**Coordinate Transform**
- **RAE** EARTH DETECTABLE RADIUS (KM)

**Degree Conversion**
- **RAE** EARTH DETECTABLE RADIUS (KM)

**Landmark Position**
- **RAE** EARTH DETECTABLE RADIUS (KM)

**Tie Calculation**
- **RAE** EARTH DETECTABLE RADIUS (KM)

**Unit Calculation**
- **RAE** EARTH DETECTABLE RADIUS (KM)

**Data Processing**
- **RAE** EARTH DETECTABLE RADIUS (KM)
DH = ASIN(UML(2)/COS(DV))

C CORRUPT THE MEASUREMENT VECTOR WITH BIAS AND NOISE AND
C COMPENSATE FOR THE BIAS KNOWLEDGE

C******************************************************************************
SL(1)*.001*SIGGCP/VMAG(MI,3) CHANGE DISTANCE TO ANGLE IN RAD.
SL(2)=SL(1)
DHCL=DH+BL(1)+GAUSS/0..SL(1))-BKL(1)
DVCL=DV+BL(2)+GAUSS/0..SL(2))-BKL(2)
C******************************************************************************
C RECONSTRUCT UNIT VECTOR FROM COMPENSATED MEASUREMENTS
C POSSIBLY NOT NEEDED AT THIS POINT
C******************************************************************************
LMU(1)=COS(DVCL)*SIN(DHCL)
LMU(2)=SIN(DVCL)
LMU(3)=COS(DVCL)*COS(DHCL)
C******************************************************************************
C FORM MEASUREMENT NOISE COVARIANCE
C******************************************************************************
DO 20 I=1,2
DO 20 J=1,2
20 RL(I,J)=0.
DO 30 I=1,2
30 RL(I,1)=SL(I)*SL(I)
RETURN
END

PROGRAM SECTIONS

Name Bytes Attributes
0 $CODE 501 PIC CON REL LCL SHR EXE RD NDWRT LONG
1 $DATA 12 PIC CON REL LCL SHR NOEXE RD NDWRT LONG
2 $LOCAL 748 PIC CON REL LCL NOSHR NOEXE RD WRT QUAD
3 ENVIR 404 PIC OVR REL GBL SHR NOEXE RD WRT LONG
4 TIME 136 PIC OVR REL GBL SHR NOEXE RD WRT LONG
5 DEBUG 8 PIC OVR REL GBL SHR NOEXE RD WRT LONG
6 LMTPAR 392 PIC OVR REL GBL SHR NOEXE RD WRT LONG
7 MEASOUT 648 PIC OVR REL GBL SHR NOEXE RD WRT LONG
8 NSTATE 112 PIC OVR REL GBL SHR NOEXE RD WRT LONG
9 TARGETS 36 PIC OVR REL GBL SHR NOEXE RD WRT LONG

ENTRY POINTS

Address Type Name
0-00000000 LAMKT
### VARIABLES

<table>
<thead>
<tr>
<th>Address</th>
<th>Type</th>
<th>Name</th>
<th>Address</th>
<th>Type</th>
<th>Name</th>
<th>Address</th>
<th>Type</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>6-00000000</td>
<td>R=B</td>
<td>AL</td>
<td>6-00000068</td>
<td>R=B</td>
<td>B</td>
<td>6-000000CB</td>
<td>R=B</td>
<td>BK</td>
</tr>
<tr>
<td>4-00000070</td>
<td>R=B</td>
<td>DATE</td>
<td>4-000000B0</td>
<td>R=B</td>
<td>BL</td>
<td>2-00000000</td>
<td>R=B</td>
<td>DATED</td>
</tr>
<tr>
<td>2-00000138</td>
<td>R=B</td>
<td>DH</td>
<td>7-0000016D</td>
<td>R=B</td>
<td>DMCS</td>
<td>4-00000066</td>
<td>R=B</td>
<td>DELT</td>
</tr>
<tr>
<td>4-00000080</td>
<td>R=B</td>
<td>DPRINT</td>
<td>2-000001F0</td>
<td>R=B</td>
<td>DVC</td>
<td>4-000000CB</td>
<td>R=B</td>
<td>DELT</td>
</tr>
<tr>
<td>7-00000290</td>
<td>R=B</td>
<td>EDVL</td>
<td>2-0000019C</td>
<td>R=B</td>
<td>EDVS</td>
<td>7-0000011B</td>
<td>R=B</td>
<td>EDC</td>
</tr>
<tr>
<td>2-00000208</td>
<td>R=B</td>
<td>IFLAG</td>
<td>6-00000109</td>
<td>R=B</td>
<td>INIT</td>
<td>7-0000011B</td>
<td>R=B</td>
<td>EDCL</td>
</tr>
<tr>
<td>9-000000D0</td>
<td>R=B</td>
<td>LS</td>
<td>6-00000199</td>
<td>R=B</td>
<td>I4</td>
<td>5-00000064</td>
<td>I+4</td>
<td>IDEBUG</td>
</tr>
<tr>
<td>9-000000D0</td>
<td>I+4</td>
<td>LS</td>
<td>9-000000D0</td>
<td>I+4</td>
<td>HS</td>
<td>5-00000068</td>
<td>I+4</td>
<td>IS</td>
</tr>
<tr>
<td>8-000000E8</td>
<td>R=B</td>
<td>RADM</td>
<td>6-00000004</td>
<td>I+4</td>
<td>Students</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8-000000E8</td>
<td>R=B</td>
<td>RADM</td>
<td>6-000000E8</td>
<td>R=B</td>
<td>RE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6-000000E8</td>
<td>R=B</td>
<td>TGGCP</td>
<td>6-0000000E</td>
<td>R=B</td>
<td>SK</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2-00000200</td>
<td>R=B</td>
<td>TO</td>
<td>4-00000180</td>
<td>R=B</td>
<td>SIGNCP</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4-00000028</td>
<td>R=B</td>
<td>TIN</td>
<td>6-0000017E</td>
<td>R=B</td>
<td>SGGCP</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4-00000000</td>
<td>R=B</td>
<td>TNMEAT</td>
<td>4-000000EC</td>
<td>R=B</td>
<td>SK</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4-00000010</td>
<td>R=B</td>
<td>TSTOP</td>
<td>4-000000E8</td>
<td>R=B</td>
<td>S</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### ARRAYS

<table>
<thead>
<tr>
<th>Address</th>
<th>Type</th>
<th>Name</th>
<th>Bytes</th>
<th>Dimensions</th>
</tr>
</thead>
<tbody>
<tr>
<td>6-000000CB</td>
<td>R=B</td>
<td>BKL</td>
<td>16</td>
<td>2</td>
</tr>
<tr>
<td>6-0000008B</td>
<td>R=B</td>
<td>BL</td>
<td>16</td>
<td>2</td>
</tr>
<tr>
<td>7-00000130</td>
<td>R=B</td>
<td>DMCS</td>
<td>16</td>
<td>2</td>
</tr>
<tr>
<td>7-0000016D</td>
<td>R=B</td>
<td>DMCS</td>
<td>16</td>
<td>2</td>
</tr>
<tr>
<td>7-0000025B</td>
<td>R=B</td>
<td>DMCS</td>
<td>16</td>
<td>2</td>
</tr>
<tr>
<td>7-0000026B</td>
<td>R=B</td>
<td>EDVS</td>
<td>16</td>
<td>2</td>
</tr>
<tr>
<td>7-00000278</td>
<td>R=B</td>
<td>EDVS</td>
<td>49</td>
<td>6</td>
</tr>
<tr>
<td>2-00000008</td>
<td>R=B</td>
<td>EMXG</td>
<td>24</td>
<td>3</td>
</tr>
<tr>
<td>2-00000020</td>
<td>R=B</td>
<td>EMXG</td>
<td>24</td>
<td>3</td>
</tr>
<tr>
<td>7-00000510</td>
<td>R=B</td>
<td>EMXG</td>
<td>24</td>
<td>3</td>
</tr>
<tr>
<td>7-00000538</td>
<td>R=B</td>
<td>EMXG</td>
<td>24</td>
<td>3</td>
</tr>
<tr>
<td>7-00000570</td>
<td>R=B</td>
<td>EMXG</td>
<td>24</td>
<td>3</td>
</tr>
<tr>
<td>7-0000057B</td>
<td>R=B</td>
<td>EMXG</td>
<td>48</td>
<td>6</td>
</tr>
<tr>
<td>2-00000000</td>
<td>R=B</td>
<td>EMXG</td>
<td>24</td>
<td>3</td>
</tr>
<tr>
<td>3-00000050</td>
<td>R=B</td>
<td>EMXG</td>
<td>320</td>
<td>10, 4</td>
</tr>
<tr>
<td>7-00000030</td>
<td>R=B</td>
<td>EMXG</td>
<td>288</td>
<td>16, 6</td>
</tr>
<tr>
<td>7-00000120</td>
<td>R=B</td>
<td>EMXG</td>
<td>32</td>
<td>2, 2</td>
</tr>
<tr>
<td>7-00000140</td>
<td>R=B</td>
<td>EMXG</td>
<td>64</td>
<td>2, 2, 2</td>
</tr>
<tr>
<td>6-00000058</td>
<td>R=B</td>
<td>EMXG</td>
<td>16</td>
<td>2</td>
</tr>
<tr>
<td>6-00000088</td>
<td>R=B</td>
<td>EMXG</td>
<td>16</td>
<td>2</td>
</tr>
<tr>
<td>3-00000080</td>
<td>R=B</td>
<td>EMXG</td>
<td>80</td>
<td>10</td>
</tr>
<tr>
<td>2-00000000</td>
<td>R=B</td>
<td>EMXG</td>
<td>72</td>
<td>3</td>
</tr>
<tr>
<td>2-00000050</td>
<td>R=B</td>
<td>EMXG</td>
<td>72</td>
<td>3</td>
</tr>
<tr>
<td>6-0000001B</td>
<td>R=B</td>
<td>EMXG</td>
<td>72</td>
<td>3</td>
</tr>
<tr>
<td>2-0000001A</td>
<td>R=B</td>
<td>EMXG</td>
<td>72</td>
<td>3</td>
</tr>
<tr>
<td>6-0000001B</td>
<td>R=B</td>
<td>EMXG</td>
<td>72</td>
<td>3</td>
</tr>
<tr>
<td>6-0000001A</td>
<td>R=B</td>
<td>EMXG</td>
<td>72</td>
<td>3</td>
</tr>
<tr>
<td>6-00000000</td>
<td>R=B</td>
<td>EMXG</td>
<td>72</td>
<td>3</td>
</tr>
<tr>
<td>2-00000000</td>
<td>R=B</td>
<td>EMXG</td>
<td>72</td>
<td>3</td>
</tr>
<tr>
<td>6-0000001A</td>
<td>R=B</td>
<td>EMXG</td>
<td>72</td>
<td>3</td>
</tr>
<tr>
<td>2-00000000</td>
<td>R=B</td>
<td>EMXG</td>
<td>72</td>
<td>3</td>
</tr>
<tr>
<td>6-0000001A</td>
<td>R=B</td>
<td>EMXG</td>
<td>72</td>
<td>3</td>
</tr>
<tr>
<td>2-00000000</td>
<td>R=B</td>
<td>EMXG</td>
<td>72</td>
<td>3</td>
</tr>
<tr>
<td>6-0000001A</td>
<td>R=B</td>
<td>EMXG</td>
<td>72</td>
<td>3</td>
</tr>
<tr>
<td>2-00000000</td>
<td>R=B</td>
<td>EMXG</td>
<td>72</td>
<td>3</td>
</tr>
<tr>
<td>6-0000001A</td>
<td>R=B</td>
<td>EMXG</td>
<td>72</td>
<td>3</td>
</tr>
<tr>
<td>2-00000000</td>
<td>R=B</td>
<td>EMXG</td>
<td>72</td>
<td>3</td>
</tr>
<tr>
<td>6-0000001A</td>
<td>R=B</td>
<td>EMXG</td>
<td>72</td>
<td>3</td>
</tr>
<tr>
<td>2-00000000</td>
<td>R=B</td>
<td>EMXG</td>
<td>72</td>
<td>3</td>
</tr>
<tr>
<td>6-0000001A</td>
<td>R=B</td>
<td>EMXG</td>
<td>72</td>
<td>3</td>
</tr>
<tr>
<td>2-00000000</td>
<td>R=B</td>
<td>EMXG</td>
<td>72</td>
<td>3</td>
</tr>
<tr>
<td>6-0000001A</td>
<td>R=B</td>
<td>EMXG</td>
<td>72</td>
<td>3</td>
</tr>
<tr>
<td>2-00000000</td>
<td>R=B</td>
<td>EMXG</td>
<td>72</td>
<td>3</td>
</tr>
</tbody>
</table>

### Notes

- **Address**: The starting address of the variable or array.
- **Type**: The type of the variable or array.
- **Name**: The name of the variable or array.
- **Bytes**: The number of bytes allocated for the variable or array.
- **Dimensions**: The dimensions of the array, if applicable.
LABELS

<table>
<thead>
<tr>
<th>Address</th>
<th>Label</th>
<th>Address</th>
<th>Label</th>
<th>Address</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>**</td>
<td>10</td>
<td>**</td>
<td>20</td>
<td>**</td>
<td>30</td>
</tr>
</tbody>
</table>

FUNCTIONS AND SUBROUTINES REFERENCED

AMAT GAUSS MATAB MINV3 MTHS$DASIN MTHS$DCOS MTHS$DSIN UNIT VMAG

Total Space Allocated = 2997 Bytes

CC = NO QUALIFIERS

FORTRAN /LIST GCP, INDATA, MATAB, OUTDATA, RUNG, DNAV, EPHRM, TRUEA, SPREGS, OCCULT, GPERT, GCPSEQ, VISIBLE, GENENV, TREG, GYROUT, RATE, BMAT, CMA

/NOCHECK /NOBOUNDS, OVERFLOW
/DEBUG /NOSYMCS, TRACEMIM
/ONLY /NODR_FLOATING /14 /OPTIMIZE /WARNINGS /NOD_LINES /NOMACHINE_CODE /CONTINUATIONS=19

COMPILATION STATISTICS

Run Time: 2.60 seconds
Elapsed Time: 35.98 seconds
Page Faults: 400
Dynamic Memory: 160 pages
2.7 Perform State Estimation (EST)

EST incorporates the latest GPS, star tracker, and GCP measurements into an estimate of the spacecraft state using an extended Carrión Square Root filter. The extended filter incorporates propagation of the state, the state transition matrix, and the process noise through integration of non linear differential equations. The square root filter has been selected to overcome numerical difficulties associated with small spacecraft computers.

The process, illustrated in Figure A-35, begins by constructing a single composite state out of navigation and attitude states. The two portions of the state were separated because they are propagated in dramatically different ways. The navigator state, \( X \), is propagated through non-linear integration of the equations of motion while the attitude state, \( E \), is propagated by the gyros. Similarly the composite state transition and covariance matrices are constructed from the attitude and position components. The last composite array to be found is the process noise. However, the attitude portion of the process noise array is loaded with zeros because it was previously considered in gyro processing. After the composite process noise array has been loaded, the navigation process noise array is reinitialized to zero for the next measurement interval.

Following construction of the composite arrays, the covariance matrix is propagated forward to current time by the subroutine PROP. PROP uses the state transition matrix \( \Phi \) to perform this function. PROP also reinitializes the state transition matrix to the identity matrix, and EST loads the independent attitude and navigation state transition matrices.

The measurement partials and the estimated measurement are computed by MPART. MPART uses the measurement code to determine which partials to generate. These partials are used along with the measurement residual, to estimate the state of the spacecraft. This process is performed in a systematic manner. First, the measurement residual is computed by subtracting the estimated measurement vector from the true measurement vectors. Each component of the measurement residual is used sequentially in a two step process to update the state and covariance and to reload the old covariance. It is necessary to reload the old covariance between each update because the original old covariance is no longer valid after one update.

Following state update, the navigation and attitude states are loaded with the new state estimate. The attitude quaternions are unitized to insure orthogonality, and the various transformation matrices are computed.

The final procedure in EST is to insure that the attitude covariance does not drop below some minimum required to insure stability. If the covariance were allowed to converge to zero, new measurements would be ignored and the state estimate would diverge.
STATE ESTIMATION MODULE (EST)

FORM COMPOSIT STATE TRANSITION MATRIX

FORM COMPOSIT PROCESS NOISE ARRAY

PROPAGATE COVARIANCE ARRAY

COMPUTE MEASUREMENT PARTIALS

DO CASE EVENT CODE

1 2 3 4

1. COMPUTE LANDMARK MEASUREMENT RESIDUALS
2. COMPUTE GPS MEASUREMENT RESIDUALS
3. COMPUTE S/T 1 MEASUREMENT RESIDUALS
4. COMPUTE S/T 2 MEASUREMENT RESIDUALS

DEF

NULL

UPDATE STATE ESTIMATE

UNITIZE QUATERNIONS

COMPUTE A AND C MATRICES

BOUND ATTITUDE COVARIANCE ARRAY

Figure A-35
SUBROUTINE EST

SUBROUTINE EST REPLACES SUBROUTINE ESTIMAT BY PERFORMING THE
ESTIMATION PROCESS USING THE MEASUREMENTS MADE BY THE
LANDMARK TRACKER, THE GPS, AND STAR TRACKERS 1 AND 2.

THE ESTIMATION STATE VECTOR HAS BEEN AUGMENTED TO 10 WHILE
THERE ARE 12 CONSIDER PARAMETERS FOR A TOTAL COMBINED STATE
OF 22.

INPUT PARAMETERS
PN = STATE TRANSITION MATRIX INITIAL CONDITIONS
PHIN = NAVIGATION STATE TRANSITION MATRIX INTEGRATED
       TO THIS TIME POINT
PHIA = ATTITUDE STATE TRANSITION MATRIX INTEGRATED
       TO THIS TIME POINT
Q = NAVIGATION PROCESS NOISE
MCODE = FLAG DETERMINING THE MEASUREMENTS TO BE USED
DHCL = HORIZONTAL LANDMARK MEASUREMENT
DVCL = VERTICAL LANDMARK MEASUREMENT
PDHL = PARTIAL OF HORIZONTAL LANDMARK MEASUREMENT
PDVL = "  " VERTICAL "  "
SIGGCP = STANDARD DEVIATION OF LMT POSITION
         MEASUREMENT
MX = GPS MEASUREMENT VECTOR
PON = OLD NAVIGATION COVARIANCE
POA = OLD ATTITUDE COVARIANCE
E tc

REAL*8 VDOT.PHDX,PDVDX

COMMON BLOCKS
INCLUDE 'DEBUG.COM'

USER CONTROLLED PARAMETERS TO VARY DEBUG PRINT LEVEL
INTER IF 1, PRINTS WHEN ENTERS MOST SUBROUTINES
IDEBUG 0-10, HIGHER NUMBER MEANS MORE PRINT

INCLUDE 'TARGETS.COM'

COMMON /TARGETS/ MTYPE,MT,NS,ITFLAG,MCODE,PI,TPI

MEASUREMENT SPECIFICATIONS

COVA(16,16), POA(16,16), QMAX

REAL*8 PA, TA, POA, PHIA, COVA, POA, QMAX

 THESE ARE THE ATTITUDE TRANSITION AND COVARIANCE MATRICES

PA ATTITUDE STATE TRANSITION MATRIX
TA PARAMETER TRANSITION MATRIX
POA DERIVATIVE OF TRANSITION MATRICES
PHIA AGGREGATE TRANSITION MATRIX
COVA NEW COVARIANCE MATRIX
POA PREVIOUS COVARIANCE MATRIX
QMAX COVARIANCE NORM MAX

INCLUDE 'GAIN.COM'

COMMON /GAIN/GAIN(22)
REAL*8 GAIN

INCLUDE 'RESIDUALS.COM'

RESIDUALS.COM CONTAIN THE RESIDUALS VALUES FOR THE GPS,
AND STAR TRACKER MEASUREMENTS

COMMON /RESIDUALS/DZHLM, DZVLM, DZHST1, DZVST1, DZHST2, DZVST2,
DZMPS(6)
REAL*8 DZHLM, DZVLM, DZHST1, DZVST1, DZHST2, DZVST2, DZMPS

INCLUDE 'PHIN.COM'

COMMON /PHIN/ PN(6,6), PON(6,6), PHIN(6,6), COVN(6,6),
PON(6,6)
REAL*8 PN, PON, PHIN, COVN, PON

 THESE ARE THE NAVIGATION TRANSITION AND COVARIANCE ARRAYS

PN POSITION STATE TRANSITION MATRIX
PDN DERIVATIVE OF TRANSITION MATRIX
PHIN AGGREGATE TRANSITION MATRIX
COVN NEW COVARIANCE MATRIX
PON PREVIOUS COVARIANCE MATRIX

INCLUDE 'UPDT.COM'

COMMON /UPDT/QN(6), QA(16), Q(6,6), QQOT(6,6)
REAL*8 QN, QA, Q, QQOT

STATE ESTIMATION PARAMETERS

QN NAV. DYN. NOISE COVARIANCE DIAGONAL
QA MIN. VALUES FOR ATT. COVARIANCE DIAGONAL
Q CONTRIBUTION TO NAV. COV. FOR DYN. NOISE
QQOT DIFFERENTIAL OF Q
INCLUDE 'ASTATE.COM'

COMMON /STATE/ DE(4), E(4), WD(3), SF(3), D(3), O(3)
REAL*8 DE, E, WD, SF, D, DD

ATTITUDE STATE AND CONSIDERED PARAMETERS

D DIFFERENTIAL OF QUATERNIONS
E QUATERNIONS
WD GYRO ORBIT RATE (RAD/SEC)
SF GYRO SCALE FACTOR
D GYRO NON-ORTHOGONALITY (RAD)
DO GYRO RELATIVE ORIENTATION (RAD)

INCLUDE 'NSTATE.COM'

COMMON /NSTATE/ XD(6), X(6), RADM, RAE
REAL*8 XD, X, RADM, RAE

POSITION STATE AND CONSIDERED PARAMETERS

XD STATE DERIVATIVES (KM/SEC AND KM/SEC/SEC)
X STATE POSITION PARAMETERS (KM AND KM/SEC)
RAOM RADIUS OF THE MOON (KM)
RAE EARTH DETECTABLE RADIUS (KM)

INCLUDE 'TIME.COM'

COMMON /TIME/ TIME, TNEXT, TSTOP, TIA, DEL, TIN, DTN, DATE0, TZERO

REAL*8 TIME, TNEXT, TSTOP, TIA, DEL, TIN, DTN, DATE0, TIME, TIA, TIN, TSTOP, TIA, DEL, TIN, DTN, DATE0, TIME, 

THESE ARE THE TIME REFERENCE FRAMES

TIME ATOMIC TIME SINCE INITIALIZATION (SEC)
TNEXT TIME FOR NEXT POSITION INTEGRATION (SEC)
TSTOP RUN TERMINATION TIME (SEC)
TIA ATTITUDE INTEGRATION TIME (SEC)
D L " STEP SIZE (SEC)
TIN POSITION INTEGRATION TIME (SEC)
DTN " STEP SIZE (SEC)
DATE0 DATE OF FLIGHT EPOCH (JD)
DATER DATE OF 1950 EPOCH (JD)
TZERO START TIME IN SECS. SINCE DATE0
TSLW TIME NEEDED TO SLEW AND ACQUIRE (SEC)
TIS REAL WORLD REFERENCE TIME (SEC)
TISN TIME FOR NEXT RW POSITION INTEGRATION (SEC)
DTA USUALLY + DEL BUT + TSLW - TIA WHEN DEL

TOO LARGE AT MEASUREMENT TIME

TPRINT TIME QUAD PRINT (SEC)
DTPRINT INCREMENTS ON TPRINT (SEC)

INCLUDE 'TMAT.COM'

COMMON /TMAT/ A(3,3), B(3,3), C(3,3), EM(4,3)
REAL*8 A, B, C, EM
TRANSFORMATION MATRICES

A  INERTIAL TO BODY AXES
B  GYRO TO BODY AXES
C  GYRO NON-ORTHOGONAL TO GYRO AXES
D  BODY TO QUATERNIAN AXES

INCLUDE 'FILT.R.COM'
COMMON/ FILTER/ IPN(6), IPA(6)  IIPN WAS IPN(11)  JACK

FILTER DATA CONSTANTS
COMMON/ FILTER/ IPN(6), IPA(6)
IPN  ARRAY INDEX OF ESTIMATED POS PARAMETERS
IPA  " "  " "  ATT PARAMETERS

INCLUDE 'COMPOSIT.COM'
COMMON/COMPOSIT/ PHI(22,22), OT(22,22), COV(22,22), PO(22,22).
IP(22), XT(22), P(22,22)

REAL8 PHI, QT, COV, PO, XT, P

PHI  COMPOSIT STATE TRANSITION MATRIX
QT  " "  PROCESS NOISE ARRAY
COV  " "  NEWEST COVARIANCE ARRAY
PO  " "  OLD COVARIANCE ARRAY
IP  " "  ARRAY OF FLAGS INDICATING ESTIMATED AND
    " "  CONSIDERED PARAMETERS
XT  " "  COMPOSIT ESTIMATED PLUS CONSIDERED
P  " "  STATE VECTOR

INCLUDE 'ARRAYS.COM'
COMMON/ ARRAYS/ T1(3), T2(3), T3(3), T4(10), T11(3,3), T33(3,3)
    T4(4,4), T66(6,6), T77(7,7), T5(5,5), T6(6,6), T7(7,7)
REAL8 T1, T2, T3, T4, T11, T33, T4, T66, T77, T5, T6, T7

INCLUDE 'PART.COM'
COMMON/ PART/ PX(22), PY(22), PZ(22), PXD(22), PYD(22), PZD(22).
PDH(22,22), PDV(22,2), PDVL(22)
REAL8 PX, PY, PZ, PXD, PYD, PZD, PDH, PDV, PDVL

PARTIALS OF THE RESPECTIVE MEASUREMENTS MADE
FOR GPS
PX  " "  PARTIALS OF X POSITION MEASUREMENT
PY  " "  " "  Y
PZ  " "  " "  Z
PXD  " "  " "  X VELOCITY
PYD  " "  " "  Y
MEASUREMENT OUTPUT PARAMETERS

MA = POSITION/VELOCITY STATE MEASUREMENT - GPS
(KM, KM/SEC)

EMXG = ESTIMATED POSITION/VELOCITY STATE
MEASUREMENT - GPS (KMSEC**2)

RGP(6,8) = MEASURED POSITION/VELOCITY STATE
MEASUREMENT - GPS (KM, KM/SEC**2)

DMCS = ESTIMATED STAR MEASUREMENT HORIZONTAL
DEVIATION FROM BORESIGHT - START (RAD)

DVCS = ESTIMATED STAR MEASUREMENT VERTICAL
DEVIATION FROM BORESIGHT - START (RAD)

EDHS = ESTIMATED LANDMARK MEASUREMENT HORIZONTAL
DEVIATION FROM BORESIGHT (RAD)

EDVL = ESTIMATED LANDMARK MEASUREMENT VERTICAL
DEVIATION FROM BORESIGHT (RAD)

LS = LANDMARK MEASUREMENT UNIT VECTOR (SECOND
SUBSCRIPT REFERS TO TRACKER) - START

LS = LANDMARK MEASUREMENT UNIT VECTOR (SECOND
SUBSCRIPT REFERS TO TRACKER) - START

DMCL = LANDMARK MEASUREMENT HORIZONTAL
DEVIATION FROM BORESIGHT - LANDMARK (RAD)

DVCL = LANDMARK MEASUREMENT VERTICAL
DEVIATION FROM BORESIGHT - LANDMARK (RAD)

EDHL = ESTIMATED LANDMARK MEASUREMENT HORIZONTAL
DEVIATION FROM BORESIGHT (RAD)

EDVL = ESTIMATED LANDMARK MEASUREMENT VERTICAL
DEVIATION FROM BORESIGHT (RAD)

LMU = LANDMARK MEASUREMENT UNIT VECTOR - LANDMARK

RL = LANDMARK MEASUREMENT NOISE COVARIANCE
(KNOWLEDGE) - LANDMARK (RAD**2)

AL = ALTITUDE OF LANDMARK (KM)

LON = LONGITUDE OF LANDMARK (DEG)
LAT = LATITUDE OF LANDMARK (DEG)
O1100  C  TNL = ORIENTATION ARRAY FOR LANDMARK TRACKER
O1200  C  N  TNL = MISALIGNMENT ARRAY - ACTUAL
O1300  C  TRK = TRACER TO NOMINAL
O1400  C  BL = BIAS - ACTUAL (RAD)
O1500  C  SL = NOISE STANDARD DEVIATION - ACTUAL (RAD)
O1600  C  BK = BIAS - KNOWLEDGE (RAD)
O1700  C  THE = LOOK ANGLE (RAD)
O1800  C  SKL = NOISE STANDARD DEVIATION - KNOWLEDGE (RAD)
O1900  C  TIE = INITIAL EARTH FIXED TO INERTIAL
O2000  C  TNLK = MISALIGNMENT ARRAY KNOWLEDGE
O2100  C  TRACER TO NOMINAL
O2200  C  SIGGC = POSITION UNCERTAINTY DUE TO CLOUDS
O2300  C  BAD LAC = POSITION UNCERTAINTY DUE TO CLOUDS
O2400  C
O2500  C
O2600  C
O2700  C
O2800  C

C IF (ENTER .EQ. 0) WRITE (6, 1) NCODE
O2900  0253
O3000  0254

C CCONSTRUCT COMPOSIT STATE TRANSITION MATRIX AND OLD COVARIANCE
O3100  C
O3200  C
O3300  C
O3400  C
O3500  C
O3600  C
O3700  C
O3800  C

C DO 3 I=1,6
O3900  0355
O4000  0356
O4100  0357
O4200  0358
O4300  0359
O4400  0360
O4500  0361
O4600  0362
O4700  0363
O4800  0364
O4900  0365
O5000  0366
O5100  0367
O5200  0368
O5300  0369
O5400  0370
O5500  0371
O5600  0372
O5700  0373
O5800  0374
O5900  0375
O6000  0376
O6100  0377
O6200  0378
O6300  0379
O6400  0380
O6500  0381
O6600  0382
O6700  0383
O6800  0384
O6900  0385
O7000  0386
O7100  0387
O7200  0388
O7300  0389
O7400  0390
O7500  0391
O7600  0392
O7700  0393
O7800  0394
O7900  0395
O8000  0396
O8100  0397
O8200  0398
O8300  0399
O8400  0400
O8500  0401
O8600  0402
O8700  0403
O8800  0404
O8900  0405
O9000  0406
O9100  0407
O9200  0408
O9300  0409

C 0.016
C 0.017
C 0.018
C 0.019
C 0.020
C 0.021
C 0.022
C 0.023
C 0.024
C 0.025
C 0.026
C 0.027
C 0.028
C 0.029
C 0.030
C 0.031
C 0.032
C 0.033
C 0.034
C 0.035
C 0.036
C 0.037
C 0.038
C 0.039
C 0.040
C 0.041
C 0.042
C 0.043
C 0.044
C 0.045
C 0.046
C 0.047
C 0.048
C 0.049
C 0.050
C 0.051
C 0.052
C 0.053
C 0.054
C 0.055
C 0.056
C 0.057
C 0.058
C 0.059
C 0.060
C 0.061
C 0.062
C 0.063
C 0.064
C PROPAGATE COVARIANCE ARRAY
10200 C
10300 0081 C
10400 C
10500 C
10600 C
10700 C
10800 C
10900 0082 DO 80 I=1,6
11000 0083 DO 80 J=1,6
11100 0084 80 PN(I,J)=P(I,J)
11200 0085 DO 90 I=7,10
11300 0086 DO 90 J=7,10
11400 0087 90 PA(I-6,J-6)=P(I,J)
11500 C
11700 C
11800 C
11900 C
12000 0089 CALL MPART
12100 C
12200 C
12300 C
12400 C
12500 C
12600 0089 GO TO (100,200,300,400). MCODE
12700 0090 RETURN
12800 C
12900 C
13000 C
13100 C
13200 C
13300 0091 100 DZHLM=DHCL-EDHL
13400 0092 DZVL=DVCL-EDVL
13500 0093 IF(IDEBUG .LT. 4) GO TO 170
13600 0094 WRITE(6,150) DZHLM,DHCL,EDHL
13700 0095 WRITE(6,151) DZVL,DVCL,EDVL
13800 0096 150 FORMAT(' DZHLM,DHCL,EDHL ','E20.10')
13900 0097 151 FORMAT(' DZVL,DVCL,EDVL ','E20.10')
14000 0098 WRITE(6,155)
14100 0099 C
14200 0100 WRITE(6,152) (PDHL(I),I=1,10)
14300 0101 WRITE(6,153)
14400 0102 WRITE(6,154)
14500 0103 WRITE(6,152) (XT(I),I=1,10)
14600 0104 152 FORMAT('SX,SE15.7')
14700 0105 155 FORMAT('PDHL')
14800 0106 153 FORMAT('PDVL')
14900 0107 154 FORMAT('XT')
C MAKE UPDATE - LANDMARK TRACKER

C CALL UPDATE(22,10,XT,COV,PO,PDVL,RL(1,1),DZVLM,IP)
DO 160 I=1,22
DO 160 J=1,22
160 PO(I,J)=COV(I,J)
CALL UPDATE(22,10,XT,COV,PO,PDVL,RL(2,2),DZVLM,IP)
GO TO 600

C CALCULATE ACTUAL-PREDICTED MEASUREMENT - GPS

C DO 205 I=1,6
DO 210 ....
Z1=GAIN(I) CALL UPDATE(22.10.XT,COV,PO,PX,RGPS(I,I),DXMGPS(I),P)
DO 220 ....
Z2=GAIN(2) CALL UPDATE(22.10.XT,COV,PO,PZ,RGPS(2,2),DXMGPS(2),IP)
DO 230 ....
Z3=GAIN(3) CALL UPDATE(22.10.XT,COV,PO,PX,RGPS(3,3),DXMGPS(3),IP)
DO 240 ....
Z4=GAIN(4) CALL UPDATE(22.10.XT,COV,PO,PX,RGPS(4,4),DXMGPS(4),IP)
DO 250 ....
Z5=GAIN(5) CALL UPDATE(22.10.XT,COV,PO,PX,RGPS(5,5),DXMGPS(5),IP)
DO 260 ....
Z6=GAIN(6) CALL UPDATE(22.10.XT,COV,PO,PX,RGPS(6,6),DXMGPS(6),IP)
WRITE(13,599)TIME,Z1,Z2,Z3,Z4,Z5,Z6
599 FORMAT(7F14.7)
GO TO 600

C MAKE UPDATE - STAR TRACKER 1

C CALL UPDATE(22.10.XT,COV,PO,PDVL,RL(1,1),DZVLM,IP)
DO 160 I=1,22
DO 160 J=1,22
160 PO(I,J)=COV(I,J)
CALL UPDATE(22,10,XT,COV,PO,PDVL,RL(2,2),DZVLM,IP)
GO TO 600
**C**
20700 0146 300 DZHST1=DHCS(1)-EDHS(1)
20900 0147 DZVST1=DVCS(1)-EDVST1
21200 **C**
21300 **C** MAKE UPDATE - STAR TRACKER 1
21400 **C**
21500 0148 CALL UPDATE(22,10,XT,COV,PG,PDHS(1,1),RS(1,1,1),DZHST1,IP)
21600 0149 DO 310 I=1,22
21700 0150 DO 310 J=1,22
21800 0151 310 PO(I,J)=COV(I,J)
21900 0152 CALL UPDATE(22,10,XT,COV,PG,PDVS(1,1),RS(2,2,1),DZVST1,IP)
22000 0153 GO TO 600
22100 **C**
22200 **C**
22300 **C** MAKE UPDATE - STAR TRACKER 2
22400 **C**
22500 **C**
22600 0154 400 DZHST2=DHCS(2)-EDHS(2)
22700 0155 DZVST2=DVCS(2)-EDVST2
22800 **C**
22900 **C** MAKE UPDATE - STAR TRACKER 2
23000 **C**
23100 **C**
23200 0156 CALL UPDATE(22,10,XT,COV,PG,PDHS(1,2),RS(1,1,2),DZHST2,IP)
23300 0157 DO 410 I=1,22
23400 0158 DO 410 J=1,22
23500 0159 410 PO(I,J)=COV(I,J)
23600 0160 CALL UPDATE(22,10,XT,COV,PG,PDVS(1,2),RS(2,2,2),DZVST2,IP)
23700 **C**
23800 **C** FINISH
23900 **C**
24000 **C**
24100 **C**
24200 **C**
24300 0161 800 DO 650 I=1,6
24400 0162 850 X(I)=XT(I)
24500 0163 DO 700 I=7,10
24600 0164 700 E(I-6)*XT(I)
24700 0165 CALL UNIT(E,E,4)
24800 0166 CALL AMAT(E,A)
24900 0167 CALL CMAT(SF,D,C)
25000 **C**
25100 **C**
25200 **C**
25300 **C**
25400 **C**
25500 0168 CALL CRAISE(16,COVA,QA,IPA)
25600 0169 IF(IDEBUG.GT.1) WRITE(6,800) X,E
25700 0170 800 FORMAT(16X,EXITING FROM EST 'X= ',6E20.10,/,X=' ',4E20.10)
25800 **C**
25900 0171 IF(IDEBUG.GT.1) WRITE(6,900) (GAIN(I),I=1,10)
26000 0172 900 FORMAT(16X,GAIN ' ',5E15.7,/,15X,5E15.7)
26100 0173 RETURN
26200 0174 END
### PROGRAM SECTIONS

<table>
<thead>
<tr>
<th>Name</th>
<th>Bytes</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 $CODE</td>
<td>1631</td>
<td>PIC CON REL LCL</td>
</tr>
<tr>
<td>1 $DATA</td>
<td>232</td>
<td>PIC CON REL LCL</td>
</tr>
<tr>
<td>2 $LOCAL</td>
<td>664</td>
<td>PIC CON REL LCL NONOEKE</td>
</tr>
<tr>
<td>3 DEBUG</td>
<td>8</td>
<td>PIC OVR REL GBL</td>
</tr>
<tr>
<td>4 TARGETS</td>
<td>36</td>
<td>PIC OVR REL GBL</td>
</tr>
<tr>
<td>5 PHIL</td>
<td>7176</td>
<td>PIC OVR REL GBL</td>
</tr>
<tr>
<td>6 GAIN</td>
<td>176</td>
<td>PIC OVR REL GBL</td>
</tr>
<tr>
<td>7 RESIDUALS</td>
<td>96</td>
<td>PIC OVR REL GBL</td>
</tr>
<tr>
<td>8 PHIN</td>
<td>1440</td>
<td>PIC OVR REL GBL</td>
</tr>
<tr>
<td>9 UPDT</td>
<td>752</td>
<td>PIC OVR REL GBL</td>
</tr>
<tr>
<td>10 $STATE</td>
<td>160</td>
<td>PIC OVR REL GBL</td>
</tr>
<tr>
<td>11 NSTATE</td>
<td>112</td>
<td>PIC OVR REL GBL</td>
</tr>
<tr>
<td>12 TIME</td>
<td>136</td>
<td>PIC OVR REL GBL</td>
</tr>
<tr>
<td>13 TMAT</td>
<td>312</td>
<td>PIC OVR REL GBL</td>
</tr>
<tr>
<td>14 FILTER</td>
<td>88</td>
<td>PIC OVR REL GBL</td>
</tr>
<tr>
<td>15 COMPOSIT</td>
<td>19624</td>
<td>PIC OVR REL GBL</td>
</tr>
<tr>
<td>16 ARRAYS</td>
<td>1096</td>
<td>PIC OVR REL GBL</td>
</tr>
<tr>
<td>17 PART</td>
<td>2112</td>
<td>PIC OVR REL GBL</td>
</tr>
<tr>
<td>18 MEASOUT</td>
<td>648</td>
<td>PIC OVR REL GBL</td>
</tr>
<tr>
<td>19 LMTPAR</td>
<td>392</td>
<td>PIC OVR REL GBL</td>
</tr>
</tbody>
</table>

### ENTRY POINTS

<table>
<thead>
<tr>
<th>Address Type Name</th>
<th>Address Type Name</th>
<th>Address Type Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-00000000</td>
<td>EST</td>
<td></td>
</tr>
</tbody>
</table>

### VARIABLES

<table>
<thead>
<tr>
<th>Address Type Name</th>
<th>Address Type Name</th>
<th>Address Type Name</th>
<th>Address Type Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>19-00000000</td>
<td>R=B AL</td>
<td>12-0000038</td>
<td>R=B DATED</td>
</tr>
<tr>
<td>10-00001000</td>
<td>R=B DMCL</td>
<td>12-00000000</td>
<td>R=B DTA</td>
</tr>
<tr>
<td>18-00000180</td>
<td>R=B DZVLM</td>
<td>7-000000000</td>
<td>R=B DZHLM</td>
</tr>
<tr>
<td>10-00000000</td>
<td>R=B DZVST1</td>
<td>7-000000000</td>
<td>R=B DZVST2</td>
</tr>
<tr>
<td>19-00000000</td>
<td>R=B EDVY</td>
<td>7-000000000</td>
<td>R=B EDHL</td>
</tr>
<tr>
<td>13-00000019</td>
<td>R=B FIB</td>
<td>7-000000000</td>
<td>R=B FIB</td>
</tr>
<tr>
<td>19-00000008</td>
<td>R=B GDH</td>
<td>0-000000000</td>
<td>R=B GDH</td>
</tr>
<tr>
<td>10-00000000</td>
<td>R=B HDVX</td>
<td>0-000000000</td>
<td>R=B HDVX</td>
</tr>
<tr>
<td>11-00000000</td>
<td>R=B RAD</td>
<td>0-000000000</td>
<td>R=B RAD</td>
</tr>
<tr>
<td>12-00000000</td>
<td>R=B REA</td>
<td>0-000000000</td>
<td>R=B REA</td>
</tr>
<tr>
<td>12-00000000</td>
<td>R=B TPS</td>
<td>0-000000000</td>
<td>R=B TPS</td>
</tr>
<tr>
<td>12-00000000</td>
<td>R=B TPRINT</td>
<td>0-000000000</td>
<td>R=B TPRINT</td>
</tr>
<tr>
<td>0-00000000</td>
<td>R=B VDOT</td>
<td>0-000000000</td>
<td>R=B VDOT</td>
</tr>
<tr>
<td>2-00000000</td>
<td>R=B Z4</td>
<td>2-000000000</td>
<td>R=B Z4</td>
</tr>
</tbody>
</table>

