AD-E950311

(12)

AD A121622

TECHNICAL REPORT RD-CR-82-27

PERSHING II SIMULATION STUDIES

J. D. Knight; R. L. Lynch; D. A. Pyles; R. N. Seitz; T. V. Thornton
Georgia Institute of Technology
Engineering Experiment Station
Atlanta, GA 30332

PREPARED FOR
Systems Simulation and Development Directorate
US Army Missile Laboratory

July 1982

U.S. ARMY MISSILE COMMAND
Redstone Arsenal, Alabama 35809

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

DISPOSITION INSTRUCTIONS

DESTROY THIS REPORT WHEN IT IS NO LONGER NEEDED. DO NOT RETURN IT TO THE ORIGINATOR.

DISCLAIMER

THE FINDINGS IN THIS REPORT ARE NOT TO BE CONSTRUED AS AN OFFICIAL DEPARTMENT OF THE ARMY POSITION UNLESS SO DESIGNATED BY OTHER AUTHORIZED DOCUMENTS.

TRADE NAMES

USE OF TRADE NAMES OR MANUFACTURERS IN THIS REPORT DOES NOT CONSTITUTE AN OFFICIAL INDORSEMENT OR APPROVAL OF THE USE OF SUCH COMMERCIAL HARDWARE OR SOFTWARE.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER RD-CR-82-27	2. GOVT ACCESSION NO. AD-A121622	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) PERSHING II SIMULATION STUDIES		5. TYPE OF REPORT & PERIOD COVERED Final Technical Report 3 February - 28 May, 1982
		6. PERFORMING ORG. REPORT NUMBER A-3165
7. AUTHOR(s) J. D. Knight, R. L. Lynch, D. A. Pyles, R. N. Seitz, and T. V. Thornton		8. CONTRACT OR GRANT NUMBER(s) DAAH01-81-D-A003 Delivery Order 0038
9. PERFORMING ORGANIZATION NAME AND ADDRESS Georgia Institute of Technology Engineering Experiment Station Atlanta, Georgia 30332		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS U. S. Army Missile Command DRSMI-IYB/Koger Redstone Arsenal, AL 35898		12. REPORT DATE July 1982
		13. NUMBER OF PAGES 93
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) U. S. Army Missile Command DRSMI-RDF/Hallum Redstone Arsenal, AL 35898		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release, distribution unlimited

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

18. SUPPLEMENTARY NOTES

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

Pershing II, PII, U70 Simulator, White Sands Missile Range, WSMR Range Safety, Missile Flight Validation, Aerodynamic Validation, Tactical Ballistic Missile Trajectories, Maximum Likelihood Method

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

Pershing II flight simulations were performed using the U70 missile simulation program to determine (1) in the event of an accidental nozzle deflection, how fast the missile would leave its (safe) flight corridor; (2) how well the U70 aerodynamic simulation matches the actual flight data; and (3) the trajectory profiles for nine Tactical Ballistic Missile flight trajectories. Also an advanced simulation program developed by TRW, Inc., was partially converted to run on an inhouse computer.

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

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

PREFACE

This report was prepared by the Electromagnetics Laboratory of the Engineering Experiment Station, Georgia Institute of Technology, under Contract No. DAAH01-81-D-A003, Delivery Order No. 38, for the U. S. Army Missile Command. The contract technical monitor was M. M. Hallum and the contract project monitor was D. L. Cobb, DRSMI-RDF, both from the Systems Simulation and Development Directorate of the U. S. Army Missile Command. The contract period extended from February 3, 1982, through May 28, 1982.

The views and conclusions in this document are those of the authors and should not be interpreted as necessarily representing the official policies of the U. S. Army Missile Command.

The authors would like to acknowledge the valuable contributions of J. P. Billingsley, and D. L. Cobb.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

LIST OF TABLES

LIST OF FIGURES

I. INTRODUCTION

A. Statement of the Task and General Approach

This report documents the results of the work performed under the Pershing II Task Order DAAH01-81-D-A003, Delivery Order 0038 (GT/EES Project No. A-3165). The task order required that the following subtasks be performed:

- The range safety problems which would arise if the motor nozzle were unintentionally deflected during boost phase were to be evaluated.

- The aerodynamic simulation model used in a Pershing II computer-simulation program was to be validated against actual flight test data.

- Computer simulation results were to be compared with actual flight data for a set of Tactical Ballistic Missile (TBM) trajectory profiles.

The results of these comparisons were "to be presented in a form useful for evaluation".

B. Background

The task documented with this report was an outgrowth of an earlier task (Pershing II Debris Studies, DAAH01-81-D-A003, Delivery Order 0017, GT/EES Project No. A-2946). The earlier task was aimed at evaluating launch and target sites for Pershing II firings, determining the effect of varying aerodynamic conditions on the missile's flight behavior, and conducting range safety studies. A follow-on task order request is in preparation and will expand upon the work performed in support of these two projects.

C. Contents of This Report

A detailed discussion of the subtask objectives and accomplishments is given in Section II. Section III discusses the U70 and TRW simulation programs used to carry out Pershing II missile simulations [1]. It includes a description of the U70 program and the computer on which it runs, modifications that had to be made to it to support this task, and the operational procedures for running it. It also discusses the problems involved in converting the TRW program to run on the U. S. Army Missile Command's Perkin-Elmer 3220 computer. Section IV discusses the results that were obtained in the execution of this task. Finally Section V presents recommendations for future work.

II. DETAILED OBJECTIVES

A. Subtask Objectives and Accomplishments

During the performance of this task, two of the three required computer simulation studies were performed, using the U70 trajectory simulation program [1] and the TRAJ targeting program. The third study, the comparison of Pershing II computer simulation studies with actual flight data, could not be carried out because flight test data were not available during the period of performance of the task. The effort that would normally have been expended upon this comparison of flight test data and simulation results was directed instead toward the implementation of a Maximum Likelihood Method computer subroutine for data analysis.

Similar studies were to be undertaken using a new Pershing II simulation program developed by TRW, Inc., but the full computer program could not be obtained in time for use on this task. Its treatment in this report is necessarily restricted to a discussion of the conversion and documentation of the program itself rather than the results obtained using it. A more detailed description of the four subtask objectives is given below.

1. Flight Safety (Motor Nozzle Deflection) Study

The motor nozzle deflection study was a flight safety study conducted to determine the breakup point of the Pershing II missile in the event of an engine malfunction. A nozzle deflection would cause the missile to deviate from its normal flight path. The objective of the study was to determine how quickly the missile would leave its flight corridor in the event of a spurious nozzle deflection and whether Range Safety would detect this aberration in time to permit destruction of the missile before it could endanger life or property.

2. Aerodynamic Model Validation Study

A validation of the aerodynamic mathematical model used in the U70 Pershing II digital simulator was required for utilization of flight data. To compare the simulation results to the flight data, a statistical technique called the Maximum Likelihood Method was to be used. This technique de-serves special mention because of its key role in the performance of this task. This technique is used to fit theoretically predicted values of relevant variables to experimentally derived values when the experimental values contain random measurement errors. Generally, the Maximum Likelihood Method provides better estimates of the underlying true values of variables than a least-squares curve fit because, in making its estimates, the Maximum Likelihood Method uses a prior knowledge of the standard deviations of instrument measurement errors as additional information to help refine its estimates.

3. Tactical Ballistic Missile Trajectory Study

The Tactical Ballistic Missile (TBM) study took the form of a matrix of U70 runs with various flight paths along one dimension and different flight characteristics along the other dimension.

One launch site and three target sites were used to generate the three flight paths. Then for each flight path, a set of three runs was made for three assumed conditions - first, a "nominal motor" or unperturbed-baseline flight plan; second, a ballistic re-entry vehicle flight plan in which no aerodynamic corrections are applied to the missile during its terminal re-entry phase; and third, a flight plan assuming offsets in target latitude and longitude from its normal target location. Thus, a total of nine runs were made for this study.

B. Conversion and Installation of The TRW Simulator

The TRW program is a six-degree-of-freedom simulator developed for the purpose of validating Pershing II onboard software. This simulator contains aerodynamic modeling features and provides simulation capabilities not found in the U70 simulation program - capabilities which the U. S. Army Missile Command's Systems Simulation and Development Directorate, Systems Evaluation Branch, seeks to add to its repertoire of simulation programs. The job of understanding the program and converting it to the Missile Command's computer was assigned to Georgia Tech as a part of this task (A-3165). TRW delivered a copy of its program to the Systems Evaluation Branch. However, when program compilation and link loading was attempted, certain sections of the program, such as the MAIN control program and the INTERP routine were found to be missing. These could not be obtained during the period of performance of the task, and therefore, could not be converted. Insofar as possible, the remainder of the program was partially converted from TRW's Digital Equipment Corporation VAX 11/780 computer to the System Simulation and Development Directorate's Perkin-Elmer (PE) 3220 computer after making minor changes to accommodate for the differences between the FORTRAN capabilities of the two computers.

Although the installation of the TRW program on the PE 3220 computer and the incorporation of its aerodynamic model within the U70 program is not explicitly required in this task's scope of work, it is implicitly required for the proper performance of the task. Since it has consumed much of the task's effort, it will be discussed in this report as though it were a fourth task objective.

III. THE U70 AND TRW SIMULATION PROGRAMS

A. The U70 Program

1. Description of the U70 Program

The U70 trajectory program employed in the performance of this task, uses the given prelaunch, target, and flight conditions to simulate the flight of a single-stage or two-stage missile and its associated maneuvering re-entry vehicle. This program utilizes the equations and requirements found in the Pershing Launch Computer and the Pershing Airborne Computer to simulate the missile flight from launch through the boost, midcourse, and terminal portions of flight to impact [2].