### Variables in hexadecimal format

- AL: 00000038
- DATED: 00000070
- DTA: 00000030
- DZHLM: 00000000
- DZVST1: 00000028
- I: 0000014
- L: 00000000
- MCODE: 00000014
- OMAX: 00000001
- P: 00000000
- RAD: 00000000
- REA: 00000000
- TPS: 00000000
- TPRINT: 00000000
- VDOT: 00000020
- Z4: 00000030
<table>
<thead>
<tr>
<th>Address</th>
<th>Type</th>
<th>Name</th>
<th>Bytes</th>
<th>Dimensions</th>
</tr>
</thead>
<tbody>
<tr>
<td>13-00000000</td>
<td>R8</td>
<td>A</td>
<td>72</td>
<td>(3, 3)</td>
</tr>
<tr>
<td>13-00000004</td>
<td>R8</td>
<td>B</td>
<td>72</td>
<td>(3, 3)</td>
</tr>
<tr>
<td>19-00000000</td>
<td>R8</td>
<td>BKL</td>
<td>16</td>
<td>(2)</td>
</tr>
<tr>
<td>19-00000004</td>
<td>R8</td>
<td>BL</td>
<td>16</td>
<td>(2)</td>
</tr>
<tr>
<td>13-00000009</td>
<td>R8</td>
<td>C</td>
<td>72</td>
<td>(3, 3)</td>
</tr>
<tr>
<td>15-00001540</td>
<td>R8</td>
<td>COV</td>
<td>3872</td>
<td>(22, 22)</td>
</tr>
<tr>
<td>5-00000000</td>
<td>R8</td>
<td>COVA</td>
<td>2048</td>
<td>(16, 16)</td>
</tr>
<tr>
<td>6-00000000</td>
<td>R8</td>
<td>COVN</td>
<td>288</td>
<td>(6, 6)</td>
</tr>
<tr>
<td>10-00000000</td>
<td>R8</td>
<td>D</td>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>10-00000008</td>
<td>R8</td>
<td>DD</td>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>10-00000008</td>
<td>R8</td>
<td>DE</td>
<td>32</td>
<td>(4)</td>
</tr>
<tr>
<td>16-00000180</td>
<td>R8</td>
<td>DMCS</td>
<td>16</td>
<td>(2)</td>
</tr>
<tr>
<td>18-00000010</td>
<td>R8</td>
<td>DVCNS</td>
<td>16</td>
<td>(2)</td>
</tr>
<tr>
<td>10-000000230</td>
<td>R8</td>
<td>DNGPS</td>
<td>48</td>
<td>(6)</td>
</tr>
<tr>
<td>10-000000220</td>
<td>R8</td>
<td>E</td>
<td>96</td>
<td>(4, 3)</td>
</tr>
<tr>
<td>16-00000258</td>
<td>R8</td>
<td>EDMS</td>
<td>16</td>
<td>(2)</td>
</tr>
<tr>
<td>18-00000268</td>
<td>R8</td>
<td>EDVS</td>
<td>16</td>
<td>(2)</td>
</tr>
<tr>
<td>13-0000000</td>
<td>R8</td>
<td>EM</td>
<td>96</td>
<td>(4, 3)</td>
</tr>
<tr>
<td>11-00002228</td>
<td>R8</td>
<td>EMK</td>
<td>148</td>
<td>(6)</td>
</tr>
<tr>
<td>6-00000000</td>
<td>R8</td>
<td>GAIN</td>
<td>176</td>
<td>(22)</td>
</tr>
<tr>
<td>15-00000380</td>
<td>I4</td>
<td>IP</td>
<td>80</td>
<td>(22)</td>
</tr>
<tr>
<td>14-0000001B</td>
<td>I4</td>
<td>IPA</td>
<td>64</td>
<td>(16)</td>
</tr>
<tr>
<td>14-00000000</td>
<td>I4</td>
<td>IPN</td>
<td>24</td>
<td>(6)</td>
</tr>
<tr>
<td>18-00000100</td>
<td>R8</td>
<td>LMIU</td>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>18-000000170</td>
<td>R8</td>
<td>MS</td>
<td>48</td>
<td>(3, 2)</td>
</tr>
<tr>
<td>18-000000200</td>
<td>R8</td>
<td>MX</td>
<td>48</td>
<td>(6)</td>
</tr>
<tr>
<td>15-00000000</td>
<td>R8</td>
<td>P</td>
<td>3872</td>
<td>(22, 22)</td>
</tr>
<tr>
<td>5-00000000</td>
<td>R8</td>
<td>PA</td>
<td>128</td>
<td>(4, 4)</td>
</tr>
<tr>
<td>5-000000200</td>
<td>R8</td>
<td>PDA</td>
<td>512</td>
<td>(16)</td>
</tr>
<tr>
<td>17-00000060</td>
<td>R8</td>
<td>PDMH</td>
<td>176</td>
<td>(22)</td>
</tr>
<tr>
<td>17-00008300</td>
<td>R8</td>
<td>PDMS</td>
<td>352</td>
<td>(22, 2)</td>
</tr>
<tr>
<td>8-000000120</td>
<td>R8</td>
<td>PDN</td>
<td>288</td>
<td>(6, 6)</td>
</tr>
<tr>
<td>17-000000790</td>
<td>R8</td>
<td>PDVL</td>
<td>176</td>
<td>(22)</td>
</tr>
<tr>
<td>17-00000580</td>
<td>R8</td>
<td>PDVS</td>
<td>352</td>
<td>(22, 2)</td>
</tr>
<tr>
<td>15-00000000</td>
<td>R8</td>
<td>PHK</td>
<td>288</td>
<td>(6, 6)</td>
</tr>
<tr>
<td>5-00000040</td>
<td>R8</td>
<td>PHIA</td>
<td>3872</td>
<td>(22, 22)</td>
</tr>
<tr>
<td>8-000000240</td>
<td>R8</td>
<td>PHIN</td>
<td>288</td>
<td>(6, 6)</td>
</tr>
<tr>
<td>6-00000000</td>
<td>R8</td>
<td>PN</td>
<td>288</td>
<td>(6, 6)</td>
</tr>
<tr>
<td>15-00000260</td>
<td>R8</td>
<td>PO</td>
<td>3872</td>
<td>(22, 22)</td>
</tr>
<tr>
<td>5-00000140</td>
<td>R8</td>
<td>PDA</td>
<td>2048</td>
<td>(16, 16)</td>
</tr>
<tr>
<td>8-00000040</td>
<td>R8</td>
<td>PON</td>
<td>288</td>
<td>(6, 6)</td>
</tr>
<tr>
<td>17-00000000</td>
<td>R8</td>
<td>PX</td>
<td>176</td>
<td>(22)</td>
</tr>
<tr>
<td>17-00000210</td>
<td>R8</td>
<td>PKD</td>
<td>176</td>
<td>(22)</td>
</tr>
<tr>
<td>17-00000080</td>
<td>R8</td>
<td>PY</td>
<td>176</td>
<td>(22)</td>
</tr>
<tr>
<td>17-00000200</td>
<td>R8</td>
<td>PYD</td>
<td>176</td>
<td>(22)</td>
</tr>
<tr>
<td>17-00000160</td>
<td>R8</td>
<td>PZ</td>
<td>176</td>
<td>(22)</td>
</tr>
<tr>
<td>17-00000370</td>
<td>R8</td>
<td>PZD</td>
<td>176</td>
<td>(22)</td>
</tr>
<tr>
<td>9-00000000</td>
<td>R8</td>
<td>Q</td>
<td>288</td>
<td>(6, 6)</td>
</tr>
<tr>
<td>9-00000030</td>
<td>R8</td>
<td>QA</td>
<td>128</td>
<td>(16)</td>
</tr>
<tr>
<td>9-00000100</td>
<td>R8</td>
<td>QQOT</td>
<td>288</td>
<td>(6, 6)</td>
</tr>
<tr>
<td>9-00000000</td>
<td>R8</td>
<td>QN</td>
<td>48</td>
<td>(6)</td>
</tr>
<tr>
<td>15-00000020</td>
<td>R8</td>
<td>QT</td>
<td>3872</td>
<td>(22, 22)</td>
</tr>
<tr>
<td>18-00000030</td>
<td>R8</td>
<td>RGPS</td>
<td>288</td>
<td>(6, 6)</td>
</tr>
</tbody>
</table>
### Compilation Statistics

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Run Time:</td>
<td>6.96 seconds</td>
</tr>
<tr>
<td>Elapsed Time:</td>
<td>121.03 seconds</td>
</tr>
<tr>
<td>Page Faults:</td>
<td>438</td>
</tr>
<tr>
<td>Dynamic Memory:</td>
<td>160 pages</td>
</tr>
</tbody>
</table>
2.7.1 Propagate Covariances Module (PROP)

The interdependence of the state parameters cause their uncertainty to grow with time. This growth in uncertainty (covariance matrix) is propagated to the measurement time by using the state transition matrix.

The mathematical equation which is used to update the covariance matrix to the time of the measurement is

\[ P(t_m) = \phi(t_m, t_{m-1}) \cdot P(t_{m-1}) \cdot \phi(t_m, t_{m-1})^T + Q \]

where

- \( P(t_m) \) = covariance matrix at time of measurement.
- \( P(t_{m-1}) \) = covariance matrix after processing previous measurement.
- \( \phi(t_{m-1}, t_{m-1}) \) = state transition matrix from time of previous measurement to time of present measurement.
- \( Q \) = noise matrix to account for model errors.

Figure A-36 illustrates this process.
PROPAGATE COVARIANCE (PROP)

\[
\begin{align*}
I &= 1 \\
\text{DO WHILE } &I \text{ .LT. NUMBER OF ROWS IN } P \text{ (COVARIANCE MATRIX)} \\
\text{COMPUTE } &I\text{th ROW IN } P = \Phi I^2 \Phi^T \\
I &= I + 1 \\
\text{ADD PROCESS NOISE ONTO DIAGONAL ELEMENTS OF } P \text{ MATRIX} \\
\text{STORE } &P \text{ IN NEW COVARIANCE MATRIX} \\
\text{INITIALIZE } &\Phi (STATE TRANSITION MATRIX) \\
&\text{TO ZERO WITH ONE'S ON DIAGONAL}
\end{align*}
\]

Figure A-36
SUBROUTINE PROP(NS,N,PHI,PN,PO,Q,PHIP)
C DIMENSION PHI(N,N),PHI(N,N),Q(N,N),PHIP(N,N),B(22)
C REAL B, PHI, PHIP, PN, PO, Q
C C TOTAL NUMBER OF ALLOWABLE PARAMETERS
C NS TOTAL NUMBER OF STATE PARAMETERS
C PHI STAT TRANSITION MATRIX
C Q PROC SS NOISE
C PN NEW COVARIANCE MATRIX
C PO OLD COVARIANCE MATRIX
C C C
C NOTE: PN AND PO ARE THE SAME AT EXIT AND EQUAL TO
C THE UPDATED COVARIANCE. PO IS DESTROYED
C BY INTERNAL PROCESSING.
C C C
C COVARIANCE PROPAGATION
C INCLUDE 'DEBUG.COM'
C COMMON /DEBUG/ IENTER, IDEBUG
C USER CONTROLLED PARAMETERS TO VARY DEBUG PRINT LEVEL
C IENTER IF 1, PRINTS WHEN ENTERS MOST SUBROUTINES
C IDEBUG 0-10, HIGHER NUMBER MEANS MORE PRINT
C C C
C DO 20 I=1,N
C DO 20 J=1,N
C B(J) = 0.
C DO 10 K=1,N
C 10 B(J) = B(J) + PHI(I,K)*PN(K,J)
C DO 20 L=1,N
C PO(I,L) = 0.
C DO 20 M=1,N
C 20 PO(I,L) = PO(I,L) + B(M) + PHI(L,M)
C DO 30 I=1,N
C DO 30 J=1,N
C IF (I.LE.NS.AND.J.LE.NS) PO(I,J) = PO(I,J) + Q(I,J)
C DO 30 K=1,N
C 30 IF (IDEBUG.GT.2) WRITE(6,25) (Q(I,J),I=1,6,J=1,6)
C IF (IDEBUG.GT.2) WRITE(6,25) (Q(I,J),I=1,6,J=1,6)
C FORMAT((5X,'Q MATRIX IN PROP:/',(10X,6E15.8))
C C INITIALIZE STATE TRANSITION MATRIX
C C C
C DO 40 I=1,N
C ! WAS NS
C DO 40 J=1,N
C ! WAS NS
C PHIP(I,J) = 0.
C DO 40 I=1,N
C 40 PHIP(I,J) = 1.
C DO 50 I = 1,N
C 50 Q(I,J) = 0.
C C
### PROGRAM SECTIONS

<table>
<thead>
<tr>
<th>Name</th>
<th>Bytes</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 SCODE</td>
<td>526</td>
<td>PIC CON REL LCL SHR EXE RD NDWRT LONG</td>
</tr>
<tr>
<td>1 SPDATA</td>
<td>33</td>
<td>PIC CON REL LCL SHR NOEXE RD NDWRT LONG</td>
</tr>
<tr>
<td>2 SLOCAL</td>
<td>308</td>
<td>PIC CON REL LCL NOSHR NOEXE RD WRT QUAD</td>
</tr>
<tr>
<td>3 DEBUG</td>
<td></td>
<td>PIC OVR REL GBL SHR NOEXE RD WRT LONG</td>
</tr>
</tbody>
</table>

### ENTRY POINTS

<table>
<thead>
<tr>
<th>Address Type Name</th>
<th>Address</th>
<th>Type</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0-00000000</td>
<td>PROP</td>
<td></td>
</tr>
</tbody>
</table>

### VARIABLES

<table>
<thead>
<tr>
<th>Address Type Name</th>
<th>Address</th>
<th>Type</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2-00000000</td>
<td>I+4</td>
<td>I</td>
</tr>
<tr>
<td></td>
<td>2-00000000B</td>
<td>I+4</td>
<td>K</td>
</tr>
<tr>
<td></td>
<td>AP-000000004</td>
<td>I+4</td>
<td>NS</td>
</tr>
</tbody>
</table>

### ARRAYS

<table>
<thead>
<tr>
<th>Address Type Name</th>
<th>Address</th>
<th>Type</th>
<th>Dimensions</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2-00000000</td>
<td>R=B</td>
<td>B (22)</td>
</tr>
<tr>
<td></td>
<td>AP-000000000</td>
<td>R=B</td>
<td>PHI (22, 22)</td>
</tr>
<tr>
<td></td>
<td>AP-00000001C</td>
<td>R=B</td>
<td>PHP (22, 22)</td>
</tr>
<tr>
<td></td>
<td>AP-000000010</td>
<td>R=B</td>
<td>PH (22)</td>
</tr>
<tr>
<td></td>
<td>AP-000000014</td>
<td>R=B</td>
<td>PD (22)</td>
</tr>
<tr>
<td></td>
<td>AP-000000018</td>
<td>R=B</td>
<td>Q (22)</td>
</tr>
</tbody>
</table>

### LABELS

<table>
<thead>
<tr>
<th>Address</th>
<th>Label</th>
<th>Address</th>
<th>Label</th>
<th>Address</th>
<th>Label</th>
<th>Address</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>1-00000000</td>
<td>35'</td>
<td></td>
<td>30</td>
<td></td>
<td>40</td>
</tr>
</tbody>
</table>

Total Space Allocated = 875 Bytes
COMMAND QUALIFIERS

FORTRAN /LIST GCP,INDATA,MATA,OUTDATA,RUNG,NAV,EPHEM,TRUEA,SPRESS,OCULT,GPERT,GCPSEQ,VISIBLE,GENENV,TREG,GRYOUT,RATE,EMAT,CHA

/CHECK=(NOBOUNDS,OVERFLOW)
/DEBUG=(NOSYMBOLS,TRACEBACK)
/FT7 /NOG_FLOATING /14 /OPTIMIZE /WARNINGS /NO_LINES /NOWACHINE_CODE /CONTINUATIONS=19

COMPILATION STATISTICS

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Run Time:</td>
<td>1.99s</td>
</tr>
<tr>
<td>Elapsed Time:</td>
<td>29.30s</td>
</tr>
<tr>
<td>Page Faults:</td>
<td>353</td>
</tr>
<tr>
<td>Dynamic Memory:</td>
<td>160 pages</td>
</tr>
</tbody>
</table>

2.7.2 Compute Measurement Partial (MPART)

Subroutine MPART, called by EST, acts as an executive for the computation of the partials required by the Carlson Filter routine in UPDATE. This routine calls the routines HLMT, HGFS, or HSTAR depending upon the measurement to be made. This measurement is dictated by MCODE an integer flag passed through common.
**Figure A-37**

```
<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>DEF</th>
</tr>
</thead>
<tbody>
<tr>
<td>n1</td>
<td>LMT</td>
<td>GPS</td>
<td>ST1</td>
<td>ST2</td>
<td>NULL</td>
<td></td>
</tr>
<tr>
<td></td>
<td>PARTIALS</td>
<td>PARTIALS</td>
<td>PARTIALS</td>
<td>PARTIALS</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

**MPART VCLR**

**DO CASE - MCODE**
SUBROUTINE MPART
C******************************************************************************
C SUBROUTINE MPART GENERATES THE PARTIAL DERIVATIVES OF
C THE MEASUREMENT WITH RESPECT TO THE ESTIMATED STATE.
C IT ACTS AS AN EXECUTIVE CALLING OTHER SUBROUTINES
C TO PERFORM THE NECESSARY FUNCTIONS.
C
C INPUT PARAMETERS
C MCODE = THE EVENT CODE DETERMINING THE
C MEASUREMENT BEING MADE

C OUTPUT PARAMETERS
C THE RESPECTIVE PARTIALS FOR THE MEASUREMENT
C BEING MADE THROUGH COMMON

C SUBROUTINES CALLED
C HLMT TO COMPUTE THE PARTIALS FOR THE LANDMARK
C TRACKER
C HGPS TO COMPUTE THE PARTIALS FOR THE GPS
C HSTAR TO COMPUTE THE PARTIALS FOR THE KTH
C STAR TRACKER

C******************************************************************************
C COMMON BLOCKS
C INCLUDE 'TARG TS.COM'
C COMMON /TARGETS/ MTYPE,IS,NS,JFLAG,MCODE,PI,TPI
C LOGICAL JFLAG
C REAL*8 PI,TPI

C MEASUREMENT SPECIFICATIONS
C MTYPE = MEASUREMENT TYPE
C JFLAG = SET FOR STAR OBSTRUCTION
C MCODE = * MEASUREMENT PROCESSING
C PI = PI
C TPI = 2*PI

C******************************************************************************
C BRANCH TO THE APPROPRIATE MEASUREMENT SUBROUTINE
C******************************************************************************
C GO TO (10,20,30,40),MCODE

C******************************************************************************
C******************************************************************************
C******************************************************************************
C******************************************************************************
C******************************************************************************
C******************************************************************************
C******************************************************************************
C******************************************************************************
C******************************************************************************
C******************************************************************************
C******************************************************************************
MPART

04600 C COMPUTE THE PARTIALS FOR STAR TRACKER NUMBER TWO
04700 C******************************************************************************
04800 0014 40 CALL HSTAR(2)
04900 0015 RETURN
05000 0018 END

PROGRAM SECTIONS

<table>
<thead>
<tr>
<th>Name</th>
<th>Bytes</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>$CODE</td>
<td>59</td>
<td>PIC COM REL LCL SHR EXE RD NOWRT LONG</td>
</tr>
<tr>
<td>$PDATA</td>
<td>8</td>
<td>PIC COM REL LCL SHR NOEXE RD NOWRT LONG</td>
</tr>
<tr>
<td>$LOCAL</td>
<td>16</td>
<td>PIC COM REL LCL NOSHR NOEXE RD WRT LONG</td>
</tr>
<tr>
<td>TARGETS</td>
<td>36</td>
<td>PIC GVR REL GBL SHR NOEXE RD WRT LONG</td>
</tr>
</tbody>
</table>

ENTRY POINTS

<table>
<thead>
<tr>
<th>Address Type Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-00000000 MPART</td>
</tr>
</tbody>
</table>

VARIABLES

<table>
<thead>
<tr>
<th>Address Type Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>3-00000004 I4 1S</td>
</tr>
<tr>
<td>3-00000008 I4 NS</td>
</tr>
<tr>
<td>3-0000000C L4 JFLAG</td>
</tr>
<tr>
<td>3-00000000C L4 MCODE</td>
</tr>
<tr>
<td>3-000000014 R4 PI</td>
</tr>
<tr>
<td>3-000000014 R4 TPI</td>
</tr>
<tr>
<td>3-00000000A 10 20</td>
</tr>
<tr>
<td>0-0000002A 30 40</td>
</tr>
</tbody>
</table>

LABELS

<table>
<thead>
<tr>
<th>Address Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-0000001A 10</td>
</tr>
<tr>
<td>0-00000022 20</td>
</tr>
<tr>
<td>0-0000002A 30</td>
</tr>
<tr>
<td>0-00000032 40</td>
</tr>
</tbody>
</table>

FUNCTIONS AND SUBROUTINES REFERENCED

HGPS HLMT HSTAR

Total Space Allocated = 119 Bytes

COMMAND QUALIFIERS

FORTRAN /LIST GCP,INDATA,MATAB,OUTPUT.DAT,DATA,RUNG,NAV,EPHEM,TRUEA,SPRESS,OCULT.GPRT,GCPSEQ.VISIBLE,GENENV,TREG,GRGUT,RATE,IMAT,CMA
/NOCHECK= (NOBOUNDS,OVERFLOW) /DEBUG= (NOSYMBOLS,TRACEBACK) /F77 /NO_F90 /OPTIMIZE /WARNINGS /MOD_LINES /NOMACHINE_CODE /CONTINUATIONS=19
COMPILATION STATISTICS

Run Time: 0.79 seconds
Elapsed Time: 15.32 seconds
Page Faults: 301
Dynamic Memory: 160 pages
2.7.2.1 Compute GPS Measurement Partials (HGPS)

HGPS, Called by MPART, computes the sensitivity of the GPS measurements to variations in the elements of the estimated state. This routine doesn't require the support of any major routine.

Processing Requirements

The GPS measurement vector is given by

\[ \hat{Z}_G = \begin{bmatrix} \hat{x}, \hat{y}, \hat{z}, \hat{\xi}, \hat{\eta}, \hat{\zeta} \end{bmatrix}^T \]

The sensitivity of this measurement vector to estimated state elements is given by

\[ \frac{\partial \hat{Z}_G}{\partial \hat{x}_i} = 1 \text{ for } \hat{x}_i = \hat{x}, \hat{y}, \hat{z}, \hat{\xi}, \hat{\eta}, \hat{\zeta} \]

= 0 otherwise

HGPS VCLR

<table>
<thead>
<tr>
<th>SET UP ESTIMATED MEASUREMENT VECTOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZERO ARRAYS</td>
</tr>
<tr>
<td>CONVERT TO RAW-VECTOR FORM</td>
</tr>
</tbody>
</table>

Figure A-38
SUBROUTINE HGPS

SUBROUTINE HGPS GENERATES THE SPECIFIC PARTIALS RELATING TO THE GLOBAL POSITIONING SYSTEM MEASUREMENTS.

INPUT PARAMETERS

• NONE

OUTPUT PARAMETERS

PX = X POSITION MEASUREMENT PARTIALS
PY = Y
PZ = Z
PXD = X VELOCITY
PYD = Y
PZD = Z

COMMON BLOCKS

INCLUDE 'PART.COM'

COMMON /PART/ PX(22),PY(22),PZ(22),PXD(22),PYD(22),PZD(22),
           PDHS(22,2),PDVS(22,2),PDHL(22),PDVL(22)
REAL*8 PX,PY,PZ,PXD, PYD, PZD, PDHS, PDVS, PDHL, PDVL

PARTIALS OF THE RESPECTIVE MEASUREMENTS MADE

FOR GPS

PX = PARTIALS OF X POSITION MEASUREMENT
PY = Y
PZ = Z
PXD = X VELOCITY
PYD = Y
PZD = Z

FOR STAR TRACKER K (K IS THE SECOND PARAMETER)

PDHS = PARTIALS OF HORIZONTAL DIFLECTION
PDVS = VERTICAL

FOR LANDMARK TRACKER

PDHL = PARTIALS OF HORIZONTAL DIFLECTION
PDVL = VERTICAL

INCLUDE 'NSTATE.COM'

COMMON /NSTATE/ XD(6),X(6),RADM,RADE
REAL*8 XD,X,RADM,RADE

POSITION STATE AND CONSIDERED PARAMETERS

XD = STATE DERIVATIVES (KM/SEC AND KM/SEC/SEC)
X = STATE POSITION PARAMETERS (KM AND KM/SEC)
RADM = RADIUS OF THE MOON (KM)
RADE = EARTH DETECTABLE RADIUS (KM)

INCLUDE 'MEASOUT.COM'

COMMON /MEASOUT/ MX(6),RGPS(6,6),DMCS(2),DVCS(2),MS(3,2),
           RS(2,2),DHCL, DVCL, LMU(3), RL(2,2), EMXG(6),
           EDMS(2), EDVS(2), EDHL, EDVL
MEASUREMENT OUTPUT PARAMETERS

MX  = POSITION/VELOCITY STATE MEASUREMENT - GPS
EMXG = ESTIMATED POSITION/VELOCITY STATE MEASUREMENT - HGPS
RGPS = STATE MEASUREMENT NOISE COVARIANCE (KM**2,KM/SEC**2)
DHCS = STAR MEASUREMENT HORIZONTAL DEVIATION FROM BORESIGHT - START (RAD)
DVCS = STAR MEASUREMENT VERTICAL DEVIATION FROM BORESIGHT - START (RAD)
EDHS = ESTIMATED STAR MEASUREMENT HORIZONTAL DEVIATION FROM BORESIGHT (RAD)
EDVS = ESTIMATED STAR MEASUREMENT VERTICAL DEVIATION FROM BORESIGHT (RAD)
MS  = STAR MEASUREMENT UNIT VECTOR (SECOND SUBSCRIPT REFERS TO TRACKER) - START
RS  = STAR MEASUREMENT NOISE COVARIANCE (KNOWLEDGE) - START (RAD**2)
DHCL = LANDMARK MEASUREMENT HORIZONTAL DEVIATION FROM BORESIGHT - LAMKT (RAD)
DVCL = LANDMARK MEASUREMENT VERTICAL DEVIATION FROM BORESIGHT - LAMKT (RAD)
EDHL = ESTIMATED LANDMARK MEASUREMENT HORIZONTAL DEVIATION FROM BORESIGHT (RAD)
EDVL = ESTIMATED LANDMARK MEASUREMENT VERTICAL DEVIATION FROM BORESIGHT (RAD)
LMU = LANDMARK MEASUREMENT UNIT VECTOR - LAMKT
RL  = LANDMARK MEASUREMENT NOISE COVARIANCE (KNOWLEDGE) - LAMKT (RAD**2)
04500 0019 20 HG(I, I)=1.
04600 C
04700 C CONFESSION TO ROW VECTOR FORM
04800 C
04900 C
05000 C
05100 0020 DO 30 1=1,10
05200 0021 PX(I)=HG(1,1)
05300 0022 PY(I)=HG(2,1)
05400 0023 PZ(I)=HG(3,1)
05500 0024 PXD(I)=HG(4,1)
05600 0025 PYD(I)=HG(5,1)
05700 0026 30 PZD(I)=HG(6,1)
05800 0027 DO 40 1=11,12
05900 0028 PX(I)=0.
06000 0029 PY(I)=0.
06100 0030 PZ(I)=0.
06200 0031 PXD(I)=0.
06300 0032 PYD(I)=0.
06400 0033 40 PZD(I)=0.
06500 0034 RETURN
06600 0035 END

PROGRAM SECTIONS

<table>
<thead>
<tr>
<th>Name</th>
<th>Bytes</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 $CODE</td>
<td>190</td>
<td>PIC CON REL LCL SHR EXE</td>
</tr>
<tr>
<td>2 $LOCAL</td>
<td>488</td>
<td>PIC CON REL LCL MOSHR NOEXE RD WRT QRD</td>
</tr>
<tr>
<td>3 PART</td>
<td>2112</td>
<td>PIC OVR REL GBL SHR NOEXE RD WRT QRD</td>
</tr>
<tr>
<td>4 MSTATE</td>
<td>112</td>
<td>PIC OVR REL GBL SHR NOEXE RD WRT QRD</td>
</tr>
<tr>
<td>5 MEASOUT</td>
<td>648</td>
<td>PIC OVR REL GBL SHR NOEXE RD WRT QRD</td>
</tr>
</tbody>
</table>

ENTRY POINTS

<table>
<thead>
<tr>
<th>Address Type Name</th>
<th>Address Type Name</th>
<th>Address Type Name</th>
<th>Address Type Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-0000000000</td>
<td>HGPS</td>
<td>5-00000148 R=8</td>
<td>DVCL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5-00000148 R=8</td>
<td>DVCL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5-000000278 R=8</td>
<td>EDML</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5-000000280 R=8</td>
<td>EDVL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4-000000068 R=8</td>
<td>RADE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4-000000068 R=8</td>
<td>RADM</td>
</tr>
</tbody>
</table>

VARIABLES

<table>
<thead>
<tr>
<th>Address Type Name</th>
<th>Address Type Name</th>
<th>Address Type Name</th>
<th>Address Type Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>5-00000148 R=8</td>
<td>5-00000148 R=8</td>
<td>5-00000278 R=8</td>
<td>5-00000280 R=8</td>
</tr>
<tr>
<td>2-00000148 I*4</td>
<td>2-00000148 I*4</td>
<td>4-00000068 R=8</td>
<td>4-00000068 R=8</td>
</tr>
</tbody>
</table>
**ARRAYS**

<table>
<thead>
<tr>
<th>Address</th>
<th>Type</th>
<th>Name</th>
<th>Bytes</th>
<th>Dimensions</th>
</tr>
</thead>
<tbody>
<tr>
<td>5-000000150</td>
<td>R+B</td>
<td>DHCS</td>
<td>16</td>
<td>(2)</td>
</tr>
<tr>
<td>5-000000160</td>
<td>R+B</td>
<td>DVCS</td>
<td>16</td>
<td>(2)</td>
</tr>
<tr>
<td>5-000000258</td>
<td>R+B</td>
<td>EDHS</td>
<td>16</td>
<td>(2)</td>
</tr>
<tr>
<td>5-000000268</td>
<td>R+B</td>
<td>EDVS</td>
<td>16</td>
<td>(2)</td>
</tr>
<tr>
<td>5-000000228</td>
<td>R+B</td>
<td>EMXG</td>
<td>48</td>
<td>(6)</td>
</tr>
<tr>
<td>2-000000000</td>
<td>R+B</td>
<td>MG</td>
<td>480</td>
<td>(6, 10)</td>
</tr>
<tr>
<td>5-000000170</td>
<td>R+B</td>
<td>LMU</td>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>5-000000000</td>
<td>R+B</td>
<td>MS</td>
<td>48</td>
<td>(3, 2)</td>
</tr>
<tr>
<td>3-000000560</td>
<td>R+B</td>
<td>PDHL</td>
<td>176</td>
<td>(22)</td>
</tr>
<tr>
<td>3-000000420</td>
<td>R+B</td>
<td>PDMS</td>
<td>352</td>
<td>(22, 2)</td>
</tr>
<tr>
<td>3-000000790</td>
<td>R+B</td>
<td>PDVL</td>
<td>176</td>
<td>(22)</td>
</tr>
<tr>
<td>3-000000580</td>
<td>R+B</td>
<td>PDVS</td>
<td>352</td>
<td>(22, 2)</td>
</tr>
<tr>
<td>3-000000210</td>
<td>R+B</td>
<td>PX</td>
<td>176</td>
<td>(22)</td>
</tr>
<tr>
<td>3-000000880</td>
<td>R+B</td>
<td>PY</td>
<td>176</td>
<td>(22)</td>
</tr>
<tr>
<td>3-000000220</td>
<td>R+B</td>
<td>PYD</td>
<td>176</td>
<td>(22)</td>
</tr>
<tr>
<td>3-000000160</td>
<td>R+B</td>
<td>PZ</td>
<td>176</td>
<td>(22)</td>
</tr>
<tr>
<td>3-000000370</td>
<td>R+B</td>
<td>PZD</td>
<td>176</td>
<td>(22)</td>
</tr>
<tr>
<td>5-000000030</td>
<td>R+B</td>
<td>RGPS</td>
<td>288</td>
<td>(6, 6)</td>
</tr>
<tr>
<td>5-000000208</td>
<td>R+B</td>
<td>RL</td>
<td>32</td>
<td>(2, 2)</td>
</tr>
<tr>
<td>5-000001400</td>
<td>R+B</td>
<td>RS</td>
<td>64</td>
<td>(2, 2, 2)</td>
</tr>
<tr>
<td>4-000000030</td>
<td>R+B</td>
<td>X</td>
<td>48</td>
<td>(6)</td>
</tr>
<tr>
<td>4-000000000</td>
<td>R+B</td>
<td>XD</td>
<td>48</td>
<td>(6)</td>
</tr>
</tbody>
</table>

**LABELS**

<table>
<thead>
<tr>
<th>Address</th>
<th>Label</th>
<th>Address</th>
<th>Label</th>
<th>Address</th>
<th>Label</th>
<th>Address</th>
<th>Label</th>
<th>Address</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>**</td>
<td>5</td>
<td>**</td>
<td>10</td>
<td>**</td>
<td>20</td>
<td>**</td>
<td>30</td>
<td>**</td>
<td>40</td>
</tr>
</tbody>
</table>

Total Space Allocated = 3550 Bytes

**COMMAND QUALIFIERS**

```
FORTRAN /LIST GCP,INDATA,MATAOUT,OUTDATA,RUNG,NAV,EPHEM,TRUEA,SPRESS,OCULT,GPRT,GPSEQ,VISIBLE,GENENV,TREG,GRDUT,RATE,BMAT,CM
/CHECK=(NOBOUNDS,OVERFLOW)
/DEBUG=(NOSYMBOULS,TRACEBACK)
//777 /NOG FLOATING /14 /OPTIMIZE /WARNINGS /MOD_LINES /NOMEMachine_CODE /CONTINUATIONS=19
```

**COMPILATION STATISTICS**

<table>
<thead>
<tr>
<th>Run Time:</th>
<th>1.42 seconds</th>
</tr>
</thead>
<tbody>
<tr>
<td>Elapsed Time:</td>
<td>16.58 seconds</td>
</tr>
<tr>
<td>Page Faults:</td>
<td>321</td>
</tr>
<tr>
<td>Dynamic Memory:</td>
<td>160 pages</td>
</tr>
</tbody>
</table>
2.7.2.2 COMPUTE LANDMARK TRACKER MEASUREMENT PARTIALS (HLMT)

HLMT, called by MPART, computes the sensitivity of the landmark tracker measurement vector sensitivity to the individual elements of the estimated state vector. This is done with the support of subroutine SPETBI, SPENMI, and PMEAS.

Processing Requirements

A measurement vector from the spacecraft to the landmark is developed by subtracting the estimated spacecraft position (\( \hat{\mathbf{P}} \)) vector from the landmark position vector (\( \mathbf{L} \)). The resulting measurement vector is transformed to landmark tracker coordinates (\( \hat{\mathbf{M}} \)) and converted to a unit vector (\( \hat{\mathbf{u}} \)). The partials of this unit vector with respect to the estimated state vector elements are given by

\[
\frac{\partial \hat{\mathbf{u}}}{\partial x_i} = \frac{\frac{\partial \hat{\mathbf{M}}}{\partial x_i} - \hat{\mathbf{M}} \frac{\partial \mathbf{M}}{\partial x_i}}{||\hat{\mathbf{M}}||^2}
\]

where the terms of the equation are computed by

\[
||\hat{\mathbf{M}}|| = (\hat{\mathbf{M}}^T \hat{\mathbf{M}})^{1/2}
\]

\[
\frac{\partial \hat{\mathbf{M}}}{\partial x_i} = (a_{ij}) \left[ \frac{\partial}{\partial x_j} (q_{ij}) (\mathbf{L} - \hat{\mathbf{P}}) + (q_{ij}) \frac{\partial}{\partial x_j} (\mathbf{L} - \hat{\mathbf{P}}) \right], \quad i = 1, 2, \ldots, 10
\]

\[
= 0 \text{ otherwise}
\]

\[
\frac{\partial \hat{\mathbf{M}}}{\partial x_i} = (\hat{\mathbf{M}}^T \frac{\partial}{\partial x_i} \hat{\mathbf{M}})^{-1}, \quad i = 1, 2, \ldots, 10
\]

\[
= 0 \text{ otherwise}
\]

where

\( \hat{\mathbf{M}} \) = the estimated measurement vector in landmark tracker coordinates

\( \{a_{ij}\} \) = transformation from body to landmark tracker coordinates

\( \{q_{ij}\} \) = transformation from inertial body coordinates.

\( \hat{\mathbf{P}} \) = Spacecraft estimated position vector in inertial coordinates.

\( \mathbf{L} \) = landmark position vector in inertial coordinates.

\( x_i \) = estimated ith estimation state vector element.
<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Compute LM position vector-inertial space</td>
</tr>
<tr>
<td>2</td>
<td>Compute estimated measurement-inertial space</td>
</tr>
<tr>
<td>3</td>
<td>Transform estimated measurement to LMT coordinates</td>
</tr>
<tr>
<td>4</td>
<td>Convert estimated measurement to unit vector</td>
</tr>
<tr>
<td>5</td>
<td>Compute partial of estimated inertial to body transformation WRT estimated state (SPETBI)</td>
</tr>
<tr>
<td>6</td>
<td>Compute partial of estimated measurement vector in inertial space WRT estimated state (SPEMI)</td>
</tr>
<tr>
<td>7</td>
<td>Compute partial of estimated measurement vector in LMT coordinates WRT estimated state</td>
</tr>
<tr>
<td>8</td>
<td>Compute partial of magnitude estimated measurement in LMT coordinates WRT estimated state</td>
</tr>
<tr>
<td>9</td>
<td>Compute partial of estimated unit vector in LMT coordinates WRT estimated state</td>
</tr>
<tr>
<td>10</td>
<td>Convert unit vector partials to measurement partials (PMEAS)</td>
</tr>
<tr>
<td>11</td>
<td>Compute measurement partials for states 1 through 10</td>
</tr>
<tr>
<td>12</td>
<td>Fill all other measurement partial elements (11 through 22) with 0</td>
</tr>
</tbody>
</table>

Figure A-39
SUBROUTINE HLMT

SUBROUTINE HLMT COMPUTES THE PARTIALS OF THE LANDMARK TRACKER MEASUREMENT VECTOR WITH RESPECT TO THE ESTIMATED STATE VECTOR.

INPUT PARAMETERS

- RE = AVERAGE RADIUS OF THE EARTH
- AL = ALTITUDE OF THE LANDMARK ABOVE RE
- LOM = LONGITUDE OF THE LANDMARK
- LAT = LATITUDE
- T = PRESENT TIME
- TO = INITIAL TIME
- TIEO = INITIAL EARTH FIXED TO INERTIAL TRANSFORMATION
- TBNL = LANDMARK TRACKER TRANSFORMATION FROM NOMINAL TO BODY
- TNLK = KNOWLEDGE TRANSFORMATION FROM LANDMARK TRACKER TO NOMINAL (MISALIGNMENT ARRAY)
- E = ESTIMATED QUATERNIAN ARRAY
- P = ESTIMATED POSITION VECTOR IN INERTIAL SPACE

OUTPUT PARAMETERS

- PHEL = PARTIALS OF HORIZONTAL DEFLECTION
- PDVL = VERTICAL

COMMON BLOCKS

- INCLUDE 'TIME.COM'
- COMMON TIME, TNEQ, TSTOP, TIA, DEL, TIN, DTN, DATEO, TZERO
- COMMON TIME, TMEAS, TRACK, TIS, TISN, DTA, DATER, TPRINT, DTPRINT
- COMMON TIME, TNEQ, TSTOP, TIA, DEL, TIN, DTN, DATEO, TMEAS, TRACK, TIS, TISN, DTA, DATER, TPRINT, DTPRINT

These are the time reference frames.
ATTITUDE STATE AND CONSIDERED PARAMETERS

COMMON /ASTATE/ DE(4), E(4), WD(3), SF(3), D(3), DD(3)
REAL*8 DE, E, WD, SF, D, DD

REAL*8 OE(5), O.00

ATTITUDE STATE AND CONSIDERED PARAMETERS

DIFFERENTIAL OF QUATERNIONS
QUATERNIONS
GYRO DRIFT RATE (RAD/SEC)
GYRO SCALE FACTOR
GYRO NON-ORTHOGONALITY (RAD)
GYRO RELATIVE ORIENTATION (RAD)

COMMON /NSTATE/ X(6), R(6), RGE, RAE
REAL*8 X, R, RGE, RAE

POSITION STATE AND CONSIDERED PARAMETERS

STATE DERIVATIVES (KM/SEC AND KM/SEC/SEC)
STATE POSITION PARAMETERS (KM AND KM/SEC)
RADIUS OF THE MOON (KM)
EARTH DETECTABLE RADIUS (KM)

COMMON /LMTPAR/ AL, LON, LAT, TBML(3,3), TNL(3,3), BL(2), SL(2),
SKL(2), TNLK(3,3), TIED(3,3), SIGGCP, THET
REAL*8 AL, TBML, TNL, BL, SL, SKL, TNLK, TIED, SIGGCP, LAT, LON,
THET

LANDMARK TRACKER PARAMETERS

ALTITUDE OF LANDMARK (KM)
LONGITUDE OF LANDMARK (DEG)
LATITUDE OF LANDMARK (DEG)
ORIENTATION ARRAY FOR LANDMARK TRACKER
NORMAL TO BODY
MISALIGNMENT ARRAY - ACTUAL
TRACKER TO NORMAL
BIAS - ACTUAL (RAD)
BIAS - KNOWLEDGE (RAD)
LOOK ANGLE (RAD)
NOISE STANDARD DEVIATION - KNOWLEDGE (RAD)
INITIAL EARTH FIXED TO INERTIAL
TRANSFORMATION
MISALIGNMENT ARRAY KNOWLEDGE
TRACKER TO NORMAL
POSITION UNCERTAINTY DUE TO CLOUDS

COMMON /PART/ PX(22), PY(22), PZ(22), PXD(22), PYD(22), PZD(22),
PDHS(22,2), PDVS(22,2), PDHL(22), PDVL(22)
REAL*R PX, PY, PZ, PXD, PYD, PZD, PDHS, PDVS, PDVL, PDV

PARTIALS OF THE RESPECTIVE MEASUREMENTS MADE

FOR GPS

PX  =  PARTIALS OF X POSITION MEASUREMENT
PY  =  *  Y  *
PZ  =  *  Z  *
PXZ =  *  X VELOCITY  *
PYZ =  *  Y  *
PZV =  *  Z  *

FOR STAR TRACKER K (K IS THE SECOND PARAMETER)

PDHS =  PARTIALS OF HORIZONTAL DILLECTION
PDVS =  *  VERTICAL  *
PDVL =  *  VERTICAL  *

FOR LANDMARK TRACKER

PDHL =  PARTIALS OF HORIZONTAL DILLECTION
PDVL =  *  VERTICAL  *

INCLUDE 'TARG TS.COM'

COMMON /TARGETS/ MTYPE, IS, NS, NFLAG, MCODE, PI, TPI

REAL*R PI, TPI

MEASUREMENT SPECIFICATIONS

MTYPE  MEASUREMENT TYPE
NFLAG  SET FOR STAR OBSTRUCTION
MCODE  *  MEASUREMENT PROCESSING
PI    PI
TPI  2*PI

INCLUDE 'DEBUG.COM'

COMMON /DEBUG/ ENTER, IDEBUG

USER CONTROLLED PARAMETERS TO VARY DEBUG PRINT LEVEL

ENTER IF 1, PRINTS WHEN ENTERS MOST SUBROUTINES
IDEBUG 0-10, HIGHER NUMBER MEANS MORE PRINT

INCLUDE 'MEASOUT.COM'

COMMON /MEASOUT/ RX(6), RGPX(6), DMCX(3), DMCS(2), MS(3,2),
             RS(2,2,2), DMCL, DVCL, LWU(3), RL(2,2), EMAK(6),
             EMDS(5), EDVS(3), EDVL, EDVL

REAL*R RX, RGPX, DMCX, DMCS, MS, RS, DMCL, DVCL, LWU, RL, EMAK,
       EDMS, EDVS, EDVL, EDVL

MEASUREMENT OUTPUT PARAMETERS

RAX =  POSITION/VELOCITY STATE MEASUREMENT - GPS
      (KM, KM/SEC)
REMG =  ESTIMATED POSITION/VELOCITY STATE
PPIK =  MEASUREMENT - GPS
PPIK =  MEASUREMENT NOISE COVARIANCE

DMCS =  STAR MEASUREMENT HORIZONTAL DEVIATION
DVCS =  STAR MEASUREMENT VERTICAL DEVIATION

FROM BORESIGHT - START (RAD)
01800  C  FROM BORESIGHT - START (RAD)
01900  C  ESTIMATED STAR MEASUREMENT HORIZONTAL
02000  C  DEVIATION FROM BORESIGHT (RAD)
02100  C  ESTIMATED STAR MEASUREMENT VERTICAL
02200  C  DEVIATION FROM BORESIGHT (RAD)
02300  C  STAR MEASUREMENT UNIT VECTOR (SECOND
02400  C  SUBSCRIPT REFERS TO TRACKER) - START
02500  C  STAR MEASUREMENT NOISE COVARIANCE
02600  C  (KNOWLEDGE) - START (RAD**2)
02700  C  LANDMARK MEASUREMENT HORIZONTAL DEVIATION
02800  C  FROM BORESIGHT - LAMKT (RAD)
02900  C  LANDMARK MEASUREMENT VERTICAL DEVIATION
03000  C  FROM BORESIGHT - LAMKT (RAD)
03100  C  ESTIMATED LANDMARK MEASUREMENT HORIZONTAL
03200  C  DEVIATION FROM BORESIGHT (RAD)
03300  C  ESTIMATED LANDMARK MEASUREMENT VERTICAL
03400  C  DEVIATION FROM BORESIGHT (RAD)
03500  C  LANDMARK MEASUREMENT UNIT VECTOR - LAMKT
03600  C  LANDMARK MEASUREMENT NOISE COVARIANCE
03700  C  (KNOWLEDGE) - LAMKT (RAD**2)
03800  C  EQUVALENCE (RE+RADE). (T, TIME). (EP, X)
04000  0027  DIMENSION LE3, L(3), ETBL(3, 3), ETBL(3, 3), ETLB(3, 3),
04100  0027  EML(3), ETI1(3, 3), EMIL(3), PEUI(3, 3), PEUI(3, 3),
04200  0027  DM2(3), DM1(3), PEUI(3, 3), PEUI(3, 3),
04300  0027  REAL8, NML, NML, NML, NML, ETLB(3, 3),
04400  0028  REAL8, NML, NML, NML, NML, ETLB(3, 3),
04500  0028  EUL, L, LE, PEM, PEM, PEM, TIE, TIE, TIE, TIE, TIE, TIE,
04600  0028  RE
04700  C******************************************************************************
04800  C  COMPUTE LE - LANDMARK POSITION VECTOR IN EARTH FIXED
04900  C  COORDINATES
05000  C******************************************************************************
05100  C******************************************************************************
05200  0029  LE1 (RE+RADE)*COS(LON+PI/180.) + COS(LAT+PI/180.)
05300  0036  LE2 (RE+RADE)*SIN(LON+PI/180.) + COS(LAT+PI/180.)
05400  0031  LE3 (RE+RADE)*SIN(LAT+PI/180.)
05500  C******************************************************************************
05600  C  COMPUTE TIE - EXACT TRANSFORMATION FROM EARTH FIXED TO
05700  C  INERTIAL SPACE
05800  C******************************************************************************
05900  C******************************************************************************
06000  0032  CALL WETTIE(TIE)
06100  C******************************************************************************
06200  C  COMPUTE L - LANDMARK POSITION IN INERTIAL SPACE
06300  C******************************************************************************
06400  C******************************************************************************
06500  C  COMPUTE ETBL - ESTIMATED BODY TO LANDMARK TRACKER
06600  C  TRANSFORMATION
06700  C******************************************************************************
06800  C******************************************************************************
06900  0034  CALL MATA(ETBL, L, 3, 3, 3)
07000  0035  CALL MINY3(ETBL, ETBL)
07100  C******************************************************************************
07200  C******************************************************************************
07300  0038  CALL AMAT(E, ETBL)
07400  C******************************************************************************
07500  C  COMPUTE EMI - ESTIMATED MEASUREMENT IN INERTIAL SPACE
DO 10 I = 1, 3
EMI(I) = L(I) - EP(I)

C TRANSFORM TO LANDMARK TRACKER COORDINATES
CALL MATAA(ETLB,ETBI,ETLI,3,3,3)
CALL MATAB(ETLI,EMI,EML,3,3,1)

C COMPUTE MEML - MAGNITUDE OF EML
MEML = VMAG(EML,3)

C COMPUTE EUL - ESTIMATED UNIT VECTOR ALONG ESTIMATED MEASUREMENT VECTOR
CALL UNIT(EML,EUL,3)
EDVL = ASIN(EUL(1))
EDHL = ASIN(EUL(2)/COS(EDVL))
IF(IDEBUG.LT.4) GO TO 15
WRITE(6,42)
WRITE(6,12)
12 Fermat('ETBI')
WRITE(6,13) ( _TBI (I,J), J='', I='', 3, 3, 1, 3, 3, 1)
13 FORMAT(10X,3EI7.7)
15 CONTINUE

C COMPUTE MEASUREMENT PARTIALS
C
DO 100 J = 1, 10
C
CALL SPETBI(J,PETBI)
CALL SPEMI(J,PEMI)
CALL MATAB(ETBI,PEMI,DM2,3,3,1)
CALL MATAB(PETBI,EMI,OMI,3,3,1)
DO 20 I = 1, 3
20 PEMBI(I) = PEMBI(I) + DM1(I)
CALL MATAB(ETLB,PEMBI,PEML,3,3,1)

C COMPUTE PMEML - PARTIAL OF MAGNITUDE OF EML
PMEML = 0.0
DO 30 I = 1, 3
30 PEMML = PEMML + (PEML(I) * EML(I) / MEML)

C COMPUTE PEUL - PARTIAL OF ESTIMATED UNIT VECTOR
DC 40 I = 1, 3
40 PEUL(I) = (PEML(I) * EML(I) / MEML**2)
```
13300  0065   IF(IDEBUG .LT. 4) GO TO 50
13400  0066   WRITE(6,42)
13500  0067   WRITE(6,43) J
13700  0069   WRITE(' FROM HLM ',')
13800  0070   WRITE(6,44) M, L.
l.

147900  0076   CONTINUE
14500  0077   CALL PMEAS(PEUL.EUL.PDHL(J),PDVL(J),2)
14900  0078   DO 200 I=11,22
15000  0079   PDVL(I)=0.
15100  0080   200 PDVL(I)=0.
15200  0081   RETURN
15300  0082   END

C**************************************************************
C CONVERT UNIT VECTOR PARTIALS TO MEASUREMENT PARTIALS
C**************************************************************

PROGRAM SECTIONS

A-2/8

<table>
<thead>
<tr>
<th>Name</th>
<th>Bytes</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCODE</td>
<td>812</td>
<td>PIC CNJ REL LCL SHR EKE RD NOWRT LONG</td>
</tr>
<tr>
<td>SPDATA</td>
<td>167</td>
<td>PIC CON REL LCL SHR NOEKE RD NOWRT LONG</td>
</tr>
<tr>
<td>SLOCAL</td>
<td>1052</td>
<td>PIC CON REL LCL NOSHR NOEKE RD WRT QUAD</td>
</tr>
<tr>
<td>TIME</td>
<td>136</td>
<td>PIC OVR REL GBL NOEKE RD WRT LONG</td>
</tr>
<tr>
<td>ASTATE</td>
<td>160</td>
<td>PIC OVR REL GBL NOEKE RD WRT LONG</td>
</tr>
<tr>
<td>NSTATE</td>
<td>112</td>
<td>PIC OVR REL GBL NOEKE RD WRT LONG</td>
</tr>
<tr>
<td>UMTPAR</td>
<td>392</td>
<td>PIC OVR REL GBL NOEKE RD WRT LONG</td>
</tr>
<tr>
<td>TART</td>
<td>2112</td>
<td>PIC OVR REL GBL NOEKE RD WRT LONG</td>
</tr>
<tr>
<td>PART</td>
<td>36</td>
<td>PIC OVR REL GBL NOEKE RD WRT LONG</td>
</tr>
<tr>
<td>TARGETS</td>
<td>8</td>
<td>PIC OVR REL GBL NOEKE RD WRT LONG</td>
</tr>
<tr>
<td>DEBUG</td>
<td>648</td>
<td>PIC OVR REL GBL NOEKE RD WRT LONG</td>
</tr>
</tbody>
</table>

ENTRY POINTS

<table>
<thead>
<tr>
<th>Address Type Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-00000000     MLMT</td>
</tr>
</tbody>
</table>

VARIABLES

<table>
<thead>
<tr>
<th>Address Type Name</th>
<th>Address Type Name</th>
<th>Address Type Name</th>
<th>Address Type Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>6-000000000 R+8 AL</td>
<td>3-000000030 R+8 DATE0</td>
<td>3-000000070 R+8 DATER</td>
<td>3-000000020 R+8 DEL</td>
</tr>
<tr>
<td>10-000000100 R+8 DHCL</td>
<td>3-000000068 R+8 DTA</td>
<td>3-000000030 R+8 DTN</td>
<td>3-000000080 R+8 DPRINT</td>
</tr>
<tr>
<td>10-000001000 R+8 DVCL</td>
<td>10-000000278 R+8 EDHL</td>
<td>10-000000280 R+8 EDVL</td>
<td>2-000000200 I+4 I</td>
</tr>
<tr>
<td>9-000000004 I+4 IDWBUG</td>
<td>9-000000000 I+4 IENTER</td>
<td>8-000000014 I+4 IS</td>
<td>2-000000204 I+4 J</td>
</tr>
</tbody>
</table>
```
<table>
<thead>
<tr>
<th>Address</th>
<th>Type</th>
<th>Name</th>
<th>Bytes</th>
<th>Dimensions</th>
</tr>
</thead>
<tbody>
<tr>
<td>6-000000CB</td>
<td>R+8</td>
<td>BKL</td>
<td>16</td>
<td>(2)</td>
</tr>
<tr>
<td>6-000000AB</td>
<td>R+8</td>
<td>BL</td>
<td>16</td>
<td>(2)</td>
</tr>
<tr>
<td>4-00000070</td>
<td>R+8</td>
<td>D</td>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>4-00000083</td>
<td>R+8</td>
<td>DD</td>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>4-00000003</td>
<td>R+8</td>
<td>DF</td>
<td>32</td>
<td>(4)</td>
</tr>
<tr>
<td>10-00000150</td>
<td>R+8</td>
<td>DHC5</td>
<td>16</td>
<td>(2)</td>
</tr>
<tr>
<td>2-00000240</td>
<td>R+8</td>
<td>DM1</td>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>2-00000228</td>
<td>R+8</td>
<td>DM2</td>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>10-00000160</td>
<td>R+8</td>
<td>DVC5</td>
<td>16</td>
<td>(2)</td>
</tr>
<tr>
<td>4-00000020</td>
<td>R+8</td>
<td>E</td>
<td>32</td>
<td>(4)</td>
</tr>
<tr>
<td>10-0000025B</td>
<td>R+8</td>
<td>EDMS</td>
<td>16</td>
<td>(2)</td>
</tr>
<tr>
<td>10-0000026B</td>
<td>R+8</td>
<td>EDVS</td>
<td>16</td>
<td>(2)</td>
</tr>
<tr>
<td>2-00000150</td>
<td>R+8</td>
<td>EML</td>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>2-00000180</td>
<td>R+8</td>
<td>EML</td>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>10-0000022B</td>
<td>R+8</td>
<td>EMXG</td>
<td>48</td>
<td>(6)</td>
</tr>
<tr>
<td>5-00000030</td>
<td>R+8</td>
<td>EP</td>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>2-0000010B</td>
<td>R+8</td>
<td>ETBI</td>
<td>72</td>
<td>(2, 3)</td>
</tr>
<tr>
<td>2-00000078</td>
<td>R+8</td>
<td>ETBL</td>
<td>72</td>
<td>(3, 3)</td>
</tr>
<tr>
<td>2-000000C0</td>
<td>R+8</td>
<td>ETLB</td>
<td>72</td>
<td>(3, 3)</td>
</tr>
<tr>
<td>2-00000160</td>
<td>R+8</td>
<td>ETLI</td>
<td>72</td>
<td>(3, 3)</td>
</tr>
<tr>
<td>2-00000240</td>
<td>R+8</td>
<td>EUL</td>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>2-00000060</td>
<td>R+8</td>
<td>L</td>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>2-00000000</td>
<td>R+8</td>
<td>LE</td>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>10-00000170</td>
<td>R+8</td>
<td>MS</td>
<td>48</td>
<td>(3, 2)</td>
</tr>
<tr>
<td>10-00000000</td>
<td>R+8</td>
<td>MX</td>
<td>48</td>
<td>(6)</td>
</tr>
<tr>
<td>7-000006E0</td>
<td>R+8</td>
<td>PMHL</td>
<td>176</td>
<td>(22)</td>
</tr>
<tr>
<td>7-00000450</td>
<td>R+8</td>
<td>PDHS</td>
<td>352</td>
<td>(22, 2)</td>
</tr>
<tr>
<td>7-00000790</td>
<td>R+8</td>
<td>PDVL</td>
<td>176</td>
<td>(22)</td>
</tr>
<tr>
<td>7-00000580</td>
<td>R+8</td>
<td>PDVS</td>
<td>352</td>
<td>(22, 2)</td>
</tr>
<tr>
<td>2-00000258</td>
<td>R+8</td>
<td>PEMB</td>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>2-00000210</td>
<td>R+8</td>
<td>PEMI</td>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>2-00000270</td>
<td>R+8</td>
<td>PEML</td>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>2-000001CB</td>
<td>R+8</td>
<td>PETBI</td>
<td>72</td>
<td>(3, 3)</td>
</tr>
<tr>
<td>2-000002BB</td>
<td>R+8</td>
<td>PEUL</td>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>7-00000000</td>
<td>R+8</td>
<td>PX</td>
<td>176</td>
<td>(22)</td>
</tr>
<tr>
<td>7-00000210</td>
<td>R+8</td>
<td>PXD</td>
<td>176</td>
<td>(22)</td>
</tr>
<tr>
<td>7-00000080</td>
<td>R+8</td>
<td>PY</td>
<td>176</td>
<td>(22)</td>
</tr>
<tr>
<td>7-000002C0</td>
<td>R+8</td>
<td>PYD</td>
<td>176</td>
<td>(22)</td>
</tr>
<tr>
<td>7-00000160</td>
<td>R+8</td>
<td>PZ</td>
<td>176</td>
<td>(22)</td>
</tr>
<tr>
<td>7-00000370</td>
<td>R+8</td>
<td>PZD</td>
<td>176</td>
<td>(22)</td>
</tr>
<tr>
<td>10-00000030</td>
<td>R+8</td>
<td>RGPS</td>
<td>263</td>
<td>(6, 6)</td>
</tr>
<tr>
<td>10-0000020B</td>
<td>R+8</td>
<td>RL</td>
<td>32</td>
<td>(2, 2)</td>
</tr>
<tr>
<td>10-000001A0</td>
<td>R+8</td>
<td>RS</td>
<td>64</td>
<td>(2, 2, 2)</td>
</tr>
</tbody>
</table>
HLMT

6-Apr-1981 15:00:53      VAX-11 FORTRAN V2.0-2

<table>
<thead>
<tr>
<th>Address</th>
<th>Label</th>
<th>Address</th>
<th>Label</th>
<th>Address</th>
<th>Label</th>
<th>Address</th>
<th>Label</th>
<th>Address</th>
<th>Label</th>
<th>Address</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>4-000000058</td>
<td>R8 SF</td>
<td>24 (3)</td>
<td></td>
<td>6-000000008B</td>
<td>R8 SL</td>
<td>16 (2)</td>
<td></td>
<td>6-00000001B</td>
<td>R8 TBML</td>
<td>72 (3, 3)</td>
<td></td>
</tr>
<tr>
<td>2-000000130</td>
<td>R8 TIE</td>
<td>72 (3, 3)</td>
<td></td>
<td>6-00000000E0</td>
<td>R8 TNL</td>
<td>72 (3, 3)</td>
<td></td>
<td>4-000000040</td>
<td>R8 WD</td>
<td>24 (3)</td>
<td></td>
</tr>
<tr>
<td>5-000000030</td>
<td>R8 X</td>
<td>48 (6)</td>
<td></td>
<td>5-0000000000</td>
<td>R8 XD</td>
<td>48 (6)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

LABELS

FUNCTIONS AND SUBROUTINES REFERENCED

AMAT  MATAB  MINV3  MTH$DASIN  MTH$DCOS  MTH$DSIN  PMEAS  SPEMI
SPECTBI  UNIT  VMAG  WET

Total Space Allocated = 5635 Bytes

COMMAND QUALIFIERS

FORTRAN /LIST GCP,INDATA,MATAB,OUTDATA,RUNG,DNAV,EPHEM,TRUEA,SPRESS,OCULT,GPERT,GCPSEQ,VISIBLE,GENENV,TREG,GYROUT,RATE,BMAT,CMAT
/CHECK=(NOBOUNDS,OVERFLOW)
/DEBUG=(NOSYMBOLS,TRACEBACK)
/FT7  /NOG_FLOATING /14 /OPTIMIZE /WARNINGS /NOD_LINES /NOMACHINE_CODE /CONTINUATIONS=19

COMPILATION STATISTICS

Run Time: 3.58 seconds
Elapsed Time: 56.86 seconds
Page Faults: 429
Dynamic Memory: 160 pages
2.7.2.2.1 COMPUTE PARTIALS WITH RESPECT TO ESTIMATED TRANSFORMATION FROM INERTIAL TO BODY COORDINATES (SPETBI)

SPETBI, called by HLMT and HSTAR, computes the sensitivity of the estimated inertial to body transformation to the individual elements of the estimated state. This routine is essentially unsupported by any other major routines.

Processor Requirements

Recognizing that

\[
(q_{ij}) = \begin{bmatrix}
\dot{q}_0^2 + \dot{q}_1^2 - \dot{q}_2^2 - \dot{q}_3^2 & 2(\dot{q}_1\dot{q}_2 + \dot{q}_3\dot{q}_0) & 2(\dot{q}_1\dot{q}_3 - \dot{q}_2\dot{q}_0) \\
2(\dot{q}_1\dot{q}_2 - \dot{q}_3\dot{q}_0) & \dot{q}_0^2 - \dot{q}_1^2 + \dot{q}_2^2 - \dot{q}_3^2 & 2(\dot{q}_2\dot{q}_3 + \dot{q}_1\dot{q}_0) \\
2(\dot{q}_1\dot{q}_3 - \dot{q}_2\dot{q}_0) & 2(\dot{q}_2\dot{q}_3 - \dot{q}_1\dot{q}_0) & \dot{q}_0^2 - \dot{q}_1^2 - \dot{q}_2^2 + \dot{q}_3^2 \\
\end{bmatrix}
\]

since

\[x_i = q_i - 7 \quad \text{for } i = 7, 8, 9, \text{ and } 10\]

\[
\frac{\partial}{\partial x_i} (q_{ij}) = 2 \begin{bmatrix}
\dot{q}_0 & \dot{q}_3 & -\dot{q}_2 \\
-\dot{q}_3 & +\dot{q}_0 & \dot{q}_1 \\
\dot{q}_2 & -\dot{q}_1 & \dot{q}_0 \\
\end{bmatrix} \quad \text{for } i = 7
\]

\[
= 2 \begin{bmatrix}
\dot{q}_1 & \dot{q}_2 & \dot{q}_3 \\
\dot{q}_2 & -\dot{q}_1 & \dot{q}_0 \\
\dot{q}_3 & -\dot{q}_0 & -\dot{q}_1 \\
\end{bmatrix} \quad \text{for } i = 8
\]

\[
= 2 \begin{bmatrix}
-\dot{q}_2 & \dot{q}_1 & -\dot{q}_0 \\
\dot{q}_1 & \dot{q}_2 & \dot{q}_3 \\
\dot{q}_0 & \dot{q}_3 & -\dot{q}_2 \\
\end{bmatrix} \quad \text{for } i = 9
\]

\[
= 2 \begin{bmatrix}
-\dot{q}_3 & \dot{q}_0 & \dot{q}_1 \\
-\dot{q}_0 & -\dot{q}_3 & \dot{q}_2 \\
\dot{q}_1 & \dot{q}_2 & \dot{q}_3 \\
\end{bmatrix} \quad \text{for } i = 10
\]

and

\[
\frac{\partial}{\partial x_i} \{q_{ij}\} = \begin{bmatrix}
0
\end{bmatrix} \quad \text{for } i < 7 \text{ and } i > 10
\]
**SPBII (I,Q)**

<table>
<thead>
<tr>
<th>1 thru 6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Return zero partial array</td>
<td>Compute partial WRT E(1)</td>
<td>Compute partial WRT E(2)</td>
<td>Compute partial WRT E(3)</td>
<td>Compute partial WRT E(4)</td>
</tr>
</tbody>
</table>

Multiply all partial elements by 2

Figure A-40
SUBROUTINE SPETBI(I,Q)

C*********************************************************
C SUBROUTINE SPETBI COMPUTES THE PARTIAL OF THE ESTIMATED
C TRANSFORMATION FROM INERTIAL TO BODY COORDINATES.
C INPUT PARAMETERS
C I = THE INTEGER CONTROLLING THE SPECIFIC
C PARTIAL BEING COMPUTED
C (RANGE 1-10)
C E = THE ESTIMATED QUATERNIAN ARRAY
C OUTPUT PARAMETERS
C Q = THE OUTPUT PARTIAL ARRAY (3,3)
C CALLED BY HSTAR AND HLMT
C*********************************************************

C COMMON BLOCKS
INCLUDE 'ASTATE.COM'
C COMMON /ASTATE/ DE(4),E(4),WD(3),SF(3),D(3),DD(3)
C REAL*8 DE,E,WD,SF,D,DD
C ATTITUDE STATE AND CONSIDERED PARAMETERS
C DE DIFFERENTIAL OF QUATERNIONS
C E QUATERNIONS
C WD GYRO DRIFT RATE (RAD/SEC)
C SF GYRO SCALE FACTOR
C D GYRO NON-ORTHOGANALITY (RAD)
C DD GYRO RELATIVE ORIENTATION (RAD)

DIMENSION Q(3,3)
REAL*8 Q
C*********************************************************
C BRANCH TO THE APPROPRIATE SECTOR ON I
C*********************************************************
GO TO (10,10,10,10,10,20,30,40,50),I
C*********************************************************
C ZERO Q ARRAY - POSITION OR VELOCITY ELEMENT PARTIAL
C*********************************************************
10 DD 15 J=1,3
10 DD 15 K=1,3
0010 15 Q(J,K)=0.
RETURN
C*********************************************************
C COMPUTE PARTIALS WRT Q(0)
C*********************************************************
20 Q(1,1)=E(1)
0013 Q(1,2)=E(4)
0014 Q(1,3)=E(3)
0015 Q(2,1)=E(4)
0016 Q(2,2)=E(1)
0017 Q(2,3)=E(2)
0018 Q(3,1)=E(3)
0019 Q(3,2)=E(2)
0020 Q(3,3)=E(1)
GO TO 100
C******************************************************************************
C COMPUTE PARTIALS WRT Q(1)
C******************************************************************************
0022     30      Q(1,1)=E(2)
0023      Q(1,2)=E(3)
0024      Q(1,3)=E(4)
0025      Q(2,1)=E(3)
0026      Q(2,2)=E(2)
0027      Q(2,3)=E(1)
0028      Q(3,1)=E(4)
0029      Q(3,2)=E(1)
0030      Q(3,3)=E(2)
0031
GO TO 100
C******************************************************************************
C COMPUTE PARTIALS WRT Q(2)
C******************************************************************************
0032     40      Q(1,1)=E(3)
0033      Q(1,2)=E(2)
0034      Q(1,3)=E(1)
0035      Q(2,1)=E(2)
0036      Q(2,2)=E(3)
0037      Q(2,3)=E(4)
0038      Q(3,1)=E(1)
0039      Q(3,2)=E(4)
0040      Q(3,3)=E(3)
0041
GO TO 100
C******************************************************************************
C COMPUTE PARTIALS WRT Q(3)
C******************************************************************************
0042     50      Q(1,1)=E(4)
0043      Q(1,2)=E(1)
0044      Q(1,3)=E(2)
0045      Q(2,1)=E(1)
0046      Q(2,2)=E(4)
0047      Q(2,3)=E(3)
0048      Q(3,1)=E(2)
0049      Q(3,2)=E(3)
0050      Q(3,3)=E(4)
0051      100     DO 200 J=1,3
0052      200     DO 200 K=1,3
0053      200     Q(J,K)=2.*Q(J,K)
0054     RETURN
0055     END
### PROGRAM SECTIONS

<table>
<thead>
<tr>
<th>Name</th>
<th>Bytes</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 SCCODE</td>
<td>349</td>
<td>PIC CON REL LCL  SHR  EXE  RD  NOWR   LONG</td>
</tr>
<tr>
<td>2 SLOCAL</td>
<td>28</td>
<td>PIC CON REL LCL  MOSHR NOEXE RD  WRT   LONG</td>
</tr>
<tr>
<td>3 ASTATE</td>
<td>160</td>
<td>PIC OVR REL GBL  SHR  NOEXE RD  WRT   LONG</td>
</tr>
</tbody>
</table>

### ENTRY POINTS

<table>
<thead>
<tr>
<th>Address</th>
<th>Type</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-00000000</td>
<td>SPETBI</td>
<td></td>
</tr>
</tbody>
</table>

### VARIABLES

<table>
<thead>
<tr>
<th>Address Type</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>AP-0000000040</td>
<td>I+4</td>
</tr>
<tr>
<td>2-00000000</td>
<td>1+4</td>
</tr>
<tr>
<td>2-00000004</td>
<td>1+4</td>
</tr>
</tbody>
</table>

### ARRAYS

<table>
<thead>
<tr>
<th>Address Type</th>
<th>Name</th>
<th>Bytes</th>
<th>Dimensions</th>
</tr>
</thead>
<tbody>
<tr>
<td>3-0000000000</td>
<td>R-B</td>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>3-0000000000</td>
<td>R-B</td>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>3-0000000000</td>
<td>R-B</td>
<td>32</td>
<td>(4)</td>
</tr>
<tr>
<td>3-0000000000</td>
<td>R-B</td>
<td>32</td>
<td>(4)</td>
</tr>
<tr>
<td>AP-0000000000</td>
<td>R-B</td>
<td>72</td>
<td>(3, 3)</td>
</tr>
<tr>
<td>3-0000000000</td>
<td>R-B</td>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>3-0000000000</td>
<td>R-B</td>
<td>24</td>
<td>(3)</td>
</tr>
</tbody>
</table>

### LABELS

<table>
<thead>
<tr>
<th>Address</th>
<th>Label</th>
<th>Address</th>
<th>Label</th>
<th>Address</th>
<th>Label</th>
<th>Address</th>
<th>Label</th>
<th>Address</th>
<th>Label</th>
<th>Address</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-00000000</td>
<td>10</td>
<td>**</td>
<td>15</td>
<td>0-00000051</td>
<td>20</td>
<td>0-00000008C</td>
<td>30</td>
<td>0-0000000C6</td>
<td>40</td>
<td>0-00000100</td>
<td>50</td>
</tr>
</tbody>
</table>

Total Space Allocated = 537 Bytes

### COMMAND QUALIFIERS