The U70 trajectory program runs on a Perkin-Elmer 3220 computer system equipped with 800 kilobytes of random access memory, a nine-track tape drive, three 67-megabyte disk drives, and five demand terminals (including the system console). The simulation is coded in FORTRAN VII-D, although it is probably compatible with other versions of FORTRAN.

The program is able to restart the missile simulation at any time during the terminal re-entry portion of flight upon entry of data such as time, position, velocity, and orientation of the missile at the desired restart time, plus the restart data acquired from a previous simulation.

2. Modifications to the U70 Program

a. Modifications for the Motor Nozzle Deflection Study

For this study, modifications were made to the U70 program's BAPLT (Boost Autopilot) subroutine to enable desired deflection angles to be input at some time during boost. These modifications are as follows:

```
     MODIFICATIONS TO BAPLT

470  CONTINUE
     IPND = ICON(50)
     IYND = ICON(51)
     IF(IPND .EQ. 0)  GO TO 453
     IF (TIMC .LT. CIN(382)) GO TO 453
     DNV (1, 1) = CIN (380) * DTOR

453  CONTINUE
     IF(IYND .EQ. 0) GO TO 454
     IF(TIMC .LT. CIN(382)) GO TO 454
     DNV (1, 2) = CIN(381) * DTOR

454  CONTINUE
     RETURN
     END
```

b. Aerodynamic Model Flight Validation

As mentioned previously, the Maximum Likelihood Method (MLM) of data correlation was intended to help fit the simulator-generated aerodynamic flight data to the actual test-flight telemetry data, i.e., to "average out" random telemetry instrumentation errors in order to provide more meaningful comparisons between the simulated and experimental results. The machinery for accomplishing this was incorporated into a new U70 subroutine called DATACORR. The addition of DATACORR to the U70 program required the creation of a new link procedure, TU70LINK.CSS, which builds a new task called TU70.

The TU70 task requires as inputs the instantaneous inertial-frame X(t), Y(t), and Z(t) position coordinates and $\phi(t)$, $\theta(t)$, and $\psi(t)$ (roll, pitch, and yaw) angular coordinates of the missile and their time derivatives (where t represents time). For program checkout purposes, constant offsets have been programmed into the inertial position and rotation inputs. Since the MLM uses an iterative technique, an upper limit must be entered on the number of iterations that the computer is to attempt before giving up (in case the iteration sequence does not converge). Finally, a 6 x 6 matrix of standard deviations and cross-correlations of measurement noise must be entered for the three position and the three rotation coordinates at t = 0, e.g., at launch (Table 1).

This matrix contains in its [1, 1] location, the variance (σ_θ^2) of the instrumentation measurement errors of the missile's x-coordinate at t = 0. In its [1, 2] location, it contains the mean square error in X due to an error in Y ($\sigma_{\theta\phi}^2$), and so forth, for all 36 elements. These TU70 input requirements are summarized in Table 2. The coding for the DATACORR subroutine is reproduced in Appendix A.

c. Data Format Modifications for the Tactical Ballistic Missile Simulations

Modifications were made for the Tactical Ballistic Missile trajectory simulations to provide the same output format for all stages of flight. Different variables were output for the boost and for the re-entry stages; these variables were combined to produce one block of output data for both stages. This required changes in the BOUT (Boost Output) and RVOUT (Re-entry Vehicle Output) subroutines. These changes are documented in Appendix B.

TABLE 1. COVARIANCE MATRIX OF MEASUREMENT NOISE, N

$$
\begin{bmatrix}
\sigma^2_{\theta} & \sigma^2_{\theta\phi} & \sigma^2_{\theta\psi} & \sigma^2_{\theta x} & \sigma^2_{\theta y} & \sigma^2_{\theta z} \\
\sigma^2_{\phi\theta} & \sigma^2_{\phi} & \sigma^2_{\phi\psi} & \sigma^2_{\phi x} & \sigma^2_{\phi y} & \sigma^2_{\phi z} \\
\sigma^2_{\psi\theta} & \sigma^2_{\psi\phi} & \sigma^2_{\psi} & \sigma^2_{\psi x} & \sigma^2_{\psi y} & \sigma^2_{\psi z} \\
\sigma^2_{x\theta} & \sigma^2_{x\phi} & \sigma^2_{x\psi} & \sigma^2_{x} & \sigma^2_{xy} & \sigma^2_{xz} \\
\sigma^2_{y\theta} & \sigma^2_{y\phi} & \sigma^2_{y\psi} & \sigma^2_{yx} & \sigma^2_{y} & \sigma^2_{yz} \\
\sigma^2_{z\theta} & \sigma^2_{z\phi} & \sigma^2_{z\psi} & \sigma^2_{zx} & \sigma^2_{zy} & \sigma^2_{z}
\end{bmatrix}
$$

TABLE 2. TU70 INPUTS FOR THE DATA CORRELATION FEATURE

INDICATORS:

CIN(75) - Contains an on-off "switch" for the data correlation feature (0 = Off, 1 = On)

CIN(70) - Contains an on-off "switch" for the computation of the transpose of the sensitivity matrix, S (0 = Off, 1 = On)

CONSTANTS:

CIN(499) - Contains the number of experimental data points

CIN(498) = σ_x^2 CIN(497) = σ_y^2 CIN(496) = σ_z^2 CIN(495) = σ_θ^2 CIN(494) = σ_ϕ^2 CIN(493) = σ_ψ^2	Contains the initial inputs of the covariance matrix of the measurement noise (assumes a diagonal matrix)
CIN(492) = E_θ CIN(491) = E_ϕ CIN(490) = E_ψ CIN(489) = E_x CIN(488) = E_y CIN(487) = E_z	Contains the error-vector differences between the measured and computed values of x, y, z, θ, ϕ, and ψ

---

3. Instructions for Using the U70 Program

A description of the procedures for compiling, running, linking, restarting the program, and making output tapes is given in Appendix C, together with a list of the .CSS files used to accomplish these tasks, and a list of the U70 source files.

The TU70 program contains an array of input constants and indicators called CIN. Table 2 contains definitions for the constants and indicators which serve as inputs for the DATACORR subroutines for the TU70 task.

B. The TRW Program

1. Compiling the TRW Simulation Program

The TRW program was originally set up for interactive compilation, program linkage, and execution. However, in converting the program to the Perkin-Elmer 3220 computer, it became necessary to switch to batch mode compilation and linkage. The reason for turning to batch mode operation was that not every file or function required by the program is present in the program itself. Some of the necessary information is stored in independent files, and operating in batch mode permits these independent files to be found during compilation and to be linked to the MAIN program during the program linkage phase. To accomplish this result, the $BATCH command had to be placed at the beginning of the program, the $BEND command at the end of the program, and the $PROG declarations put at the beginning of each subroutine.

Next, a "compile file" was produced by modifying a copy of the systems file F7CAE.CSS (which contains the FORTRAN VII compiler and the linking loader) and then using it to compile the TRW program. Compilation time was saved by compiling one subroutine at a time, producing an object code image and saving it in a separate file, and then linking it together with other object code subroutines during the link phase. This meant that only those subroutines which had been updated had to be recompiled.

F7CAE.CSS is a procedure which compiles and links any program. The compiler in F7CAE.CSS uses system file F7D.TSK to produce an object file and F7D.ERR to record errors found during the compile phase. The F7CAE.CSS link loader processes object files generated by the compiler and creates a task file for execution. The link loader also creates a load map and provides a record of any undefined external references or symbols. Appendix D lists the file names used by the TRW program and Appendix E lists the program's subroutine names and calling sequences.

Of the many error messages which surfaced during the first compilation, a substantial fraction was associated with VAX-peculiar FORTRAN INCLUDE statements. INCLUDE is a VAX FORTRAN compile-time command which causes the compiler to copy a disk file into a program - typically, a file containing COMMON and EQUIVALENCE statements. This permits a simple INCLUDE statement to be substituted in subroutines for the extensive COMMON and EQUIVALENCE statements that would ordinarily be found there instead. This practice not only reduces the amount of effort required to write and update the program but also reduces the chances of making a mistake in coding, since the COMMON and EQUIVALENCE statements only have to be updated at one place in the program rather than in every subroutine in which they are referenced.

Another VAX-peculiar FORTRAN VII enhancement is the DO WHILE command. Since the DO WHILE statement is not recognized by Perkin-Elmer's FORTRAN VII compiler, it was necessary to replace DO WHILE's in the original program with ordinary DO statements in the Perkin-Elmer version of the program.

The above modifications constitute the principal programming changes that have so far had to be made to the TRW subroutines in order to convert them from the VAX 11/780 computer to the Perkin-Elmer 3220 computer.