```
FORTRAN /LIST GCP,INDATA,MATAB,OUTDATA,RUNG,DNAV,EPHEM,TRUEA,SPRESS,OCCULT,GPERT,GCPSEQ,VISIBLE,GENENV,TREG,GYROUT,RATE,BMAT,CMA
/CHECK=(NOBOUNDS,OVERFLOW)
/DEBUG=(NOSYMBOLS,TRACEBACK)
/F77 /NOG_FLOATING /14 /OPTIMIZE /WARNINGS /NOD_LINES /NOMACHINE_CODE /CONTINUATIONS=19
```
COMPILATION STATISTICS

Run Time: 2.07 seconds
Elapsed Time: 38.65 seconds
Page Faults: 324
Dynamic Memory: 160 pages
2.7.2.2.2 COMPUTE PARTIALS OF ESTIMATED MEASUREMENT VECTOR IN INERTIAL SPACE (SPEMI)

SPEMI called by HLMT, computes the sensitivity of the estimated landmark tracker measurement vector in inertial space with respect to the elements of the estimated state. This routine is unsupported by other major routines.

Processing Requirements

Recognizing that
\[
\hat{M}_T = L - \hat{P}
\]
\[
\frac{\partial \hat{M}_T}{\partial x_1} = \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix}
\]
for \( \hat{x}_1 = \hat{x} \)
\[
\frac{\partial \hat{M}_T}{\partial x_1} = \begin{bmatrix} 0 \\ -1 \\ 0 \end{bmatrix}
\]
for \( \hat{x}_1 = \hat{y} \)
\[
\frac{\partial \hat{M}_T}{\partial x_1} = \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix}
\]
for \( \hat{x}_1 = \hat{z} \)

and
\[
\frac{\partial \hat{M}_T}{\partial x_1} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}
\]
otherwise

where
\( \hat{M}_T \) = Estimated measurement vector in inertial space
\( L \) = Landmark position vector in inertial space
\( \hat{P} = \begin{bmatrix} \hat{x} \\ \hat{y} \\ \hat{z} \end{bmatrix} \) = Spacecraft estimated position vector in inertial space.
SPEMI (I, X)

<table>
<thead>
<tr>
<th>T</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>I Greater than 3</td>
<td></td>
</tr>
<tr>
<td>Set all partial elements to zero</td>
<td>Initialize all partial elements to zero</td>
</tr>
<tr>
<td></td>
<td>Set X(I) = -1</td>
</tr>
</tbody>
</table>

Figure A-41
SUBROUTINE SPXE[I, X]

SUBROUTINE SPXE COMPTE. IH PARTIAL OF THE MEASUREMENT VECTOR IN INERTIAL SPACE OF THE ESTIMATED STATE

INPUT PARAMETERS
I = THE INTEGER CONTROLLING THE INDIVIDUAL PARTIAL (RANGE 1-10)

OUTPUT PARAMETERS
X = THE OUTPUT PARTIAL VECTOR (3,1)
CALLED BY HLMT

DIMENSION X(3)
REAL*8 X
DO 20 J=1,3
20 X(I)=0.
DO 50 I=1,3
50 RETURN
DO 60 J=1,3
60 X(I)=1.
DO 10 J=1,3
10 X(I)=0.
RETURN
END

PROGRAM SECTIONS

Name Bytes Attributes
0 $CODE 66 PIC CON REL LCL SHR EXE RD NDWR) LONG
2 $LOCAL 24 PIC CON REL LCL NOSHR NODXEXE RD WRT LON.

ENTRY POINTS

Address Type Name
0-00000000 SPXE

VARIABLES

Address Type Name Address Type Name
AP-000000000 1=4 I 2-00000000 1=4 J
**SPENI**

**ARRAYS**

<table>
<thead>
<tr>
<th>Address Type</th>
<th>Name</th>
<th>Bytes</th>
<th>Dimensions</th>
</tr>
</thead>
<tbody>
<tr>
<td>AP-00000000</td>
<td>R8</td>
<td>24</td>
<td>(3)</td>
</tr>
</tbody>
</table>

**LABELS**

<table>
<thead>
<tr>
<th>Address</th>
<th>Label</th>
<th>Address</th>
<th>Label</th>
<th>Address</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>20</td>
<td>0-00000034</td>
<td>50</td>
<td>60</td>
<td></td>
</tr>
</tbody>
</table>

Total Space Allocated = 90 Bytes

**COMMAND QUALIFIERS**

- FORTRAN /LIST GCP,INDATA,MATAB,OUTDATA,RUNPG,DNAV,EPHEM,TRUEA,SPRESS,occult,GPERT,GCSEQ,Visible,GENENV,REG,GYROUT,Rate,GRAT,CNA
- /CHECK=(NOBOUNDS,OVERFLOW)
- /DEBUG=(NOSYMBOLS,TRACEBACK)
- F77 /NOF_FLOATING /14 /OPTIMIZE /WARNINGS /NOLINES /NONMACHINE_CODE /CONTINUATIONS=19

**COMPILATION STATISTICS**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Run Time</td>
<td>0.56</td>
</tr>
<tr>
<td>Elapsed Time</td>
<td>11.04</td>
</tr>
<tr>
<td>Page Faults:</td>
<td>305</td>
</tr>
<tr>
<td>Dynamic Memory:</td>
<td>180 pages</td>
</tr>
</tbody>
</table>
2.7.2.3 MEASUREMENT PARTIALS FOR STAR TRACKER (K) (HSTAR)

HSTAR, called by MPART, computes the sensitivity of the star tracker measurement to variations in the individual elements of the estimated state vector. This is done with the support of subroutines SPETBI and PMEAS.

Processing Requirements

A unit vector along the optical axis at each star tracker (K) is established and transformed to inertial space through the estimated transformation from star tracker coordinates to inertial space ($\bar{U}_s$). The partials of this unit vector with respect to the element of the estimated state are given by

$$\frac{\partial \bar{U}_s}{\partial x_i} = \begin{bmatrix} \{b_{ij}\} & \frac{\partial}{\partial x_i} \{q_{ij}\} & \{a_{ij}\} \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

where

- $\{b_{ij}\} = \text{Transformation from body to star tracker coordinates.}$
- $\{q_{ij}\} = \text{Estimated transformation from inertial to body coordinates.}$
- $\{a_{ij}\} = \text{Ideal transformation from star tracker to inertial coordinates.}$
<table>
<thead>
<tr>
<th>Step</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compute star unit vector in inertial coordinates</td>
</tr>
<tr>
<td>Compute estimated transformation from star tracker to inertial space</td>
</tr>
<tr>
<td>Computed estimated unit vector in ST coordinates</td>
</tr>
<tr>
<td>Compute estimated measurement</td>
</tr>
<tr>
<td>Compute partial of estimated transformation from inertial to body WRT estimated state (SPETBI)</td>
</tr>
<tr>
<td>Compute partials of estimated unit vector</td>
</tr>
<tr>
<td>Compute unit vector partials to measurement partials</td>
</tr>
<tr>
<td>Compute partials of unit vector with respect to states 7 through 10</td>
</tr>
<tr>
<td>Fill all other measurement partial elements (1 through 6) and (11 through 22) with zero</td>
</tr>
</tbody>
</table>

Figure A-42
SUBROUTINE HSTAR(K)

C*******************************************************************************

SUBROUTINE HSTAR COMPUTES THE PARTIALS FOR THE KTH STAR TRACKER MEASUREMENT VECTOR.

INPUT PARAMETERS
K = SELECTOR OF THE KTH STAR TRACKER
STATE = ELEMENTS 7-10 ARE USED AS THE IDEAL INERTIAL TO BODY QUATERNIAN ELEMENTS
TBNS = STAR TRACKER NOMINAL TO BODY TRANSFORMATION
TNS = " " STAR TRACKER TO NOMINAL TRANSFORMATION (MISALIGNMENT ARRAY)
E = ESTIMATED QUATERNIAN ARRAY

OUTPUT PARAMETERS
PDHS = PARTIAL OF THE STAR TRACKER HORIZONTAL DEFLECTION
PDVS = PARTIAL OF THE STAR TRACKER VERTICAL DEFLECTION

*******************************************************************************

COMMON BLOCKS
INCLUDE 'PART.COM'
COMMON /PART/PX(22),PY(22),PZ(22),PXD(22),PYD(22),PZD(22),PDHS(22,2),PDVS(22,2),PDHL,PDVL
REAL*8 PX,PY,PZ,PXD,PYD,PZD,PDHS,PDVS,PDHL,PDVL

PARTIALS OF THE RESPECTIVE MEASUREMENTS MADE
FOR GPS
PX = PARTIALS OF X POSITION MEASUREMENT
PY = " " Y " "
PZ = " " Z " "
PXD = " " X VELOCITY " "
PYD = " " Y " "
PZD = " " Z " "

FOR STAR TRACKER K (K IS THE SECOND PARAMETER)
PDHS = PARTIALS OF HORIZONTAL DEFLECTION
PDVS = " " VERTICAL " 

FOR LANDMARK TRACKER
PDHL = PARTIALS OF HORIZONTAL DEFLECTION
PDVL = " " VERTICAL " 

*******************************************************************************

INCLUDE 'STARPAR.COM'
COMMON /STARPAR/BSK(2,2),SSK(2,2),TNSK(3,3,2),TBNSK(3,3,2)
REAL*8 BS,SS,TNS,TBNS

*******************************************************************************

STAR TRACKER PARAMETERS
IN EACH CASE THE LAST SUBSCRIPT REFERS TO THE TRACKER USED
BS = BIAS - ACTUAL (RAD)
SS = NOISE STANDARD DEVIATION - ACTUAL (RAD)
TNS = MISALIGNMENT ARRAY - TRANSFORMATION FROM STAR TRACKER TO NOMINAL
TBNS = ORIENTATION ARRAY - TRANSFORMATION FROM
<table>
<thead>
<tr>
<th>Line</th>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0260</td>
<td>C</td>
<td>NOMINAL TO BODY</td>
</tr>
<tr>
<td>0260</td>
<td>C</td>
<td>BS K = BIAS - KNOWLEDGE (RAD)</td>
</tr>
<tr>
<td>0260</td>
<td>C</td>
<td>SS K = NOISE STANDARD DEVIATION - KNOWLEDGE (RAD)</td>
</tr>
<tr>
<td>0260</td>
<td>C</td>
<td>TNS K = MISALIGNMENT KNOWLEDGE ARRAYS - TRANSFORMATION</td>
</tr>
<tr>
<td>0260</td>
<td>C</td>
<td>FROM STAR TRACKER TO NOMINAL</td>
</tr>
<tr>
<td>0260</td>
<td>0008</td>
<td>INCLUDE 'ENVIR.COM'</td>
</tr>
<tr>
<td>0200</td>
<td>0009</td>
<td>COMMON /ENVIR/, STATE(10), PROFILE(10,4), INIT</td>
</tr>
<tr>
<td>0270</td>
<td>0010</td>
<td>REAL*8 STATE, PROFILE</td>
</tr>
<tr>
<td>0270</td>
<td>C</td>
<td>REAL WORLD STATE PARAMETERS</td>
</tr>
<tr>
<td>0270</td>
<td>C</td>
<td>STATE STATE VALUES: X,Y,Z,XO,YO,ZO,E0,E1,E2,E3</td>
</tr>
<tr>
<td>0270</td>
<td>C</td>
<td>PROFILE ATTITUDE PROFILE - TIME (SEC) VS</td>
</tr>
<tr>
<td>0270</td>
<td>C</td>
<td>INIT INTEGRATION INITIALIZATION KEY (-1)</td>
</tr>
<tr>
<td>0270</td>
<td>0011</td>
<td>INCLUDE 'ASTATE.COM'</td>
</tr>
<tr>
<td>0200</td>
<td>0012</td>
<td>COMMON /ASTATE/, DE(4), E(4), WD(3), SF(3), D(3), DD(3)</td>
</tr>
<tr>
<td>0280</td>
<td>0013</td>
<td>REAL*8 DE, E, WD, SF, D, DD</td>
</tr>
<tr>
<td>0280</td>
<td>C</td>
<td>ATTITUDE STATE AND CONSIDERED PARAMETERS</td>
</tr>
<tr>
<td>0280</td>
<td>C</td>
<td>D. DIFFERENTIAL OF QUATERNIONS</td>
</tr>
<tr>
<td>0280</td>
<td>C</td>
<td>E QUATERNIONS</td>
</tr>
<tr>
<td>0280</td>
<td>C</td>
<td>WD GYRO DRIFT RATE (RAD/SEC)</td>
</tr>
<tr>
<td>0280</td>
<td>C</td>
<td>SF GYRO SCALE FACTOR</td>
</tr>
<tr>
<td>0280</td>
<td>C</td>
<td>D GYRO NON-ORTHOGONALITY (RAD)</td>
</tr>
<tr>
<td>0280</td>
<td>C</td>
<td>DD GYRO RELATIVE ORIENTATION (RAD)</td>
</tr>
<tr>
<td>0280</td>
<td>0014</td>
<td>INCLUDE 'MEASOUT.COM'</td>
</tr>
<tr>
<td>0010</td>
<td>0015</td>
<td>COMMON /MEASOUT/, MX(6), RGPS(6, 6), DMC(2), DVC(2), MS(3, 2),</td>
</tr>
<tr>
<td>0200</td>
<td></td>
<td>RS(2, 2, 2), DMCXV, LMU(3), RL(2, 2), EMEG(6),</td>
</tr>
<tr>
<td>0030</td>
<td></td>
<td>EMD(2), EDAH(2), EDL, EDVL</td>
</tr>
<tr>
<td>0040</td>
<td>0016</td>
<td>REAL*8 MX, RGPS, DMC, DVC, MS, RS, DMCXV, LMU, RL, EMEG,</td>
</tr>
<tr>
<td>0050</td>
<td></td>
<td>EMD, EDH, EDVL</td>
</tr>
<tr>
<td>0060</td>
<td>C</td>
<td>MEASUREMENT OUTPUT PARAMETERS</td>
</tr>
<tr>
<td>0080</td>
<td>C</td>
<td>MX = POSITION VELOCITY STATE MEASUREMENT - GPS</td>
</tr>
<tr>
<td>0090</td>
<td>C</td>
<td>(KM, KM/SEC)</td>
</tr>
<tr>
<td>0100</td>
<td>C</td>
<td>EMXG = ESTIMATED POSITION VELOCITY STATE MEASUREMENT - HGPS</td>
</tr>
<tr>
<td>0120</td>
<td>C</td>
<td>RGPS = STATE MEASUREMENT NOISE COVARIANCE</td>
</tr>
<tr>
<td>0130</td>
<td>C</td>
<td>(KNOWLEDGE) - GPS (KM+2, KM/SEC**2)</td>
</tr>
<tr>
<td>0140</td>
<td>C</td>
<td>DMC = STAR MEASUREMENT HORIZONTAL DEVIATION FROM BORESIGHT - START (RAD)</td>
</tr>
<tr>
<td>0150</td>
<td>C</td>
<td>DVC = STAR MEASUREMENT VERTICAL DEVIATION FROM BORESIGHT - START (RAD)</td>
</tr>
<tr>
<td>0160</td>
<td>C</td>
<td>EMD = ESTIMATED STAR MEASUREMENT HORIZONTAL</td>
</tr>
<tr>
<td>0170</td>
<td>C</td>
<td>DEVIATION FROM BORESIGHT (RAD)</td>
</tr>
<tr>
<td>0180</td>
<td>C</td>
<td>EDAH = ESTIMATED STAR MEASUREMENT VERTICAL</td>
</tr>
<tr>
<td>0190</td>
<td>C</td>
<td>DEVIATION FROM BORESIGHT (RAD)</td>
</tr>
<tr>
<td>0200</td>
<td>C</td>
<td>MS = STAR MEASUREMENT UNIT VECTOR (SECOND)</td>
</tr>
<tr>
<td>0220</td>
<td>C</td>
<td>SUBSCRIPT REFERS TO TRACKER</td>
</tr>
<tr>
<td>0240</td>
<td>C</td>
<td>RS = STAR MEASUREMENT NOISE COVARIANCE</td>
</tr>
</tbody>
</table>
C (KNOWLEDGE) - START (RAD==2)

C DHCL = LANDMARK MEASUREMENT HORIZONTAL DEVIATION FROM BORESIGHT - LAMKT (RAD)
C DVCL = LANDMARK MEASUREMENT VERTICAL DEVIATION FROM BORESIGHT - LAMKT (RAD)
C EDHL = ESTIMATED LANDMARK MEASUREMENT HORIZONTAL DEVIATION FROM BORESIGHT (RAD)
C EDVL = ESTIMATED LANDMARK MEASUREMENT VERTICAL DEVIATION FROM BORESIGHT (RAD)
C LMU = LANDMARK MEASUREMENT UNIT VECTOR - LAMKT (KNOWLEDGE) - LAMKT (RAD==2)
C RL = LANDMARK MEASUREMENT NOISE COVARIANCE (KNOWLEDGE) - LAMKT

DIMENSION TBI(3,3), TBS(3,3), US(3), UB(3), UI(3),
       ETBS(3,3), ETSB(3,3), ETSI(3,3), ETSJI(3,3), EU(3),
       PEUS(3,10), PETBI(3,3), UIIP(3), ATEM(4), T1(3,3),
       T2(3,3), EU(3), TEM(3)

REAL*8 EU, ETBI, ETBS, ETSB, ETSI, EU, PETBI, PEUS, TBI, TBS, UB,
       UI, UIIP, US, ATEM, T1, T2, TEMP

C COMPUTE TIB - TRUE BODY TO INERTIAL TRANSFORMATION
C
DO 5 I=1,4
ATEMP(I)=STAT(I)+6
C
CONTINUE
CALL AMAT(ATEMP, TBI)
CALL MTNV3(TBI, TIB)

C COMPUTE TBS - TRUE STAR TRACKER TO BODY TRANSFORMATION
C
DO 400 1=1,3
DO 100 J=1,3

T1(I, J)=TBS(I, J, K)
CALL MATAB(T1, T2, TBS, 3, 3)

DO 100 J=1,3
DO 200 K=1,3

T2(I, J)=TNS(I, J, K)
CALL MATAB(T1, T2, TBS, 3, 3)

C COMPUTE UNIT VECTOR TO STAR IN TRACKER COORDINATES
C
US(1)=0.
US(2)=0.
US(3)=1.

CALL MATAB(TBS, US, UB, 3, 3, 1)
CALL MATAB(TIB, UB, UI, 3, 3, 1)

C COMPUTE ETSB - ESTIMATE OF TSB
C
DO 110 I=1,3
DO 110 J=1,3

T1(I, J)=TBS(I, J, K)
CALL MATAB(T1, T2, ETBS, 3, 3)

CALL MINV3(ET8S, ET5B)

VAX-11 FORTRAN V2.0-2
_DBAO:[D11R.GCP]HSTAR.FOR;10
6-Apr-1981 15:03:18
9-Feb-1981 18:15:38

Page 3
HSTAR

07300 C COMPUTE EUI - ESTIMATED UNIT VECTOR IN STAR TRACKER
07400 C COORDINATES
07500
07600 0040 CALL AMAT(E,ETBI)
07700 0041 CALL MATAB(ETSB,ETBI,ETS1,3,3,3)
07800 0042 CALL MATAB(ETS1,UI,EUS,3,3,1)
07900 C
08000 C CALCULATE ESTIMATED MEASUREMENT
08100 C
08200 C
08300 C EDVS(K)=ASIN(U5(I))
08400 C EDHS(K)=ASIN(U5(2)/COS(EDVS(K)))
08500 C
08600 C
08700 C
08800 C
08900 0045 DO 10 I=1,3
09000 0046 DO 10 J=1,10
09100 0047 10 PEUS(I,J)=0.
09200 0048 DO 20 J=1,10
09300 C
09400 C COMPUTE PARTIALS OF EBI WRT Q(J), J=7,10
09500 C
09600 0049 CALL SPETBI(J,PETBI)
09700 C
09800 C SET UP ARRAY FOR PARTIALS OF ESTIMATED UNIT VECTOR
09900 C
10000 0050 CALL MATAB(PETBI,UI,UIP,3,3,1)
10100 0051 CALL MATAB(ETSB,UIP,TEMP,3,3,1)
10200 0052 DO 25 I=1,3
10300 0053 25 PEUS(I,J)=TEMP(I)
10400 C
10500 C CONVERT UNIT VECTOR PARTIALS TO MEASUREMENT PARTIALS
10600 C
10700 0054 20 CALL PMEAS(Temp,EUS,PDHS(J,K),PDYS(J,K),1)
10800 0055 DO 30 I=1,32
10900 0056 PDHS(I,K)=0.
11000 0057 30 PDYS(I,K)=0.
11100 0058 DO 40 I=1,6
11200 0059 PDHS(I,K)=0.
11300 0060 40 PDYS(I,K)=0.
11400 0061 RETURN
11500 0062 END
## PROGRAM SECTIONS

<table>
<thead>
<tr>
<th>Name</th>
<th>Bytes</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 $CODE</td>
<td>485</td>
<td>PIC CON REL LCL</td>
</tr>
<tr>
<td>1 $DATA</td>
<td>6</td>
<td>PIC CON REL LCL</td>
</tr>
<tr>
<td>2 $LOCAL</td>
<td>147</td>
<td>PIC CON REL LCL</td>
</tr>
<tr>
<td>3 PART</td>
<td>2112</td>
<td>PIC OVR REL GBL</td>
</tr>
<tr>
<td>4 STARPAR</td>
<td>560</td>
<td>PIC OVR REL GBL</td>
</tr>
<tr>
<td>5 ENVIR</td>
<td>404</td>
<td>PIC OVR REL GBL</td>
</tr>
<tr>
<td>6 ASTATE</td>
<td>160</td>
<td>PIC OVR REL GBL</td>
</tr>
<tr>
<td>7 MEASOUT</td>
<td>648</td>
<td>PIC OVR REL GBL</td>
</tr>
</tbody>
</table>

## ENTRY POINTS

<table>
<thead>
<tr>
<th>Address</th>
<th>Type</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-00000000</td>
<td></td>
<td>HSTAR</td>
</tr>
</tbody>
</table>

## VARIABLES

<table>
<thead>
<tr>
<th>Address</th>
<th>Type</th>
<th>Name</th>
<th>Address</th>
<th>Type</th>
<th>Name</th>
<th>Address</th>
<th>Type</th>
<th>Name</th>
<th>Address</th>
<th>Type</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>7-000001E0</td>
<td>R+B</td>
<td>DHCL</td>
<td>7-000001E8</td>
<td>R+B</td>
<td>DVCCL</td>
<td>7-0000027B</td>
<td>R+B</td>
<td>EML</td>
<td>7-0000028C</td>
<td>R+B</td>
<td>EDVL</td>
</tr>
<tr>
<td>2-00003488</td>
<td>I+4</td>
<td>1</td>
<td>B-00000190</td>
<td>I+4</td>
<td>INIT</td>
<td>2-000048C</td>
<td>I+4</td>
<td>J</td>
<td>AP-0000004B</td>
<td>I+4</td>
<td>K</td>
</tr>
</tbody>
</table>

## ARRAYS

<table>
<thead>
<tr>
<th>Address</th>
<th>Type</th>
<th>Name</th>
<th>Bytes</th>
<th>Dimensions</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-0000034B</td>
<td>R+B</td>
<td>ATEMP</td>
<td>32</td>
<td>(4)</td>
</tr>
<tr>
<td>4-00000000</td>
<td>R+B</td>
<td>BS</td>
<td>32</td>
<td>(2, 2)</td>
</tr>
<tr>
<td>4-00000160</td>
<td>R+B</td>
<td>BSK</td>
<td>32</td>
<td>(2, 2)</td>
</tr>
<tr>
<td>6-00000070</td>
<td>R+B</td>
<td>D</td>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>6-0000008B</td>
<td>R+B</td>
<td>DD</td>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>6-00000000</td>
<td>R+B</td>
<td>DE</td>
<td>32</td>
<td>(4)</td>
</tr>
<tr>
<td>7-00000150</td>
<td>R+B</td>
<td>DMCS</td>
<td>16</td>
<td>(2)</td>
</tr>
<tr>
<td>7-00000160</td>
<td>R+B</td>
<td>DVC5</td>
<td>16</td>
<td>(2)</td>
</tr>
<tr>
<td>6-00000020</td>
<td>R+B</td>
<td>E</td>
<td>32</td>
<td>(4)</td>
</tr>
<tr>
<td>7-0000025B</td>
<td>R+B</td>
<td>EDH5</td>
<td>16</td>
<td>(2)</td>
</tr>
<tr>
<td>7-0000215B</td>
<td>R+B</td>
<td>EDV5</td>
<td>16</td>
<td>(2)</td>
</tr>
<tr>
<td>7-0000222B</td>
<td>R+B</td>
<td>EML</td>
<td>48</td>
<td>(6)</td>
</tr>
<tr>
<td>2-00000180</td>
<td>R+B</td>
<td>ETBI</td>
<td>72</td>
<td>(3, 3)</td>
</tr>
<tr>
<td>2-00000120</td>
<td>R+B</td>
<td>ETBS</td>
<td>72</td>
<td>(3, 3)</td>
</tr>
<tr>
<td>2-000016B</td>
<td>R+B</td>
<td>ETSA</td>
<td>72</td>
<td>(3, 3)</td>
</tr>
<tr>
<td>2-00001240</td>
<td>R+B</td>
<td>EUI</td>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>2-00004580</td>
<td>R+B</td>
<td>EUS</td>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>7-000021F0</td>
<td>R+B</td>
<td>LMU</td>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>7-0000037B</td>
<td>R+B</td>
<td>MS</td>
<td>48</td>
<td>(3, 2)</td>
</tr>
<tr>
<td>7-00000000</td>
<td>R+B</td>
<td>MX</td>
<td>48</td>
<td>(6)</td>
</tr>
<tr>
<td>3-000006E0</td>
<td>R+B</td>
<td>PDKL</td>
<td>176</td>
<td>(22)</td>
</tr>
<tr>
<td>3-00000420</td>
<td>R+B</td>
<td>PDHS</td>
<td>352</td>
<td>(22, 2)</td>
</tr>
<tr>
<td>3-00000730</td>
<td>R+B</td>
<td>PDVL</td>
<td>176</td>
<td>(22)</td>
</tr>
</tbody>
</table>
COMPILATION STATISTICS

Run Time: 2.72 seconds
Elapsed Time: 45.11 seconds
Page Faults: 398
Dynamic Memory: 160 pages
2.7.3 Make Measurement Updates

A Carlsson Square Root filter shall be used to estimate the spacecraft attitude and position. The filter calculates a new estimate for the state and model parameters each time a measurement becomes available. The correction depends on the difference between the actual measurement and the predicted measurement and the value of the gain matrix computed from the covariance of the measurement noise and the covariance of the state estimate. Every time the state and model parameters are corrected by the filter, the covariance matrix for the state and model parameters is also changed to reflect the additional knowledge contained in the measurement.

A triangular formulation is used to avoid the possibility of a covariance matrix which is not positive semi-definite and the resulting divergence of the state estimate and failure of the recursive algorithm. The square root formulation provides twice the effective numerical precision of Kalman's original formulation and guarantees a positive definite covariance matrix.

The square root \( S \) of the covariance matrix \( P \) is defined by

\[
P = S S^T
\]

The Cholesky method for calculating the square root of a matrix shall be used to calculate the square root \( S \) of the covariance matrix \( P \) at each measurement update. The equations which define the Cholesky method are

\[
S_{j1} = \left( P_{11} - \sum_{k=1+1}^{n} S_{1k}^2 \right)^{1/2},
\]

\[
S_{j1} = \left[ P_{j1} - \sum_{k=1+1}^{n} S_{jk} S_{ik} \right] S_{i1}^{-1}, \quad j = i-1,1
\]

where

- \( n \) = dimension of covariance matrix \( P \).
- \( P_{ij} \) = element of \( P \) matrix.
- \( S_{ij} \) = element of \( S \) matrix.

The matrix \( S \) is upper triangular. The equation for the elements of \( S \) must be evaluated with the initial index \( i = n \). The index \( i \) is decremented by 1 until the final value \( i = 1 \) is achieved. Evaluation in this order causes no references to uncalculated elements of the \( S \) matrix.

The filter equations shall use the difference between the reference measurement and the actual measurement to update the state measurement. It uses the current covariance matrix which represents the estimate uncertainties in the estimated and considered parameters.
The Carlson formulation does not treat "consider" parameters. Therefore, this formulation has been slightly modified to take non-estimated uncertain model parameters into consideration. Very briefly stated, these modifications simply require that the augmented state covariance (including the measurement consider parameters) replace the spacecraft's position and velocity covariance in Carlson's algorithm.

However, the state and state covariance of the measurement consider parameters are not themselves affected. In this way, the uncertainties in the measurement consider parameters propagate into the estimation of the spacecraft's position vector as if the consider parameters were an extension of the solution vector, but stopping short of actually computing their corrections.

The classic Kalman formulation for a measurement update can be written in three steps

\[ P^+ = P - KGP \]
\[ X^+ = X + K \cdot dy \]
\[ K = PG^T(GPG^T + r)^{-1} \]

where the (+) superscript denotes the state or state covariance following the measurement update,

- \( P = P' \), the augmented state covariance matrix (algorithm NR-2)
- \( X \) = State vector (position and velocity, possibly augmented with the moon's mean radius)
- \( K \) = Kalman gain matrix
- \( G = G' \), the augmented observation matrix (see algorithm NO-2)
- \( r \) = Measurement "noise" covariance matrix
- \( dy \) = Observed minus computed measurement discrepancy vector.

We may express the state covariance matrix as the product of its square-root matrix, \( S \), which has the upper triangular form:

\[ P = SS^T \]

With which we may write
\[
P^+ = SS^T - SS^T G^T (FPG^T + r)^{-1} GSS^T \\
= S \left[ I - \left( S^T G S / \alpha \right) \right] S^T
\]

Since the Space Sextant measurement is scalar,
\[
\alpha = (GP^T + r)
\]
is also scalar, obviating the need for a matrix inversion.

We observe that the square-root of \( P \) may be updated independently
\[
S^+ = S \left[ I - ff^T / \alpha \right] ^{1/2}
\]
where, as in Carlson's notation, we use \( f = ST \) \( GT \).

The essence of Carlson's algorithm is the technique for taking the square-root of the bracketed quantity which is done without directly resorting to the Cholesky decomposition.

With each measurement compute the square-root of the augmented state covariance which has been integrated up to the time of the measurement
\[
S = \sqrt{P}
\]
where \( S \) has the upper triangular form by Cholesky's method. Compute the column vector \( f \)
\[
f = ST \cdot G^T
\]
where the segmented \( G \) matrix (see algorithm NO-2) is a \((1 \times n)\),
n = m (state variables + 1 ("consider" variables)

Compute the diagonal elements of the \( a^D \) and \( c^D \) matrices:
\[
\begin{align*}
\alpha_0 &= r \\
\alpha_i &= \alpha_{i-1} + f_{i-1}^2 \\
a_i &= \left( \alpha_i / d_i \right)^{1/2} \\
C_i &= f_i / \left( \alpha_{i-1} \alpha_i \right)^{1/2}
\end{align*}
\]
and the \( f^* \) matrix
\[
\begin{align*}
f_1^* &= 0 \\
f_i^* &= \left[ \begin{array}{c} (i-1) \\ f_1 f_2 \cdots f_{i-1} \end{array} \right] / \left( \begin{array}{c} n-i+1 \\ 0 \end{array} \right)^T, \quad i=2,n
\end{align*}
\]
and form the square-root of the matrix
\[
A = a^D - f^* c^D
\]
The updated square-root state covariance matrix is thus
\[ S(+) = SA \]
from which a fully reconstituted state covariance is computed in temporary storage, including the pseudo-updated considered parameters,

\[ P_o = S(+) S(+)^T \]

The actual updated state covariance is constructed from only those elements of \( P_o \) which apply to the real estimation variables. The remaining elements of \( P^o \) (i consider parameters) being unchanged.

\[
\begin{align*}
P_{ij}^+ &= P_{ij}^o \quad i = 1, m \\
P_{ij}^+ &= P_{ij}^o \quad j = 1, n \\
P_{ij}^+ &= P_{ij}^o \quad i = m+1, n \\
P_{ij}^+ &= P_{ij} \quad j = m+1, n
\end{align*}
\]

The state vector itself is updated

where \[ x^+ = x + \frac{1}{\alpha} dy \]

\[
\begin{align*}
b_1 &= S_{11} f_1 + S_{12} f_2 + \cdots + S_{1j} f_j \quad j = 1, m \\
b_2 &= S_{22} f_2 + S_{23} f_3 + \cdots + S_{2j} f_j \quad j = 1, m \\
&\vdots \quad : \\
&\vdots \\
b_i &= S_{ij} f_j \quad j = 1, m \quad i = 1, m
\end{align*}
\]
MAKE MEASUREMENT UPDATE VCLR DESCRIPTION

The following three pages constitute the VCLR for the module. The following area and data descriptions will prove helpful for following it.

Area A:

Cholesky Decomposition.
Finds the square root of the covariance matrix and puts it into a one dimensional array (Reference: Carlson, Neal A., "Fast Triangular Formation of the Square Root Filter," AIAA Journal)

Area B:

Update parameters with are specified to be estimated.
(Reference: Same as for Area A)

Area C:

Put single indexed arrays corresponding to square roots of the covariance into double dimensional arrays.

Area D:

Multiply double dimensional arrays to update covariance; however, do not update covariance when neither the row nor the column element was estimated.

Definition of data names necessary for understanding of VCLR:

N = Maximum number of considered parameters
S = Single dimensional square root of the covariance matrix
R = Measurement noise variance
H = Measurement partials array
IPT = Array specifying if parameter is to be estimated
   0 = Considered only
   1 = Estimated and considered
X = Array of parameters
PO = Two dimensional square root of covariance matrix
PT = PO transpose
PN = New covariance matrix
DZ = Actual-predicted measurement
UPDATE VCLR (continued)

\[
L = \frac{MN+M}{2+1}
\]

\[
II = 1
\]

**DOWHILE** II .LT. N + 1

\[
I = N-II+1
\]

\[
LL = L-1
\]

\[
JJ = 1
\]

**DOWHILE** JJ .LT. I + 1

\[
J = I - JJ + 1
\]

\[
L = L - 1
\]

\[
T = PN(J,I)
\]

\[
T = I.EQ.N
\]

\[
LJ = 1
\]

\[
LI = LL
\]

\[
N1M1 = N - 1
\]

\[
K = 1
\]

**DOWHILE** K .LT. N1M1 + 1

\[
LJ = LJ + K
\]

\[
LI = LI + K
\]

\[
T = T-S(LJ)*S(LI)
\]

\[
K = K + 1
\]

\[
L = EQL.LL
\]

\[
A = 0
\]

**T.LE.0**

\[
ULL
\]

\[
A = 1./SQRT(T)
\]

\[
JJ = JJ + 1
\]

\[
II = II + 1
\]

\[
S(L) = T*A
\]

**Figure A-43**

A-275
UPDATE VCLR (continued)

\[ L = 0 \]
\[ AL = R \]
\[ I = 1 \]

**DO WHILE I .LT. M + 1**

\[ T = 0. \]
\[ K = 1 \]

**DO WHILE K .LT. I + 1**

\[ L = L + 1 \]
\[ T = T + S(L) \times H(K) \]
\[ F(I) = T \]
\[ K = K + 1 \]
\[ I = I + 1 \]

**DO WHILE I .LT. M + 1**

\[ B(I) = 0. \]
\[ BT = AL \]
\[ AL = AL + F(I) \times 2 \]
\[ GM = \text{SQRT}(AL \times BT) \]
\[ A = BT/GM \]
\[ C = F(I)/GM \]
\[ J = 1 \]

**DO WHILE J .LT. I + 1**

\[ L = L + 1 \]
\[ T = S(L) \]
\[ S(L) = A \times T - C \times B(J) \]
\[ B(J) = B(J) + T \times F(I) \]
\[ J = J + 1 \]
\[ I = I + 1 \]

\[ T = \text{IPT}(I) .EQ. 0 \]

\[ X(I) = X(I) + B(I) \times DZ/AL \]
\[ I = I + 1 \]

Figure A-43
SUBROUTINE UPDATE(N,NS,X,PN,PO,H,R,DZ,IPT)

DIMENSION X(N),PN(N,N),PO(N,N),H(N),IPT(N)

DIMENSION S(253),PT(22,22),F(22),B(22)


INCLUDE 'GAIN,COM'

COMMON /GAIN/GAIN(22,REAL*B :iAIN

THE FORMAL PARAMETERS ARE:

N LARGEST NUMBER OF CONSIDERED PARAMETERS
NS NUMBER OF STATE PARAMETERS
X ARRAYS OF ESTIMATED PARAMETERS
PN NEW COVARIANCE MATRIX
PO OLD COVARIANCE MATRIX
H MEASUREMENT PARTIALS ARRAYS
R MEASUREMENT NOISE VARIANCE
DZ ACTUAL - PREDICTED MEASUREMENT DIFFERENCE
SHIP STATE TRANSITION MATRIX
IEST ARRAYS SPECIFYING IF A PARAMETER IS TO BE
ESTIMATED OR CONSIDERED ONLY

NOTE: PO AND PN ENTER WITH THE SAME VALUES AND PO
IS DESTROYED BY INTERNAL PROCESSING.

USER CONTROLLED PARAMETERS TO VARY DEBUG PRINT LEVEL

TIME, TNEXT, TSTOP, TIA, DEL, DIN, DATEO, TMEAS, TRACK, TIS, TISN, DTA, DATEO, TPRINT, DTPRINT

REAL+8 TIME, TNEXT, TSTOP, TIA, DEL, DIN, DATEO, TMEAS, TRACK, TIS, TISN, DTA, DATEO, TPRINT, DTPRINT

THESE ARE THE TIME REFERENCE FRAMES

TIME ATOMIC TIME SINCE INITIALIZATION (SEC)
TNEXT TIME FOR NEXT POSITION INTEGRATION (SEC)
TSTOP RUN TERMINATION TIME (SEC)
TIA ATTITUDE INTEGRATION TIME (SEC)
D.L STEP SIZE (SEC)
TIN POSITION INTEGRATION TIME (SEC)
DTIN STEP SIZE (SEC)
DATEO DATE OF FLIGHT EPOCH (JD)
DATER DATE OF 1950 EPOCH (JD)
UPDATE

01800 * C
01900 * C
02000 * C
02100 * C
02200 * C
02300 * C
02400 * C
02500 * C
02600 * C
02700
02800
02900
03000
03100
03200
03300
03400
03500
03600
03700 0013
03800 0014
03900 0015
04000 0016
04000 0017
04100
04200
04400
04500
04600
04700
04800
04900
05000
05100
05200
05300
05400
05500
05600
05700
05800
05900
06000
06100
06200
06300
06400
06500
06600
06700
06800
06900
07000
07100
07200 0047
07300 0048
07400 0049
07500 0050

* C

TZERO START TIME IN SECS. SINCE DATE
TSLEW TIME NEEDED TO SLEW AND ACQUIRE (SEC)
TIS REAL WORLD REFERENCE TIME (SEC)
TISN TIME FOR NEXT RW POSITION INTEGRATION (SEC)
DATA TOO LARGE AT MEASUREMENT TIME
TPRINT TIME FOR PRINT (SEC)
DTPRINT INCREMENT ON TPRINT (SEC)

COLESKY DECOMPOSITION S = SQRT(P)

CARLSON SQRT FORMULATION
P = SST

CHOLESKY DECOMPOSITION
S = SQUARE ROOT OF P

IF(L.EQ.1) WRITE(6,10)

IF(I.DEBUG .GT. 1) WRITE(I,20)

IF(I.EQ.I) WRITE(6,21)

IF(I.DEBUG .GT. 1) WRITE(6,22)

IF(I.DEBUG .GT. 1) WRITE(6,23)

IF(I.EQ.H) GO TO 70

T = T-S(LJ)*S(LI)

IF(T.LE.0.) GO TO 80
A = 1./SQRT(T)
S(I) = T*A
C COVARIANCE AND STATE UPDATE
C S+ = S
L = 0
AL = R
DO 100 I=1,N
T = 0.
DO 90 K=1,1
L = L+1
T = T+S(L)*H(K)
F(I) = T
L = 0
DO 110 I=1,N
B(I) = 0.
BT = AL
AL = AL+F(I)**2
GM = SQRT(AL*BT)
A = BT/GM
C = F(I)/GM
DO 100 J=1,1
L = L+1
T = T+S(J)
S(L) = A*T-C*B(J)
B(J) = B(J)+F(I)
DO 120 I=1,N
IF(IP(I).LE.0) GO TO 120
CORREC = B(I)*DZ/AL
GAIN(I) = B(I)/AL
I TEMPORARY TO LOOK AT GAIN - JACK
10700 0080 115 FORMAT('UPDADTE GAIN AND CORRECTION FOR ELEMENT ',I2,' = ',F6.11)
12000 0090 2E20.10
12100 0091 CONTINUE
12200 0092 C**********
12300 0093 C TRANSFORM UPPER TRAPEZOIDAL COVARIANCE MATRIX (S) INTO
12400 0094 C CONVENTIONAL SQUARE MATRIX (PN)
12500 0095 C**********
12600 0096 L = 0
12700 0097 C**********
12800 0098 C PUT SINGLE ARRAY S INTO DOUBLE ARRAYS PO AND PT WHERE
12900 0099 PT IS THE TRANSPOSE
13000 0099 C**********
13100 0099 C*** DO 160 I=1,N
13200 0099 C*** DO 160 J=1,N
13300 0099 C*** IF(J.GT.I) GO TO 150
13400 0099 C*** L = L+1
13500 0099 C*** PT(I,J) = S(L)
13600 0099 C*** GO TO 160
13700 0099 C*** 150 PT(I,J) = 0.
13800 0099 C*** 160 PO(J,I) = PT(I,J)
13900 0099 C*** *** C MULTIPY SST TO GET PN, DO NOT UPDATE THE COVARIANCE
14000 0099 C*** WHERE THE ROWS AND COLUMNS CROSS
14100 0099 C*** C**********
14200 0099 C*** DO 190 I=1,N
14300 0099 C*** DO 180 J=1,N
14400 0099 C**********
UPDATER

13300  0093 IF(IPT(I).EQ.0.AND.IPT(J).EQ.0) GO TO 190
13400  0094 T = 0.
13500  0095 DO 170 K = 1,N
13600  0096 170 T = T + PT(I,K) * PT(K,J)
13700  0097 PN(I,J) = T
13800  0098 CONTINUE
13900  0099 190 RETURN
14000  0100 END

PROGRAM SECTIONS

Name                     Bytes Attributes
0 $CODE                  1447     PIC CON REL LCL SHR EXE RD NWRT LONG
1 $PODATA                224      PIC CON REL LCL SHR NOEXE RD NWRT LONG
2 $LOCAL                 6456     PIC CON REL LCL NO SHR NO EXE RD WRT QUAD
3 GAIN                    176      PIC OVR REL GBL SHR NO EXE RD WRT LONG
4 DEBUG                   128      PIC OVR REL GBL SHR NO EXE RD WRT LONG
5 TIME                    136      PIC OVR REL GBL SHR NO EXE RD WRT LONG

ENTRY POINTS

Address Type Name
0-00000000 UPDATE

VARIABLES

Address Type Name   Address Type Name   Address Type Name   Address Type Name
2-00001868 R=8 A    2-00001870 R=8 AL   2-00001878 R=8 BT    2-00001880 R=8 C
2-00001888 R=8 DOREC 5-00000038 R=8 DATE0 5-00000070 R=8 DATER 5-00000020 R=8 DEL
5-00000060 R=8 DTA   5-00000030 R=8 DIN   5-00000080 R=8 DPRINT AP-00000020 R=8 DZ
2-00001890 R=8 GM    2-000018A0 I=4 I    4-00000004 I=4 IDEBUG
2-000018AC I=4 II    2-000018A4 I=4 J    4-00000004 I=4 IENTER
2-000018B0 I=4 L    2-000018B4 I=4 JJ    2-000018B8 R=8 LJ
AP-00000004 I=4 N    2-000018BC I=4 K
2-00001898 R=8 T    5-00000018 R=8 TIME
5-00000058 R=8 TIS   5-00000000 R=8 TSTOP AP-00000000 R=8 R
5-00000078 R=8 TPRINT 5-00000050 R=8 TRACK

ARRAYS

Address Type Name   Bytes Dimensions
2-00001786 R=8 B    176 (22)
2-00001708 R=8 C    176 (22)
3-00000000 R=8 GAIN  176 (22)
AP-00000100 R=8 H    176 (22)
AP-00000020 R=8 IPT   88 (22)
UPDATE
6-Apr-1981 15:04:05 VAX-11 FORTRAN V2.0-2

AP-000000010@ R+8 PN 3872 (22, 22)
AP-000000014@ R+8 PO 3872 (22, 22)
2-00000078 R+8 PT 3872 (22, 22)
2-000000000 R+8 S 2024 (253)
AP-000000000@ R+8 X 176 (22)

LABELS
Address Label Address Label Address Label Address Label
1-00000000 10' 1-00000010A 20' 1-00000035 21' 1-00000095 22' ** 60 0-000002E4 70
0-00000030 80 ** 90 ** 100 ** 110 1-000000A5 115' 0-00000490 120
1-0000006A 122' 1-00000075 123' 1-00000080 124' 1-0000008B 125' 0-00000D6 150 0-000004E3 160
** 170 ** 180 0-0000059D 190

FUNCTIONS AND SUBROUTINES REFERENCED
MTHSOSORT

Total Space Allocated = 8447 Bytes

COMMAND QUALIFIERS
FORTRAN /LIST GCP,INDATA,OUTDATA,RUNG,GNP,LNAV,EHEN,TRUEA,SPRESS,OCCLUT,GPERG,GCPSEQ,VISIBLE,GENENV,TREG,GRYOUT,RATE,EMAT,CM
/CHECK=(NOBOUNDS,OVERFLOW)
/DEBUG=(NOSYMBOLS,TRACEBACK)
/F77 /NOG_FLOATING /14 /OPTIMIZE /WARNINGS /NO_LINES /NOMACHINE_CODE /CONTINUATIONS=19

COMPILATION STATISTICS
Run Time: 4.71 seconds
Elapsed Time: 86.74 seconds
Page Faults: 386
Dynamic Memory: 160 pages
2.8.0 PRINT PERFORMANCE DATA (OUTDATA)

The purpose of this module is to prepare and print output data pertaining to each attitude and navigation time and measurement update.

The Print Performance Module is straightforward as described in the VCLR shown in Figure A-44. However, one portion which needs explanation, is the section involving the T44 transformation matrix. This is explained below.

The error in the quaternion is found through the equation:

\[ \epsilon = R \cdot F^* \]

where

- \( \epsilon \) = error in the quaternion
- \( R \) = real world quaternion
- \( F \) = filter (or predicted) quaternion

This corresponds intuitively to "subtracting" the predicted values of the quaternion from the real world values.\(^1\)

The covariance matrix associated with the error quaternion is calculated by

\[ P_\epsilon = [T_{44}] \begin{bmatrix} P_F \end{bmatrix} [T_{44}]^T \]

where

- \( P_\epsilon \) = uncertainty in rotation about each axis
- \( T_{44} \) = transition matrix
- \( P_F \) = filter covariance

The T44 matrix is found by taking the partials of \( \epsilon \) with respect to each component of the real world quaternion.

\(^1\) An alternate equation, \( \epsilon = F^* \cdot R \) could have been used. However, this equation is slightly less intuitive and was not chosen for this reason. It should be noted that these two equations yield slightly different results due to the fact that quaternion multiplication is not commutative.
\[
\begin{align*}
( F_1 R_1 + F_2 R_2 + F_3 R_3 + F_4 R_4 )
\quad &\quad + 1 \left( -F_2 R_1 + F_1 R_2 - F_4 R_3 + F_3 R_4 \right) \\
&\quad + j \left( -F_3 R_1 + F_4 R_2 + F_1 R_3 - F_2 R_4 \right) \\
&\quad + k \left( -F_4 R_1 - F_3 R_2 + F_2 R_3 + F_1 R_4 \right)
\end{align*}
\]

\[
\tau_{44} = \frac{\partial \varepsilon}{\partial R}
\]

\[
\begin{bmatrix}
\frac{\partial \varepsilon_0}{\partial R_1} & \frac{\partial \varepsilon_0}{\partial R_2} & \frac{\partial \varepsilon_0}{\partial R_3} & \frac{\partial \varepsilon_0}{\partial R_4} \\
\frac{\partial \varepsilon_1}{\partial R_1} & \frac{\partial \varepsilon_1}{\partial R_2} & \frac{\partial \varepsilon_1}{\partial R_3} & \frac{\partial \varepsilon_1}{\partial R_4} \\
\frac{\partial \varepsilon_2}{\partial R_1} & \frac{\partial \varepsilon_2}{\partial R_2} & \frac{\partial \varepsilon_2}{\partial R_3} & \frac{\partial \varepsilon_2}{\partial R_4} \\
\frac{\partial \varepsilon_3}{\partial R_1} & \frac{\partial \varepsilon_3}{\partial R_2} & \frac{\partial \varepsilon_3}{\partial R_3} & \frac{\partial \varepsilon_3}{\partial R_4}
\end{bmatrix}
\]

\[
\begin{bmatrix}
F_1 & F_2 & F_3 & F_4 \\
-F_2 & F_1 & -F_4 & F_3 \\
-F_3 & F_4 & F_1 & -F_2 \\
-F_4 & -F_3 & F_2 & F_1
\end{bmatrix}
\]
It should be noted that
\[ \epsilon = R \cdot F^* = T_{44} \quad R \]

Therefore, \( T_{44} \) can be used both to find the uncertainty in the rotation about each axis (\( P_\epsilon \)) and the error in the actual vs predicted value of the quaternion.

Next, the uncertainty in the error quaternion \( \epsilon \) is transformed to an uncertainty about each axis as follows:

\[
P = \begin{bmatrix}
0 & 0 & 0 & 2 \\
0 & 0 & 2 & 0 \\
0 & 2 & 0 & 0 \\
\end{bmatrix} \quad P_\epsilon = \begin{bmatrix}
0 & 0 & 0 \\
0 & 0 & 2 \\
2 & 0 & 0 \\
\end{bmatrix} \]

\[
= 4 \begin{bmatrix}
P_{\epsilon 44} & P_{\epsilon 43} & P_{\epsilon 42} \\
P_{\epsilon 34} & P_{\epsilon 33} & P_{\epsilon 32} \\
P_{\epsilon 24} & P_{\epsilon 23} & P_{\epsilon 22} \\
\end{bmatrix}
\]

The standard deviations of the uncertainty about each axis are the square roots of the appropriate diagonal elements of the \( P \) matrix. The RSS of these standard deviations is both printed and plotted.
<table>
<thead>
<tr>
<th>T</th>
<th>POST-FLIGHT OPERATION</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>INITIALIZE REAL WORLD QUATERNIANS</td>
</tr>
<tr>
<td></td>
<td>DETERMINE REAL WORLD POSITION AND VELOCITY</td>
</tr>
<tr>
<td>T</td>
<td>TIME FOR PRINT</td>
</tr>
<tr>
<td></td>
<td>INCREMENT PRINT TIME</td>
</tr>
<tr>
<td></td>
<td>PRINT HEADER, MEASUREMENT TYPE, ACTUAL AND FILTER STATES</td>
</tr>
<tr>
<td></td>
<td>COMPUTE $T_{44}$</td>
</tr>
<tr>
<td></td>
<td>COMPUTE ERROR STATE</td>
</tr>
<tr>
<td></td>
<td>PRINT ERROR STATE</td>
</tr>
<tr>
<td></td>
<td>ICALL = 0</td>
</tr>
<tr>
<td></td>
<td>PRINT EARTH, MOON, AND STAR VECTORS</td>
</tr>
<tr>
<td>T</td>
<td>MEASUREMENT FOR ATTITUDE REFERENCE</td>
</tr>
<tr>
<td></td>
<td>NULL</td>
</tr>
<tr>
<td></td>
<td>COMPUTE AND PRINT NAV. STANDARD DEV.</td>
</tr>
<tr>
<td></td>
<td>COMPUTE AND PRINT ATTITUDE STANDARD DEVIATION</td>
</tr>
<tr>
<td></td>
<td>PRINT STATE COVARIANCE</td>
</tr>
<tr>
<td></td>
<td>PRINT ERROR PARAMETERS</td>
</tr>
<tr>
<td></td>
<td>PRINT STATE TRANSITION MATRIX</td>
</tr>
<tr>
<td></td>
<td>WRITE PLOT FILE</td>
</tr>
</tbody>
</table>

Figure A-44
SUBROUTINE OUTDATA(ICALL)
C
C FORMAL PARAMETERS
C ICALL = OUTPUT PRINT INDICATOR
0 INDICATES CALLED AFTER A MEASUREMENT
1 INDICATES CALLED AFTER PROPAGATION
C
C INCLUDE 'DEBUG.COM'
COMMON /DEBUG / ENTER.IDEBUG
C
C USER CONTROLLED PARAMETERS TO VARY DEBUG PRINT LEVEL
C I NTER IF 1, PRINTS WHEN ENTERS MOST SUBROUTINES
C I DEBUG 0-10, HIGHER NUMBER MEANS MORE PRINT
C
C
C INCLUDE 'RESIDUALS.COM'
*
C**********************************************************************************************
C
C INCLUDE 'RESIDUALS/DZHL.M.DZVL.M.DZHS.T1.DZVS.T1.DZHS.T2.DZVS.T2.
C REAL=0 DZHL.M.DZVL.M.DZHS.T1.DZVS.T1.DZHS.T2.DZVS.T2.DXMGPS
C
C INCLUDE 'CONTAL.COM'
COMMON /CONTAL/ MOP.TINT
REAL=8 TINT
C
PROGRAM CONTROL DESCRIPTORS FOR MULTIPLE RUNS
C
MOP MODE OF OPERATION
1 = PREFLIGHT SIMULATION
2 = POSTFLIGHT SIMULATION
3 = MONTE CARLO SIMULATION
C
TINT NUMBER OF SECONDS OF FULL OPERATION PER CYCLE
C
C
C
C INCLUDE 'RVEC.COM'
COMMON /RVEC/ R(3),RM(3),RO(3),RSM(3),RSO(3),RSS(3),SB(3)
REAL=8 R,RF,RM,RO,RSO,RSS,SB,RA,R2,R3,RSMA,RTG
C
C
C THESE ARE RADIUS VECTORS IN ECI AND BODY COORDINATES
C
R EARTH CENTER TO ECI (KM)
RM MOON ECI (KM)
RO SUN ECI (KM)
RSM SPACECRAFT TO MOON ECI (KM)
RSO SUN ECI (KM)
RSS EARTH CENTER TO STAR ECI
RA ABSOLUTE OF VEOR (KM)
R2 SQUARE OF RA (KM 2)
R3 CUBE OF RA (KM 3)
RSMA ABSOLUTE OF RSM (KM)
INCLUDE 'PHIN.COM'

COMMON /PHIN/ PN(6,6),PDN(6,6),PHIN(6,6),COVN(6,6),

REAL PN,PDN,PHIN,COVN,PON

THESE ARE THE NAVIGATION TRANSITION AND COVARIANCE ARRAYS

PN   POSITION STATE TRANSITION MATRIX
PDN  DERIVATIVE OF TRANSITION MATRICES
PHIN AGGREGATE TRANSITION MATRIX
COVN NEW COVARIANCE MATRIX
PON  PREVIOUS COVARIANCE MATRIX

INCLUDE 'ARRAYS.COM'

COMMON /ARRAYS/ T1(3),T2(3),T3(3),T4(10),T11(3,3),T33(3,3),

REAL T1,T2,T3,T4,T11,T33,T44,T66,T77,T8,TS

THESE ARE TEMPORARY STORAGE ARRAYS FOR USE BY ALL MODULES

T1 - T4 SINGLE DIMENSION ARRAYS
T11 - T77 DUAL DIMENSIONED ARRAYS
T11 DUAL ARRAY; OFF DIAGONAL SET TO ZERO

INCLUDE 'TIME.COM'

COMMON /TIME/ TIME,TNEXT,TSTOP,TIA,DEL,TIN,DTN,DATEO,TZERO,

TMEAS,TRACK,TIS,TISN,TDA,TDP,TIP,DPRT,TIPRT

REAL T,TIME,TNEXT,TSTOP,TIA,DEL,TIN,DTN,DATEO,TMEAS,TRACK,TIS,

TISN,TDA,TZERO,TDATE,TIP,DPRT

THESE ARE THE TIME REFERENCE FRAMES

TIME ATOMIC TIME SINCE INITIALIZATION (SEC)
TMEAS TIME FOR NEXT POSITION INTEGRATION (SEC)
TSTOP RUN TERMINATION TIME (SEC)
TIA ATITUDE INTEGRATION TIME (SEC)
TMEAS,TRACK TIME NEEDED TO SLEW AND ACQUIRE (SEC)
TIS REAL WORLD REFERENCE TIME (SEC)
TISN TIME FOR NEXT RW POSITION INTEGRATION (SEC)
TMEAS,TRACK TIA WHEN DEL
INCLUDE 'UPDT.COM'

COMMON /UPDT/ QN(16), QA(16), QDOT(6,6)
REAL*8 QN, QA, QDOT

STATE STIMATION PARAMETERS

QN NAV. DYN. NOISE COVARIANCE DIAGONAL
QA MIN. VALUES FOR ATT. COVARIANCE DIAGONAL
Q CONTRIBUTION TO NAV. COV. FOR DYN. NOISE
QDOT DIFFERENTIAL OF Q

INCLUDE 'PLOT.COM'

COMMON /PLOT/ TP1, TP2
REAL*8 TP1, TP2

PLOTTING INFORMATION

TP1 LOWER ASSCISSA VALUE - TIME (MIN)
TP2 UPPER

INCLUDE 'COMPOSIT.COM'

COMMON /COMPOSIT/ PHI(22,22), QT(22,22), COV(22,22), PD(22,22), IP(22), XT(22), P
REAL*8 PHI, QT, COV, PD, IP, XT, P

PHI = COMPOSIT STATE TRANSITION MATRIX
QT = PROCESS NOISE ARRAY
COV = NEWEST COVARIANCE ARRAY
PD = OLD COVARIANCE ARRAY
IP = ARRAY OF FLAGS INDICATING ESTIMATED AND
CONSIDERED PARAMETERS
XT = COMPOSIT ESTIMATED PLUS CONSIDERED
STATE VECTOR
P = INITIALIZED TRANSITION MATRIX FOR NEXT
INTERVAL

INCLUDE 'TITL.COM'

COMMON /TITL/ ATITLE(40)
LOGICAL*1 ATITLE

ATITLE IS THE TITLE PRINTED AT EACH PRINT TIME
AS WELL AS THE TOP OF EACH PLOT

INCLUDE 'LMTPAR.COM'

COMMON /LMTPAR/ AL, LON, LAT, TBNL(3,3), TNL(3,3), BL(2), SL(2),
BK(2), SKL(2), TNLK(3,3), TIEO(3,3), SIGGCP, THET
REAL*8 AL, TBNL, TNL, BL, SL, BK, SKL, TNLK, TIEO, SIGGCP, LON,
THET

LANDMARK TRACKER PARAMETERS
OUTDATA

00800  * C   AL = ALTITUDE OF LANDMARK (KM)
00900  * C   LON = LONGITUDE OF LANDMARK (DEG)
01000  * C   LAT = LATITUDE OF LANDMARK (DEG)
01100  * C   TBLN = ORIENTATION ARRAY FOR LANDMARK TRACKER
01200  * C   NOMINAL TO BODY
01300  * C   TNL = MISALIGNMENT ARRAY - ACTUAL
01400  * C   TRACER TO NOMINAL
01500  * C   BL = BIAS - ACTUAL (RAD)
01600  * C   SL = NO:SE STANDARD DEVIATION - ACTUAL (RAD)
01700  * C   BAL = BIAS - KNOWLEDGE (RAD)
01800  * C   THET = LOOK ANGLE (RAD)
01900  * C   SKL = NO:SE STANDARD DEVIATION - KNOWLEDGE (RAD)
02000  * C   TIEO = INITIAL EARTH FIXED TO INERTIAL
02100  * C   TRANSFORMATION
02200  * C   TNLK = MISALIGNMENT ARRAY KNOWLEDGE
02300  * C   TRACKER TO NOMINAL
02400  * C   SIGGCP = POSITION UNCERTAINTY DUE TO CLOUDS
02500  * C   
02500  C *** ICALL = 1 TIME UPDATE
02600  C *** ICALL = 0 MEASUREMENT UPDATE
02700  C
02800  0053
02900  0054
03000  0055
03100  0056
03200  0057
03300  DATA TMLAST /0,
03400  C******C CALL AEROSPACE ROUTINE TO GET REAL WORLD ATTITUDE (TEMPORARY)
03500  C*******C DETERMINE REAL WORLD POSITION AND VELOCITY
03600  C
03700  0058
03800  0059
03900  0060
04000  0061
04100  0062
04200  0063
04300  0064
04400  0065
04500  GO TO 10
04600  C
04700  C
04800  13 DELTAT = TIME - TMLAST
04900  0066
05000  0067
05100  0068
05200  CALL RUNG(INIT, STATE, DSTATE, TMLAST, DELTAT)
05300  TMLAST = TIME
05400  0070
05500  0071
05600  0072
05700  0073
05800  0074
05900  0075
06000  0076
06100  0077
06200  0078
06300  0079
06400  0080
06500  0081
06600  C
06700  C
06800  14 FORMAT(/,5X,7H1,ME = ,F9.3)
06900  15 GROUP TO (150,160,170,180),MCODE
07000  145 FORMAT(/,5X,14N0 MEASUREMENT)
07100  150 WRITE(6,155)
07200  0081
07300  1155 FORMAT(/,5X,'LOOK ANGLE = ',E1.7)
06400 0082 WRITE(6,1156) LON, LAT, AL
06500 0083 1156 FORMAT(/,5X,'LON, LAT, AL = ',3(3X,E15.7))
06600 0084 GO TO 15
06700 0085 155 FORMAT(/,5X,44HLANDMARK TRACKER MEASUREMENTS HAVE BEEN MADE)
06800 0086 160 WRITE(6,165)
06900 0087 GO TO 15
07000 0088 165 FORMAT(/,5X,31HPGPS MEASUREMENTS HAVE BEEN MADE)
07100 0089 170 WRITE(6,175)
07200 0090 GO TO 15
07300 0091 175 FORMAT(/,5X,41HSTARTRACKER 1 MEASUREMENTS HAVE BEEN MADE)
07400 0092 180 WRITE(6,185)
07500 0093 185 FORMAT(/,5X,41HSTARTRACKER 2 MEASUREMENTS HAVE BEEN MADE)
07600 0094 190 WRITE(6,195)
07700 0095 195 FORMAT(/,5X,50HACTUAL/FILTER STATES -- X,Y,Z,XD,YD,ZD,EO,El,E2,E3)
07900 0096 220 CONTINUE
08000 0097 T4(1,1) = STAT(7)
08100 0098 T4(2,2) = STAT(7)
08200 0099 T4(3,3) = STAT(7)
08300 0100 T4(4,4) = STAT(7)
08400 0101 T4(4,1) = STATE(8)
08500 0102 T4(3,4) = STATE(8)
08600 0103 T4(1,2) = STAT(8)
08700 0104 T4(4,3) = STAT(8)
08800 0105 T4(3,1) = STATE(9)
08900 0106 T4(4,2) = STATE(9)
09000 0107 T4(1,3) = STAT(9)
09100 0108 T4(2,4) = STAT(9)
09200 0109 T4(4,1) = STATE(10)
09300 0110 T4(2,3) = STATE(10)
09400 0111 T4(1,4) = STAT(10)
09500 0112 T4(3,2) = STAT(10)
09600 0113 CALL MATAB(144,1,1,7,4,4,1)
09700 0114 C***********************************************************************
09800 0115 C PERFORM QUATERNION MULTIPLY
09900 0116 C EPS=(FILTERQUATERNION)XREALWORLDQUATERNIONSTAR
09900 0117 C***********************************************************************
10000 0118 1000 CALL MATAB(T44,1,1,7,3,1,4,1)
10100 0119 C***********************************************************************
10200 0120 C MTYPE = 1 FOR NAVIGATION
10300 0121 C 2 FOR ATTITUDE REFERENCE
10400 0122 C 3 FOR CALIBRATION
10500 0123 C 4 FOR CRUISE FLIGHT
10600 0124 C***********************************************************************
10700 0125 GO 20 I=1,6
10800 0126 20 T4(I) = STATE(I)-X/I
10900 0127 IFR (TIME .LT. TPAINT) GO TO 120
11000 0128 TPRINT=TPRINT+DTPRINT
11100 0129 30 WRITE(6,40) T4
11200 0129 40 FORMAT(5X,18HSTATE ERROR VECTOR/2X,3E14.7,7E12.4,/)  
11300 0129 C***********************************************************************
11400 0130 C WRITE OUT DATA FOR PLOTTING ROUTINES
11500 0130 C***********************************************************************
11600 0131 IF (CALLEQ.0) WRITE(6,115) RM, RO, RSS
11700 0132 115 FORMAT(4X,'1 EARTH TO MOON ',7X,3E22.14,/.4X,  
11800 0132 1 EARTH TO SUN ',7X,3E22.14,/.4X,  
11900 0132 1 UNIT VECTOR TO STAR ',3E22.14)
12000 0132 IF (MTYPE.EQ.2.AND.ICALL.EQ.0) GO TO 70
OUTOATA

12100 0123 PNAV = SORT(COV(1,1) + COV(2,2) + COV(3,3))
12200 0124 DO 50 I = 1, 3
12300 0125 50 T(i) = SORT(COV(i,i))
12400 0126 WRITE(6,60) T1(i), PNAVF
12500 0127 60 FORMAT(/5X, 29HNAVIGATION STANDARD DEVIATION,F7.0, 4E16.6)
12600 0128 ENAV = SQRT(T4(i) + 2 * T4(2) + 2 * T4(3))
12700 0129 IF (I.EQ.1 .AND. CALL.EQ.0) GO TO 120
12800 0130 70 DO 80 I = 1, 4
12900 0131 DO 80 J = 1, 4
13000 0132 T4(i,j) = T4(i, j)
13100 0133 80 PATT(i,j) = COVA(i,j)
13200 0134 CALL MATAB(T4T, PATT, PATT, 4, 4)
13300 0135 CALL MATAB(T4, PATT, PATT, 4, 4)
13400 0136 PATTF = 2.0 * SQRT(PATT(2,2) + PATT(3,3) + PATT(4,4))
13500 0137 DO 90 I = 2, 4
13600 0138 90 PATT(i, i) = 2.0 * SQRT(PATT(i, i))
13700 0139 WRITE(6, 100) PATT(i, i), 2.0 * SQRT(PATT(i, i))
13800 0140 100 FORMAT(/5X, 27HATTITUDE STANDARD DEVIATION, F7.0, 4F16.9)
13900 0141 WRITE(6, 992)
14000 0142 992 FORMAT(/4X, 9HSTATE COVARIANCE ')
14100 0143 DO 105 I = 1, 22
14200 0144 II I = I
14300 0145 IF (I.GT.8) II = 8
14400 0146 WRITE(6, 103) (COV(i,j), J = 1, II)
14500 0147 103 FORMAT(5X, 8E15.8)
14600 0148 105 CONTINUE
14700 0149 WRITE(6, 106)
14800 0150 106 FORMAT(15H1./5X, '----- (CONTINUED) ')
14900 0151 DO 107 I = 9, 22
15000 0152 II I = I
15100 0153 IF (I.GT.16) II = 16
15200 0154 WRITE(6, 103) (COV(i,j), J = 9, II)
15300 0155 WRITE(6, 108)
15400 0156 108 FORMAT(/5X, '----- (CONTINUED) ')
15500 0157 DO 109 I = 17, 22
15600 0158 109 WRITE(6, 103) (COV(i,j), J = 17, I)
15700 0159 WRITE(6, 110) (E(I), I = 1, 16)
15800 0160 110 FORMAT(/5X, 19HATTITUDE PARAMETERS/5X, 11HQUATERNIONS, 9X, 4E16.9/
15900 5X, 10HGYRO DRIFT, 10X, 3E16.9/5X, 17HGYRO SCALE FACTOR, 3X, 3E16.9/
16000 5X, 20HGYRO NON-ORTH ORIENTATION, 5X, 3E16.9/5X, 20HRELATIVE ORIENTATION.
16100 1600 16100 110 FORMAT(/5X, 'STATE TRANSITION MATRIX ')
16200 0161 EATT = 2.0 * SQRT(T4(8) + 2 * T4(9) + 2 * T4(10))
16300 0162 WRITE(6, 111)
16400 0163 111 FORMAT(/5X, ' STATE TRANSITION MATRIX ')
16500 0164 DO 112 I = 1, 10
16600 0165 II I = I
16700 0166 IF (I.GT.8) II = 8
16800 0167 112 WRITE(6,103) (PHI(i,j), J = 1, II)
16900 0168 WRITE(6, 108)
17000 0169 113 I = 9, 10
17100 0170 II I = I
17200 0171 IF (J.GT.10) II = 10
17300 0172 113 WRITE(6, 103) (PHI(i,j), J = 9, II)
17400 0173 C**************************************************************
17500 C CREATE FILE LISTING TIME, MEASUREMENT TYPE JUST TAKEN.
17600 C ESTIMATE OF QUATERNION, DIFFERENCE BETWEEN MEASUREMENT AND PREDICTED
17700 C MEASUREMENT, ESTIMATE OF POSITION AND VELOCITY AT CERTAIN TIME PTS.
OUTDATA

17800 0173 120 WRITE(12,45) TIME, T4, (SORT(COV(J1,J1)), J1=1,6),
18000 0173 120 (PATT(J1,J1), J1=1,4),
18100 0173 120 D2HLM.D2VLM,D2HST1,D2VST1,D2HST2,D2VST2,DXMGPS,
18200 0173 120 (STAT(J1,J1), J1=1,6)
18300 0174 45 IF (ICALL.EQ.0) GO TO 130
18400 0175 145 IF (PTIME.EQ.TPUNCH) GO TO 130,
18500 0176 145 PTIME = TIA
18600 0177 IF (MTYPE.EQ.1) PTIME = TIN
18700 0178 IF (PTIME.LT.TPUNCH) GO TO 130,
18800 0179 145 C WRITE(11,125) PTIME, MTYPE, MSET,
18900 0179 145 C (E(I).I=1,4),DZ,(X(I).I=1,6)
19000 0179 125 FORMAT(1X,F10.1,1X,2(I2,13X).4(F11.9,1X),E11.5,1X,/,
19100 0180 130 RETURN
19200 0180 130 RETURN
19300 0181 END

PROGRAM SECTIONS

Name Bytes Attributes

0 $CODE 2002 PIC CON REL LCL SHR EXE RD NOWRT LONG
1 $PDATA 851 PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2 $LOCAL 912 PIC CON REL LCL MOSHR NOEXE RD WRT QUAD
3 DEHRG 6 PIC OVR REL GBL SHR NOEXE RD WRT LONG
4 RESIDUALS 96 PIC OVR REL GBL SHR NOEXE RD WRT LONG
5 CONTRO 12 PIC OVR REL GBL SHR NOEXE RD WRT LONG
6 RVEC 224 PIC OVR REL GBL SHR NOEXE RD WRT LONG
7 ENVIR 404 PIC OVR REL GBL SHR NOEXE RD WRT LONG
8 ASTATE 160 PIC OVR REL GBL SHR NOEXE RD WRT LONG
9 TARGETS 36 PIC OVR REL GBL SHR NOEXE RD WRT LONG
10 NSTATE 112 PIC OVR REL GBL SHR NOEXE RD WRT LONG
11 PHIA 7176 PIC OVR REL GBL SHR NOEXE RD WRT LONG
12 PHIN 1440 PIC OVR REL GBL SHR NOEXE RD WRT LONG
13 ARRAYS 1096 PIC OVR REL GBL SHR NOEXE RD WRT LONG
14 TIME 136 PIC OVR REL GBL SHR NOEXE RD WRT LONG
15 UPGT 752 PIC OVR REL GBL SHR NOEXE RD WRT LONG
16 PLOT 16 PIC OVR REL GBL SHR NOEXE RD WRT LONG
17 COMPOSIT 19624 PIC OVR REL GBL SHR NOEXE RD WRT LONG
18 TITLE 40 PIC OVR REL GBL SHR NOEXE RD WRT LONG
19 LMTPAR 392 PIC OVR REL GBL SHR NOEXE RD WRT LONG

ENTRY POINTS

Address Type Name

0-00000000 OUTDATA
### Variables

<table>
<thead>
<tr>
<th>Address</th>
<th>Type</th>
<th>Name</th>
<th>Address</th>
<th>Type</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>19-00000000</td>
<td>R+B</td>
<td>AL</td>
<td>14-00000000</td>
<td>R+B</td>
<td>DATEO</td>
</tr>
<tr>
<td>2-0000023B</td>
<td>R+B</td>
<td>DELTAT</td>
<td>14-00000009</td>
<td>R+B</td>
<td>DTA</td>
</tr>
<tr>
<td>2-00000240</td>
<td>R+B</td>
<td>DZ</td>
<td>4-00000000</td>
<td>R+B</td>
<td>DZHLM</td>
</tr>
<tr>
<td>4-00000008</td>
<td>R+B</td>
<td>DZVLM</td>
<td>4-00000018</td>
<td>R+B</td>
<td>DZVST1</td>
</tr>
<tr>
<td>2-0000027B</td>
<td>R+B</td>
<td>ENAV</td>
<td>2-00000294</td>
<td>I+4</td>
<td>I</td>
</tr>
<tr>
<td>3-00000000</td>
<td>I+4</td>
<td>ENTER</td>
<td>2-0000028C</td>
<td>I+4</td>
<td>II</td>
</tr>
<tr>
<td>9-00000000</td>
<td>I+4</td>
<td>IS</td>
<td>2-00000288</td>
<td>I+4</td>
<td>J</td>
</tr>
<tr>
<td>19-00000010</td>
<td>R+B</td>
<td>LAT</td>
<td>19-00000808</td>
<td>R+B</td>
<td>LON</td>
</tr>
<tr>
<td>9-00000000</td>
<td>I+4</td>
<td>MTYPE</td>
<td>9-00000008</td>
<td>I+4</td>
<td>N5</td>
</tr>
<tr>
<td>2-0000025B</td>
<td>R+B</td>
<td>PNAVF</td>
<td>2-00000260</td>
<td>R+B</td>
<td>TIME</td>
</tr>
<tr>
<td>6-00000088</td>
<td>R+B</td>
<td>PR1</td>
<td>6-00000048</td>
<td>R+B</td>
<td>RA</td>
</tr>
<tr>
<td>6-00000060</td>
<td>R+B</td>
<td>RSMA</td>
<td>19-00000178</td>
<td>R+B</td>
<td>SIGCP</td>
</tr>
<tr>
<td>14-00000000</td>
<td>R+B</td>
<td>TIME</td>
<td>14-00000208</td>
<td>R+B</td>
<td>TIN</td>
</tr>
<tr>
<td>14-00000050</td>
<td>R+B</td>
<td>TISN</td>
<td>14-00000408</td>
<td>R+B</td>
<td>TMEAS</td>
</tr>
<tr>
<td>16-00000000</td>
<td>R+B</td>
<td>TPI</td>
<td>16-00000008</td>
<td>R+B</td>
<td>TP2</td>
</tr>
<tr>
<td>2-00000270</td>
<td>R+B</td>
<td>TPUNCH</td>
<td>14-00000050</td>
<td>R+B</td>
<td>TRACK</td>
</tr>
</tbody>
</table>

### Arrays

<table>
<thead>
<tr>
<th>Address</th>
<th>Type</th>
<th>Name</th>
<th>Bytes</th>
<th>Dimensions</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-00000180</td>
<td>R+B</td>
<td>AT</td>
<td>72</td>
<td>(3, 3)</td>
</tr>
<tr>
<td>18-00000000</td>
<td>L+1</td>
<td>ATITLE</td>
<td>40</td>
<td>(40)</td>
</tr>
<tr>
<td>19-000000CB</td>
<td>R+B</td>
<td>BKL</td>
<td>16</td>
<td>(2)</td>
</tr>
<tr>
<td>19-000002BA</td>
<td>R+B</td>
<td>BL</td>
<td>16</td>
<td>(2)</td>
</tr>
<tr>
<td>17-00001E40</td>
<td>R+B</td>
<td>COV</td>
<td>3872</td>
<td>(22, 22)</td>
</tr>
<tr>
<td>11-00000C00</td>
<td>R+B</td>
<td>COVA</td>
<td>2048</td>
<td>(16, 16)</td>
</tr>
<tr>
<td>12-000003E0</td>
<td>R+B</td>
<td>COVN</td>
<td>2048</td>
<td>(16, 16)</td>
</tr>
<tr>
<td>8-00000070</td>
<td>R+B</td>
<td>D</td>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>8-00000088</td>
<td>R+B</td>
<td>DO</td>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>8-00000000</td>
<td>I+8</td>
<td>DE</td>
<td>32</td>
<td>(4)</td>
</tr>
<tr>
<td>2-0000025B</td>
<td>R+B</td>
<td>DSTATE</td>
<td>80</td>
<td>(10)</td>
</tr>
<tr>
<td>4-00000000</td>
<td>R+B</td>
<td>DVMPS</td>
<td>48</td>
<td>(6)</td>
</tr>
<tr>
<td>8-00000020</td>
<td>R+B</td>
<td>E</td>
<td>32</td>
<td>(4)</td>
</tr>
<tr>
<td>17-000003C0</td>
<td>I+4</td>
<td>IP</td>
<td>88</td>
<td>(22)</td>
</tr>
<tr>
<td>17-000003C0</td>
<td>I+4</td>
<td>IP</td>
<td>88</td>
<td>(22)</td>
</tr>
<tr>
<td>17-000003E8</td>
<td>R+B</td>
<td>P</td>
<td>3872</td>
<td>(22, 22)</td>
</tr>
<tr>
<td>11-00000000</td>
<td>R+B</td>
<td>PA</td>
<td>128</td>
<td>(4, 4)</td>
</tr>
<tr>
<td>2-00000100</td>
<td>R+B</td>
<td>PAT</td>
<td>128</td>
<td>(4, 4)</td>
</tr>
<tr>
<td>2-00000200</td>
<td>R+B</td>
<td>PATT</td>
<td>128</td>
<td>(4, 4)</td>
</tr>
<tr>
<td>11-00000200</td>
<td>R+B</td>
<td>POA</td>
<td>512</td>
<td>(4, 16)</td>
</tr>
<tr>
<td>17-00000120</td>
<td>R+B</td>
<td>PON</td>
<td>288</td>
<td>(6, 6)</td>
</tr>
<tr>
<td>11-00000400</td>
<td>R+B</td>
<td>PHI</td>
<td>3872</td>
<td>(22, 22)</td>
</tr>
<tr>
<td>11-00000400</td>
<td>R+B</td>
<td>PHIA</td>
<td>2048</td>
<td>(16, 16)</td>
</tr>
<tr>
<td>12-00000240</td>
<td>R+B</td>
<td>PHIN</td>
<td>288</td>
<td>(6, 6)</td>
</tr>
<tr>
<td>12-00000200</td>
<td>R+B</td>
<td>PN</td>
<td>288</td>
<td>(6, 6)</td>
</tr>
<tr>
<td>17-000020B0</td>
<td>R+B</td>
<td>PQ</td>
<td>3872</td>
<td>(22, 22)</td>
</tr>
<tr>
<td>11-00001400</td>
<td>R+B</td>
<td>POA</td>
<td>2048</td>
<td>(16, 16)</td>
</tr>
<tr>
<td>12-00000480</td>
<td>R+B</td>
<td>PON</td>
<td>288</td>
<td>(6, 6)</td>
</tr>
<tr>
<td>7-00000050</td>
<td>R+B</td>
<td>PROFILE</td>
<td>320</td>
<td>(10, 4)</td>
</tr>
<tr>
<td>15-00000080</td>
<td>R+B</td>
<td>Q</td>
<td>288</td>
<td>(6, 6)</td>
</tr>
<tr>
<td>15-00000030</td>
<td>R+B</td>
<td>QA</td>
<td>128</td>
<td>(16)</td>
</tr>
<tr>
<td>15-00000100</td>
<td>R+B</td>
<td>QDOT</td>
<td>288</td>
<td>(6, 6)</td>
</tr>
</tbody>
</table>
OUTDATA

15-00000000 R=8 QN 48 (6)
17-000000F20 R=8 QT 3872 (22, 22)
6-00000000 R=8 R 24 (3)
6-00000018 R=8 RM 24 (3)
6-00000030 R=8 RD 24 (3)
6-00000048 R=8 RSM 24 (3)
6-00000060 R=8 RSO 24 (3)
6-00000078 R=8 RSS 24 (3)
6-000000CR R=8 RTG 24 (3)
6-00000050 R=8 SB 24 (7)
6-00000058 R=8 SF 16 (2)
19-00000000 DBAO: [D11R.GCP] OUTDATA.FOR: 54
19-000000DB R=8 SKL 16 (2)
2-000001CB R=8 ST 32 (4)
7-00000000 R=8 STATE 80 (10)
13-00000000 R=8 T1 24 (3)
13-00000098 R=8 T11 72 (3, 3)
13-00000018 R=8 T2 24 (3)
13-00000030 R=8 T3 24 (3)
13-000000E0 R=8 T33 72 (3, 3)
13-00000048 R=8 T4 80 (10)
13-00000128 R=8 T44 128 (4, 4)
2-00000000 R=8 T44T 128 (4, 4)
13-00000038 R=8 T5 32 (4)
13-00000049 R=8 T6 32 (4)
13-000001AB R=8 T66 20U (6, 6)
13-00000428 R=8 T7 32 (4)
13-000002CB R=8 T77 288 (6, 6)
11-00000000 R=8 TA 384 (4, 12)
19-00000018 R=8 TNBL 72 (3, 3)
19-00000130 R=8 TIE0 72 (3, 3)
19-00000060 R=8 TNL 72 (3, 3)
19-000000EB R=8 TNLK 72 (3, 3)
19-00000097 R=8 WD 24 (3)
19-00000040 R=8 X 48 (6)
10-00000030 R=8 XD 48 (6)
10-00000000 R=8 XT 176 (22)

LABELS

Address Label Address Label
0-00000000 5' 1-00000000 14' 0-00000010 15 1-00000019 18'
** 20 1-000000049 13 1-0000000C 45' 1-00000001BE 30' 1-0000001E0 60'
** 30 1-00000165 40' 1-000000345 50' ** 50 1-00000024E 103'
0-00000387 70 1-00000029B 109' 1-00000024E 105' 1-000000325 111'
1-00000256 106' ** 109 1-000000265 110' 1-000000325 155'
** 112 1-0000002E6 115' ** 125' 1-000000265 111'
1-00000001D 145' 0-00000064 155' 1-00000000BE 155' 1-00000000BE 1155'
1-0000009B 175' 1-000000006A 185' 1-0000000234 992' 1-000000321 1155'
1-00000047 1155' 1-000000001F0 220 1-000000234 992' 1-000000321 1155'

Address Label Address Label
0-00000070 10 1-000000019 15 1-000000119 18'
0-00000049 13 1-000000345 45' 1-00000001E0 60'
1-00000001BE 30' ** 50 1-00000024E 103'
** 30 1-00000165 40' 1-00000024E 105' 1-000000325 111'
0-00000387 70 1-00000029B 109' 1-000000265 110' 1-000000325 111'
1-00000256 106' ** 109 1-000000265 110' 1-000000325 111'
** 112 1-0000002E6 115' ** 125' 1-000000265 111'
1-00000001D 145' 0-00000064 155' 1-00000000BE 155' 1-000000321 1155'
1-0000009B 175' 1-000000006A 185' 1-0000000234 992' 1-000000321 1155'
1-00000047 1155' 1-000000001F0 220 1-000000234 992' 1-000000321 1155'
FUNCTIONS AND SUBROUTINES REFERENCED

MATHLIB, MTHDSQRT, RUNG

Total Space Allocated = 35409 Bytes

COMMAND QUALIFIERS

FORTRAN /L OUTDATA

/CHECK=(NOBOUNDS,OVERFLOW)
/DEBUG=(NOSYMBOLS,TRACEBACK)
/F77 /NOG_FLOATING /14 /OPTIMIZE /WARNINGS /NOD_LINES /NOMACHINE_CODE /CONTINUATIONS=19

COMPILED STATISTICS

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Run Time:</td>
<td>7.31 seconds</td>
</tr>
<tr>
<td>Elapsed Time:</td>
<td>24.43 seconds</td>
</tr>
<tr>
<td>Page Faults:</td>
<td>504</td>
</tr>
<tr>
<td>Dynamic Memory:</td>
<td>169 pages</td>
</tr>
</tbody>
</table>
2.9 Plotting and Utility Routines

The following routines are included for completeness. They are all routines associated with GCPSIM as plotting and utility routines.
SUBROUTINE CLOTBL(X,PCNT)

C SUBROUTINE CLOTBL COMPUTES A PERCENT OF CLOUD COVER
C BASED ON THE RANDOM NUMBER X.

C INPUT PARAMETERS
C X = A RANDOM NUMBER BETWEEN 0-1
C
C OUTPUT PARAMETERS
C PCNT = THE PERCENT CLOUD COVER CALLED BY MEASURE
C
INCLUDE 'CLOUD.COM'

COMMON /CLOUDBL(12)
REAL PCNT THE PERCENTAGE OF CLOUD COVER

INCLUDE 'MODE.COM'

COMMON /MODE/ MODE(10)

MODE(1) = LANDMARK TRACKER SWEEP MODE
0 = RANDOM
1 = FIXED AT INPUT THE
2 = NO DEFAULT TO STAR TRACKER

MODE(2) = CLOUD SELECTION MODE
0 = RANDOM CLOUD DENSITIES BASED ON INPUT TABLES CLOTBL
1 = FIXED DENSITY AT NO CLOUDS
2 = NO CLOUDS WITH 100% CLOUD COVER FOR A SPECIFIED PERIOD (CLOTBL(11,12))

MODE(3-10) NOT SPECIFIED AT PRESENT

INCLUDE 'TIME.COM'

COMMON /TIME/ TIME,TNEXT,TSTOP,TIA,DEL,TIN,DIN,DATEO,TZERO

REAL TIME,TNEXT,TSTOP,TIA,DEL,TIN,DIN,DATEO,TZERO,TMEAS,TRACK,TIS,TISN,DTA,DTA,TPrinter,DTPrinter

REAL+8 TIME,TNEXT,TSTOP,TIA,DEL,TIN,DIN,DATEO,TMEAS,TRACK

TIS,TISN,DTA,DTA,TPrinter,DTPrinter

THESE ARE THE TIME REFERENCE FRAMES

TIME ATOMIC TIME SINCE INITIALIZATION (SEC)
TNEXT TIME FOR NEXT POSITION INTEGRATION (SEC)
TSTOP RUN TERMINATION TIME (SEC)
TIA ATTITUDE INTEGRATION TIME (SEC)
DEL " STEP SIZE (SEC)
TIN POSITION INTEGRATION TIME (SEC)
DTA " STEP SIZE (SEC)
DATEO DATE OF FLIGHT EPOCH (JD)
DATER DATE OF 1950 EPOCH (JD)
TZERO START TIME IN SEC. SINCE DATEO
TSLEW TIME NEEDED TO SLEW AND ACQUIRE (SEC)
TIS REAL WORLD REFERENCE TIME (SEC)
TISN TIME FOR NEXT RW POSITION INTEGRATION (SEC)
CLDTBL  20-Apr-1981 15:05:15  VAX-11 FORTRAN V2.0-2  
               _DBCS:([D11R.GCP]CLDTBL.FOR;16  

Page 3

ARcA_\s

Address  Type  Name  Bytes  Dimensions
    3-00000000  R=8  CLDTBL  96  (12)
    4-00000000  I=4  MODE     40  (10)

LABELS

Address  Label  Address  Label  Address  Label  Address  Label
    0-00000000  10  0-00000000  20  0-00000000  30  0-00000000  100

Total Space Allocated = 427 Bytes

COMMAND QUALIFIERS

FORTRAN /L CLDTBL

/!CHECK=(NOREARFLOW)
/!DEBUG=(NOSYMBOLS,TRACEBACK)
/!77   /!NOFLOATING /!OPTIMIZE /!WARNINGS /!NO_LINES /!NOMACHINE_CODE /!CONTINUATIONS=19

COMPIIATION STATISTICS

Run Time:  1.19 seconds
Elapsed Time:  57.87 seconds
Page Faults:  141
Dynamic Memory:  30 pages
$:COMP.COM
$!
$! COMP.COM COMPILE: ALL THE NECESSARY ROUTINES OR EXECUTION
$! OF THE GCP SIMULATION PROGRAM
$!
$FOR/LIST- I/DEBUG/NOOPTIMIZE-
GCP,-
  INDATA,- MATAB,-
  OUTDATA,- RUNG,-
  DNAV,-
    EPHM,- TRUEA,-
    SPRESS,- OCCULT,-
    GPERT,-
  MATAB,-
  DNAV,-
    EPHM,- TRUEA,-
    SPRESS,- OCCULT,-
    GPERT,-
  GCPSEQ,-
    VISIBLE,- OCCULT,-
  GENENV,-
    TREG,-
    GYROUT,-
    RATE,-
    BMAT,-
    CMAT,-
    MATAB,-
    MINV3,-
    GAUSS,-
    KATT,-
  BMAT,-
  CMAT,-
  MATAB,-
  OMULT,-
  UNIT,-
  RUNG,-
  DNAV,-
    EPHM,- TRUEA,-
    SPRESS,- OCCULT,-
    GPERT,-
  GYRO,-
    KATT,-
    BMAT,-
    CMAT,-
    MATAB,-
    OMULT,-
    UNIT,-
    PDATT,-
    MATAB,-
    PATT,-
    MATAB,-
    MEASURE,-
    RANDU,-
    CDOTBL,-
SUBROUTINE CRAISE (N, COV, COVMIN, IPT)
C**********************************************
C RAISES DIAGONAL ELEMENTS OF THE COVARIANCE MATRIX IF THEY DROP
BETWEEN CERTAIN VALUES AND THEY ARE BEING ESTIMATED.
C
C FORMAL PARAMETERS
N = NUMBER OF ROWS OR COLS IN COVARIANCE MATRIX
COV = COVARIANCE MATRIX
COVMIN = MINIMUM VALUES FOR EACH VARIANCE
IPT = ARRAY OF 1'S AND 0'S, A 1 INDICATING THAT A PARTICULAR
PARAMETER IS BEING ESTIMATED
C
C DIMENSION COV(N,N), COVMIN(N), IPT(N)
C
DIMENSION COV(22,22), COVMIN(22), IPT(22)

DO 100 I = 1, N
 IF (IPT(I).EQ.0) GO TO 100
 IF (COV(I,I).LT.COVMIN(I)) COV(I,I) = COVMIN(I)
 CONTINUE
100 CONTINUE
RETURN
END

PROGRAM SECTIONS

Name Bytes Attributes
0 $CODE 103 PIC CON REL LCL SHR EXE RD NOWRT LONG
2 $LOCAL 64 PIC CON REL LCL NOSHR NOEXE RD WRT LONG

ENTRY POINTS

Address Type Name
0 00000000 CRAISE

VARIABLES

Address Type Name Address Type Name
2 00000000 I*4 I AP-00000004 I*4 N

ARRAYS

Address Type Name Bytes Dimensions
AP-00000000B R*8 COV 3872 (22, 22)
AP-0000000CB R*8 COVMIN 176 (22)
AP-00000010B I*4 IPT 88 (22)
LABELS

<table>
<thead>
<tr>
<th>Address</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-0000000F</td>
<td>100</td>
</tr>
</tbody>
</table>

Total Space Allocated = 167 Bytes

Command Qualifiers