2. Subroutines Missing From the TRW Program

As mentioned in Section II.E., the TRW simulation program was delivered with some major portions of the program missing. The most critical missing subroutine was INTERP(Interpolation) which, by interpolating large data tables, would have provided the thrust characteristics, mass properties, and aerodynamic data needed to simulate the flight. The main control program, MAIN, was also missing, although a program SDCTRL (Six-Degree-of-Freedom Control) was found which seems to perform similar functions. Essential portions of the program that were required to simulate steering, guidance, and navigation in the Pershing Airborne Computer were also missing. Finally, an error message routine ERRMSG was missing.

a. INTERP(Interpolation)

With the aid of a former developer of the TRW INTERP routine, a prior version of INTERP was located in an earlier edition of the TRW simulator and efforts were made to understand it and use it in the current version of the TRW program. At the heart of the INTERP routine is the above mentioned involved set of aerodynamic tables, organized in a way that minimizes storage requirements without unduly increasing run times. A discussion of what has been learned about this multi-dimensional data table is presented in Appendix F.

b. MAIN and SDCTRL (Six-Degree-of-Freedom-Control) Programs

After finding that the MAIN routine was missing from the TRW simulator, efforts were begun to develop such a function-controlling routine. An earlier version of MAIN was found in a program listing and the listing was used to help understand what was needed to recreate a current MAIN routine. A flowchart of this early version MAIN routine is contained in Appendix G. Appendix H contains a flowchart of the SDCTRL routine.

c. The Output Processor

The TRW simulation program has an output processor which takes advantage of large arrays to store values of variables for output. These arrays are saved in three groups named BIGØ1R, BIGØ1I, and BIGØ1D, which contain Real, Integer, and Double Precision variables, respectively. These output-variable values are transferred to the BIGØ1 arrays for use by the output processor through EQUIVALENCE statments located in each subroutine [1]. Each variable stored in BIGØ1R, BIGØ1I, and BIGØ1D is stored on a disk file BIGØ1.DAT. The names of these variables are listed in Appendix I.

3. Tape-to-Disk Conversion of the TRW Program

TRW delivered its simulation program to the System Evaluation Branch on a nine-track 1600 bit-per-inch tape. This was loaded onto a Perkin-Elmer 3220 disk pack using the following procedure:

Mount the tape on a 1600 bpi tape drive.

```
> COPY
> AL FILE1.FTN,IN,80
> OUT FILE1.FTN
> COPY *,*
```

IV. RESULTS

A. Flight Safety (Motor Nozzle Deflection) Results

As mentioned earlier in Section II,B, this Pershing II missile simulation was carried out using the U70 simulator to determine what would happen if, through some hardware failure, the nozzle were accidently deflected during a live missile firing. In carrying out these simulated flight tests, the missile was allowed to "fly" unperturbed (in the computer) for a short time. Then the motor nozzle was deflected by a pre-determined angle and the simulation continued until either the missile's total angle of attack exceeded 15° or the normal (perpendicular to the body of the missile) acceleration exceeded 5 g's. Either one of these occurences was assumed to generate sufficiently unstable conditions that breakup of the missile would occur, so the simulation was terminated at that point.

The spurious nozzle deflections were assumed to occur at 30 seconds and at 49 seconds into the flight. The 30-second flight time was chosen because, at 30 seconds, the aerodynamic forces acting on the missile would be at or near their maximum, i.e, a worst-case condition. The 49-second flight time was chosen because it is almost at first-stage burnout, and is a time when missile failure is likely. Three nozzle deflection angles were tried, 0.5°, 2°, and 7.6°, first in pitch and then in yaw, leading to a total of twelve cases (two deflection times and three nozzle deflection angles, first in pitch and then in yaw). All the cases were simulated flights from McGregor, New Mexico, to McDonald's Well, New Mexico, with the targeting conditions given in Table 3 below. Before presenting these results in detail, some data comparison problems need to be discussed.

TABLE 3. LAUNCH SITE AND TARGET

McGregor, NM	LAUNCH SITE
	Latitude = 32.09575°
	Longitude = -106.2035°
	Altitude = 1251.0000 m
McDonald's Well, NM	TARGET SITE
	Latitude = 33.113580°
	Longitude = -106.35897°
	Altitude = 1228.00000 m

In order to provide a standard by which to compare the deviated-nozzle runs, a nominal or baseline simulation was run in which it was assumed that the missile followed a nominal, undeflected flight plan. Since the deviated nozzle results were printed at 0.1 second intervals, it would have been advantageous to have also printed the baseline results at 0.1 second intervals for purposes of comparison. Unfortunately, the large volume of printout generated by the U70 program made it impractical to print results at 0.1 second intervals, and the results were available only at 1.0 second intervals. Consequently, the baseline results had to be interpolated at 0.1 second intervals in order to compare them to the deviated-nozzle results, i.e., a direct comparison was not possible.

A second data comparison problem resides in the fact that, for unknown reasons, the first data points in the deviated-nozzle simulation printouts (the results for 30.0 seconds and the results for 49.0 seconds) are invalid. These 30- and 49-second numbers are not random but seem to come from some earlier time in the flight. Subsequent results appear to be correct when compared to the unperturbed flight results. For example, the interpolated unperturbed flight results for 30.1 seconds agree to four decimal places with the deviated-nozzle results, with subsequent data departing from the baseline data in a regular and reasonable way. It is the authors' conclusion that the faulty first point in each set of numbers was the result of a print-routine malfunction rather that an error in the deviated nozzle results. However, the reader needs to be aware of this characteristic in the data.

Figure 1 depicts the missile's flight path for an unperturbed flight plan and for the three different yaw-axis nozzle deflection angles looking down from above when the nozzle deflections occur 30.0 seconds into the flight. Figure 1 is a plot of downrange coordinates versus crossrange coordinates. The "straight line" in Figure 1 represents the path of an unperturbed missile. (It curves about 1° to the right but the curvature is too small to be visible in the plot.) For all three deflections, the missile disintegrates because the yaw angle of attack exceeds 15° rather than because the normal acceleration exceeds 5 g's. Note that for the smaller the nozzle deflection angle, the farther the missile can travel before it reaches the critical angle of 15°.

Figure 2 shows a similar plot when the nozzle deflection occurs 49 seconds into the flight. Note that the flight path deflections appear to be smaller than they were when the nozzle deflection occurred at 30 seconds (Figure 2). In reality, the deflections are the same but the velocity of the missile is greater, and this increases the horizontal scale of the plot. (The results for a nozzle deflection of 7.6° are missing because of a faulty computer run.)

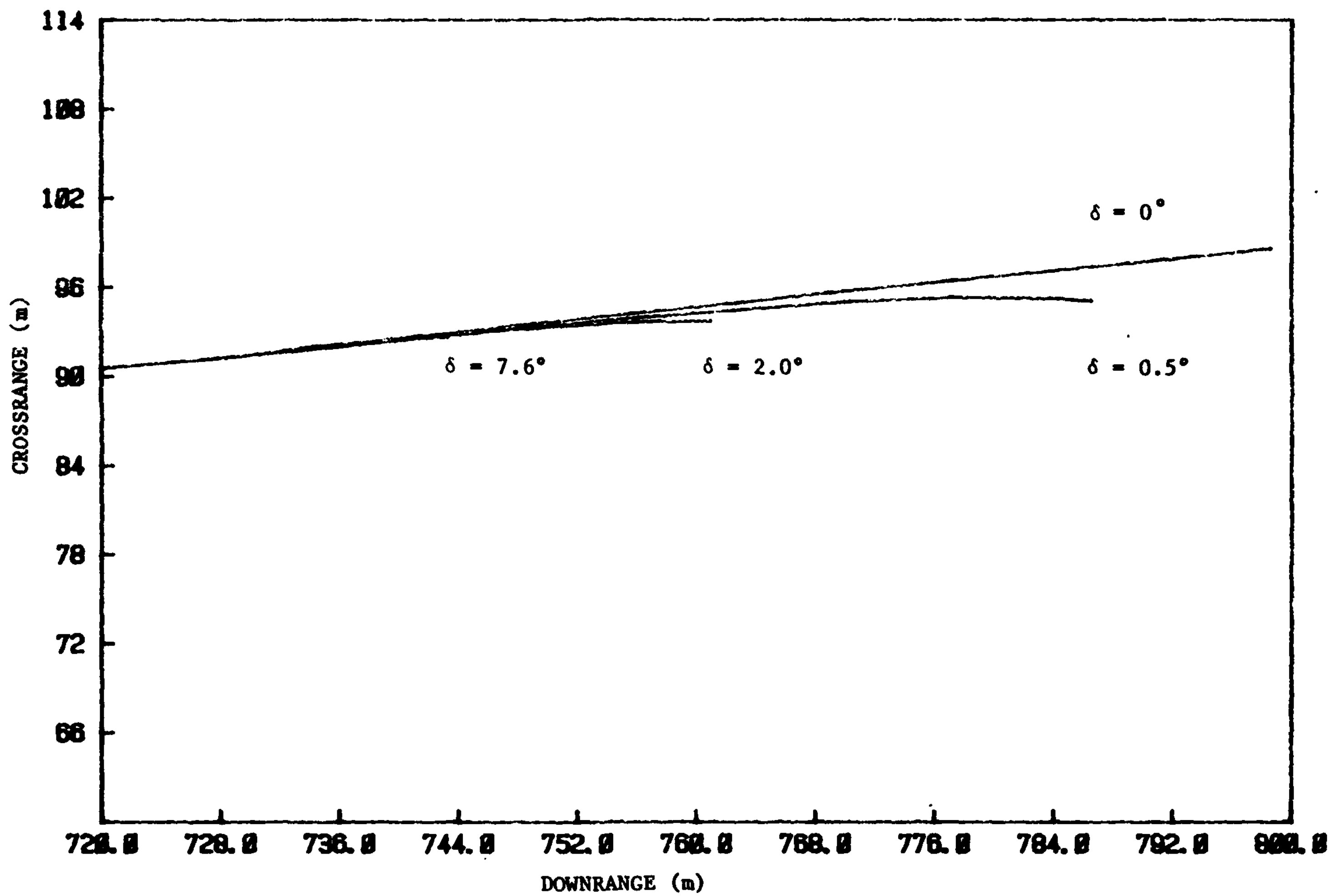

Figure 1. Crossrange vs. Downrange for Yaw Nozzle Deflections (δ's) Occurring at 30 Seconds into the Flight.