```
FORTRAN /LIST GCP,INDATA,MATAB,OUTDATA, RUNG, DNAV,EPHEM, TRUEA, SPRESS, OCCJLT, GPERT, GCPSRO, VISIBLE, GENV, TREG, GYROUT, RATE, BMAT, CMA

/CHECK+(NOBOUNDS,OVERFLOW)
/DEBUG+(NOSYMBS,TRACEBACK)
/F77 /NOFLOATING /W14 /OPTIMIZE /WARNINGS /NO_LINES /NOMACHINE_CODE /CONTINUATIONS=10
```

Compilation Statistics

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Run Time</td>
<td>0.59 seconds</td>
</tr>
<tr>
<td>Elapsed Time</td>
<td>14.93 seconds</td>
</tr>
<tr>
<td>Page Faults</td>
<td>274 pages</td>
</tr>
<tr>
<td>Dynamic Memory</td>
<td>160 pages</td>
</tr>
</tbody>
</table>
SUBROUTINE CRESIF(XTRANS,XIN,XTRANP,XOUT)

THIS ROUTINE TRANSFORMS ERROR VECTORS FROM INERTIAL TO FLIGHT

REAL*8 XIN(3),XOUT(3),XTRANS(3,3),XTRANP(3,3),XTEMP(3)

DO 10 I=1,3
10 RSUM=0
   DO J=1,3
      RSUM=RSUM+XTRANS(J,I)*XIN(J)
   CONTINUE
   XTEMP(I)=RSUM
   CONTINUE
   RSUM=0
   DO J=1,3
      RSUM=RSUM+XTRANP(J,I)*XTEMP(J)
   CONTINUE
   XOUT(I)=RSUM
   CONTINUE
RETURN
END

PROGRAM SECTIONS

<table>
<thead>
<tr>
<th>Name</th>
<th>Bytes</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>$CODE</td>
<td>169</td>
<td>PIC CON REL LCL SHR EXE RD NOWRT LONG</td>
</tr>
<tr>
<td>$LOCAL</td>
<td>116</td>
<td>PIC CON REL LCL NOSHR NOEXE RD WRT QUAD</td>
</tr>
</tbody>
</table>

ENTRY POINTS

Address  Type  Name
0-0000000  CRESIF

VARIABLES

<table>
<thead>
<tr>
<th>Address</th>
<th>Type</th>
<th>Name</th>
<th>Address</th>
<th>Type</th>
<th>Name</th>
<th>Address</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-00000018</td>
<td>I+4</td>
<td>I</td>
<td>2-00000020</td>
<td>I+4</td>
<td>J</td>
<td>2-0000001C</td>
<td>R+4</td>
</tr>
</tbody>
</table>

ARRAYS

<table>
<thead>
<tr>
<th>Address</th>
<th>Type</th>
<th>Name</th>
<th>Bytes</th>
<th>Dimensions</th>
</tr>
</thead>
<tbody>
<tr>
<td>AP-00000008</td>
<td>R+8</td>
<td>XIN</td>
<td>2/ (3)</td>
<td></td>
</tr>
<tr>
<td>AP-00000010</td>
<td>R+8</td>
<td>XOUT</td>
<td>21 (3)</td>
<td></td>
</tr>
<tr>
<td>2-00000000</td>
<td>R+8</td>
<td>XTEMP</td>
<td>24 (3)</td>
<td></td>
</tr>
</tbody>
</table>
CRESIF

8-Apr-1981 07:38:51
VAX-11 FORTRAN V2.0-2
"DBAo:[DIR.GCP]CRESIF.FOR"

Page 2

AP-00000000C R8 XTRANP
AP-00000000C R8 XTRANS

72 (3, 3)
72 (3, 3)

LABELS

<table>
<thead>
<tr>
<th>Address</th>
<th>Label</th>
<th>Address</th>
<th>Label</th>
<th>Address</th>
<th>Label</th>
<th>Address</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>**</td>
<td>5</td>
<td>**</td>
<td>10</td>
<td>**</td>
<td>15</td>
<td>**</td>
<td>20</td>
</tr>
</tbody>
</table>

Total Space Allocated = 285 Bytes
COMMAND QUALIFIERS

FORTRAN CRESIF/LIS

/CHECK=(NOBOUNDS,OVERFLOW)
/DEBUG=(NOSYMBOLS,TRACEBACK)
/F77 /NOG_FLOATING /A3 /OPTIMIZE /WARNINGS /NOD_LINES /NOMACHINE_CODE /CONTINUATIONS=19

COMPILATION STATISTICS

Run Time: 0.79 seconds
Elapsed Time: 2.00 seconds
Page Faults: 106
Dynamic Memory: 28 pages
SUBROUTINE CRTRAN(XTRANS,XIN,XOUT)
  THIS ROUTINE TRANSFORMS ERROR VECTORS FROM INERTIAL TO FLIGHT
  REAL*8 XIN(3),XOUT(3),XTRANS(3,3),RSUM
  DO 10 I=1,3
      RSUM=0
  DO 5 J=1,3
      RSUM=RSUM+XTRANS(J,I)*XIN(J)
  CONTINUE
  LTOUT(I)=RSUM
  CONTINUE
  RETURN
  END

PROGRAM SECTIONS

<table>
<thead>
<tr>
<th>Name</th>
<th>Bytes</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>CODE</td>
<td>100</td>
<td>PIC CON REL LCL SHR EXE RD NOWRT LONC</td>
</tr>
<tr>
<td>LOCAL</td>
<td>76</td>
<td>PIC CON REL LCL NOSHR NOEXE RD WRT QUAD</td>
</tr>
</tbody>
</table>

ENTRY POINTS

<table>
<thead>
<tr>
<th>Address Type Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-00000000 CRTRAN</td>
</tr>
</tbody>
</table>

VARIABLES

<table>
<thead>
<tr>
<th>Address Type Name</th>
<th>Address Type Name</th>
<th>Address Type Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-00000000 I=4 I</td>
<td>2-00000000C I=4 J</td>
<td>2-00000000 R=8 RSUM</td>
</tr>
</tbody>
</table>

ARRAYS

<table>
<thead>
<tr>
<th>Address Type Name</th>
<th>Bytes</th>
<th>Dimensions</th>
</tr>
</thead>
<tbody>
<tr>
<td>AP-00000000 R=8 XIN</td>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>AP-00000000C R=8 XOUT</td>
<td>24</td>
<td>(3)</td>
</tr>
<tr>
<td>AP-000000004 R=8 XTRANS</td>
<td>72</td>
<td>(3, 3)</td>
</tr>
</tbody>
</table>
CRTRAN

8-Apr-1981 07:38:42  VAX-11 FORTRAN V2.0-2

<table>
<thead>
<tr>
<th>Address</th>
<th>Label</th>
<th>Address</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td>5</td>
<td>*</td>
<td>10</td>
</tr>
</tbody>
</table>

Total Space Allocated = 176 Bytes
COMMA QUALIFIERS

FORTRAN CRTRAN/LIS

/CHECK=(NOBOUNDS,OVERFLOW)
/DEBUG=(NOSYMBOLS,TRACEBACK)
/F77 /NOG_FLOATING /i /OPTIMIZE /WARNINGS /NOLINES /NOMACHINE_CODE /CONTINUATIONS=19

COMPILATION STATISTICS

Run Time: 0.54 seconds
Elapsed Time: 2.24 seconds
Page Faults: 99
Dynamic Memory: 28 pages
0001  REAL 4 X(1000,6), TIME(1000), KARRAY(1000)
0002  OPEN (UNIT=1, NAME='GAIN.CAT', TYPE='OLD', RECORDSIZE=98)
0003  I=1
0004  2 CONTINUE
0005    READ (1,3,END=4) TIME(I), (X(I,J), J=1,6)
0006    3 FORMAT (7F14.7)
0007    I=I+1
0008    GO TO 2
0009    4 CONTINUE
0010    ITNUM=ITNUM+1
0011    6 CONTINUE
0012    WRITE (5,10)
0013    10 FORMAT (' INPUT PLOT NUMBER (-1=EXIT) : ',$)
0014    READ (5,20) ITYPE
0015    20 FORMAT (5)
0016    IF (ITYPE.EQ.-1 OR ITYPE.GT.6) GO TO 9000
0017    DO 30 I=1, ITNUM
0018       XARRAY(I)=X(I, ITYPE)
0019    30 CONTINUE
0020    CALL WPLT21
0021    CALL WQMP (0...0...6)
0022    CALL LINPLT (TIME, KARRAY, ITNUM)
0023    GO TO 6
0024    9000 CONTINUE
0025    END
**PROGRAM SECTIONS**

<table>
<thead>
<tr>
<th>Name</th>
<th>Bytes</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 $CODE</td>
<td>244</td>
<td>PIC CON REL LCL SHR EXE RD NOWRT LONG</td>
</tr>
<tr>
<td>1 $PODATA</td>
<td>62</td>
<td>PIC CON REL LCL SHR NOEXE RD NOWRT LONG</td>
</tr>
<tr>
<td>2 $LOCAL</td>
<td>32080</td>
<td>PIC CON REL LCL NOSHR NOEXE RD WRT LONG</td>
</tr>
</tbody>
</table>

**ENTRY POINTS**

<table>
<thead>
<tr>
<th>Address</th>
<th>Type</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-00000000</td>
<td>GAIN$MAIN</td>
<td></td>
</tr>
</tbody>
</table>

**VARIABLES**

<table>
<thead>
<tr>
<th>Address</th>
<th>Type</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-00007000</td>
<td>1*4</td>
<td>I</td>
</tr>
<tr>
<td>2-00007008</td>
<td>I*4</td>
<td>ITNUM</td>
</tr>
<tr>
<td>2-0000700C</td>
<td>I*4</td>
<td>ITYPE</td>
</tr>
<tr>
<td>2-00007D04</td>
<td>I*4</td>
<td>J</td>
</tr>
</tbody>
</table>

**ARRAYS**

<table>
<thead>
<tr>
<th>Address</th>
<th>Type</th>
<th>Name</th>
<th>Bytes</th>
<th>Dimensions</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-000050C0</td>
<td>R*4</td>
<td>TIME</td>
<td>4000</td>
<td>(1000)</td>
</tr>
<tr>
<td>2-000050C0</td>
<td>R*4</td>
<td>X</td>
<td>24000</td>
<td>(1000,6)</td>
</tr>
<tr>
<td>2-000050C0</td>
<td>R*4</td>
<td>XARAY</td>
<td>4000</td>
<td>(1000)</td>
</tr>
</tbody>
</table>

**LABELS**

<table>
<thead>
<tr>
<th>Address</th>
<th>Label</th>
<th>Address</th>
<th>Label</th>
<th>Address</th>
<th>Label</th>
<th>Address</th>
<th>Label</th>
<th>Address</th>
<th>Label</th>
<th>Address</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-00000014</td>
<td>2</td>
<td>0-00000011</td>
<td>3'</td>
<td>0-00000067</td>
<td>4</td>
<td>0-0000006C</td>
<td>6</td>
<td>1-00000017</td>
<td>10'</td>
<td>1-0000003B</td>
<td>20'</td>
</tr>
<tr>
<td><strong>++</strong></td>
<td>30</td>
<td>0-000000F0</td>
<td>9000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**FUNCTIONS AND SUBROUTINES REFERENCED**

FOR$OPEN
HP7221
LINPLT
QQMP

Total Space Allocated = 32386 Bytes

**COMPILER OPTIONS**

/CHECK=(NOBOUND,OVERFLOW)
/DEBUG=(NOSYMBOLS,TRACEBACK)
/OPTIMIZE /WARNINGS /14 /MOD_LINES
FUNCTION GAUSS(MEAN, SIGMA)
C*---------------------------------------------------------------*
C RANDOM NOISE GENERATOR USING PARAMETER MEAN AND STANDARD DEV
C*---------------------------------------------------------------*
C*---------------------------------------------------------------*
05500 0002   INCLUDE 'DEBUG.COM'
05600 0003   COMMON /DEBUG/ IENTR, IDEBUG
05700 0004   USER CONTROLLED PARAMETERS TO VARY DEBUG PRINT LEVEL
05800 0005   I NTER IF 1, PRINTS WHEN ENTERS MOST SUBROUTINES
05900 0006   I DEBUG 0-10, HIGHER NUMBER MEANS MORE PRINT
06000 0007   REAL*8 MEAN, SIGMA, GAUSS.RAN
06100 0008   DATA 11,12/12345,54321/
06200 0009   GAUSS = 0.
06300 0010   DO 10 I=1,12
06400 0011   GAUSS = GAUSS + RAN(11,12)
06500 0012   GAUSS = MEAN + SIGMA*(GAUSS-8.)
06600 0013   RETURN
06700 0014   END

PROGRAM SECTIONS

<table>
<thead>
<tr>
<th>Name</th>
<th>Bytes</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>$CODE</td>
<td>52</td>
<td>PIC CON REL LCL SHR EXE RD NOWAT LONG</td>
</tr>
<tr>
<td>LOCAL</td>
<td>32</td>
<td>PIC CON REL LCL MOSHR MOEAE RD WRT QUAD</td>
</tr>
<tr>
<td>DEBUG</td>
<td>8</td>
<td>PIC OVR REL GBL SHR MOEAE RD WRT LONG</td>
</tr>
</tbody>
</table>

ENTRY POINTS

<table>
<thead>
<tr>
<th>Address</th>
<th>Type</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-00000000</td>
<td>R=8</td>
<td>GAUSS</td>
</tr>
</tbody>
</table>

VARIABLES

<table>
<thead>
<tr>
<th>Address</th>
<th>Type</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-00000010</td>
<td>I=4</td>
<td>I</td>
</tr>
<tr>
<td>2-00000008</td>
<td>I=4</td>
<td>1</td>
</tr>
<tr>
<td>2-00000000</td>
<td>R=8</td>
<td>MEAN</td>
</tr>
<tr>
<td>3-00000000</td>
<td>R=8</td>
<td>SIGMA</td>
</tr>
</tbody>
</table>

LABELS

<table>
<thead>
<tr>
<th>Address</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>::</td>
<td>10</td>
</tr>
</tbody>
</table>
FUNCTIONS AND SUBROUTINES REFERENCED

FORTRAN

Total Space Allocated = 92 Bytes

COMMAND QUALIFIERS

FORTRAN /LIST GCP,INDATA, MATAB, OUTDATA, RUNG, DNAV, EPHEM, TRUEA, SPRESS, OCCULT, GPERT, GCPSEQ, VISIBLE, GENENV, TREG, GYROUT, RATE, BMAT, CMA

/DEBUG = (NOSYMBOLS, TRACEBACK)

/F77 /NOG_FLOATING /I4 /OPTIMIZE /WARNINGS /NOD_LINES /NOMACHINE_CODE /CONTINUATIONS=19

COMPILATION STATISTICS

Run Time: 0.58 seconds
Elapsed Time: 6.23 seconds
Page Faults: 337
Dynamic Memory: 160 pages
**C**

-- GCPPLOT (GCP PLOT ROUTINES)

* C

0001 INCLUDE 'ARRAY.COM'

* C

COMMON BLOCK USED FOR PLOTTING ROUTINE

* C

0002 COMMON /ARRAY/ X(3000,39), INUM

0003 REAL*4 X,X1(3000,39),X2(3000,39)

0004 INTEGER*2 INUM

0005 EQUIVALENCE (X(1,1),X(1,1)),(X2(1,1),X(1,1,2))

* C

0006 CHARACTER*64 MENS((23,6))

0007 CHARACTER*20 XIT,YIT(4)

0008 CHARACTER*60 XREF(2)

0009 CHARACTER*60 CITLE(38)

0010 LOGICAL*1 ATYPE,AXTIT(20),AYIT(80),ATITLE(1200),ABELLS(10)

0011 LOGICAL*1 AITIT(64),ATYPE,AREF(80),AFRAME(2)

0012 INTEGER*2 IOPTS(4),ITYPE,MAX(4),IOFF(4),IOPTD(38),IFRAME

0013 REAL*4 TIME(3000),YARRAY(12000)

0014 EQUIVALENCE (AXTIT(1),XTIT),(AYIT(1),YTIT(1))

0015 EQUIVALENCE (ATITLE(1),CTITLE(1)),(AREF(1),XREF(1))

0016 DATA IFRAME/1/.

0017 DATA ABELLS/7,7,7,7,7,7,7,7,7/.

0018 DATA IOPTD/2,3,4,2,12,13,14,15,6,7,5,15,16,17,15,9,10,11,9,

0019 19,20,21,19,22,23,24,25,26,27,28,29,30,28,31,32,33.

0020 DATA IMAX/8,8,8,14/.

0021 DATA IOFF/0,8,16,24/.

* C

**DATA MENS/**

' WELCOME to the GCP simulation plot routines.'

* C

These plot routines allow the user to overlay up
to 4 plots as long as the plots to be generated
are of the same units. The user will be prompted
by the command OPTION>> or by the sentence Hit
RETURN to continue >>. The user has the option
of using the HP7221 Hewlett Packard plotter (H)
or the 4014 Tektronix plotter (T). If the user is
using the HP7221, before this task may be run he
must type in ASSIGN TD7: FOR010 so that the
HP7221 will be properly assigned.

* C

The following are the type(s) of plot(s) which
may be generated:

1 POSITION ERRORS (KM)

2 VELOCITY ERRORS (KM/SEC)

3 ATTITUDE ERRORS (ARC SEC)

4 RESIDUALS (KM OR KM/SEC)

EXIT PLOT ROUTINES

**POSITION ERROR PLOTS**

1 NAVIGATION X POSITION
NAVIGATION POSITION ERROR RSS
STANDARD DEV. X POSITION ERROR
STANDARD DEV. Y POSITION ERROR
STANDARD DEV. Z POSITION ERROR
NAVIGATION X VELOCITY ERROR
NAVIGATION Y VELOCITY ERROR
NAVIGATION Z VELOCITY ERROR
NAVIGATION VELOCITY ERROR RSS
STANDARD DEV. X VELOCITY ERROR
STANDARD DEV. Y VELOCITY ERROR
STANDARD DEV. Z VELOCITY ERROR
NAVIGATION VELOCITY ERROR RSS
ATTITUDE ERROR ROLL
ATTITUDE ERROR PITCH
ATTITUDE ERROR YAW
STANDARD DEV. ATTITUDE ERROR ROLL
STANDARD DEV. ATTITUDE ERROR PITCH
STANDARD DEV. ATTITUDE ERROR YAW
L.M. TRACKER RESIDUAL HORIZONTAL
L.M. TRACKER RESIDUAL VERTICAL
STAR TRACKER 1 RESIDUAL HORIZONTAL
STAR TRACKER 1 RESIDUAL VERTICAL
STAR TRACKER 2 RESIDUAL HORIZONTAL
STAR TRACKER 2 RESIDUAL VERTICAL
GPS X POSITION RESIDUAL
GPS Y POSITION RESIDUAL
GPS Z POSITION RESIDUAL
GPS RSS POSITION RESIDUAL
GPS X VELOCITY RESIDUAL
GPS Y VELOCITY RESIDUAL
GPS Z VELOCITY RESIDUAL
GPS RSS VELOCITY RESIDUAL

--- DEFINITION OF X ---

C

X(1,1) = TIME
X(1,2-4) = X,Y,Z POSITION ERRORS
X(1,5-7) = X,Y,Z VELOCITY ERRORS
X(1,8-11) = QUATERNION ATTITUDE ERRORS
X(1,12-14) = STANDARD DEV. X,Y,Z POSITION ERRORS
X(1,15-17) = STANDARD DEV. X,Y,Z VELOCITY ERRORS
X(1,18-21) = STANDARD DEV. QUATERNION ATTITUDE ERRORS
X(1,22-23) = L.M. HORIZ.,VERT. RESIDUALS
X(1,24-25) = STAR TRACKER 1 HORIZ.,VERT. RESIDUALS
X(1,26-27) = STAR TRACKER 2 HORIZ.,VERT. RESIDUALS
X(1,28-30) = GPS X,Y,Z POSITION RESIDUALS
X(1,31-33) = GPS X,Y,Z VELOCITY RESIDUALS
X(n,n,1) = COORDINATES IN INERTIAL FRAME
X(n,n,2) = COORDINATES IN FLIGHT FRAME
C C C--- START PLOTS
C C C--- GET OUTPUT PLOTTING DEVICE TYPE
C 0026 WRITE(5,908)
0027 READ(5,903)AOTYPE
C C C--- OPEN INPUT DATA FILE AND READ THE PLOT TITLE
C 0028 OPEN UNIT=2,NAME='PLOT.DAT',TYPE='OLD',RECORDSIZE=780)
0029 READ(2,907)(ATIT(I),I=1,64)
0030 I=1
0031 10 CONTINUE
0032 READ(2.900,END=20)(X(I,J,1),J=1,39)
0033 I=I+1
0034 GO TO 10
0035 20 CONTINUE
0036 INUM=I-1
0037 WRITE(5,2)INUM
0038 FORMAT(' TOTAL POINTS READ IN : ',14)
C C C--- WRITE OUT FIRST MENU
C 0039 WRITE(5,901)(MENUS(I,1),I=1,23)
0040 WRITE(5,912)
0041 WRITE(5,913)
0042 READ(5,903)AFRAME(1)
0043 WRITE(5,914)
0044 READ(5,903)AFRAME(2)
C C C--- SET UP FLIGHT COORDINATE SIDE OF X IF NECESSARY
C 0045 IF(AFRAME(2).NE.'Y')GO TO 25
0046 WRITE(5,915)
0047 CALL INEFLT
0048 25 CONTINUE
C C C--- PUT UP SECOND MENU
C 0049 CONTINUE
0050 WRITE(5,901)(MENUS(I,2),I=1,23)
0051 WRITE(5,904)
0052 READ(5,905)ITYPE
C C C--- ERROR CHECK ON ITYPE
C 0053 IF(ITYPE.LT.0)GO TO 9999
C 0054 IF(ITYPE.GT.0.AND.ITYPE.LE.4)GO TO 25
0055 WRITE(5,920)ABELLS
0056 WRITE(5,902)
0057 READ(5,903)AOTYPE
0058 GO TO 30
GCPPLTSMAIN

0059 35 CONTINUE
C
0060 40 CONTINUE
C
0061 WRITE(5,901)(MENUS(I,ITYPE+2),I=1,23)
0062 WRITE(5,906)
0063 READ(5,905)(IOPTS(I),I=1,4)
0064 CONTINUE
0065 IF(ITYPE.EQ.3.OR.AFRA ME(2).NE.'Y')GO TO 45
0066 WRITE(5,916)
0067 READ(5,917)IFRAME
0068 IF(IFRAME.LT.1.OR.IFRAME.GT.2)GO TO 43
0069 45 CONTINUE
C
C--- ERROR CHECK IOPT
C
0070 DO 60 I=1,4
0071 IF(IOPTS(I).GE.0.AND.IOPTS(I).LE.IMAX(ITYPE))GO TO 50
0072 WRITE(5,920)ABELLS
0073 WRITE(5,921)
0074 READ(5,922)
0075 GO TO 40
0076 50 CONTINUE
0077 60 CONTINUE
C
C--- DETERMINE THE NUMBER OF PLOT OVERLAYS
C
0078 ITOT=0
0079 DO 70 I=1,4
0080 IF(IOPTS(I).NE.0)ITOT=ITOT+1
0081 70 CONTINUE
C
C--- DETERMINE THE PLOTS UNITS
C
0082 UNIT=1
0083 IF(ITYPE.EQ.2)UNIT=2
0084 IF(ITYPE.EQ.3)UNIT=3
C
C--- IF THIS IS A RESIDUAL PLOT, CHECK FOR UNITS CONFORMITY
C
C--- CHECK FOR EMBEDDED ZERO IN OPTION STRING
C
0085 IF(ITOT.EQ.4)GO TO 68
0086 DO 65 I=ITOT+1,4
0087 IF(IOPTS(I).EQ.0)GO TO 63
0088 WRITE(5,922)ABELLS
0089 WRITE(5,921)
0090 READ(5,922)ATYP.
0091 GO TO 40
0092 63 CONTINUE
0093 65 CONTINUE
0094 68 CONTINUE
C
0095 IERR=0
0096 IF(ITYPE.NE.4)GO TO 80
0097 IT=1
0098 IF(IOPTS(1).GT.10) I=2
0099 IF(I.EQ.2) GO TO 75
0100 DO 73 I=1,ITOT
0101 73 CONTINUE
0102 IF(IOPTS(1).GT.10) I=2
0103 IUNIT=1
0104 GO TO 78
0105 75 CONTINUE
0106 DO 77 I=1,ITOT
0107 IF(IOPTS(1).LT.11) I=2
0108 77 CONTINUE
0109 IUNIT=2
0110 78 CONTINUE
0111 80 CONTINUE
0112 IF(I.EQ.4 .AND. (IOPTS(1).EQ.1 .OR. IOPTS(1).EQ.2)) IUNIT=4
0113 WRITE(5,110) IUNIT
0114 1 FORMAT(' IUNIT=', I10)
0115 IF(I.EQ.0) GO TO 90
0116 WRITE(5,921) ABESSL
0117 WRITE(5,902) ABEELL
0118 READ(5,903) ATYPE
0119 GO TO 40
0120 90 CONTINUE
C--- EVERYTHING CHECKS OUT GOOD SO FAR
C
0121 C--- GET THE TIME FRAME USED FOR THIS PLOT
0122 100 CONTINUE
0123 WRITE(5,909) X(INUM,1,1)
0124 READ(5,910) RSTART,RSTOP
0125 IF(RSTART.LT.RSTOP) GO TO 110
0126 WRITE(5,911) RSTART
0127 110 CONTINUE
0128 IF(RSTOP.LT.X(INUM,1,1)) RSTOP=X(INUM,1,1)
0129 IF(RSTART.LT.X(INUM,1,1)) RSTART=X(INUM,1,1)
C--- GET THE BOUNDS ON THE TIMES
C
0130 DO 120 I=1,INUM-1
0131 IF(RSTART.GE.X(I,1,1), AND RSTOP.LE.X(I+1,1,1)) GO TO 130
0132 120 CONTINUE
0133 130 CONTINUE
0134 ITSTART=I
0135 DO 140 I=1,INUM-1
0136 IF(RSTOP.GE.X(I,1,1), AND RSTART.LE.X(I+1,1,1)) GO TO 150
0137 140 CONTINUE
0138 150 CONTINUE
0139 ITSTOP=I+1
0140 INUM=ITSTOP-ITSTART+1
C--- SET UP TIME ARRAY
C
0141 DO 160 I=1,INUM
0142 160 TIME(I)=X(I,1,1)+ITSTART-1,1,1)
GCPLOT$MAIN

0143 160  CONTINUE
C*********************************************************************
C CREATE THE PLOTS
C*********************************************************************
C
C--- INITIALIZE THE PLOTTING DEVICE
C
0144  IF(AOTYPE.EQ.'T')GO TO 170
0145   CALL HP7221
0146   CALL QMHP(0.,0.,6)
   1 FIRST ARG IS PEN TYPE
0147   0. = BLACK
0148     1. = RED
0149     2. = BLUE
0150     3. = GREEN
0147  GO TO 180
0148  170  CONTINUE
0149   CALL TK
0150  180  CONTINUE
0151   CALL PAGE(8.5,11.)
0152   CALL PHYSOR(1.5,1.)
0153   CALL TITLE(' ',1.,AXIT,20.,AYIT((IUNIT-1)*20+1),20,6.,8.)
C
C--- PUT UP TITLE AND LEGEND
C
0154   CALL HEIGHT(.20)
0155   CALL MESSAG(AXIT,40.,0.,9.7)
0156   WRITE(5,828)(IFRAME,)
0157  828  FORMAT(' IFRAME = ',I6)
0158   CALL MESSAG(AREF((IFRAME-1)*40+1),40.,0.,9.4)
0159   CALL HEIGHT(.15)
0160   DO 190 I=1,ITOT
0161     ITEMP=IOFF(I)OPTS(I)
0162     IF(AOTYPE.EQ.'H')CALL QHMP(FLDAT(I-1),0.,6)
0163     CALL MESSAG(ATITLE(I),ITEMP-1)*50+1),50,2.,9.-(I-1)*.25)
0164  182  CONTINUE
0165  184  CONTINUE
0166   CALL RESAP('D')
0167   GO TO 189
0168  188  CONTINUE
0169  190  CONTINUE
0170   CALL DOT
0171   GO TO 189
0171  192  CONTINUE
0172   CALL DASH
0173  194  CONTINUE
0174   CALL VECTOR(0.,-(I-1)*.25+.08,1.5,9.-(I-1)*.25+.08,0)
0176  196  CONTINUE
0177   CALL QMHP(0.,0.,6)
C
C--- PROCESS CURVES NOW. STOR. VALUES INTO YARRAY
C
0181   DO 500 I=1,ITOT
0182     IPTR=IOFF(I)OPTS(I)
0183     ITEMP=IOPTOF(IPTR)
GO TO THE PROPER OPTION

DO 210 J=1,ITNUM
    YARRAY((I-1)*3000+J) = X(J+ITSTRT-1,ITEMP,IFRAME)
    CONTINUE
GO TO 500

DO 250 J=1,ITNUM
    YARRAY((I-1)*3000+J) = SQRT(X(J+ITSTRT-1,ITEMP,IFRAME)**2 +
                              X(J+ITSTRT-1,ITEMP+1,IFRAME)**2 +
                              X(J+ITSTRT-1,ITEMP+2,IFRAME)**2)
    CONTINUE
GO TO 500

DO 310 J=1,ITNUM
    YARRAY((I-1)*3000+J) = (X(J+ITSTRT-1,ITEMP,IFRAME)**2 +
                              X(J+ITSTRT-1,ITEMP+1,IFRAME)**2 +
                              X(J+ITSTRT-1,ITEMP+2,IFRAME)**2) / 4.848E-5
    CONTINUE

DO 350 J=1,ITNUM
    YARRAY((I-1)*3000+J) = (2.*SQRT(X(J+ITSTRT-1,ITEMP,IFRAME)**2 +
                              X(J+ITSTRT-1,ITEMP+1,IFRAME)**2 +
                              X(J+ITSTRT-1,ITEMP+2,IFRAME)**2)) / 4.848E-5
    CONTINUE

FINO MIN AND MAX.

RMAX = 3000000000.
RMIN = 3000000000.

DO 700 I=1,ITOT
    IF(YARRAY((I-1)*3000+J) .LT. RMIN) RMIN = YARRAY((I-1)*3000+J)
    IF(YARRAY((I-1)*3000+J) .GT. RMAX) RMAX = YARRAY((I-1)*3000+J)

CONTINUE
C--- PLOT GRID AND AXIS
C
0214 CALL GRAF(TIME(1), 'SCALE', TIME(ITNUM), RMIN, 'SCALE', RMAX)
0215 CALL GRID(2,2)

C--- DRAW CURVES
C
0216 DO 800 I=1,1IT
0217 CALL QHMF(FLOAT(I-1), 0., 6)
0218 GO TO (710, 720, 730, 740) I
0219 CALL RESET('DOT')
0220 CALL RESJT('DOT')
0221 GO TO 750
0222 CALL DASH
0223 CALL DASH
0224 GO TO 750
0225 CALL DASH
0226 CALL DASH
0227 CALL DASH
0228 CALL DASH
0229 CALL CURVE(TIME, YARRAY(I-1)*3000+1, ITNUM, 0)
0230 CALL CURVE(TIME, YARRAY(I-1)*3000+1, ITNUM, 0)
0231 CALL CURVE(TIME, YARRAY(I-1)*3000+1, ITNUM, 0)
0232 CALL CURVE(TIME, YARRAY(I-1)*3000+1, ITNUM, 0)
0233 CALL CURVE(TIME, YARRAY(I-1)*3000+1, ITNUM, 0)
0234 CALL CURVE(TIME, YARRAY(I-1)*3000+1, ITNUM, 0)
0235 CALL CURVE(TIME, YARRAY(I-1)*3000+1, ITNUM, 0)
0236 CALL CURVE(TIME, YARRAY(I-1)*3000+1, ITNUM, 0)
0237 CALL CURVE(TIME, YARRAY(I-1)*3000+1, ITNUM, 0)
0238 CALL CURVE(TIME, YARRAY(I-1)*3000+1, ITNUM, 0)
0239 CALL CURVE(TIME, YARRAY(I-1)*3000+1, ITNUM, 0)
0240 CALL CURVE(TIME, YARRAY(I-1)*3000+1, ITNUM, 0)
0241 CALL CURVE(TIME, YARRAY(I-1)*3000+1, ITNUM, 0)
0242 CALL CURVE(TIME, YARRAY(I-1)*3000+1, ITNUM, 0)
0243 CALL CURVE(TIME, YARRAY(I-1)*3000+1, ITNUM, 0)
0244 CALL CURVE(TIME, YARRAY(I-1)*3000+1, ITNUM, 0)
0245 CALL CURVE(TIME, YARRAY(I-1)*3000+1, ITNUM, 0)
0246 CALL CURVE(TIME, YARRAY(I-1)*3000+1, ITNUM, 0)
0247 CALL CURVE(TIME, YARRAY(I-1)*3000+1, ITNUM, 0)
0248 CALL CURVE(TIME, YARRAY(I-1)*3000+1, ITNUM, 0)
0249 CALL CURVE(TIME, YARRAY(I-1)*3000+1, ITNUM, 0)
0250 CALL CURVE(TIME, YARRAY(I-1)*3000+1, ITNUM, 0)
0251 CALL CURVE(TIME, YARRAY(I-1)*3000+1, ITNUM, 0)
0252 CALL CURVE(TIME, YARRAY(I-1)*3000+1, ITNUM, 0)
0253 CALL CURVE(TIME, YARRAY(I-1)*3000+1, ITNUM, 0)
0254 CALL CURVE(TIME, YARRAY(I-1)*3000+1, ITNUM, 0)
0255 CALL CURVE(TIME, YARRAY(I-1)*3000+1, ITNUM, 0)
0256 CALL CURVE(TIME, YARRAY(I-1)*3000+1, ITNUM, 0)
0257 END
### PROGRAM SECTIONS

<table>
<thead>
<tr>
<th>Name</th>
<th>Bytes</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 SCODE</td>
<td>2990</td>
<td>PIC CON REL LCL SHR EXE RD NOWRT LONG</td>
</tr>
<tr>
<td>1 SPDATA</td>
<td>878</td>
<td>PIC CON REL LCL SHR NOEXE RD NOWRT LONG</td>
</tr>
<tr>
<td>2 SLOCAL</td>
<td>71544</td>
<td>PIC CON REL LCL NO SHR NOEXE RD WRT LONG</td>
</tr>
<tr>
<td>3 ARRAY</td>
<td>936002</td>
<td>PIC OVR REL GBL SHR NOEXE RD WRT LONG</td>
</tr>
</tbody>
</table>

### ENTRY POINTS

<table>
<thead>
<tr>
<th>Address</th>
<th>Type</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-00000000</td>
<td></td>
<td>GCPPLOT$MAIN</td>
</tr>
</tbody>
</table>

### VARIABLES

<table>
<thead>
<tr>
<th>Address</th>
<th>Type</th>
<th>Name</th>
<th>Address</th>
<th>Type</th>
<th>Name</th>
<th>Address</th>
<th>Type</th>
<th>Name</th>
<th>Address</th>
<th>Type</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-000015B8 L*1</td>
<td>ADTYPE</td>
<td>2-000015B8 L*1</td>
<td>ATYPE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2-000015B8</td>
<td>I*2</td>
<td>IFRAME</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2-000015B8</td>
<td>I*4</td>
<td>ITYPE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2-000015B8 R*4</td>
<td>RMAX</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2-000015B8 CHAR XITIT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### ARRAYS

<table>
<thead>
<tr>
<th>Address</th>
<th>Type</th>
<th>Name</th>
<th>Bytes</th>
<th>Dimensions</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-000015B8 L*1</td>
<td>ABELLS</td>
<td>10</td>
<td>(10)</td>
<td></td>
</tr>
<tr>
<td>2-000015B8</td>
<td>AFRAME</td>
<td>2</td>
<td>(2)</td>
<td></td>
</tr>
<tr>
<td>2-00000000</td>
<td>AREF</td>
<td>80</td>
<td>(80)</td>
<td></td>
</tr>
<tr>
<td>2-00000000</td>
<td>ATIT</td>
<td>64</td>
<td>(64)</td>
<td></td>
</tr>
<tr>
<td>2-00000000</td>
<td>ATITLE</td>
<td>1900</td>
<td>(1900)</td>
<td></td>
</tr>
<tr>
<td>2-00000000</td>
<td>AXTIT</td>
<td>20</td>
<td>(20)</td>
<td></td>
</tr>
<tr>
<td>2-00000000</td>
<td>AYITIT</td>
<td>80</td>
<td>(80)</td>
<td></td>
</tr>
<tr>
<td>2-00000000</td>
<td>CTITLE</td>
<td>1900</td>
<td>(38)</td>
<td></td>
</tr>
<tr>
<td>2-00000000</td>
<td>I*2</td>
<td>IMAX</td>
<td>8</td>
<td>(4)</td>
</tr>
<tr>
<td>2-00000000</td>
<td>I*2</td>
<td>IOFF</td>
<td>8</td>
<td>(4)</td>
</tr>
<tr>
<td>2-00000000</td>
<td>I*2</td>
<td>IOPTOF</td>
<td>76</td>
<td>(38)</td>
</tr>
<tr>
<td>2-00000000</td>
<td>I*2</td>
<td>IOPTS</td>
<td>8</td>
<td>(4)</td>
</tr>
<tr>
<td>2-00000000</td>
<td>I*2</td>
<td>CHAR MENUS</td>
<td>8832</td>
<td>(23, 6)</td>
</tr>
<tr>
<td>2-00000000</td>
<td>R*4</td>
<td>TIME</td>
<td>12000</td>
<td>(3000)</td>
</tr>
<tr>
<td>3-00000000</td>
<td>R*4</td>
<td>X</td>
<td>936000</td>
<td>(3000, 39, 2)</td>
</tr>
<tr>
<td>3-00000000</td>
<td>R*4</td>
<td>X1</td>
<td>468000</td>
<td>(3000, 39)</td>
</tr>
<tr>
<td>3-00000000</td>
<td>R*4</td>
<td>X2</td>
<td>468000</td>
<td>(3000, 39)</td>
</tr>
<tr>
<td>3-00000000</td>
<td>CHAR XREF</td>
<td>80</td>
<td>(2)</td>
<td></td>
</tr>
<tr>
<td>2-00000000</td>
<td>R*4</td>
<td>YARRAY</td>
<td>48000</td>
<td>(12000)</td>
</tr>
<tr>
<td>2-00000000</td>
<td>CHAR YITIT</td>
<td>80</td>
<td>(4)</td>
<td></td>
</tr>
</tbody>
</table>
### LABELS

<table>
<thead>
<tr>
<th>Address</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-0000001D</td>
<td>'1'</td>
</tr>
<tr>
<td>0-000002A9</td>
<td>'35'</td>
</tr>
<tr>
<td>0-000004BE</td>
<td>'63'</td>
</tr>
<tr>
<td></td>
<td>'77'</td>
</tr>
<tr>
<td></td>
<td>'120'</td>
</tr>
<tr>
<td>0-000006DD</td>
<td>'180'</td>
</tr>
<tr>
<td></td>
<td>'190'</td>
</tr>
<tr>
<td></td>
<td>'310'</td>
</tr>
<tr>
<td></td>
<td>'700'</td>
</tr>
<tr>
<td></td>
<td>'800'</td>
</tr>
<tr>
<td>1-00000063</td>
<td>'904'</td>
</tr>
<tr>
<td>1-00000104</td>
<td>'910'</td>
</tr>
<tr>
<td>1-000001F2</td>
<td>'916'</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Address</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-000002A9</td>
<td>'35'</td>
</tr>
<tr>
<td>0-0000032E</td>
<td>'43'</td>
</tr>
<tr>
<td>0-00000492</td>
<td>'68'</td>
</tr>
<tr>
<td>0-000004F4</td>
<td>'80'</td>
</tr>
<tr>
<td></td>
<td>'140'</td>
</tr>
<tr>
<td>0-0000076</td>
<td>'184'</td>
</tr>
<tr>
<td></td>
<td>'210'</td>
</tr>
<tr>
<td></td>
<td>'350'</td>
</tr>
<tr>
<td></td>
<td>'710'</td>
</tr>
<tr>
<td></td>
<td>'828'</td>
</tr>
<tr>
<td>1-00000039</td>
<td>'900'</td>
</tr>
<tr>
<td></td>
<td>'906'</td>
</tr>
<tr>
<td></td>
<td>'911'</td>
</tr>
<tr>
<td></td>
<td>'917'</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Address</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>'20'</td>
</tr>
<tr>
<td></td>
<td>'500'</td>
</tr>
<tr>
<td></td>
<td>'600'</td>
</tr>
<tr>
<td></td>
<td>'720'</td>
</tr>
<tr>
<td>1-00000081</td>
<td>'720'</td>
</tr>
<tr>
<td>1-0000003F</td>
<td>'901'</td>
</tr>
<tr>
<td>1-00000077</td>
<td>'906'</td>
</tr>
<tr>
<td>1-00000089</td>
<td>'907'</td>
</tr>
<tr>
<td>1-00000140</td>
<td>'912'</td>
</tr>
<tr>
<td>1-00000177</td>
<td>'913'</td>
</tr>
<tr>
<td>1-00000280</td>
<td>'920'</td>
</tr>
</tbody>
</table>

### FUNCTIONS AND SUBROUTINES REFERENCED

<table>
<thead>
<tr>
<th>CHNDOT</th>
<th>CURVE</th>
<th>DASH</th>
<th>NOT</th>
<th>ENDPL</th>
<th>FORRCEN</th>
<th>GRAF</th>
<th>GRID</th>
</tr>
</thead>
<tbody>
<tr>
<td>HEIGHT</td>
<td>WP7221</td>
<td>INELT</td>
<td>MESS</td>
<td>MTH$SORT</td>
<td>PAGE</td>
<td>PHYSS</td>
<td>QOHP</td>
</tr>
<tr>
<td>RESLT</td>
<td>TITLE</td>
<td>TK</td>
<td>VECTOR</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Total Space Allocated = 1011414 Bytes

### COMMAND QUALIFIERS

FORTRAN /LIS/CO:77 GCPLOT

/CHECK=(NOBOUNDS,OVERFLOW)
/DEBUG=(NOSYMBOLS,TRACEBACK)
/F77 /MOD_FLOATING /14 /OPTIMIZE /MARK:VGS /MOD_LINES /NONMACHINE_CODE /CONTINUATIONS=77

### COMPILATION STATISTICS

Run Time: 9.48 seconds
Elapsed Time: 54.60 seconds
Page Faults: 592
Dynamic Memory: 143 pages
SUBROUTINE INEFLT

C---- THIS ROUTINE TRANSFORMS THINGS IN X FROM INERTIAL TO FLIGHT COORDINATES.
C
C INCLUDE 'ARRAY.COM'
C COMMON BLOCK USED FOR FLOATING ROUTINE
C
C COMMON /ARRAY/ X(3000,39,2), NUM
C REAL*4 X,X(3000,39) X(3000,39)
C INTEGER*2 NUM
C EQUIVALENCE (X(1,1),X(1,1,1)),(X(2,1),X(1,1,2))
C
C REAL*8 XTRANS(3,3,3000), XTRANP(3,3,3000)
C REAL*8 XTEMP(3),XTEMP2(3)
C
C---- CREATE TRANSFORMATION ARRAYS
C
C CALL TRANSF(XTRANS,XTRANP)
C
C---- TRANSFER THINGS IN X THAT WILL NOT BE TRANSFORMED
C
DO 10 I=1,NUM
0011 X(I,1,2)=X(I,1,1)
0012 X(I,1,2)=X(I,1,1)
0013 X(I,2,2)=X(I,1,1)
0014 X(I,2,2)=X(I,1,1)
0015 X(I,2,2)=X(I,1,1)
0016 X(I,2,2)=X(I,1,1)
0017 X(I,2,2)=X(I,1,1)
0018 X(I,2,2)=X(I,1,1)
0019 X(I,2,2)=X(I,1,1)
0020 X(I,2,2)=X(I,1,1)
0021 X(I,2,2)=X(I,1,1)
0022 10 CONTINUE
C
C---- CREATE TRANSFORMED ERRORS
C
DO 20 I=1,NUM
0024 XTEMP(1)=X(1,2,1)
0025 XTEMP(2)=X(1,3,1)
0026 XTEMP(3)=X(1,4,1)
0027 CALL CTRAN(XTRANS,XTEMP(1),XTEMP(1))
0028 XTEMP(1)=XTEMP(1)
0029 XTEMP(1)=XTEMP(2)
0030 XTEMP(1)=XTEMP(3)
0031 XTEMP(1)=XTEMP(1)
0032 XTEMP(2)=X(1,6,1)
0033 XTEMP(3)=X(1,7,1)
0034 CALL CTRAN(XTRANS,XTEMP(1),XTEMP(1))
0035 XTEMP(2)=XTEMP(1)
0036 XTEMP(2)=XTEMP(2)
0037 XTEMP(2)=XTEMP(3)
0038 20 CONTINUE
C
C---- CREATE RESIDUALS TRANSFORMED
C
program sections

name                  bytes    attributes
0 $C000               350       PIC CON REL LCL SHR EXE RD NOWRT LONG
2 $LCAL              43200     PIC CON REL LCL NOCHR NOEXE RD WRT QUAD
3 ARRAY              936002     PIC OVR REL GBL SHR NOEXE RD WRT LONG

entry points

address  type  name
0-00000000            INFLT

variables

address  type  name          address  type  name
2-00009780            I*4 I          3-000E4840            I*2 INUM

arrays

address  type  name          bytes    dimensions
3-00000000            R 1 X          936000 (3000, 39, 2)
3-00000000            R*4 XI         468000 (3000, 39)
3-00000000            R*4 X2         468000 (3000, 39)
2-00009780            R*8 XTENP1       24 (3)
2-00009780            R*8 XTENP2       24 (3)
2-000348BC            R*8 XTRANP       216000 (3, 3, 3000)
2-00000000            R*8 XTRANS       216000 (3, 3, 3000)

labels

address  label          address  label          address  label
10                  20
FUNCTIONS AND SUBROUTINES REFERENCED

CRTAN       TRANFM

Total Space Allocated = 1368432 Bytes

COMMAND QUALIFIERS

FORTRAN /LIS :INEFLT

/CHECK=(NOUNDEF,OVERFLOW)
/DEBUG=(NOSYMBOLS,TRACEBACK)
/NOFREE /NOG_FLOATING /14 /OPTIMIZE /WARNINGS /NOMACHINE_CODE /CONTINUATIONS=19

COMPILATION STATISTICS

Run Time: 1.46 seconds
Elapsed Time: 3.50 seconds
Page Faults: 168
Dynamic Memory: 51 pages
$!LINKUP.COM
$!
LINKUP.COM LINKS ALL THE NECESSARY ROUTINES FOR EXECUTION
$!
OF THE GCP SIMULATION PROGRAM
$!
$!LINK/MAP=GCP.MAP/FULL/CROSS_REFERENC. - !/DEB-
  
GCP, 
  
INDATA, 
  
MATAB, 
  
OUTDATA, 
  
RUNG, 
  
DNAV, 
  
EPHEM, 
  
SPRESS, 
  
OCCULT, 
  
GPERT, 
  
MATAB, 
  
EPHEM, 
  
TRUEA, 
  
SPRESS, 
  
OCCULT, 
  
GPERT, 
  
GCPSEQ, 
  
VISIBLE, 
  
OCCULT, 
  
GENENV, 
  
TREG, 
  
GYROUT, 
  
RATE, 
  
BMAT, 
  
CMAT, 
  
MATA B, 
  
MINV3, 
  
GAUSS, 
  
KATT, 
  
BMAT, 
  
CMAT, 
  
MATA B, 
  
QMULT, 
  
UNIT, 
  
RUNG, 
  
DNAV, 
  
EPHEM, 
  
TRUEA, 
  
SPRESS, 
  
OCCULT, 
  
GPERT, 
  
GYRO, 
  
KATT, 
  
BMAT, 
  
CMAT, 
  
MATA B, 
  
QMULT, 
  
UNIT, 
  
PDA T, 
  
MATA B, 
  
PATT, 
  
MATA B, 
  
MEASURE, 
  
RANDU, 
  
CLC-8.
SUBROUTINE MINV3(X,XINV) 
C THIS SUBROUTINE FINDS THE INVERSE OF A 3X3 MATRIX 
C
DIMENSION X(3,3),XINV(3,3),TCOFAC(3,3),EL(4)
REAL*8 DET,SIGN,EL,TCOFAC,X,XINV
C
C FIND TRANSPOS OF COFACTOR MATRIX AND DETERMINANT
C
DO 400 I = 1,3
1000 X(II) = X(II)
1010 DO 400 J = 1,3
1020 I = I
1030 DO 200 I = 1,3
1040 IF (II.EQ.I.OR.JJ.EQ.J) GO TO 200
1050 EL(ICOUNT) = X(II,JJ)
1060 ICOUNT = ICOUNT + 1
200 CONTINUE
300 TCOFAC(J,I) = SIGN*(EL(1)*EL(4)-EL(3)*EL(2))
400 CONTINUE
C FIND DETERMINANT
C
DO 500 J = 1,3
500 DET = X(1,1)*TCOFAC(1,1) + X(2,1)*TCOFAC(1,2) + X(3,1)*TCOFAC(1,3)
500 CONTINUE
C INVERSE = (TRANSPOSE OF COFACTOR MATRIX)/DETERMINANT
C
DO 600 J = 1,3
600 XINVRS(I,J) = TCOFAC(I,J)/DET
600 CONTINUE
RETURN
END
### PROGRAM SECTIONS

<table>
<thead>
<tr>
<th>Name</th>
<th>Bytes</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 $CODE</td>
<td>213</td>
<td>PIC CON REL LCL SHR EXE RD NOWRT LONG</td>
</tr>
<tr>
<td>2 $LOCAL</td>
<td>184</td>
<td>PIC CON REL LCL NOSHR NOEXE RD WRT QUAD</td>
</tr>
</tbody>
</table>

### ENTRY POINTS

<table>
<thead>
<tr>
<th>Address Type</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-00000000</td>
<td>MINV3</td>
</tr>
</tbody>
</table>

### VARIABLES

<table>
<thead>
<tr>
<th>Address Type</th>
<th>Name</th>
<th>Address Type</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-00000068</td>
<td>R8</td>
<td>2-00000078</td>
<td>I*4</td>
</tr>
<tr>
<td>2-0000007C</td>
<td>I*4</td>
<td>2-00000088</td>
<td>J*4</td>
</tr>
</tbody>
</table>

### ARRAYS

<table>
<thead>
<tr>
<th>Address Type</th>
<th>Name</th>
<th>Type</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-0000048</td>
<td>R8</td>
<td>EL</td>
<td></td>
</tr>
<tr>
<td>2-00000000</td>
<td>R8</td>
<td>TCOFAC</td>
<td></td>
</tr>
<tr>
<td>AP-0000004#</td>
<td>R8</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>AP-0000008#</td>
<td>R8</td>
<td>XINVRS</td>
<td></td>
</tr>
</tbody>
</table>

### LABELS

<table>
<thead>
<tr>
<th>Address Type</th>
<th>Name</th>
<th>Address</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-00000059</td>
<td></td>
<td>200</td>
<td>**</td>
</tr>
<tr>
<td></td>
<td></td>
<td>300</td>
<td>**</td>
</tr>
<tr>
<td></td>
<td></td>
<td>400</td>
<td>**</td>
</tr>
<tr>
<td></td>
<td></td>
<td>500</td>
<td>**</td>
</tr>
</tbody>
</table>

Total Space Allocated = 397 Bytes

### COMMAND QUALIFIERS

FORTRAN /LIST GCP,INDATA,MATAB,OUTDATA,RUNG,DNAV,EPHEM,TRUEA,SPRESS,RECCOMPUTE,GPERT,GCPSEQ,VISIBLE,GENENV,TREG,GRYOUT,RATE,BMAT,CMA

/DEBUG=NOBOUND,OVERFLOW
/DEBUG=NOBOUND,OVERFLOW
/F77 /NOG_FLOATING /14 /OPTIMIZE /WARNINGS /NOD_LINES /NOMACHINE_CODE /CONTINUATIONS=19

6-Apr-1981 14:50:37  VAX-11 FORTRAN V2.0-2

Page 2
COMPILATION STATISTICS

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Run Time:</td>
<td>1.08 seconds</td>
</tr>
<tr>
<td>Elapsed Time:</td>
<td>11.19 seconds</td>
</tr>
<tr>
<td>Page Faults:</td>
<td>325</td>
</tr>
<tr>
<td>Dynamic Memory:</td>
<td>160 pages</td>
</tr>
</tbody>
</table>
REAL+4 RSTART,RSTOP

DATA MENU(23)

CHARACTER+40 TITLE(26)

LOGICAL+1 ATIT(1040),YTIT(1040),ATYPE

CHARACTER+40 YTITL(26)

EQUIVALENCE (ATIT(1),TITLE(1)),(YTITL(1),YTIT(1))

DATA MENU/

• DISPLAY TYPES:
• 1 NAVIGATION ERROR X POS 15 STAR TRACKER 2 RES. HORZ.
• 2 NAVIGATION ERROR Y POS 16 STAR TRACKER 2 RES. VERT.
• 3 NAVIGATION ERROR Z POS 17 L.M. TRACKER RESID. HORZ.
• 4 NAVIGATION ERROR RSS POS 18 L.M. TRACKER RESID. VERT.
• 5 NAVIGATION ERROR X VEL. 19 GPS X POSITION RESID.
• 6 NAVIGATION ERROR Y VEL. 20 GPS Y POSITION RESID.
• 7 NAVIGATION ERROR Z VEL. 21 GPS Z POSITION RESID.
• 8 NAVIGATION ERROR RSS VEL. 22 GPS RSS POSITION RESID.
• 9 ATTITUDE ERROR ROLL 23 GPS X VELOCITY RESID.
• 10 ATTITUDE ERROR PITCH 24 GPS Y VELOCITY RESID.
• 11 ATTITUDE ERROR YAW 25 GPS Z VELOCITY RESID.
• 12 ATTITUDE ERROR RSS 26 GPS RSS VELOCITY RESID.
• 13 STAR TRACKER 1 RES. HORZ -1 EXIT
• 14 STAR TRACKER 1 RES. VERT

DATA TITLE/

• NAVIGATION ERROR X POS (KM.)
• NAVIGATION ERROR Y POS (KM.)
• NAVIGATION ERROR Z POS (KM.)
• NAVIGATION ERROR RSS POS (KM.)
• NAVIGATION ERROR X VEL. (KM./SEC.)
• NAVIGATION ERROR Y VEL. (KM./SEC.)
• NAVIGATION ERROR Z VEL. (KM./SEC.)
• NAVIGATION ERROR RSS VEL. (KM./SEC.)
• ATTITUDE ERROR ROLL (ARC SEC.)
• ATTITUDE ERROR PITCH (ARC SEC.)
• ATTITUDE ERROR YAW (ARC SEC.)
• ATTITUDE ERROR RSS (ARC SEC.)
• S.T. 1 HORZ. RESIDUAL (ARC SEC.)
• S.T. 1 VERT. RESIDUAL (ARC SEC.)
• S.T. 2 HORZ. RESIDUAL (ARC SEC.)
• S.T. 2 VERT. RESIDUAL (ARC SEC.)
• L.M. TRACKER HORZ. RES. (ARC SEC.)
• L.M. TRACKER VERT. RES. (ARC SEC.)
• GPS X POS RESIDUAL (KM.)
• GPS Y POS RESIDUAL (KM.)
• GPS Z POS RESIDUAL (KM.)
• GPS RSS POS RESIDUAL (KM.)
• GPS X VEL RESIDUAL (KM./SEC.)
• GPS Y VEL RESIDUAL (KM./SEC.)
• GPS Z VEL RESIDUAL (KM./SEC.)
• GPS RSS VEL RESID. (KM./SEC.)

DATA TITLES/

• NAVIGATION ERROR X POS. VS. TIME
• NAVIGATION ERROR Y POS. VS. TIME
• NAVIGATION ERROR Z POS. VS. TIME
• NAVIGATION RSS POS. ERROR VS. TIME
**NAVIGATION ERROR X VEL. VS. TIME**
**NAVIGATION ERROR Y VEL. VS. TIME**
**NAVIGATION ERROR Z VEL. VS. TIME**
**NAVIGATION RSS VEL. ERROR VS. TIME**
**ATTITUDE ERROR ROLL VS. TIME**
**ATTITUDE ERROR PITCH VS. TIME**
**ATTITUDE ERROR YAW VS. TIME**
**ATTITUDE ERROR RSS VS. TIME**
**STAR TRACKER 1 HORIZ. RES. VS. TIME**
**STAR TRACKER 1 VERT. RES. VS. TIME**
**STAR TRACKER 2 HORIZ. RES. VS. TIME**
**STAR TRACKER 2 VERT. RES. VS. TIME**
**LANDMARK TRACKER HORIZ. RES. VS. TIME**
**LANDMARK TRACKER VERT. RES. VS. TIME**
**GPS X POSITION RESIDUAL VS. TIME**
**GPS Y POSITION RESIDUAL VS. TIME**
**GPS Z POSITION RESIDUAL VS. TIME**
**GPS RSS POSITION RESIDUAL VS. TIME**
**GPS X VELOCITY RESIDUAL VS. TIME**
**GPS Y VELOCITY RESIDUAL VS. TIME**
**GPS Z VELOCITY RESIDUAL VS. TIME**
**GPS RSS VELOCITY RESIDUAL VS. TIME**
0020 2 FORMAT(' TOTAL DATA VALUES READ IN WAS ',I8)
0021 2 WRITE(5,908)
0022 30 READ(5,909)ATYPE
0023 40 CONTINUE
0024 40 CONTINUE
0025 40 WRITE(5,901)(MENU(I),I=1,123)
0026 40 WRITE(5,902)
0027 40 READ(5,903,ERR=40)IYPE
0028 40 IF(ITYPE.EQ.-1)GO TO 890
0029 40 WRITE(5,904)
0030 40 READ(5,905,ERR=40)IRFNUM
0031 40 WRITE(5,905)(X(I,1),X(INUM,1))
0032 40 READ(5,906,ERR=40)RSTART,RSTOP
0033 40 IF(RSTART.LT.RSTOP)GO TO 45
0034 40 WRITE(5,907)
0035 40 GO TO 40
0036 45 CONTINUE
0037 45 IF(RSTOP.GT.X(INUM,1))RSTOP=X(INUM,1)
0038 45 IF(RSTART.LT.X(INUM,1))RSTART=X(INUM,1)
0039 45 IF(ITYPE.EQ. 'H')GO TO 47
0040 45 CALL TK
0041 45 GO TO 48
0042 47 CONTINUE
0043 47 CALL HP7221
0044 47 CALL CMP(0,0,0,0,6)
0045 47 CONTINUE
0046 47 CALL QPMLT
0047 47 CALL XLABEL(' TIME (SEC.) ',14)
0048 47 CALL YLABEL(YTIT(ITYPE-1)+40+1),40)
0049 C--- FIND THE BOUNDS ON THE TIME
0050 DO 50 I=1,INUM-1
0051 IF(RSTART.GE.X(I,1).AND.RSTART.LE.X(I+1,1))GO TO 60
0052 WRITE(5,111)RSTART,X(I,1),X(I+1,1),3F15.7)
0053 50 CONTINUE
0054 60 CONTINUE
0055 ITSTRT=I
0056 DO 70 I=1,INUM-1
0057 IF(RSTOP.GE.X(I,1).AND.RSTOP.LE.X(I+1,1))GO TO 80
0058 70 CONTINUE
0059 80 CONTINUE
0060 ITSSTP=I
0061 ITNUM=ITSSTP-ITSTRT+1
0062 DO 90 I=1,ITNUM
0063 90 TIME(I)=X(I+ITSTRT-1,1)
0064 90 CONTINUE
0065 GO TO (100,150,100,200,100,100,200,300,300,300,400,500,500,
500,500,500,500,500,500,500,500,600,500,500,500,500,500,600)ITYPE
0066 100 CONTINUE
C--- NAVIGATIONAL ERROR TYPE OF PLOT
C ITYPE=ITYPE
0067 ITYPE=ITYPE
0068 IF(ITYPE.LE.3)ITYPE=ITYPE+1
0069 DO 110 I=1,ITNUM
0070      VARAY(I)=X(I+ITSTRT-1,ITYPE1)
0071      CONTINUE
0072      GO TO 800
0073      CONTINUE

C   C--- RSS ERROR TYPE OF PLOT
C
0074      ITYPE1=4
0075      IF (ITYPE.EQ.8) ITYPE1=7
0076      DO 210 I=1,ITNUM
0077      VARAY(I)=SORT(X(I+ITSTRT-1,ITYPE1-2)**2*
*                        X(I+ITSTRT-1,ITYPE1-1)**2*
                        X(I+ITSTRT-1,ITYPE1)**2)
210      CONTINUE

0078      GO TO 800
0079      CONTINUE
0080      CONTINUE

C   C--- ATTITUDE ERROR
C
0081      DO 310 I=1,ITNUM
0082      VARAY(I)=(X(I+ITSTRT-1,ITYPE)*2)/.4848E-5
0083      CONTINUE
0084      GO TO 800
0085      CONTINUE

C   C--- RSS ATTITUDE ERROR
C
0086      DO 410 I=1,ITNUM
0087      VARAY(I)=(2*SQR(X(I+ITSTRT-1,9)**2* X(I+ITSTRT-1,10)**2+X(ITSTRT-1,1,11)**2))/.4848E-5
410      CONTINUE
0088      GO TO 800
0089      CONTINUE

C   C--- RESIDUAL TYPE OF PLOT
C
0090      CONTINUE
0091      ITYPE=EITYPE-1
0092      IF (ITYPE.GE.23) ITYPE1=ITYPE-2
0093      DO 510 I=1,ITNUM
0094      VARAY(I)=X(I+ITSTRT-1,ITYPE1-1)
0095      CONTINUE
0096      GO TO 800

C   C--- RSS TYPE FOR RESIDUALS
C
0097      CONTINUE
0098      ITYPE=18
0099      IF (ITYPE.EQ.26) ITYPE1=21
0100      DO 610 I=1,ITNUM
0101      VARAY(I)=2.*SORT(X(I+ITSTRT-1,ITYPE1)**2*
                        X(I+ITSTRT-1,ITYPE1+1)**2*
                        X(I+ITSTRT-1,ITYPE1+2)**2)
610      CONTINUE

0102      GO TO 800
0103      CALL LINPLT(TIME,YARAY,ITNUM)
0104      CONTINUE
0105      CALL QOHP(1.0,0.0,0.0)
0106      IF (ITYPE.EQ.'H') CALL QOHP(1.0,0.0,0.0)
CALL GRID(3,2)
0106 IF(ATYPE.EQ.'H')CALL QQP(0,0,6)
0108 CALL HEA(IN(AYTI((ITYPE-1)40+1),-40,-2,1)
0109 CALL ENDPL(0)
0110 GO TO 40
0111 CONTINUE
0112 890 FORMAT(33F20.10)
0113 900 FORMAT(460)
0114 901 FORMAT(460)
0115 902 FORMAT(' INPUT DISPLAY TYPE NUMBER : ','$)
0116 903 FORMAT(' INPUT REFERENCE FRAME NUMBER : ','$)
0117 904 FORMAT(' INPUT START,STOP TIME FOR PLOT : ','$)
0118 905 FORMAT(' MIN MAX TIMES ARE ':2F10.3,5,$)
0119 906 FORMAT(2F15.6)
0120 907 FORMAT(' BAD START,STOP TIMES WERE INPUT ')
0121 908 FORMAT(' INPUT OUTPUT DEVICE (T OR M) : ','$)
0122 909 FORMAT(A1)
0123 END

PROGRAM SECTIONS

Name Bytes Attributes
0 $CODE 1516 PIC CON REL LCL SHR EXE 7D NOWR' LONG.
1 $DATA 1360 PIC CON REL LCL SHR NOEXE RD NOWR LONG
2 $LOCAL 20372 PIC CON REL LCL NOHR NOEXE RD WRT LONG

ENTRY POINTS

Address Type Name
0-00000000 PLOTSMAIN

VARIABLES

Address Type Name Address Type Name Address Type Name Address Type Name
2-00031B26 L=ATYPE 2-00031B30 I=4 I 2-00031B3C I=4 INUM 2-00031B44 I=4 IFRMUM
2-00031B50 I=4 ITNUM 2-0031B4C I=4 ITSTOP 2-00031B48 I=4 ITSTAT 2-00031B40 I=4 ITYPE
2-00031B54 I=4 IYPE 2-00031B34 I=4 J 2-00031B38 I=4 K 2-00031B28 R=4 RSTART
2-00031B2C R=4 RSTOP

ARRAYS

Address Type Name Bytes Dimensions
2-00000410 L=ATIT 1040 (1040)
2-000031594 CHAR MENU1 1426 (23)
2-00020600 R=4 TIME 8000 (2000)
2-00000410 CHAR TITLES 164000 (2000, 23)
2-00000820 R=4 X
PLTSMAIN

2-00031560 #4 XFILL
2-00027620 #4 YARRAY
2-00000000 #4 YTIT
2-00000000 CHAR YTITLE

PLOTS

2-00031560 8000 (2000)
2-00027620 1040 (1040)
2-00000000 1040 (26)

LABELS

Address Label Address Label Address Label Address Label Address Label Address Label
1-00000000 2 0-00000015 10 0-00000072 20 ** 30 0-000000CE 40 0-00000208 45
0-0000002C 47 0-00000248 ** 50 0-000002DF 60 ** 70 0-00000313 80
** 90 0-00000385 ** 110 1-00000024 111' 0-00000386 200 ** 210
** 300 ** 310 0-00000476 400 ** 410 0-00000406 500 ** 510
0-0000051C 600 ** 610 0-00000590 600 0-000005EB 890 1-00000040 900' 1-00000046 901'
1-00000049 902' 1-00000060 903' 1-00000070 904' 1-00000094 905' 1-00000007 906' 1-0000000D 907'
1-00000101 908' 1-00000125 909'

FUNCTIONS AND SUBROUTINES REFERENCED

ENDPL FORSOPEN GRID HEADIN HP1221 LMPLT MNSSORT OPNPLT
QQHM TK XLABEL YLABEL

Total Space Allocated = 205640 Bytes

COMMAND QUALIFIERS

FORTRAN CO:40 PLOT/LIS

/COMPIIION STATISTICS

Run Time: 4.87 seconds
Elapsed Time: 19.17 seconds
Page Faults: 294
Dynamic Memory: 22 pages
SUBROUTINE PM AS(PU,U,PDH,PDV,IFLAG)

C******************************************************************************
C SUBROUTINE PM AS COMPUTES THE PARTIALS OF THE MEASUREMENT
C WITH RESPECT TO THE ESTIMATED STATE USING THE PARTIAL OF
C THE UNIT VECTOR (PU) WRT THE ESTIMATED STATE AND THE
C UNIT VECTOR ITSELF.
C INPUT PARAMETERS
C 
C  PU = THE PARTIALS OF THE UNIT VECTOR WRT THE
C  ESTIMATED STATE
C  U = THE MEASUREMENT UNIT VECTOR ITSELF
C OUTPUT PARAMETERS
C  PDH = THE PARTIALS OF THE HORIZONTAL DIFLECTION
C  PDV = THE PARTIALS OF THE VERTICAL DIFLECTION
C CALLED BY HSTAR AND HLMT
C******************************************************************************
C
DIMENSION PU(3),U(3)
REAL*8 PDH,PDV,PU,U,A,B,C

IF(IFLAG .GT. 1) GO TO 100

C COMPUTE PARTIALS OF MEASUREMENT VECTOR (APPROXIMATE)
C
PDH=PU(2)
PDV=PU(1)
RETURN

C COMPUTE EXACT PARTIALS OF MEASUREMENT
C
A=1./SQRT(1.-U(1)*U(1))
PDV*A=PU(1)
B=COS(ASIN(U(1)))
C=1./SQRT(1.-(U(2)/B)**2)
PDH=C*(B*PU(2)+U(2)*U(1)*A*PU(1))/(B*B)
RETURN

END
PROGRAM SECTIONS

Name                          Bytes  Attributes
0 $CODE                        163     PIC CON REL LCL SHR EXE RD NOWRT LONG
2 $LOCAL                       64      PIC CON REL LCL NOSHR NOEXE RD WHT QUAD

ENTRY POINTS

Address  Type  Name
0-0:000000  PMEAS

VARIABLES

Address  Type  Name          Address  Type  Name          Address  Type  Name          Address  Type  Name
2-00000000  R+B  A            2-00000000  R+B  B            2-00000010  R+B  C            AP 10000001  I+4  IFLAG
AP-00000000  R+B  PDH          AP-00000010  R+B  PDV

ARRAYS

Address  Type  Name          Bytes  Dimensions
AP-00000000  R+B  PU          24 (3)
AP-00000000  R+B  U           24 (3)

LABELS

Address  Label
0-0000034  100

FUNCTIONS AND SUBRoutines REFERENCED

MTH$DASIN  MTH$DCOS  MTH$DSORT

Total Space Allocated = 227 Bytes

COMMAND QUALIFIERS

FORTTRAN /LIST  GCP,INDATA,MATAB,OUTDATA,RUNG,NAV,EPHEM,TRUEA,SPRESS,OCULT,GPERT,GCPSEQ,VISIBLE,GENENV,TREG,GYROUT,RATE,GMAT,CMA
/CHECK=(NOBOUNDS,OVERFLOW)
/DEBUG=(NOSYM801S,TRACEBACK)
/F77 /NOG_FLOATING /I4 /OPTIMIZE /WARNINGS /NOD_LINES /NOMACHINE_CODE /CONTINUATIONS=19
COMPILATION STATISTICS

Run Time: 0.86 seconds
Elapsed Time: 18.87 seconds
Page Faults: 323
Dynamic Memory: 180 pages
SUBROUTINE OMULT (A,B,C)

C THIS SUBROUTINE DOES A QUAT MULTIPLY AND RETURNS ANSWER IN C

C

DIMENSION A(4),B(4),C(4),D(4)

REAL*8 A,B,C,D

D(1) = A(1)*B(1) - A(2)*B(2) - A(3)*B(3) - A(4)*B(4)
D(2) = A(1)*B(2) + A(2)*B(1) + A(3)*B(4) - A(4)*B(3)
D(3) = A(1)*B(3) - A(2)*B(4) + A(3)*B(1) + A(4)*B(2)
D(4) = A(1)*B(4) + A(2)*B(3) - A(3)*B(2) + A(4)*B(1)
C(1) = D(1)
C(2) = D(2)
C(3) = D(3)
C(4) = D(4)

RETURN

END

PROGRAM SECTIONS

Name Bytes Attributes
0  SCODE  210 PIC CON REL LCL SHR EKE RD NOWRT LONG
2  SLOCAL  84 PIC CON REL LCL NOSHR NOEAE RD WRT QUAD

ENTRY POINTS

Address Type Name
0-00000000  QMULT

ARRAYS

Address Type Name Bytes Dimensions
AP-0000000004 R+B 32 (4)
AP-0000000008 R+B A  32 (4)
AP-0000000012 R+B C  32 (4)
2-00000000 R+B D  24 (3)

Total Space Allocated = 294 Bytes

COMMAND QUALIFIERS

FORTAN /LIST GCP,INDATA,MATAB,OUTDATA,RUNO,EPHEM,TRUEA,SPRESS,OCULT,GPERT,GCPS,PRINT,SHOW,SHOWENV,SHOWTREG,YOURG,RATE,SBMAT,CMAP
/CHECK={NOBND,OVERFLOW}
/DEBUG={NOSYM,TBAK}
/F77 /NOFLOAT /I4 /OPTIMIZE /WARNINGS /NOLINES /NOMACHINE_CODE /CONTINUATIONS=19
COMPILATION STATISTICS

Run Time: 1.07 seconds
Elapsed Time: 14.16 seconds
Page Faults: 293
Dynamic Memory: 160 pages
SUBROUTINE RDOTAT,T,THET,AL

SUBROUTINE RDOTATA READS THE VALUE OF THET AND AL FROM A DATA TABLE AT THE TIME CLOSEST TO TIME T.

INPUT DATA T = TIME OF MEASUREMENT

OUTPUT DATA THET = LOOK ANGLE TO LANDMARK (RAD)
AL = ALTITUDE OF LANDMARK ABOVE EARTH MEAN SURFACE (KM)

CALLED BY RDCGP

THET IS INITIALLY A FLAG
THET .GT. 90. DEG PICK RANDOM THET AND AL
THET .LT. 90. DEG KEEP THET, SET AL=0.

INCLUDE 'MODE.COM'
COMMON /MODE/ MODE(10)

MODE(1) = LANDMARK TRACKER SWEEP MODE
0 = RANDOM
1 = FIXED AT INPUT THET
2 = NO DEFAULT TO STAR TRACKER

MODE(2) = CLOUD SELECTION MODE
0 = RANDOM CLOUD DENSITIES BASED ON INPUT TABLES CLOTBL
1 = FIXED DENSITY AT NO CLOUDS
2 = NO CLOUDS WITH 100% CLOUD COVER FOR A SPECIFIED PERIOD (CLOTBL(11,12))
3-10 = NOT SPECIFIED AT PRESENT

DATA 11,12,IFLAG/12345,54321,0/
REAL+8 AL,T,THET,RAN,GAUSS,X,THETP
IFIIFLAG .GT. 0) GO TO 10
IFIIFLAG .LT. 0) GO TO 20

IFIIFLAG .GE. 1) GO TO 20
IFIIFLAG .LT. 0) GO TO 10

THETP=THET+3.1415926536/180.
GO TO (20,20),MODE(1)

THET=(RAN(11,12))+2.*THETP
X=GAUSS(0.,.5)

AL=ABS(X)
GO TO 100

THETP=THET

IFIIFLAG .GE. 1) GO TO 20
IFIIFLAG .LT. 0) GO TO 10

RETURN
END
PROGRAM SECTIONS

<table>
<thead>
<tr>
<th>Name</th>
<th>Bytes</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 CODE</td>
<td>110</td>
<td>PIC CON REL LCL</td>
</tr>
<tr>
<td>1 SPDATA</td>
<td>8</td>
<td>PIC CON REL LCL</td>
</tr>
<tr>
<td>2 SLOCAL</td>
<td>52</td>
<td>PIC CON REL LCL</td>
</tr>
<tr>
<td>3 MODE</td>
<td>40</td>
<td>PIC OVR REL GBL</td>
</tr>
</tbody>
</table>

ENTRY POINTS

<table>
<thead>
<tr>
<th>Address</th>
<th>Type</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-00000000</td>
<td>RDDTA</td>
<td></td>
</tr>
</tbody>
</table>

VARIABLES

<table>
<thead>
<tr>
<th>Address</th>
<th>Type</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>AP-0000000000</td>
<td>R*8</td>
<td>AL</td>
</tr>
<tr>
<td>AP-0000000004</td>
<td>R*8</td>
<td>T</td>
</tr>
<tr>
<td>AP-0000000010</td>
<td>I*4</td>
<td>I1</td>
</tr>
</tbody>
</table>

ARRAYS

<table>
<thead>
<tr>
<th>Address</th>
<th>Type</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>3-00000000</td>
<td>I*4</td>
<td>MODE</td>
</tr>
</tbody>
</table>

LABELS

<table>
<thead>
<tr>
<th>Address</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-0000002C</td>
<td>10</td>
</tr>
<tr>
<td>0-00000065</td>
<td>20</td>
</tr>
<tr>
<td>0-00000080</td>
<td>100</td>
</tr>
</tbody>
</table>

FUNCTIONS AND SUBROUTINES REFERENCED

FORTRAN GAUSS

Total Space Allocated = 210 Bytes

COMMAND QUALIFIERS

FORTRAN /L RDDTA

/CHECK=(NOBOUNDS,OVERFLOW)
/DEBUG=(NOSYMBS,TRACEBACK)
/FF7 /NOG_FLOAT /14 /OPTIMIZE /WARNINGS /MOD_LINES /NOMACHINE_CODE /CONTINUATIONS=19
COMPILATION STATISTICS

Run Time: 0.83 seconds
Elapsed Time: 7.44 seconds
Page Faults: 149
Dynamic Memory: 39 pages
SUBROUTINE RDGCP

SUBROUTINE RDGCP READS FROM THE DATA BASE LONGITUDE AND ALTITUDE FOR THE INTENDED LANDMARK

INPUT PARAMETERS
THET = LOOK ANGLE ABOUT LMT X-AXIS (RAD)
AL = LANDMARK ALTITUDE ABOVE MEAN (KM)

OUTPUT PARAMETERS
LON = LANDMARK LONGITUDE (DEG)
LAT = LATITUDE (DEG)
AL = ALTITUDE ABOVE MEAN (KM)

CALLED BY MEASURE


INCLUDE 'LMTPAR.COM'
COMMON /LMTPAR/ AL, LON, LAT, TBNL(3,3), TNL(3,3), BL(2), SL(2), BKL(2), SKL(2), TNLK(3,3), TIEO(3,3), SIGGCP, THET

REAL*8, AL, LON, LAT, BL, SL, BKL, SKL, TNLK, TIEO, SIGGCP

LANDMARK TRACKER PARAMETERS
AL = ALTITUDE OF LANDMARK (KM)
LON = LONGITUDE OF LANDMARK (DEG)
LAT = LATITUDE OF LANDMARK (DEG)
TBNL = ORIENTATION ARRAY FOR LANDMARK TRACKER NOMINAL TO BODY
TNL = ORIENTATION ARRAY FOR LANDMARK TRACKER ACTUAL

BL = BIAS - ACTUAL (RAD)
SL = NOISE STANDARD DEVIATION - ACTUAL (RAD)
BKL = BIAS - KNOWLEDGE (RAD)
THET = LOOK ANGLE (RAD)
SKL = NOISE STANDARD DEVIATION - KNOWLEDGE (RAD)
TIEO = INITIAL EARTH FIXED TO INERTIAL TRANSFORMATION

TNLK = MISALIGNMENT ARRAY KNOWLEDGE TRACKER TO NOMINAL
SIGGCP = POSITION UNCERTAINTY DUE TO CLOUDS

INCLUDE 'ENVIR.COM'
COMMON /ENVIR/ STATE(10), PROFILE(10,4), INIT

REAL*8, STATE, PROFILE

REAL WORLD STATE PARAMETERS
STATE VALUES: X, Y, Z, XD, YD, ZD, E0, E1, E2, E3
PROFILE ATTITUDE PROFILE - TIME (SEC) VS INERTIAL ANGULAR RATES (RAD/SEC)
INIT INTEGRATION INITIALIZATION KEY (-1)
INCLUDE 'ARRAYS.COM'

COMMON /ARRAYS/ T1(3),T2(3),T3(3),T4(4),T11(3,3),T33(3,3)

REAL*8 T1,T2,T3,T4,T11,T33

THESE ARE TEMPORARY STORAGE ARRAYS FOR USE BY ALL MODULES

T1 - T4 SINGLE DIMENSION ARRAYS
T11 - T33 DUAL DIMENSIONED ARRAYS

INCLUDE 'TARGETS.COM'

COMMON /TARGETS/ MTYPE,IS,NS,JFLAG,MCODE,PI,TPI

REAL*8 PI,TPI

MEASUREMENT SPECIFICATIONS

MTYPE MEASUREMENT TYPE
JFLAG SET FOR STAR OBSTRUCTION
MCODE "*" MEASUREMENT PROCESSING

INCLUDE 'TIME.COM'

COMMON /TIME/ TIME,TNEXT,TSTOP,TIA,DEL,TIN,DIN,DATED,TZERO

REAL*8 TIME,TNEXT,TSTOP,TIA,DEL,TIN,DIN,DATED,TZERO

THESE ARE THE TIME REFERENCE FRAMES

TIME ATOMIC TIME SINCE INITIALIZATION (SEC)
TNEXT TIME FOR NEXT POSITION INTEGRATION (SEC)
TSTOP RUN TERMINATION TIME (SEC)
TIA ATTITUDE INTEGRATION TIME (SEC)
D L STEP SIZE (SEC)
TIN POSITION INTEGRATION TIME (SEC)
DTN STEP SIZE (SEC)
DATED DATE OF FLIGHT EPOCH (JD)
DATER DATE OF 1950 EPOCH (JD)
TZERO START TIME IN SECS. SINCE DATED
TSLEW TIME NEEDED TO SLEW AND ACQUIRE (SEC)
TIS REAL WORLD REFERENCE TIME (SEC)
TIN TIME FOR NEXT RW POSITION INTEGRATION (SEC)
DTA USUALLY + DEL BUT + TSL EW - TIA WHEN DEL
TPRINT TIME FOR PRINT (SEC)
DPRTIN INCREMENT ON TPRINT (SEC)

INCLUDE 'NSTATE.COM'

COMMON /NSTATE/ XD(6),X(6),RADE

REAL*8 XD,X,RADE,RAD
POSITION STATE AND CONSIDERED PARAMETERS

STATE DERIVATIVES (KM/SEC AND KM/SEC/SEC)

STATE POSITION PARAMETERS (KM AND KM/SEC)

RADIUS OF THE MOON (KM)

EARTH DETECTABLE RADIUS (KM)

DIMENSION UL(3), TLL(3,3), ULL(3,3), TIB(3,3), TIL(3,3),
M(3), L(3), TLE(3,3), TLI(3,3), TEL(3,3), TIL(3,3).

EQUIVALENCE (RE,RAE)

GENERATE UNIT VECTOR ALONG LMT BORESIGHT

UL(1)=0.
UL(2)=0.
UL(3)=1.

READ A LOOK ANGLE TO AND AN ALTITUDE OF THE LANDMARK

FROM THE DATA FILE

LOOK ANGLE IS RELATIVE TO BODY AXES ABOUT THE
X-BODY AXIS

CALL RDTA(TIME,THET,AL)

TRANSFORM UNIT VECTOR BY LOOK ANGLE

CALL AMAT(STATE(7),TBI)

CALL MINV3(TBI,TIB)

CALL MATA8(TBL,TLL,T33,3,3)

CALL MATAB(TIB,T33,TIL,3,3)

CALL MATAI(TIL,ULL,ULI,3,3,1)

TRANSFORM UNIT VECTOR TO INERTIAL COORDINATES

CALL AMAT(STATE(7),TBI)

TRANSFORM UNIT VECTOR TO INERTIAL COORDINATES

Determine angle of unit vector along landmark

DIRECTION RELATIVE TO LOCAL DOWN
C******************************************************************************
C DO 15 I=1,3
C 15 DOWN(I)=STAT(I)
C CALL VCRS(ULI,DOWN,UCD)
C THETP=VMAG(UCD,3)/VMAG(DOWN,3)
C THETP=ASIN(THETP)
C******************************************************************************
C******************************************************************************
C FIND LENGTH OF VECTOR FROM S/C TO LM
C******************************************************************************
C******************************************************************************
C B=SQRD(VMAG(STATE,3)**2+(RE+AL)**2-2.*VMAG(STATE,3)*
C (RE+AL)*COS(-THETP+ASIN(VMAG(STATE,3)))+
C SIN(THETP)/(RE+AL)**2))
C******************************************************************************
C******************************************************************************
C ESTABLISH VECTOR
C******************************************************************************
C******************************************************************************
C DO 20 I=1,3
C 20 M(I)=B=UL(I)
C******************************************************************************
C******************************************************************************
C COMPUTE LM POSITION VECTOR IN INERTIAL SPACE
C******************************************************************************
C******************************************************************************
C DO 30 I=1,3
C 30 LI(I)=STATE(I)+M(I)
C******************************************************************************
C******************************************************************************
C TRANSFORM TO EARTH FIDUCIAL COORDINATES
C******************************************************************************
C******************************************************************************
C CALL WET(TIEO.TIE)
C CALL MIP=3(TIE..TLE)
C CALL MATAB(TIE..TLE,3.3.1)
C******************************************************************************
C******************************************************************************
C SOLVE FOR LATITUDE AND LONGITUDE (DEG)
C******************************************************************************
C******************************************************************************
C LAT=ASIN(LED)//(RE+AL))
C LON=180.*ASIN(LED)/((RE+AL)*COS(LAT))/PI
C LAT=LAT*180./PI
C RETURN END
<table>
<thead>
<tr>
<th>Name</th>
<th>Bytes</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 $CODE</td>
<td>505</td>
<td>PIC CON REL LCL SHR EXE RD NOWRT LONG</td>
</tr>
<tr>
<td>1 $PODA</td>
<td>0</td>
<td>PIC CON REL LCL SHR NOEXE RD NOWRT LONG</td>
</tr>
<tr>
<td>2 $LOCAL</td>
<td>929</td>
<td>PIC CON REL LCL NO SHR NOEXE RD WRT QUAD</td>
</tr>
<tr>
<td>3 $MPAR</td>
<td>392</td>
<td>PIC OVR REL GBL SHR NOEXE RD WRT LONG</td>
</tr>
<tr>
<td>4 $ENVIR</td>
<td>404</td>
<td>PIC OVR REL GBL SHR NOEXE RD WRT LONG</td>
</tr>
<tr>
<td>5 $ARRAYS</td>
<td>1096</td>
<td>PIC OVR REL GBL SHR NOEXE RD WRT LONG</td>
</tr>
<tr>
<td>6 $TARGETS</td>
<td>136</td>
<td>PIC OVR REL GBL SHR NOEXE RD WRT QUAD</td>
</tr>
<tr>
<td>7 $TIME</td>
<td>112</td>
<td>PIC OVR REL GBL SHR NOEXE RD WRT QUAD</td>
</tr>
<tr>
<td>8 $STATE</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Address</th>
<th>Type</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>3-00000000</td>
<td>R+8</td>
<td>AL</td>
</tr>
<tr>
<td>7-00000020</td>
<td>R+8</td>
<td>DEL</td>
</tr>
<tr>
<td>2-0000290</td>
<td>I+4</td>
<td>IFLAG</td>
</tr>
<tr>
<td>6-00000000</td>
<td>I+4</td>
<td>MTYPE</td>
</tr>
<tr>
<td>8-00000060</td>
<td>R+8</td>
<td>RADM</td>
</tr>
<tr>
<td>2-0000278</td>
<td>R+8</td>
<td>TO</td>
</tr>
<tr>
<td>7-00000000</td>
<td>R+8</td>
<td>TIME</td>
</tr>
<tr>
<td>7-00000048</td>
<td>R+8</td>
<td>THEAS</td>
</tr>
<tr>
<td>7-00000050</td>
<td>R+8</td>
<td>TRACK</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Address</th>
<th>Type</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>7-00000038</td>
<td>R+8</td>
<td>DATED</td>
</tr>
<tr>
<td>7-00000030</td>
<td>R+8</td>
<td>DTIN</td>
</tr>
<tr>
<td>6-00000004</td>
<td>I+4</td>
<td>IS</td>
</tr>
<tr>
<td>3-00000008</td>
<td>R+8</td>
<td>LON</td>
</tr>
<tr>
<td>6-00000014</td>
<td>R+8</td>
<td>PI</td>
</tr>
<tr>
<td>3-00000178</td>
<td>R+8</td>
<td>SGGCP</td>
</tr>
<tr>
<td>2-0000280</td>
<td>R+8</td>
<td>THETP</td>
</tr>
<tr>
<td>7-00000018</td>
<td>R+8</td>
<td>TIA</td>
</tr>
<tr>
<td>7-00000060</td>
<td>R+8</td>
<td>TISN</td>
</tr>
<tr>
<td>7-00000078</td>
<td>R+8</td>
<td>TPRINT</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Address</th>
<th>Type</th>
<th>Name</th>
<th>Dimensions</th>
</tr>
</thead>
<tbody>
<tr>
<td>3-0000000C0</td>
<td>R+8</td>
<td>SKL</td>
<td>16 (2)</td>
</tr>
<tr>
<td>3-0000000A8</td>
<td>R+8</td>
<td>SL</td>
<td>16 (2)</td>
</tr>
<tr>
<td>2-0000240</td>
<td>R+8</td>
<td>DOWN</td>
<td>24 (3)</td>
</tr>
<tr>
<td>2-0000048</td>
<td>R+8</td>
<td>LE</td>
<td>24 (3)</td>
</tr>
<tr>
<td>2-0000060</td>
<td>R+8</td>
<td>LI</td>
<td>24 (3)</td>
</tr>
<tr>
<td>2-0000078</td>
<td>R+8</td>
<td>M</td>
<td>24 (3)</td>
</tr>
<tr>
<td>4-00000L50</td>
<td>R+8</td>
<td>PROFILE</td>
<td>372 (10, 4)</td>
</tr>
<tr>
<td>3-0000006D</td>
<td>R+8</td>
<td>SKL</td>
<td>16 (2)</td>
</tr>
<tr>
<td>3-000008B</td>
<td>R+8</td>
<td>SL</td>
<td>16 (2)</td>
</tr>
<tr>
<td>4-00000000</td>
<td>R+8</td>
<td>STATE</td>
<td>80 (10)</td>
</tr>
<tr>
<td>5-00000000</td>
<td>R+8</td>
<td>T1</td>
<td>24 (3)</td>
</tr>
<tr>
<td>5-0000009B</td>
<td>R+8</td>
<td>T11</td>
<td>72 (3, 3)</td>
</tr>
<tr>
<td>5-0000018B</td>
<td>R+8</td>
<td>T2</td>
<td>24 (3)</td>
</tr>
<tr>
<td>5-0000030A</td>
<td>R+8</td>
<td>T3</td>
<td>24 (3)</td>
</tr>
<tr>
<td>5-0000033</td>
<td>R+8</td>
<td>T33</td>
<td>72 (3, 3)</td>
</tr>
</tbody>
</table>
### FUNCTIONS AND SUBROUTINES REFERENCED

<table>
<thead>
<tr>
<th>ANAT</th>
<th>MATAB</th>
<th>MINT3</th>
<th>MTHSDASIN</th>
<th>MTHSDCOS</th>
<th>MTHSDSIN</th>
<th>MTHDSORT</th>
<th>RODATA</th>
</tr>
</thead>
<tbody>
<tr>
<td>VKAG</td>
<td>VMAG</td>
<td>VETAG</td>
<td>VETAG</td>
<td>VETAG</td>
<td>VETAG</td>
<td>VETAG</td>
<td>VETAG</td>
</tr>
</tbody>
</table>

Total Space Allocated = 3609 Bytes

### COMMAND QUALIFIERS

- FORTRAN /LIST GCP,INDATA,MATAB,OUTDATA,RUNG,DAV,EPHEM,TRUEA,SPRESS,occult,GPERR,GPSEQ,VISIBLE,GENENV,TREG,GTRQ,FRATE,RMAT,CMA
- /CHECK=(NOBOUNDS,OVERFLOW)
- /DEBUG=(NOSYMBOLS,TRACEBACK)
- /FF77 /NOG Floating /14 /OPTIMIZE /WARNINGS /NOM_LINES /NOMACHINE_CODE /CONTINUATIONS=19

### COMPILATION STATISTICS

- Run Time: 2.57 seconds
- Elapsed Time: 31.83 seconds
- Page Faults: 395
- Dynamic Memory: 180 pages
SUBROUTINE TRANFM(XTRANS,XTRANPI)

C--- THIS ROUTINE CREATES THE TRANSFORMATION MATRICES FOR THE TRANSFORMATION FROM INERTIAL TO FLIGHT COORDINATES.

C

INCLUDE 'ARRAY.COM'

COMMON /ARRAY/ X(3000,39,21), INUM

REAL*4 X(1:3000,39),X2(3000,39)

INTEGER*2 INUM

EQUIVALENCE (X(1,1),X(1,1,1)),(X2(1,1),X(1,1,2))

REAL*8 XTRANS(3,3,3000),XTRANP(3,3,3000)

REAL*8 UXF(3,1000),UYF(3,1000),UZF(3,1000),TUVF(3,1000)

REAL*8 TEMP1(3),TEMP2(3)

C--- CREATE Uzf

DO 10 I=1,NINUM
   XTEMP1(I)=X(I,34,1)
   XTEMP2(I)=X(I,35,1)
   XTEMP3(I)=X(I,36,1)
   CALL UNIT(XTEMP1(I),UZF(I,1),3)
   CONTINUE

C--- NEGATE ALL ELEMENTS OF Uzf

DO 15 I=1,NINUM
   UZF(I,1)=-UZF(I,1)
   UZF(I,2)=-UZF(I,2)
   UZF(I,3)=-UZF(I,3)
   CONTINUE

C--- CREATE Uyf

DO 20 I=1,NINUM
   XTEMP2(I)=X(I,37,1)
   XTEMP3(I)=X(I,38,1)
   CALL VCRS(XTEMP2(I),XTEMP1(I),TUVF(I,1))
   CALL UNIT(TUVF(I,1),UYF(I,1),3)
   CONTINUE

C--- CREATE Uyf

DO 30 I=1,NINUM
   CALL VCRS(UYF(I,1),UZF(I,1),UXF(I,1))
   CONTINUE

C--- MERGE INTO XTRANS

DO 40 I=1,NINUM
   DO 35 J=1,3
      XTRANS(I,J,1)=UXF(J,I)
      XTRANS(I,J,2)=UYF(J,I)
      XTRANS(I,J,3)=UZF(J,I)
      CONTINUE
   CONTINUE

END
TRANFM

0036 35 CONTINUE
0037 40 CONTINUE
C C--- CREATE THE INVERSE
C
0038 DO 50 I=1,INUM
0039 CALL MINV3(XTRANS(1,1,1),XTRANS(1,1,1))
0040 50 CONTINUE
C C

C C--- CREATE THE TRANSPOSE
C
0041 DO 70 I=1,INUM
0042 DO 60 J=1,3
0043 XTRANS(I,J,1)=XTRANS(J,1,1)
0044 XTRANS(3,J,1)=XTRANS(J,3,1)
0045 XTRANS(3,J,1)=XTRANS(J,3,1)
0046 60 CONTINUE
0047 /0 CONTINUE
0048 /0 CONTINUE
0049 END

PROGRAM SECTIONS

Name       Bytes Attributes
0 $CGC       527 PIC CON REL LCL SHR EXE RD NOXRT LONG
1 $DATA      4 PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2 $LOCAL     288176 PIC CON REL LCL NOSHR NOEXE RD WRT QUAD
3 $ARRAY     936002 PIC OVR REL GBL SHR NOEXE RD WRT LONG

ENTRY POINTS

Address Type Name
0-00000000 TRANFM

VARIABLES

Address Type Name Address Type Name Address Type Name
2-00046530 I+4 I 3-0004840 I+2 INUM 2-00046534 I+4 J

ARRAYS

Address Type Name Bytes Dimensions
2-00034800 R=B TUFF 72000 (3, 3000)
2-00000000 R=B UXF 72000 (3, 3000)
2-00011940 R=B UYF 72000 (3, 3000)
2-00023280 R=B UZF 72000 (3, 3000)
TRANFM

8-Apr-1981 07:41:05  _DBAO:[D11R.GCP]TRANFM.FOR;13

3-00000000 R+4 X  936000 (3000, 39, 2)
3-00000000 R+4 X1  468000 (3000, 39)
3-00072420 R+4 X2  468000 (3000, 39)
2-00004500 R+B XTEMP1  24 (3)
2-00004518 R+B XTEMP2  24 (3)
AP-00000008 R+B XTRANP  216000 (3, 3, 3000)
AP-00000004 R+B XTRANS  216000 (3, 3, 3000)

LABELS

<table>
<thead>
<tr>
<th>Address</th>
<th>Label</th>
<th>Address</th>
<th>Label</th>
<th>Address</th>
<th>Label</th>
<th>Address</th>
<th>Label</th>
<th>Address</th>
<th>Label</th>
<th>Address</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>**</td>
<td>10</td>
<td></td>
<td>**</td>
<td>15</td>
<td>**</td>
<td>20</td>
<td>**</td>
<td>30</td>
<td>**</td>
<td>35</td>
</tr>
<tr>
<td></td>
<td>**</td>
<td>50</td>
<td></td>
<td>**</td>
<td>60</td>
<td>**</td>
<td>70</td>
<td>**</td>
<td></td>
<td>**</td>
<td>40</td>
</tr>
</tbody>
</table>

FUNCTIONS AND SUBROUTINES REFERENCED

MINV3  UNIT  VCRS

Total Space Allocated = 1224769 Bytes

COMMAND QUALIFIERS

FORTRAN TRANFM/LIS

/CHECK=(NOBOUNDS,OVERFLOW)
/DEBUG=(NOSYMBOALS,TRACEBACK)
/FT77 /NOG_FLOATING /14 /OPTIMIZE /WARNINGS /NOD_LINES /NO_MACHINE_CODE /CONTINUATIONS=19

COMPILATION STATISTICS

Run Time:  2.15 seconds
Elapsed Time:  4.93 seconds
Page Faults:  187
Dynamic Memory:  59 pages
```
0001 SUBROUTINE UNIT(R,V,N)
  C**********************************************************************
  C UNITIZE THE VCTGR v13)
  C**********************************************************************
 0002 INCLUDE 'DEBUG.COM'
0010 0003 COMMON /DEBUG/ ENTER,IDEBUG
0020 0003 C USER CONTROLLED PARAMETERS TO VARY DEBUG PRINT LEVEL
0030 0003 C ENTER IF 1, PRINTER WHEN ENTERS MOST SUBROUTINES
0040 0003 C IDEBUG 0-10, HIGHER NUMBER MEANS MORE PRINT
0050 0003 C DIMENSION R(N),V(N)
0060 0004 DIMENSION R(22),V(22)
0070 0005 REAL*8 D,R,V,VMAG
0080 0006 D = VMAG(R,N)
0090 0007 DO 10 I=1,N
0090 0008 10 V(I) = R(I)/D
0100 0009 RETURN
0110 0010 END