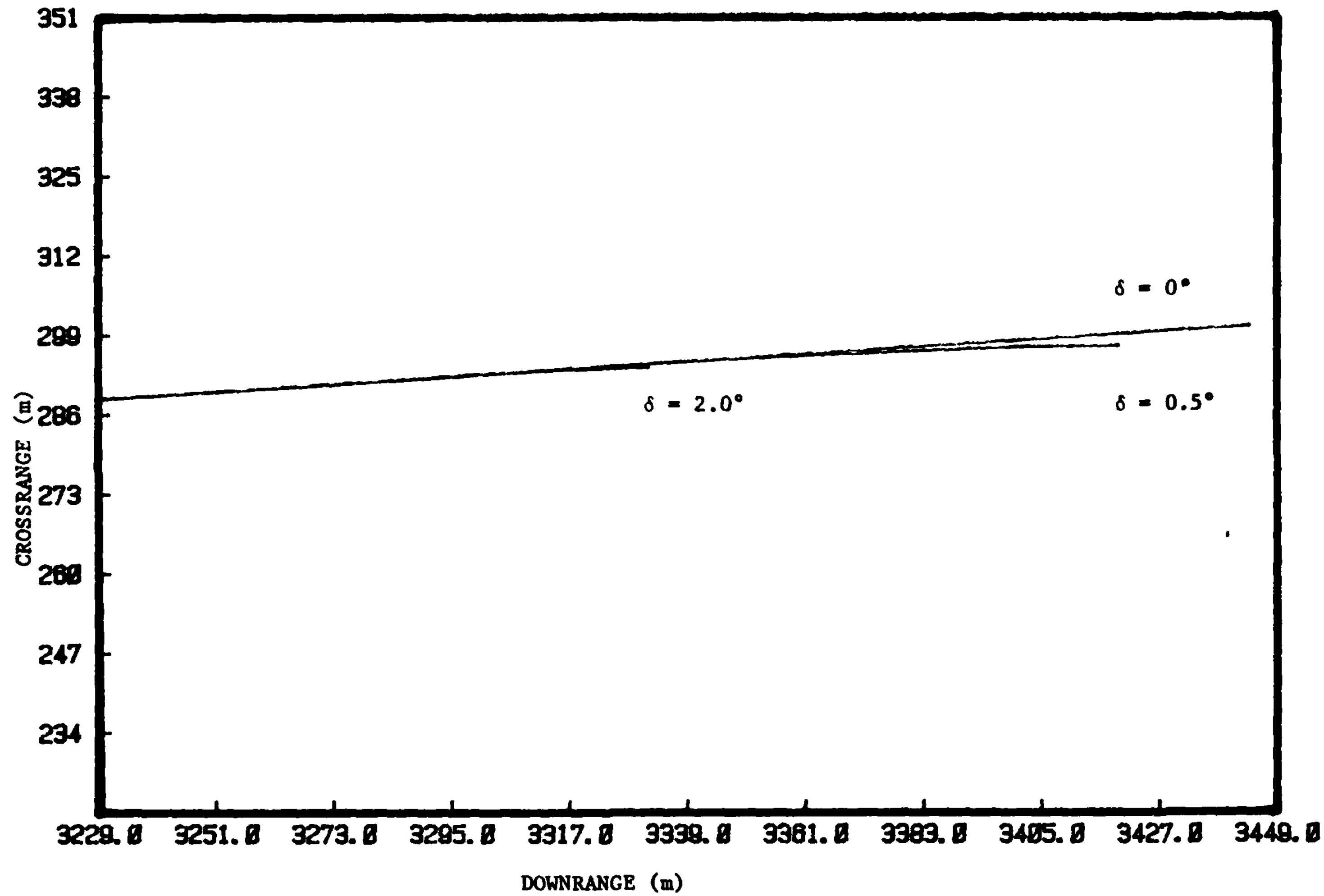

Figure 2. Crossrange vs. Downrange for Yaw Nozzle Deflections (δ's) Occurring at 49 Seconds into the Flight.

Figure 3 shows a plot of the missile vertical pitch angle versus its downrange coordinates when a nozzle deflection in pitch occurs at 30 seconds, and Figure 4 depicts similar plots for the 49 second pitch nozzle deflection.

Figures 5 through 8 show how the normal (lateral) accelerations of the missile vary with time. An examination of these curves confirms that in every case, the simulated flights terminated before the normal accelerations exceeded the destructive limit of 5 g's, i.e., the flights terminated because the missile's total angle of attack exceeded 15°. There is an interesting tendency for these normal accelerations to rise, dip, and then rise again. No simple explanation has been advanced for this behavior, although the Pershing II guidance system is sufficiently complex that a complex response might be expected in the event of a major nozzle deflection malfunction. (The results for a nozzle deflection of 7.6° are missing because of the faulty computer run mentioned previously in connection with Figure 2.)

These simulations show that the missile would leave its allowable flight corridor and exceed the allowable angle of attack within approximately one second.

B. Aerodynamic Model Validation Results

As mentioned in Section II.B., this study could not be completed because Pershing II flight test data were not available in time.

In support of this task, the DATACORR subroutine, incorporating the Maximum Likelihood Method of data correlation, was written, compiled, and, insofar as possible, was tested. However, one important part of this subroutine, the derivation of sensitivity coefficients, was not completed.

C. Tactical Ballistic Missile Trajectory Results

As mentioned in Section II.C., nine U70 runs were made for this study for nine input trajectory profiles. Three of these profiles entailed offsets in target latitude and longitude.

When the study was completed, the results were output on tape and delivered to the sponsor.

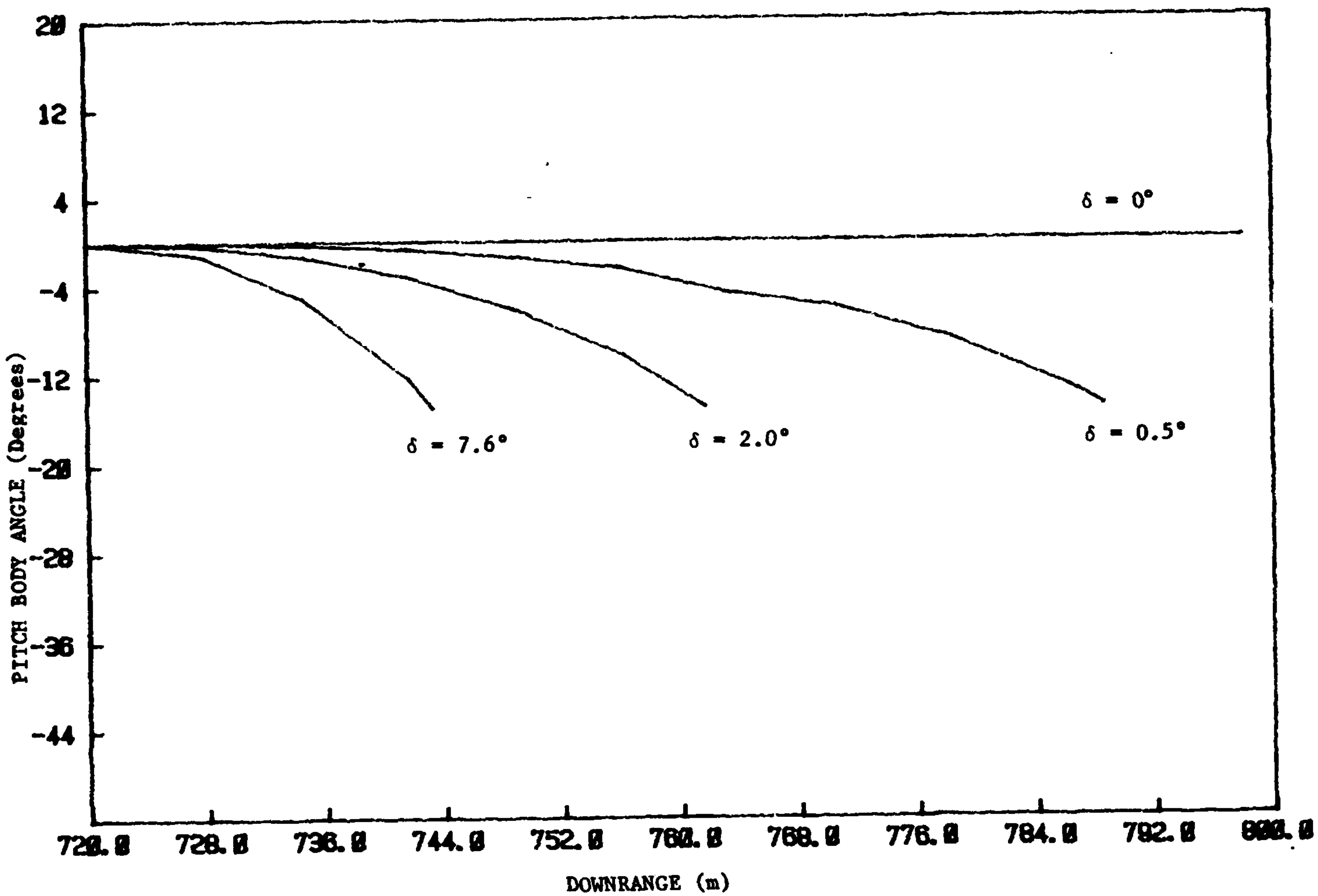

Figure 3. Pitch Body Angle vs. Downrange for Pitch Nozzle Deflections (δ's) Occurring at 30 Seconds into the Flight.