PROGRAM SECTIONS

<table>
<thead>
<tr>
<th>Name</th>
<th>Bytes</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>$CODE</td>
<td>87</td>
<td>PIC CON REL LCL SHR EXE RD NQWRAT LONG</td>
</tr>
<tr>
<td>LOCAL</td>
<td>64</td>
<td>PIC CON REL LCL NOSHR NOEXE RD WRT QUAD</td>
</tr>
<tr>
<td>DEBUG</td>
<td>8</td>
<td>PIC OVR REL GBL SHR NOEXE RD WRT LONG</td>
</tr>
</tbody>
</table>

ENTRY POINTS

<table>
<thead>
<tr>
<th>Address Type</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-00000000</td>
<td>UNIT</td>
</tr>
</tbody>
</table>

VARIABLES

<table>
<thead>
<tr>
<th>Address Type</th>
<th>Name</th>
<th>Address Type</th>
<th>Name</th>
<th>Address Type</th>
<th>Name</th>
<th>Address Type</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-00000000</td>
<td>R=8</td>
<td>2-00000008</td>
<td>I=4</td>
<td>3-00000004</td>
<td>I=4</td>
<td>3-00000000</td>
<td>I=4</td>
</tr>
<tr>
<td>AP-00000000</td>
<td>INT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

ARRAYS

<table>
<thead>
<tr>
<th>Address Type</th>
<th>Name</th>
<th>Bytes</th>
<th>Dimensions</th>
</tr>
</thead>
<tbody>
<tr>
<td>AP-00000004</td>
<td>R=8</td>
<td>176</td>
<td>(22)</td>
</tr>
<tr>
<td>AP-00000008</td>
<td>R=8</td>
<td>176</td>
<td>(22)</td>
</tr>
</tbody>
</table>
```

UNIT

LABELS

Address Label
** 10

FUNCTIONS AND SUBROUTINES REFERENCED

VMAG

Total Space Allocated = 159 Bytes

COMMAND QUALIFIERS

FORTRAN /LIST GCP,INDATA,MATAB,OUTDATA,RUNG,DNAV,EPHEM,TRUEA,SPRESS,OCCLUD,GPERT,GCPSEQ,VISIBLE,GENENV,TREG,GYROUT,RATE,BMAT,CMA

/CHECK=(NOBOUNDS,OVERFLOW)
/DEBUG=(NOSYMBOLS,TRACEBACK)
/F77 /NOG_FLOATING /14 /OPTIMIZE /WARNINGS /NOO_LINES /NOMACHINE_CODE /CONTINUATIONS=19

COMPILATION STATISTICS

Run Time: 0.59 seconds
Elapsed Time: 8.42 seconds
Page Faults: 333
Dynamic Memory: 160 pages
SUBROUTINE VCRS(A,B,C)

INCLUDE 'DEBUG.COM'

C

C USER CONTROLLED PARAMETERS TO VARY DEBUG PRINT LEVEL
C

C ENTER IF 1. PRINTS WHEN ENTERS MOST SUBROUTINES
C DEBUG 0-10. HIGHER NUMBER MEANS MORE PRINT
C

C DIMENSION A(3),B(3),C(3)
C REAL A,B,C
C
C VECTOR CROSS PRODUCT (A) X (B) = (C)
C
C C(1) = A(2)*B(3)-A(3)*B(2)
C C(2) = A(3)*B(1)-A(1)*B(3)
C C(3) = A(1)*B(2)-A(2)*B(1)
C
RETURN
C
END

PROGRAM SECTIONS

Name      Bytes Attributes
0 SCODE   106 PIC CON REL LCL SHR EXE RD NOWRT LONG
2 SLOCAL  60 PIC CON REL LCL NOSHR NOEXE RD WRT LONG
3 DEBUG   8 PIC OVR REL GBL SHR MOEXE RD WRT LONG

ENTRY POINTS

Address Type Name
0-00000000 VCRS

VARIABLES

Address Type Name Address Type Name
3-00000004 I+4 IDEBUG 3-00000000 I+4 IENTER

ARRAYS

Address Type Name Bytes Dimensions
AP-000000040 R=8 A 24 (3)
AP-000000080 R=8 B 24 (3)
AP-0000000C0 R=8 C 24 (3)

Total Space Allocated = 174 Bytes
COMMAND QUALIFIERS

FORTRAN /LIST GCP,INDATA,MATAB,OUTDATA,RUNG,NAV,EPHEM,TRUEA,SPRESS,OCUL,T,GPERT,GPCSEQ,VISIBLE,GENENV,TREG,GYROUT,RATE,INTAT,CMAT

/CHECK=(NOBOUNDS,OVERFLOW)
/DEBUG=(NOSYMBOLS,TRACEBACK)
/FF77 /NOSG_FLOAT /14 /OPTIMIZE /WARNINGS /NOditions /NOMACHINE_CODE /continuations=19

COMPILATION STATISTICS

Run Time: 0.63 seconds
Elapsed Time: 10.88 seconds
Page Faults: 165
Dynamic Memory: 160 pages
FUNCTION VDOT(A,B,N)
C************************************************************************C
C VECTOR DOT PRODUCT (A) DOT (B) = VDOT
C************************************************************************C

INCLUDE 'DEBUG.COM'
COMMON /DEBUG/ ENTER, IDEBUG

C USER CONTROLLED PARAMETERS TO VARY DEBUG PRINT LEVEL
C
INTER IF 1, PRINTS WHEN ENTERS MOST SUBROUTINES
IDEBUG 0-10, HIGHER NUMBER MEANS MORE PRINT

DIMENSION A(N),B(N)
DIMENSION A(22),B(22)
REAL*8 VDOT

DO 10 I=1,N
VDOT = VDOT + A(I) * B(I)
10 RETURN

END
VOOT

6-Apr-1981 15:06:03  VAX-11 FORTRAN V2.0-2

LABELS

Address Label
-- 10

Total Space Allocated = 136 Bytes

COMMAND QUALIFIERS

FORTRAN /LIST GCP,INDATA,MATAB,OUTDATA,RUNG,EPHEM,TRUEA,SPRESS,GPRT,GPSEQ,VISIBLE,GENENV,TREG,GROUT,RATE,DMAT,CMA

/ CHECK=(NOBOUNDS,OVERFLOW)
/ DEBUG=(NOSYMBOLS,TRACEBACK)
/ F77 /NOG_FLOATING /14 /OPTIMIZE /WARNINGS /NOG_LINES /NOMACHINE_CODE /CONTINUATIONS=19

COMPILATION STATISTICS

Run Time: 0.59 seconds
Elapsed Time: 15.48 seconds
Page Faults: 280
Dynamic Memory: 180 pages
FUNCTION VMAG(A,N)

C ABSOLUTE MAGNITUDE OF A(N) VECTOR

C***************************************************************

0002 INCLUDE 'DEBUG.COM'

0010 0003  * COMMON /DEBUG/ IENTER,DEBUG

0020  * C USER CONTROLLED PARAMETERS TO VARY DEBUG PRINT LEVEL

0030  * C

0040  * C

0050  * C I NTER IF 1, PRINTS WHEN ENTERS MOST SUBROUTINES

0060  * C IDEBUG 0-10, HIGHER NUMBER MEANS MORE PRINT

0070  * C

0004 REAL=8 VMAG,A,VDOT

0005 VMAG = SQRT(VDOT(A,A,N))

0006 RETURN

0007 END

PROGRAM SECTIONS

<table>
<thead>
<tr>
<th>Name</th>
<th>Bytes</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 CODE</td>
<td>38</td>
<td>PIC CON REL LCL  SHR  EXE  RD  NOWRT  LONG</td>
</tr>
<tr>
<td>2 LOCAL</td>
<td>24</td>
<td>PIC CON REL LCL  NO  SHR  NOEXE  RD  WRT  QUAD</td>
</tr>
<tr>
<td>3 DEBUG</td>
<td>8</td>
<td>PIC OVR REL GBL  SHR  NOEXE  RD  WRT  LONG</td>
</tr>
</tbody>
</table>

ENTRY POINTS

<table>
<thead>
<tr>
<th>Address</th>
<th>Type</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-00000000</td>
<td>R8</td>
<td>VMAG</td>
</tr>
</tbody>
</table>

VARIABLES

<table>
<thead>
<tr>
<th>Address</th>
<th>Type</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>AP-00000004</td>
<td>R8</td>
<td>A</td>
</tr>
<tr>
<td>3-00000004</td>
<td>I+4</td>
<td>IDEBUG</td>
</tr>
<tr>
<td>3-00000000</td>
<td>I+4</td>
<td>IENTER</td>
</tr>
<tr>
<td>AP-00000000</td>
<td>I+4</td>
<td>N</td>
</tr>
</tbody>
</table>

FUNCTIONS AND SUBROUTINES REFERENCED

MTHDOSQRT VDOT

Total Space Allocated = 70 Bytes
VMAG

6-Apr-1981 15:06:20  VAX-11 FORTRAN V2.0-2
1-Dec-1980 07:15:24  _DBAO:[DIR.GCP]VMAG.FOR:

COMMAND QUALIFIERS

FORTRAN /LIST GCP,INDATA,MATAB,OUTDATA,RUNG,NAV,EPHEM,TRUEA,SPRESS,OCULT,GPRT,GPSEQ,VISIBLE,GENENV,TREG,GRYOUT,RATE,GMAT,CMA

/CHECK=(NOBOUNDS,OVERFLOW)
/DEBUG=(NOSYMBOLS,TRACEBACK)
/F77 /NOG_FLOATING /14 /OPTIMIZE /WARNINGS /NOD_LINES /NONMACHINE_CODE /CONTINUATIONS=19

COMPILATION STATISTICS

Run Time: 0.45 seconds
Elapsed Time: 10.41 seconds
Page Faults: 279
Dynamic Memory: 160 pages
SUBROUTINE WET (TIEO, TIE)

SUBROUTINE WET CREATES THE TRANSFORMATION FROM EARTH
FIXED TO INERTIAL COORDINATES

INPUT VARIABLES

TIEO = INITIAL TRANSFORMATION

OUTPUT VARIABLES

TIE = EARTH FIXED TO INERTIAL TRANSFORMATION
AT TIME T

PROGRAMMED BY JACK MYERS 16 JUNE 1980
EXT 4443

INCLUDE 'TIME.COM'

COMMON /TIME/ TIME, TNext, TSTOP, TIA, DEL, TIN, DTN, DATEO, TZERO

TIME, TNext, TSTOP, TIA, DEL, TIN, DTN, DATEO, TZERO

REAL=8 TIME, TNext, TSTOP, TIA, DEL, TIN, DTN, DATEO, TMEAS, TRACK, TIS,

TIS, DTN, Tzero, DATEO, TPRINT, DPRINT

THESE ARE THE TIME REFERENCE FRAMES

TIME

ATOMIC TIME SINCE INITIALIZATION (SEC)

TNext

TIME FOR NEXT POSITION INTEGRATION (SEC)

TSTOP

RUN TERMINATION TIME (SEC)

TIA

ATTITUDE INTEGRATION TIME (SEC)

DL

STEP SIZE (SEC)

TIN

POSITION INTEGRATION TIME (SEC)

DTN

STEP SIZE (SEC)

DATED

DATE OF FLIGHT EPOCH (JD)

DATEO

DATE OF 1950 EPOCH (JD)

TZERO

START TIME IN SEC. SINCE DATEO

TSLW

TIME NEEDED TO SLEW AND ACQUIRE (SEC)

TIS

REAL WORLD REFERENCE TIME (SEC)

TIN

TIME FOR NEXT RW POSITION INTEGRATION (SEC)

DTA

REAL = DEL BUT + TSLW - TIA WHEN DEL

TOO LARGE AT MEASUREMENT TIME

TPRINT

TIME FOR PRINT (SEC)

DPRINT

INCREMENT ON TPRINT (SEC)

DATA WE/7.2921152E-5/ 'RAD/SEC

DIMENSION A(3,3)

REAL=8 TIE, TIA, DEL, DJS, TE

COMPUTE TIME CHANGE SINCE DATEO=JANUARY 1 1950

DO 10 I=1, 9

DO 10 I=1, 9
02500  0011  DO 10  J=1,3
02600  0012  10    A(I,J)=0.
02700  0013  A(1,1)=COS(W+TE)
02800  0014  A(1,2)=SIN(W+TE)
02900  0015  A(2,1)=-A(1,2)
03000  0016  A(2,2)=A(1,1)
03100  0017  A(3,3)=1.
03110  C*********************************************************
03120  C   SET UP TOTAL TRANSFORMATION
03130  C*********************************************************
03200  0018  CALL MATAB(A,TIE0,TIE,3,3,3)
03300  0019  RETURN
03400  0020  END

PROGRAM SECTIONS

Name        Bytes Attributes
0  SCODE     149 PIC CON REL LCL SHR EKE RD NOWRT LONG
1  SPDATA    4  PIC CON REL LCL SHR NOME RD NOWRT LONG
2  BLOCAL    132 PIC CON REL LCL NOSHR NOME RD WRT QUAD
3  TIME      136 PIC OVR REL GBL SHR NOME RD WRT LONG

ENTRY POINTS

Address  Type  Name
0-00000000  0-WET

VARIABLES

Address  Type  Name    Address  Type  Name    Address  Type  Name    Address  Type  Name    Address  Type  Name
3-00000038 R8  DATE0  3-00000070 R8  DATER    3-0000020  R8  DEL    2-00000050 R8  BUJS
3-00000068 R8  DTA    3-0000030 R8  DTM     3-0000080 R8  DPRINT    2-00000060 I=4  I
2-00000064 I=4  J     2-00000058 R8  TE      3-0000018 R8  TIA     AP-C00000000 R8  TIE
3-00000060 R8  TIE0   3-00000000 R8  TIME    3-0000028 R8  TIN     3-00000050 R8  TIS
3-00000050 R8  TISH   3-00000018 R8  THEAS    3-00000080 R8  TTEXT    3-00000078 R8  TPRINT
3-00000010 R8  TRACK  3-00000040 R8  TSTOP   3-00000090 R8  TZERO
2-00000000 R8  A          72 (3,3)
FUNCTIONS AND SUBROUTINES REFERENCED

FORTAN MTHSDCOS MTHSDSIN

Total Space Allocated = 421 Bytes

COMMAND QUALIFIERS

FORTAN /LIST GCP,INDATA,MATAB,OUTDATA,RUNG,DMY,EPHEM,TRUEA,SPRESS,OCULT,GPERT,GCPSEQ,VISIBLE,GENENV,TREG,GYROUT,RATE,GMAT,CMA

/CHECK=(NOBOUNDS,OVERFLOW)
/DEBUG=(NOSYMBOLS,TRACEBACK)
/FF7 /NOG_FLOATING /14 /OPTIMIZE /WARNINGS /NOO_LINES /NOMACHINE_CODE /CONTINUATIONS=19

COMPILATION STATISTICS

Run Time: 0.98 seconds
Elapsed Time: 13.67 seconds
Page Faults: 346
Dynamic Memory: 169 pages
3.0 SIMULATION MODE 2

The second mode of GCPSIM operation provides for extraction of GCP's directly from LANDSAT imagery. The main portion of GCPSIM is identical to that discussed previously. The primary difference is that several additional programs are hosted on the PDP 11/70 and communicate with GCPSIM, which is hosted on the VAX 11/780, through a parallel link (Figure A-45).

Two primary programs exist. The first called LMEX allows a landmark data base to be constructed independently from GCPSIM. The other main program performs the scrolling of the imagery, extraction of image data, and correlations from the stored GCP. These two programs are discussed separately in the following sections.

Figure A-45. Implementation Strategy of GCPSIM
Landmark Extraction Mode
3.1 LANDMARK (GCP) EXTRACTION PROGRAM (LMEX)

The Landmark Extraction Program, LMEX, was written to allow the user to interactively extract ground control points from LANDSAT imagery. These extracted GCP's will be used for correlation purposes on later LANDSAT imagery of the same area.

LMEX created a data base that will be used with the correlation-scroll programs. LMEX can create a new data file or can be used to modify an existing data file.

The landmark data base file is a direct access file of record size 256 bytes. The user can name this file any name desired. The file setup is as follows:

<table>
<thead>
<tr>
<th>Header block</th>
</tr>
</thead>
<tbody>
<tr>
<td>(4 records)</td>
</tr>
<tr>
<td>Landmark data</td>
</tr>
<tr>
<td>(1 record per</td>
</tr>
<tr>
<td>landmark)</td>
</tr>
<tr>
<td>.</td>
</tr>
<tr>
<td>.</td>
</tr>
<tr>
<td>.</td>
</tr>
</tbody>
</table>

The first four records are header records containing information about each extracted landmark. Each landmark extracted uses up 16 bytes of header space, therefore up to 63 landmarks may be extracted. Word 1 of the header contains the number of landmarks in the data base. Each landmarks image data consists of 256 bytes. This allows each landmarks data to be stored in one record.

The LANDSAT imagery is displayed on a RAMTEK 9100 B/W graphics display system. The data displayed is either 7-bit or 6-bit grey scale imagery. The data being displayed will be scrolled from the bottom of the screen to the top, thus simulating the satellite motion.

The user may run LMEX on a previously stored landmark data base or create a new one. Landmarks may be defined at any time by using the joystick. The only restriction on landmark definition is that adjacent landmarks must be at least 200 lines apart. The LMEX programs checks for this occurring and will not allow it.
Is this a new data base file to be used?

T

Initialize landmark no. & other variables as well as the header

F

Read in the landmark no. and header information

Get a line of image data and scroll it

T

Is a previously-defined landmark on the screen?

F

Draw a box around the landmark

NULL

T

Does the user wish to define a landmark?

F

Allow the user to position a cursor at the center of the landmark

Determine header offsets and other landmark information

Determine where in the header this landmark fits

NULL

T

Is this a valid landmark?

F

Store the landmark data & header

NULL

Draw a box around the landmark

Do until all the lines of imagery have been scrolled and the user wishes to exit

Write back the updated header block

Figure A-46

A-374
3.2 CORRELATION - SCROLL PROGRAMS

The correlation-scroll programs consist of three separate tasks. These tasks are installed and run concurrently. Inter-task communication is performed by using a shareable global area. Inter-task synchronization is performed by using global event flags. The three tasks are:

1. **SCRLL** - This task scrolls the image data.

2. **SCOUNT** - This task updates the landmark data base and decides when a correlation is to be performed.

3. **CORR** - This task performs the actual correlation and returns the registration vector.

These tasks run at different priorities. SCRLL is set at the highest priority, next is SCOUNT, and finally CORR. SCRLL will scroll a line of image data, signal for SCOUNT to become active, and then suspend itself until SCOUNT reawakes SCRLL. SCOUNT in turn, determines whether to awake CORR to perform landmark correlation.

The inter-relationship of the tasks may be seen in Figure A-46.
CORRELATION - SCROLL PROGRAM STRUCTURE

SHAREABLE GLOBAL AREA

EVENT FLAG 40

EVENT FLAG 41

EVENT FLAG 42

SCOUNT

CORR

Figure A-47

A-376
3.2.1 SCROLL PROGRAM (SCRLL)

The SCROLL program is the driver task for the correlation-scrolI programs. This program starts the other two tasks running. This program will then begin scrolling image data. After each line of image data is displayed and scrolled, SCOUNT will be activated and SCROLL will suspend itself. SCRLL will then be activated by SCOUNT when SCOUNT has finished performing its function.
<table>
<thead>
<tr>
<th>SCRLL</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Reset The RAMTEK 9100</strong></td>
</tr>
<tr>
<td><strong>Set event flag 40. Reset flags 41 and 42.</strong></td>
</tr>
<tr>
<td><strong>Get the name of the image file to scroll &amp; the starting element</strong></td>
</tr>
<tr>
<td><strong>Create a shareable global and an address window for mapping purposes</strong></td>
</tr>
<tr>
<td><strong>Start up SCOUNT and CORR to become active</strong></td>
</tr>
<tr>
<td><strong>Open the image data file. Set line counter.</strong></td>
</tr>
<tr>
<td><strong>Get no. lines to be scrolled from GCPSIM</strong></td>
</tr>
<tr>
<td><strong>Do until end flag from GCPSIM</strong></td>
</tr>
<tr>
<td><strong>Wait until event flag 40 is set</strong></td>
</tr>
<tr>
<td><strong>Get a line of imagery and display it</strong></td>
</tr>
<tr>
<td><strong>Scroll the image, simulating motion</strong></td>
</tr>
<tr>
<td><strong>Clear event flag 40. Set event flag 41 to start SCOUNA processing.</strong></td>
</tr>
<tr>
<td><strong>Do while lines of imagery still need to be scrolled</strong></td>
</tr>
<tr>
<td><strong>Set the end flag and set event flags 41 and 42</strong></td>
</tr>
<tr>
<td><strong>Exit task off</strong></td>
</tr>
</tbody>
</table>

Figure A-48
3.2.1.1 DRAW A BOX (DRAW)

DRAW performs a similar operation as the DRABOX library utility. DRAW should be used when the imagery that is to have the box drawn has been scrolled. The differences between DRAW and DRABOX have to do with the RM9100's window and scan parameters.

The passed arguments I and J define the upper left corner of the area to have the box enclose. The box is always 18 by 18 pixels, thus enclosing a 16 x 16 pixel landmark.

FORMAT: CALL DRAW (I-J)

<table>
<thead>
<tr>
<th>DRAW</th>
</tr>
</thead>
<tbody>
<tr>
<td>DRAW THE TOP LINE</td>
</tr>
<tr>
<td>DRAW THE BOTTOM LINE</td>
</tr>
<tr>
<td>DRAW THE LEFT LINE</td>
</tr>
<tr>
<td>DRAW THE RIGHT LINE</td>
</tr>
</tbody>
</table>

Figure A-49
3.2.2 SCROLL COUNTER (SCOUNT)

SCOUNT is the main data base handler for the correlation-scroll programs. This program keeps track of the number of lines that have been scrolled and whether or not a landmark has become visible. If a landmark is visible, this program waits until the entire search area is visible and then activates the landmark correlation program CORR.

SCOUNT is activated by SCRLL when event flag 41 is set. After each line of image data has been scrolled, SCOUNT will be activated to perform its checking and then deactivate itself and activate SCRLL to begin the process over.
SCOUNT

Create an address window to map into the shareable global area

Read in the landmark data base header

Set up counter for first landmark

Suspend until event flag 41 is set

Decrement the landmark counter by 1

Is the counter greater than 0?

Is the counter ≠ 0

Get the search area size and set counter

Read in the target landmark data

Decrement the search area counter by 1

Is the search area counter equal to zero

Reset the counter to the next search area

Set event flag 2 to begin correlation

Read in a line of search area image data

Do until the end flag is set

Exit task off

Figure A-50
3.2.3 CORRELATION (CORR)

CORR is used in conjunction with SCOUNT to perform the correlation analysis on the LANDSAT imagery. CORR returns the relative registration vector of the best fit of the landmark imagery upon the target search area imagery.

CORR waits for event flag 42 to be set before any processing occurs. SCOUNT is the task which will set this flag. When SCOUNT sets this flag, the known landmark imagery has been stored in the array LM and the target search area image data is in the array WIN.

The S.S.D.A algorithm uses an exhaustive search of the landmark data versus the search area data. The best fit is where the correlation value determined by the S.S.D.A algorithm is at a minimum. The S.S.D.A algorithm is:

\[
\sum_{I = 1}^{\Sigma} \sum_{J = 1}^{\Sigma} \text{abs} \left( \text{WIN}(J,I) - \text{WIN} \text{MEAN} \right) - \left( \text{LM}(J,I) - \text{LANDMARK MEAN} \right)
\]

Once CORR has determined the best fit landmark placement, the tasks suspends itself until SCOUNT starts it up again.
Create an address window to map into the shareable global area

<table>
<thead>
<tr>
<th>Suspend until event flag 42 is set</th>
</tr>
</thead>
<tbody>
<tr>
<td>Calculate the mean of the known landmark data</td>
</tr>
<tr>
<td>Using the SSDA correlation, find the best fit location of the landmark on the target search area</td>
</tr>
<tr>
<td>Draw a box around the best fit</td>
</tr>
<tr>
<td>Reset event flag #42</td>
</tr>
</tbody>
</table>

Do until the end flag is set

Exit task off

Figure A-51
3.2.3.1 SSDA CORRELATION ROUTINE (CSDA)

CDSA implements the SSDA correlation algorithm. This routine exhaustively calculates the correlation values for a known landmark versus a target search area. The best fit is where the correlation value is at a minimum.

The algorithm is:

\[
\sum_{i=1}^{\text{# of lines}} \sum_{j=1}^{\text{# of elements}} \text{abs}((\text{WIN}(j,i)-\text{WINDOW MEAN})-(\text{LM}(j,i)-\text{LANDMARK MEAN}))
\]

FORMAT: CALL CSDA (WIN, LANMK, WM, LM, N, BFE, BFL, M)

- WIN = contains search area imagery
- LANMK = contains landmark area imagery
- WM = search area mean
- LM = landmark mean
- N = landmark size in pixels
- BFE = relative best bit element
- BFL = relative best fit line
- M = search area size
CSDA

<table>
<thead>
<tr>
<th>Set the best fit line and element to 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Determine the search area mean</td>
</tr>
<tr>
<td>Set the correlation to 0</td>
</tr>
<tr>
<td>Determine value and add into this placements corr. value</td>
</tr>
<tr>
<td>Do for the no. of elements in the landmark</td>
</tr>
<tr>
<td>Do for the no. of lines in the landmark</td>
</tr>
<tr>
<td>Is this a best fit?</td>
</tr>
<tr>
<td>T</td>
</tr>
<tr>
<td>Save best fit</td>
</tr>
<tr>
<td>F</td>
</tr>
<tr>
<td>NULL</td>
</tr>
<tr>
<td>Do for the no. of element placements</td>
</tr>
<tr>
<td>Do for the no. of line placements</td>
</tr>
</tbody>
</table>

Figure A-52

A-385
3.3 LIBRARY Routines

The library routines are those routines which perform a general function without being strictly I/O routines. These subroutines are:

1. DRABOX
2. MEAN
3. POS
3.3.1 DRAW A BOX (DRABOX)

The DRABOX routine draws a square box sized ISIZE x ISIZE. INTENS defines the intensity that this box is to be written at on the RM9100. IX and IY define the upper left hand corner of the data to be enclosed by the box. IBUF is a scratch array.

![Diagram of a box with corners labeled (IX-1, IY-1) and (IX, IY)]

**FORMAT:** CALL DRABOX (INTENS, ISIZE, IX, IY, IBUF)

**Restrictions:** 0<ISIZE<6>

<table>
<thead>
<tr>
<th>DRABOX</th>
<th>Initialize the scratch array to the output intensity</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Write the box on the RM9100.</td>
</tr>
</tbody>
</table>

Figure A-53
3.3.2 DETERMINE THE MEAN (MEAN)

The subroutine MEAN calculates the mean of a group of array elements. The array IARRAY (element, line) contains the data values. IARRAY is a square, 2 dimensional array. RWM is the calculated mean value. ISIZE is the number of elements and lines to be used. IES and ILS are the offsets into IARRAY from where the mean is to be taken.

FORMAT: CALL MEAN (IARRAY, RWM, ISIZE, ILS, IES)
MEAN

Initialize mean to zero

Sum up next element to previous sum

Do for the number of elements

Do for the number of lines

Divide the total sum by the number of elements x the number of lines

Figure A-54
3.3.3 POSITION CURSOR (POS)

POS is used to allow the user to position the cursor on the RM9100 screen. This routine allows a 9 x 9 pixel cursor to be drawn and moved around by the user. When switch 4 is flipped up, then down, the user has positioned the cursor to the desired location and control is returned to the calling routine.

IX and IY are the initial location of the center of the cursor. IX and IY also receive the final values of the cursor center. ILSTRT, ILSTOP, IESTRT and IESTOP define a rectangular area in which the cursor may move.

\[
\begin{align*}
(0,0) & \quad \text{CURSOR WINDOW} \\
\text{RM9100} & \quad \text{CURSOR WINDOW} \\
(\text{IESTRT, ILSTRT}) & \quad \text{CURSOR WINDOW} \\
(\text{IESTOP, ILSTOP}) & \quad \text{CURSOR WINDOW}
\end{align*}
\]

FORMAT: CALL POS (IX, IY, ILSTRT, ILSTOP, IESTRT, IESTOP)
POS

<table>
<thead>
<tr>
<th>T</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Is the stick in detent?</td>
<td>Read the data under where the cursor is to be drawn and save</td>
</tr>
<tr>
<td>NULL</td>
<td>Draw the cursor</td>
</tr>
<tr>
<td></td>
<td>Redraw the data</td>
</tr>
<tr>
<td></td>
<td>Do until switch 4 is up</td>
</tr>
</tbody>
</table>

Figure A-55
### 3.4 I/O Routines

The I/O routines are those routines which are used strictly for I/O purposes. These routines are:

<table>
<thead>
<tr>
<th>Subroutine Name</th>
<th>Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. ADJDAT</td>
<td></td>
</tr>
<tr>
<td>2. INIJOY</td>
<td></td>
</tr>
<tr>
<td>3. RAMRDB</td>
<td></td>
</tr>
<tr>
<td>4. RAMWRB</td>
<td></td>
</tr>
<tr>
<td>5. READJY</td>
<td></td>
</tr>
<tr>
<td>6. RESET</td>
<td></td>
</tr>
<tr>
<td>7. RMCHCK</td>
<td></td>
</tr>
<tr>
<td>8. RMRDIM</td>
<td></td>
</tr>
<tr>
<td>9. RMSET</td>
<td></td>
</tr>
<tr>
<td>10. RMSETO</td>
<td></td>
</tr>
<tr>
<td>11. RMWRIM</td>
<td></td>
</tr>
<tr>
<td>12. WIO</td>
<td></td>
</tr>
</tbody>
</table>
3.4.1 JOYSTICK I/O ROUTINES

The joystick device on the PDP 11/70 is terminal number TT4:. The joystick routines read this terminal and interpret the data. For data to be sent from TT4:, one of the switches marked 1 through 4 must be switched up.
3.4.1.1 ADJUST DATA (ADJUST)

The ADJUST subroutine is written in DEC MACRO assembly language. This routine adjusts the bits in the output data from the joystick. This routine is device dependent and called only by the subroutine READJY. The source code for ADJUST is found in the file ADJDAT.MAC.
3.4.1.2 INITIALIZE THE JOYSTICK (INIJOY)

The INIJOY subroutine assigns a logical unit number to the joystick. This subroutine is called once.

The parameter ILUN is the logical unit number to assign the joystick to. This unit number will be used by the routine READJY, to read the joystick device.

FORMAT: CALL INIJOY (ILUN)
3.4.1.3 READ THE JOYSTICK (READJY)

The READJY subroutine allows the user to read the joystick. Four words will be returned on each call. These words will be the switch word, X, Y, and Z words. The X, Y, and Z words have a range between 0 and 1023. When the stick is in detent the X and Y words are 512. When the stick is up in the upper left corner the X and Y words are 0. The Z word is set by the joystick knob. The switch word is bit oriented and follows this convention:

<table>
<thead>
<tr>
<th>Bit Location</th>
<th>15</th>
<th>14</th>
<th>13</th>
<th>12</th>
<th>11</th>
<th>10</th>
<th>9</th>
<th>8</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Switch Location

4 3 2 1

SWITCH WORD

If bit 0 is set switch 1 is up. If bit 1 is set, switch 2 is up, and so on. There are 4 switches.

The IWRDCT parameter must always be 4. ABUFF is the buffer to receive the 4 words of output. ILUN is the logical 4-word number of the joystick as initialized by INIJOY.

FORMAT: CALL READJY (ABUFF, IWRDCT, ILUN)
3.4.2 RAMTEK 9100 I/O ROUTINES

All RM9100 I/O routines use the WTQIO and GETADR system I/O subroutines. The basic structure for a RM9100 I/O routine is the following:

1. The command instruction is placed in a buffer.

2. The address of the command buffer and the length of the buffer is stored in a parameter array. This is where GETADR is used.

3. Depending on the function, the appropriate data from the parameter array is written/read to the RM9100. The operation of reading/writing is performed by using a WTQIO.

Two auxiliary routines that have been used by all RM9400 I/O subroutines are W10 and RMCHCK. These routines have been developed so that all I/O functions will utilize them and help conserve task space.
3.4.2.1 RM9100 READ IMAGE WINDOW (RAMRDB)

The RAMRDB subroutine allows the user to read a rectangular area from the RM9100 memory. This rectangular area is defined by the calling routine by setting LXSTART, LXSTOP, IYSTART and IYSTOP. Up to 2000 pixels may be read in any one call to RAMRDB. ISCAN defines the scan direction of the read. IDATA is the array in which data will be read and ILEN is the number of pixels to read into IDATA.

FORMAT: CALL RAMRDB (ISCAN, LXSTART, LXSTOP, IYSTART, IYSTOP, IDATA, ILEN)
3.4.2.2 RM9100 WRITE IMAGE WINDOW (RAMWRB)

The RAMWRB subroutine allows the user to write a rectangular area to the RM9100 memory. This rectangular area is defined by the calling routine by setting IXSTRT, IXSTOP, IYSTRT, and IYSTOP. Up to 2000 pixels may be written in any one call to RAMWRB. ISCAN is the scan direction of the write. IDATA is the array containing ILEN number of pixels to be written.

FORMAT: CALL RAMWRB (ISCAN, IXSTRT, IXSTOP, IYSTRT, IYSTOP, IDATA, ILEN)
3.4.2.3 RESET THE RM9100 (RESET)

The RESET instruction is an RM9100 system clear function identical to the power-on (or hard reset) initialization sequence. All pending operations are discarded. The refresh memory will be erased.

FORMAT: CALL RESET
3.4.2.4 CHECK RM9100 I/O (RMCHCK)

The RMCHCK subroutine checks to see if the previous RM9100 I/O was correctly handled by the RM9100. The status words are contained in the common block IO. The directive status word, IDS, is equal to 1 if the directive was sent successfully. The I/O status word, IOST(1), is 1 if the issued I/O executed correctly. If either of these variables do not equal 1, the I/O was unsuccessful.

FORMAT: CALL RMCHCK
3.4.2.5 RM9100 READ IMAGE (RMRDIM)

The RMRDIM subroutine allows the user to read a rectangular area from the RM9100 memory. Up to 2000 pixels may be read in any one call to RMRDIM. The default window of the entire screen is used.

ISCAN determines the primary and secondary scan directions.

ISCAN = 0  
primary - left to right  
secondary - top to bottom

ISCAN = 4  
primary - top to bottom  
secondary - left to right

IXS and IYS are the element and line values in memory at which the read is to begin. IDATA is the buffer in which data is to be filled and ILEN is the number of pixels to be read.

FORMAT: CALL RMRDIM (ISCAN, IXS, IYS, IDATA, ILEN)
3.4.2.6 SET UP RM9100 (RMSET)

The RMSET subroutine resets the window parameters of the RM9100 back to their original values. These are:

- IXSTRT = 0
- IXSTOP = 319
- IYSTRT = 0
- IYSTOP = 255

FORMAT: CALL RMSET
3.4.2.7 RESET RM9100 ORIGIN (RMSEO)

This routine allows the user to set the RM9100 origin parameters. Updating the y origin by one in a loop will give an illusion of vertical scrolling. This routine is used to scroll image data.

FORMAT: CALL RMSEO (IXORG, IYORG)
3.4.2.8 RM9100 WRITE IMAGE (RMWRIM)

The RMWRIM writes to any rectangular area in the RM9100, in word mode. Up to 2000 pixels may be written in any one call to RMWRIM. The default window of the entire screen is used.

In the LMEM programs, ICHNL always equals -1, ISCALE always equals 0, and IFCTN always equals 0. ISCAN is the scan direction to be used for the write.

ISCAN = 0  
primary - left to right  
secondary - top to bottom

ISCAN = 4  
primary - top to bottom  
secondary - left to right

IXS and IYS are the element, line start where the write to memory is the occur. IDATA is the buffer address of the element(s) to be written and ILEN is the number of pixels to be written.

FORMAT: CALL RMWRIM (ICHNL,ISCAN,ISCALE,IFCTN,IXS,IYS,IDATA,ILEN)
3.4.2.9 ISSUE WAIT FOR QUEUE I/O (WIO)

The WIO subroutine issues a PDP 11/70 WTQIO system directive. There are two arguments passed to this routine, IFCTN and IDEV. IFCTN specifies the I/O function to be executed and IDEV is the device in which the I/O function is to occur.

This subroutine is used for all RM9100 I/O. The common block IO contains the other parameters required by the WTQIO instruction.

FORMAT: CALL WIO (IFCTN, IDEV)

FUNCTION CODES: "410 - Write Logical block (RM9100)
"1010 - Read Logical block (RM9100)