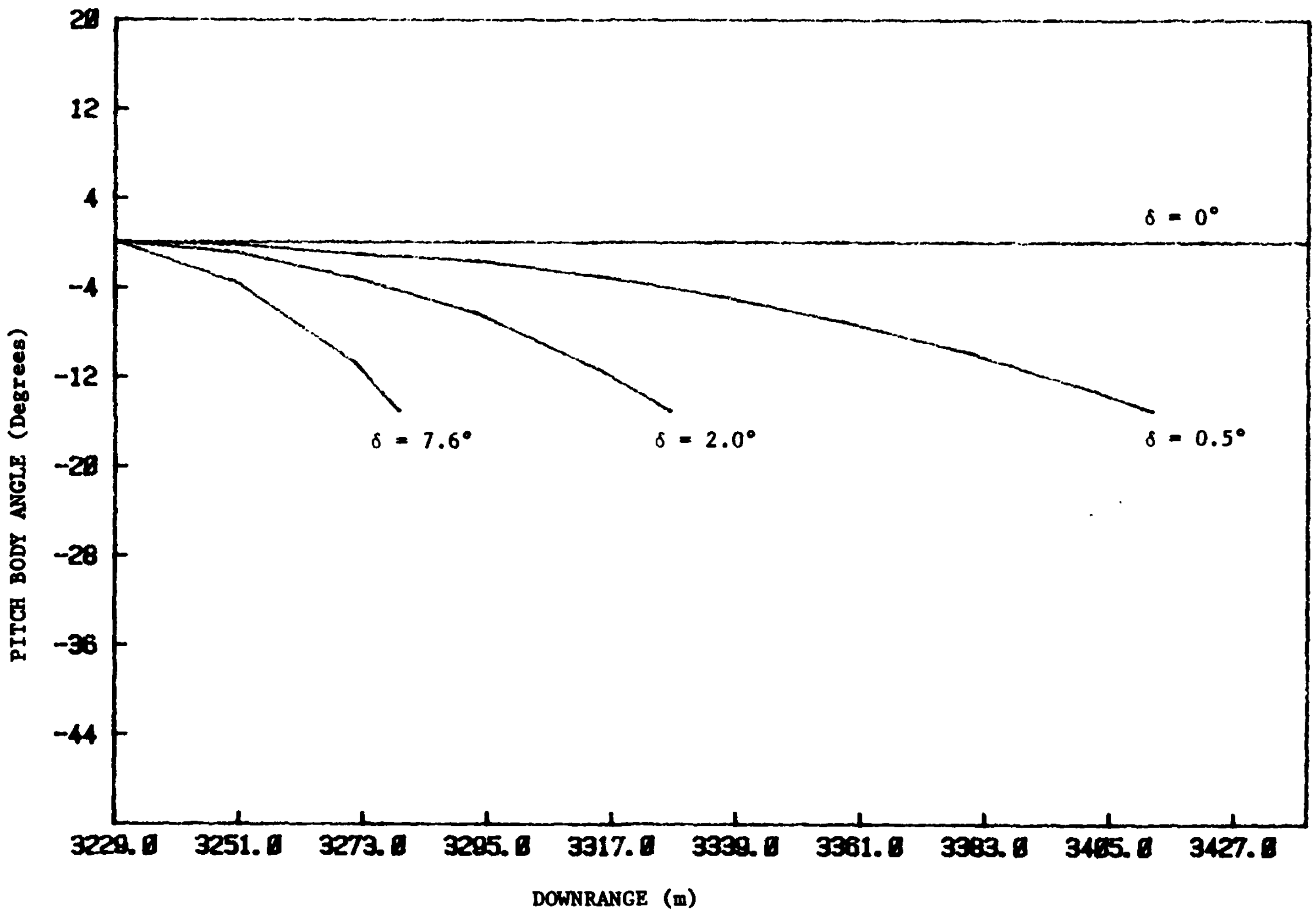

Figure 4. Pitch Body Angle vs. Downrange for Pitch Nozzle Deflections (δ's) Occurring at 49 Seconds into the Flight.

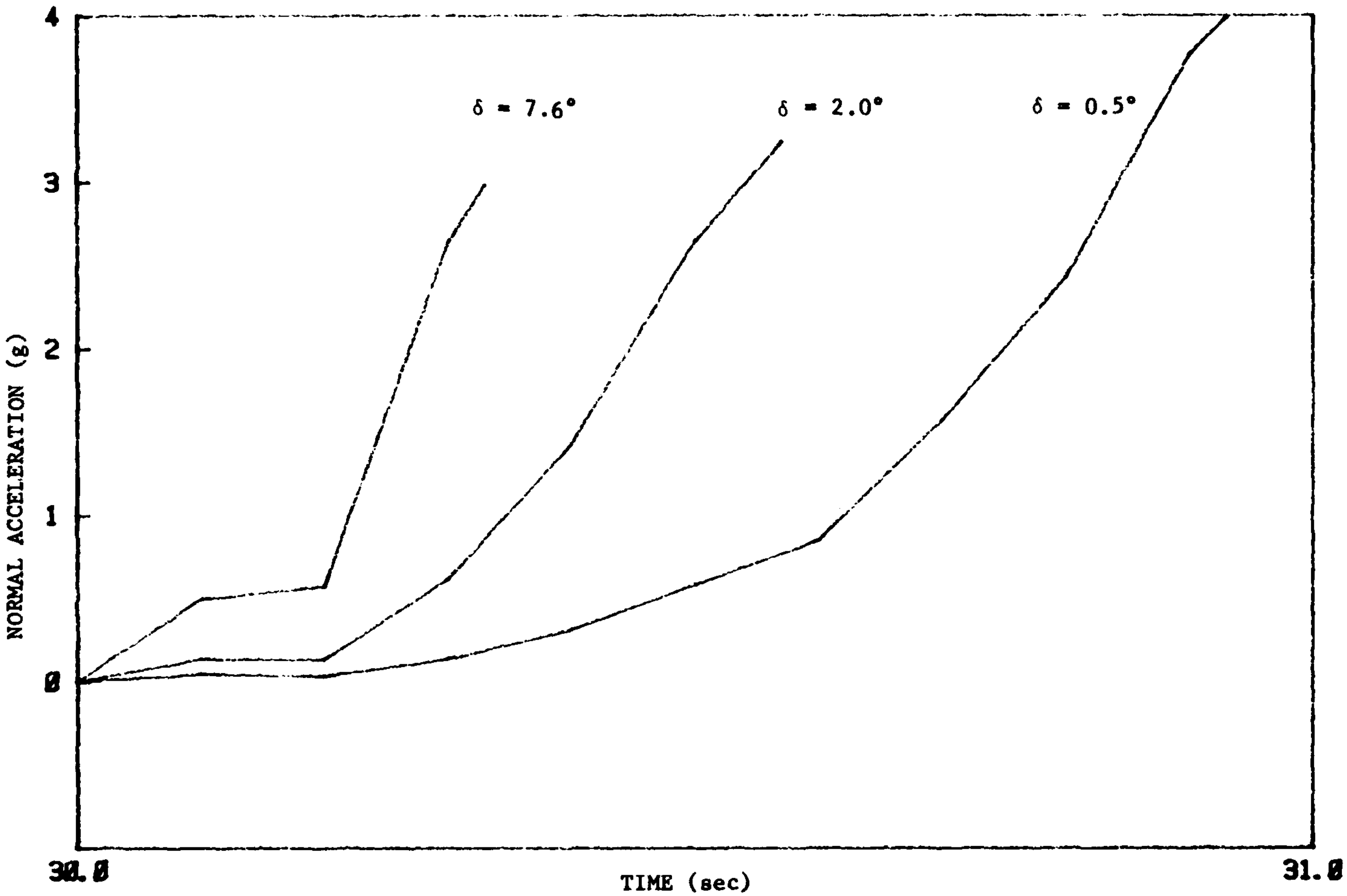

Figure 5. Normal Accelerations vs. Time for Pitch Nozzle Deflections (δ's) Occurring at 30 Seconds into the Flight.

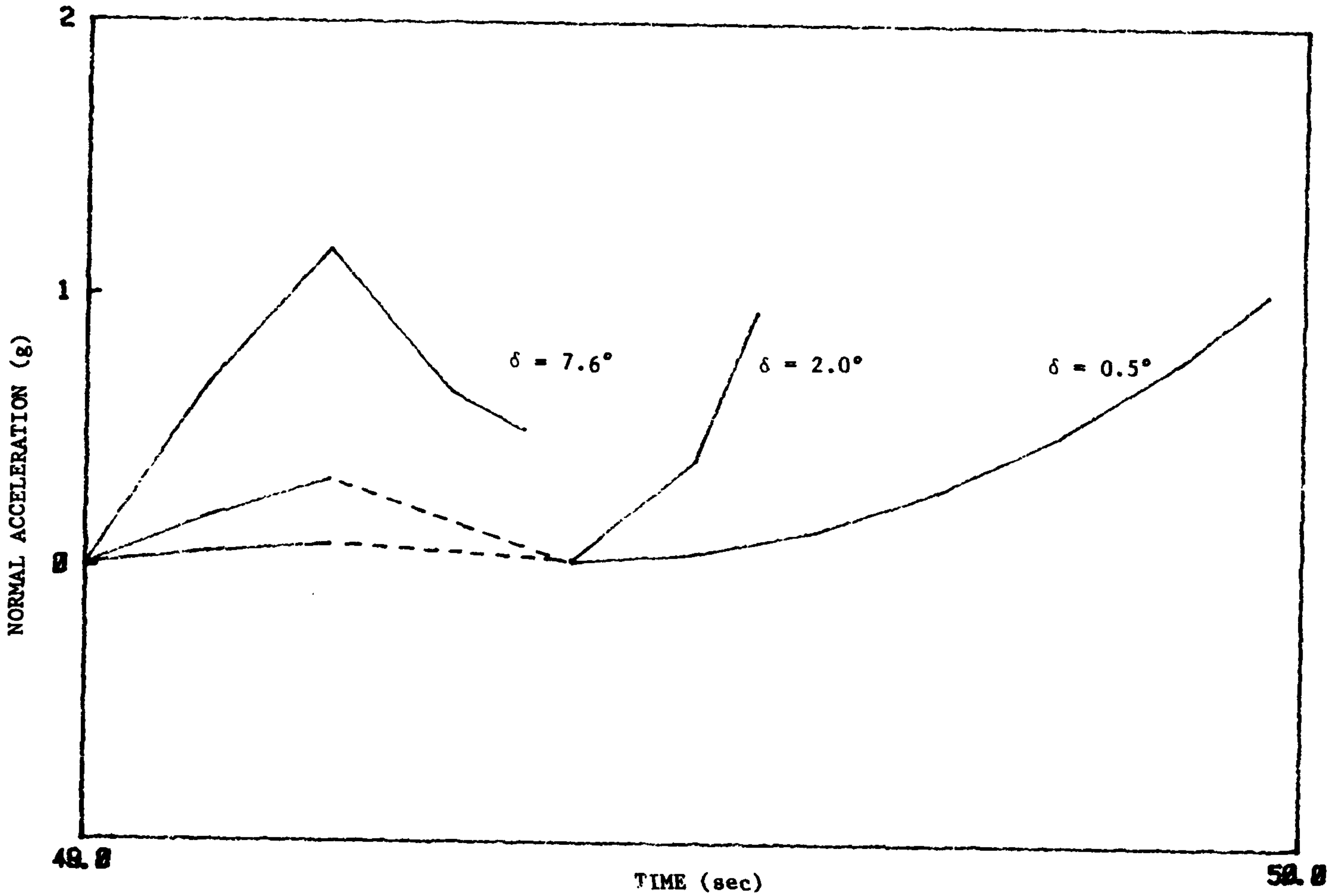

Figure 6. Normal Accelerations vs. Time for Pitch Nozzle Deflections (δ's) Occurring at 49 Seconds into the Flight.

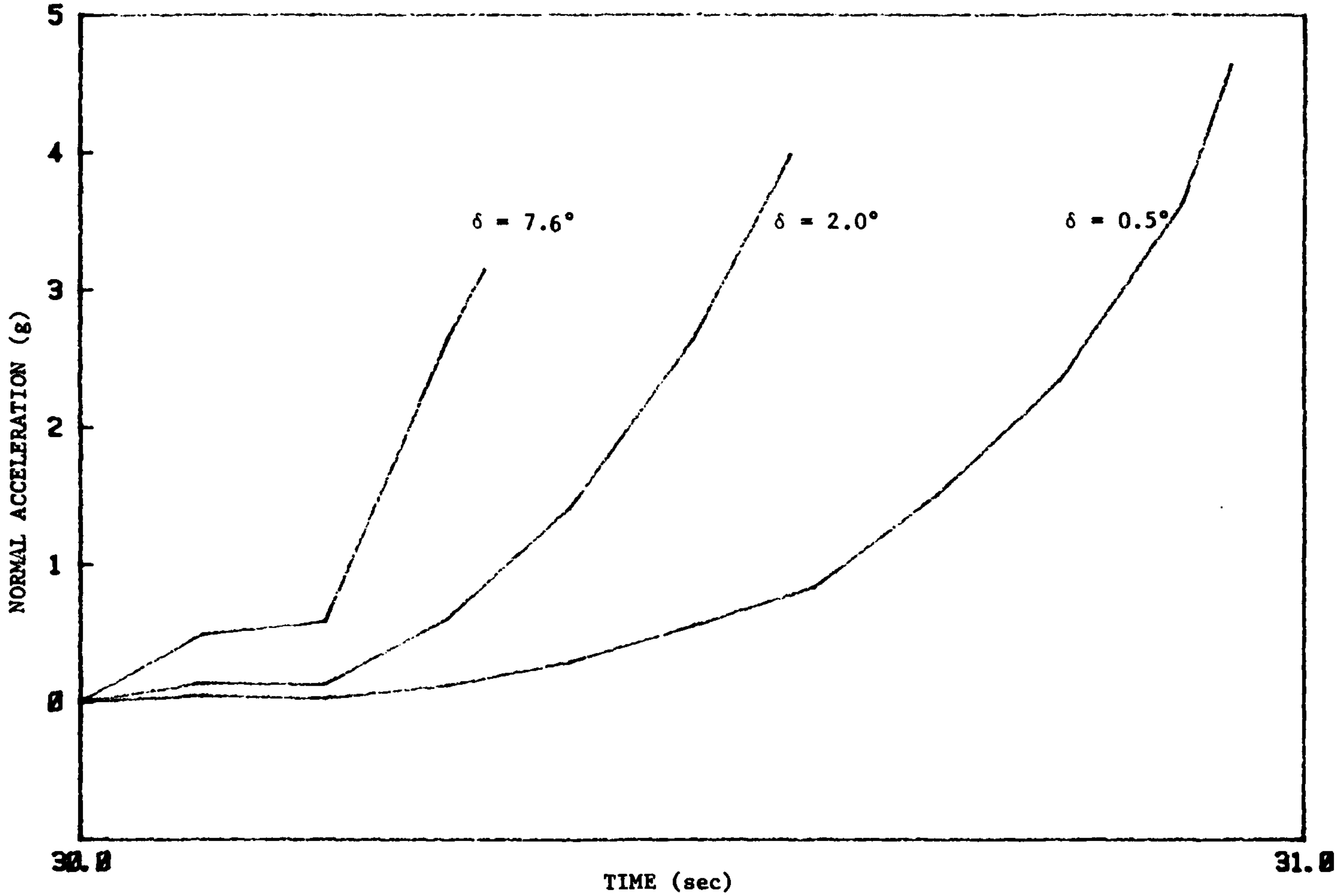

Figure 7. Normal Accelerations vs. Time for Yaw Nozzle Deflections (δ's) Occurring at 30 Seconds into the Flight.

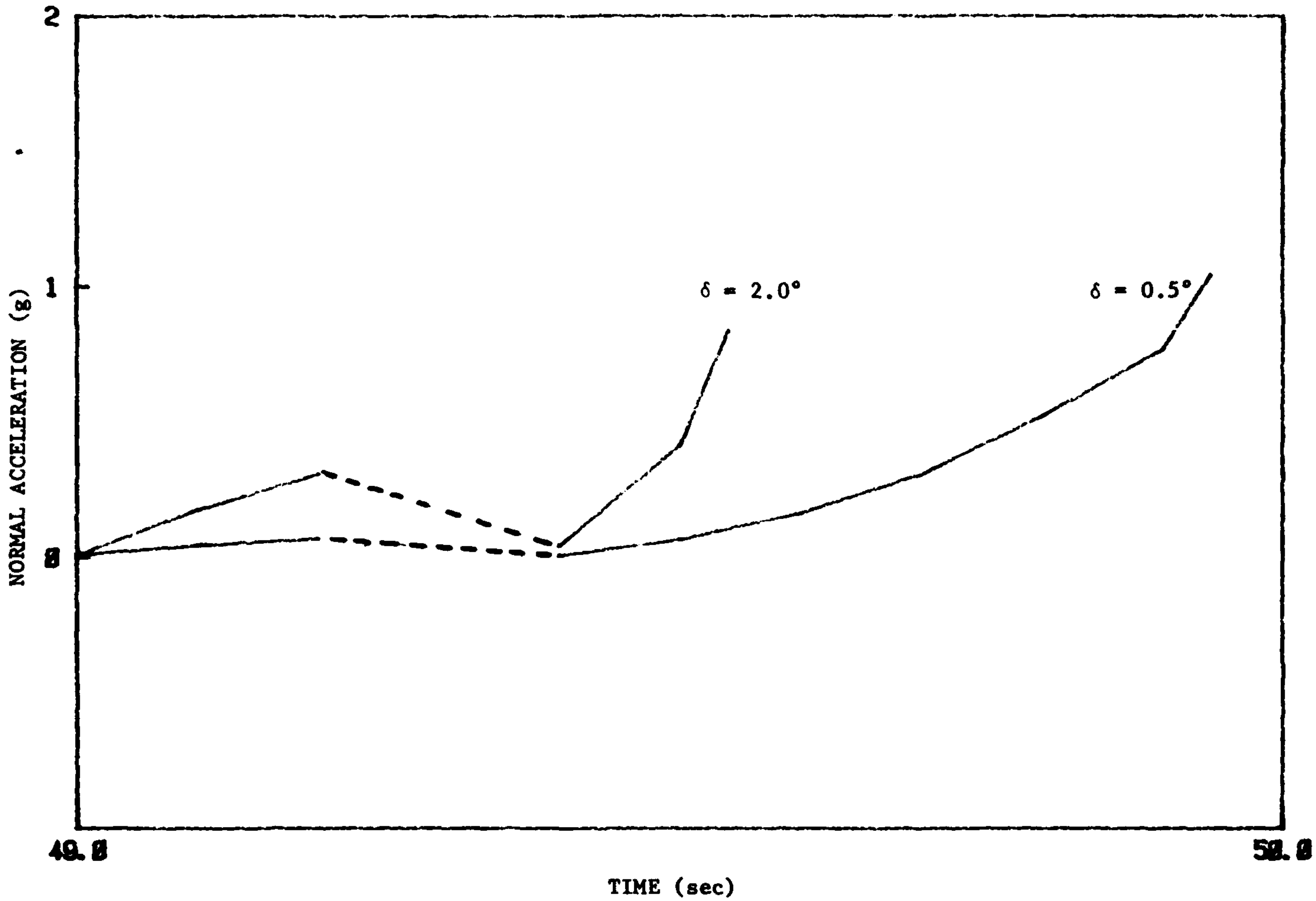

Figure 8. Normal Accelerations vs. Time for Yaw Nozzle Deflections (δ's) Occurring at 49 Seconds into the Flight.

D. Conversion and Installation of the TRW Simulation Program

Some of the details of the TRW program conversion have been discussed in Section III.B., and in Appendices D through I. A summary of the results is as follows:

1. All subroutines except those missing from the incoming program tape were converted and successfully compiled.

2. The organization of the INTERP tables was deciphered and documented (Appendix F).

3. The MAIN and SDCTRL subroutines were flowcharted (Appendices G and H).

4. A linking task builder TRWLINK.CSS and a compiler task builder DAVEF7.CSS were created to accommodate the TRW program.

V. RECOMMENDATIONS FOR FURTHER WORK

A. Flight Safety (Motor Nozzle Deflection) Recommendations

One of the recommendations for future work is to carry out these evaluations, re-running the simulation with print statements that monitor the moment-by-moment variations in the angles of attack, the down-range and cross-range coordinates, and the inertial flight path angles before, during, and after the moment when the nozzle deviation occurs.

B. Aerodynamic Model Validation Recommendations

Recommendations for future work consist of completing the DATACORR subroutine and carrying out the study using actual flight data.

Normally, in employing the Maximum Likelihood Method, all the computed constitutive forces and moments that make up the total X, Y, or Z forces or ϕ, θ, or ψ moments (constituents of lift, drag, etc.,) are compared individually, with their equivalent measured values. However, the aerodynamic model used in the U70 simulator is not yet sufficiently detailed to permit making these comparisons because of the difficulty of attributing corrections to individual components of the U70 predictions, so the comparisons are to be made at the level of overall body forces and torques. For this reason, the DATACORR subroutine doesn't presently output its force and torque corrections to the main U70 program. However, future plans call for the upgrading of the main program to provide aerodynamic modeling at the constitutive level. When that is done, the DATACORR subroutine and the TU70 task will be upgraded to output these corrections to the U70 program.

C. Tactical Ballistic Missile Trajectory Recommendations

Recommendations for further work depend upon the sponsor's requirements.

D. Conversion and Installation of the TRW Simulation Program

The Maximum Likelihood Method correlation technique being used to validate the U70 simulator predictions against actual flight data (Section II.C.) should be used to evaluate the TRW simulator.

Recommendations for future work consist of completing the conversion and installation of the TRW program, using it to simulate Pershing II test flights, and comparing its results with the actual test flight data.

REFERENCES

[1] "U70 Utilization Report," Martin-Marietta Corporation, Orlando, Florida, Revision A, January 27, 1981.

[2] "Computer Program Missile Interim Detail Specification/Performance/Design Requirements for Pershing Airborne Computer Program for Pershing II Weapon System, Engineering Development Program," Specification MIS-21748, Martin-Marietta Corporation, October 1, 1980.

APPENDIX A
CODE LISTING FOR THE DATACORR (Data Correlation) SUBROUTINE

TABLE A-1. CODE LISTING FOR THE DATACORR SUBROUTINE

```
$PROG DATACORR
      SUBROUTINE DATACORR
$INCLUDE 9,U70TIM.COM
C ***     BEGIN COMMON PACK   ***
 IMPLICIT DOUBLE PRECISION (A-H,O-Z)
 REAL C3V,CUR3V,DCB,DCI,SCD,SCI,PSAVE
 COMMON CIN(400),SCD(1100),SCI(1100),DCB(7200),DCI(640)
 COMMON TITL(12),TMAJ(30),TAB(50),TTAB(90),CUR3V(3300),DUM2(10)
 COMMON ICON(100),IDB(32),IDC(33),IDI(32),ISC(31),ISI(30),IND3V(1
 COMMON NDB(32),NDI(32),NSI(30),NSTR(30),NTV(3,10),IALIGN
 COMMON /CONST/ A13    ,A23    ,A33    ,COSBL ,COSPL ,CPSIA ,CTHA  ,
1               CXNUP ,CXNUY ,DTOR   ,D2AC2 ,F11   ,F12   ,F13   ,
2               F21   ,F22   ,F23    ,F31   ,F32   ,F33   ,RL    ,
3               RPRES ,RTOD  ,SINBL  ,SINPL ,SPSIA ,STHA  ,SXNUP ,
4               SXNUY ,S11   ,S12    ,S13   ,S21   ,S22   ,S23   ,
5               S31   ,S32   ,S33    ,TANPL ,TOLH  ,TOL1D ,TOL2D ,
6               U11   ,U21   ,U22    ,U23   ,U31   ,U33   ,PI    ,
7               COT   ,SOT   ,PIO2   ,TWOPI ,THIRD ,SQRT2 ,SQRT3 ,
8               PIO4

 COMMON ACA    ,ACCEL ,ACR   ,ACR4  ,ACR5  ,BPINT ,BYINT ,CGDE
 COMMON DGAMS  ,DGGT  ,DRHO  ,DVEL  ,GAMMC ,GAMME ,GAMRE ,IGUID
 COMMON IPCD   ,ITHM  ,IYAW  ,PHIPE ,PHIRE ,PHIYE ,PHPEC ,PHPEP
 COMMON PHYEC  ,PHYEP ,PSIC  ,PX    ,PY    ,PZ    ,RCX   ,RCY
 COMMON RCZ    ,RD    ,RDD   ,RMAG  ,SGDE  ,SGE   ,SGG   ,TFF
 COMMON TFFS   ,TGE   ,TGN   ,TGS   ,THETAC,TIMPL ,URANX ,URANY
 COMMON URANZ  ,URAX  ,URAY  ,URAZ  ,URAZX ,URAZY ,URAZZ ,URX
 COMMON URY    ,URZ   ,VE    ,VEX   ,VEXS  ,VEY   ,VEYS  ,VEZ
 COMMON VEZS   ,VIP   ,VLEX  ,VLEY  ,VLEZ  ,VR    ,VRE   ,VREH
 COMMON VREHX  ,VREHY ,VREHZ ,ZDIFF ,ZETAE ,ZETRE ,ZINTC ,ZINTP
 COMMON ZINTS

 COMMON AA1    ,AA2   ,ALDDTC,ALDDTS,ALOADF,ALOADS,ALPDDE,ALPHDD
 COMMON ALPHP  ,ALPHPD,ALPHPS,ALPHS ,ALPHT ,ALPHY ,ALPHYD,ALPHYS
 COMMON ATC    ,ATP   ,AX    ,AY    ,AZ    ,A11   ,A12   ,A21
 COMMON A22    ,A31   ,A32   ,BB2   ,BEDDTC,BEDDTS,BETADD,BETAP
 COMMON BETAR  ,BETAY ,BETAW ,BETDDE,BTC   ,BTP   ,B11   ,B12
 COMMON B13    ,B21   ,B22   ,B23   ,B31   ,B32   ,B33   ,CA
 COMMON CAAP   ,CAAY  ,CAIN  ,CLP   ,CMQ   ,CMXP  ,CN    ,CNAPD
 COMMON CNAYD  ,CNP   ,CNG   ,CNVDO ,CNVDP ,CNY   ,CNZ   ,COSPSD
 COMMON COSTHD ,CPHI  ,CPSI  ,CTHE  ,CY    ,CYP   ,CYR   ,C11
 COMMON C12    ,C13   ,DELTA1,DELTA2,DELTA3,DELTA4,DFP   ,DFR
 COMMON DFY    ,DT    ,DVTC  ,DVTP  ,DVTS  ,D11   ,D12   ,D13
 COMMON D21    ,D22   ,D23   ,D31   ,D32   ,D33   ,DELC  ,EDLC
 COMMON EDDLC  ,EPHDC ,EPHDS ,EPHIC ,EPHIP ,EPHIS ,EPSDC ,EPSDS
 COMMON EPSIC  ,EPSIP ,EPSIS ,ETA   ,ETHAC ,ETHAP ,ETHAS ,ETHDC
 COMMON ETHDS  ,FE    ,FP    ,FROLL ,FSL   ,FT    ,FX    ,FXM
 COMMON FY     ,FYAV  ,FYAW  ,FYM   ,FZ    ,FZAV  ,FZM   ,G
 COMMON GA     ,GADDTC,GADDTS,GAMDD ,GAMDDE,GR    ,GTC   ,GTP
 COMMON H      ,HDT   ,H11   ,H12   ,H13   ,H21   ,H22   ,H23
 COMMON H31    ,H32   ,H33   ,IAERO ,ICN   ,ICO   ,IDINT ,IDISC
```

TABLE A-1. CODE LISTING FOR THE DATACORR SUBROUTINE (Continued)

```
COMMON IDT    ,IDUBL ,IENV  ,IFAIL ,IFCOM ,IFINC ,IFLAG ,IGC
COMMON IHIJ   ,IHT   ,IMAJ  ,IMISS ,INDH  ,INDT  ,INT   ,IPAGE
COMMON IPLOT  ,IPNT  ,IPRNT ,IPROP ,IRJU  ,IRUN  ,ISEN  ,ISMTH
COMMON ITA    ,ITB   ,ITC   ,ITHR  ,ITOLB ,ITPOS ,ITT   ,IUNT
COMMON IUPD   ,IVAC  ,IWD   ,IWIN  ,JSTG  ,KFLAG ,KSTG  ,LBCNT
COMMON LINC   ,LINTOT,MAJ   ,MBAL  ,NTPU  ,P     ,PC    ,PDC
COMMON PDS    ,PDT   ,PHI   ,PHID  ,PHIE  ,PHILC ,PHIMC ,PP
COMMON PS     ,PSI   ,PSID  ,PSIE  ,PTIM  ,PXPC  ,PXPP  ,PXPS
COMMON PYPC   ,PYPP  ,PYPS  ,PZPC  ,PZPP  ,PZPS  ,P11   ,P12
COMMON P13    ,P21   ,P22   ,P23   ,P31   ,P32   ,P33   ,Q
COMMON QA     ,QC    ,QDC   ,QDS   ,QMIN  ,QP    ,QS    ,R
COMMON RAIM   ,RC    ,RDC   ,RDS   ,RET   ,RHO   ,RP    ,RS
COMMON SIGMAP,SIGMAY,SINPSD,SINTHD,SPHI  ,SPSI  ,STHE  ,SXGZGD
COMMON TBO    ,TBS   ,TDIFF ,TDT   ,TEMP1 ,TEMP2 ,TEMP3 ,THETA
COMMON THETAD,THETAE,THETL ,TIMC  ,TIMP  ,TIMS  ,TLIM  ,TST
COMMON TSTMAX,TSTMIN,TTT   ,UMA   ,VIXC  ,VIXP  ,VIXS  ,VIYC
COMMON VIYP   ,VIYS  ,VIZC  ,VIZP  ,VIZS  ,VMA   ,VN    ,VPRIX
COMMON VPRIY  ,VPRIZ ,VRW   ,VRXP  ,VRYP  ,VRZP  ,VS    ,VW
COMMON VWE    ,VWS   ,VX    ,VY    ,VZ    ,W     ,WACB  ,WACBS
COMMON WD     ,WDACB ,WDACBS,WDS   ,WG    ,WIN   ,WMA   ,WPB
COMMON WPBS   ,WPN   ,WPS   ,WS    ,XC    ,XDC   ,XDDC  ,XDDS
COMMON XDG    ,XDIN  ,XDP   ,XDS   ,XEL   ,XET   ,XIN   ,XIXX
COMMON XIYY   ,XIZZ  ,XKN   ,XLAM  ,XLAMM ,XLCM  ,XLCP  ,XL11
COMMON XL12   ,XL13  ,XL21  ,XL22  ,XL23  ,XL31  ,XL32  ,XL33
COMMON XM     ,XMAID ,XMASS ,XMAX  ,XMAY  ,XMAZ  ,XMAVX ,XMAVY
COMMON XMAVZ  ,XMC   ,XMFX  ,XMFY  ,XMFZ  ,XMSP  ,XMTX  ,XMTY
COMMON XMTZ   ,XMXACS,XMYACS,XMZACS,XNLOAF,XNU   ,XNV1  ,XNV2
COMMON XNV3   ,XP    ,XPC   ,XPDC  ,XPDD  ,XPDS  ,XPP   ,XPS
COMMON XS     ,YC    ,YDC   ,YDDC  ,YDDS  ,YDG   ,YDIN  ,YDP
COMMON YDS    ,YEL   ,YET   ,YMAID ,YP    ,YPC   ,YPDC  ,YPDD
COMMON YPDS   ,YPP   ,YPS   ,YS    ,ZC    ,ZDC   ,ZDDC  ,ZDDS
COMMON ZDG    ,ZDIN  ,ZDP   ,ZDS   ,ZEL   ,ZET   ,ZIN   ,ZMAID
COMMON ZP     ,ZPC   ,ZPDC  ,ZPDD  ,ZPDS  ,ZPP   ,ZPS   ,ZS
COMMON ZZ1    ,ZZ2   ,ZZ3   ,ZZ4   ,ZZ5   ,ZZ6   ,ZZ7   ,ZZ8

COMMON ACC    ,ADE11 ,ADE12 ,ADE13 ,ADE21 ,ADE22 ,ADE23 ,ADE31
COMMON ADE32  ,ADE33 ,ALDC  ,ALDCL ,ALPHPL,ALPHYL,ALPC  ,ALRC
COMMON ALYC   ,ALRFP ,ALRFY ,ALTC  ,ALTCL ,AMAX  ,APEST ,AYEST
COMMON APX    ,APY   ,APZ   ,BERR  ,BETA  ,BETASH,BETASW,CAO
COMMON CABASE,CASF  ,CNLIM ,CUN   ,CUE   ,CUD   ,DELPA ,DELRA
COMMON DELYA  ,DELPL ,DELRL ,DELYL ,DLAM  ,DLAMD ,DVN   ,DVE
COMMON DVD    ,ENSCS ,EESCS ,ERALT ,GADPC ,GADDPC,GADDPS,GDDDP
COMMON GADYC  ,GADDYC,GADDYS,GDDDY ,GAMMAD,GAMMAT,GDDC  ,GDDCL
COMMON GDPC   ,GDRC  ,GDYC  ,GDPX  ,GDPY  ,GDPZ  ,GDTC  ,GDTCL
COMMON GLIM   ,HABS  ,HDOT  ,HNAV  ,HPOS  ,IASW  ,IATM  ,IBSW
COMMON IDUM   ,IFIL  ,IGAIN ,ILAG  ,ILAST ,IRREF ,IRUP  ,ISCS
COMMON ISEQ   ,ISHD  ,ISTER ,JBSW  ,NREP  ,PHIAL ,PHICMD,PHIGD
COMMON PHITAB,PN    ,PE    ,PD    ,POS   ,PUN   ,PUE   ,PUD
COMMON RKE    ,RKGD  ,RKGT  ,RUX   ,RUY   ,RUZ   ,TEJ   ,TG
COMMON TLAM   ,TLAMD ,TPD   ,TRTN  ,UPC   ,URC   ,UYC   ,VHS
COMMON VNAV   ,VXS   ,VYS   ,VZS   ,WAP   ,XFAIL ,YFAIL ,ZFAIL
```

TABLE A-1. CODE LISTING FOR THE DATACORR SUBROUTINE (Continued)

```
 COMMON XMT     ,XMTD   ,YMT     ,YMTD   ,ZMT     ,ZMTD

     END COMMON PACK   ***
 COMMON/OTHERS/AC      ,ACM     ,ACPX    ,ACPY    ,ACPZ    ,ACX     ,ACY     ,
1ACZ     ,API(2,14)    ,APC     ,APFB    ,AYC     ,AYFB    ,CGE     ,CLDR    ,
2COSTC  ,DGAMC  ,DNV(2,6)       ,EPP     ,EYP     ,FEEP    ,GAMMCS,GAMMG ,
3IACFLG,IAPFLG,IBO      ,IDB3    ,IDB4    ,IDB5    ,IDB6    ,IDI3    ,IDI4    ,
4IDI5   ,IDI6   ,ICOF    ,IGAFLG,IPOFL  ,PMX     ,PMY     ,PMZ     ,QAP     ,
5SINGC  ,SINPC  ,SINTC  ,TGSM    ,TGSMS  ,TTIMP  ,TVACF  ,TVACP  ,T11     ,
6T12     ,T13    ,T21     ,T22     ,T23     ,T31     ,T32     ,T33     ,VG      ,
7VIS     ,VTS    ,XMAP    ,XMSAP  ,YIN     ,HANK(20)         ,ACSX
 DIMENSICN DMG(64),PMG(64)
 DIMENSION DCG(54),PCG(54)
 DIMENSICN AS(6,6),SAS(6,6),AA(6,6),AAT(6,6),AS1(6,6)
 DIMENSICN AL(6,6),AM(6,6),AM1(6,6),AE(6,1),AN(5,1)
 DIMENSICN DC(6,1),AP(6,1),AR(6,1),AAR(6,1)
 DIMENSICN X(6,12),WORK(12),IHLD(6)
 DATA DMG/5H   THE,5H   PHE,5H   PSE,5H    XE,5H    YE,5H    ZE,
1         5H   THC,5H   PHC,5H   PSC,5H   XCC,5H   YCC,5H   ZCC,
2         5H  THDC,5H  PHDC,5H  PSDC,5H  XDCC,5H  YDCC,5H  ZDCC,
3         5HTHDDC,5HPHDDC,5HPSDDC,5HXDDCC,5HYDDCC,5HZDDCC,
5         5H  AA11,5H  AA12,5H  AA13,5H  AA14,5H  AA15,5H  AA16,
6         5H  AA21,5H  AA22,5H  AA23,5H  AA24,5H  AA25,5H  AA26,
7         5H  AA31,5H  AA32,5H  AA33,5H  AA34,5H  AA35,5H  AA36,
8         5H  AA41,5H  AA42,5H  AA43,5H  AA44,5H  AA45,5H  AA46,
9         5H  AA51,5H  AA52,5H  AA53,5H  AA54,5H  AA55,5H  AA56,
A         5H  AA61,5H  AA62,5H  AA63,5H  AA64,5H  AA65,5H  AA66,
B         5H   IXX,5H   IYY,5H   IZZ,5HIMASS/
 DATA DCG/5H  AS11,5H  AS12,5H  AS13,5H  AS14,5H  AS15,5H  AS16,
1         5H  AS21,5H  AS22,5H  AS23,5H  AS24,5H  AS25,5H  AS26,
2         5H  AS31,5H  AS32,5H  AS33,5H  AS34,5H  AS35,5H  AS36,
3         5H  AS41,5H  AS42,5H  AS43,5H  AS44,5H  AS45,5H  AS46,
4         5H  AS51,5H  AS52,5H  AS53,5H  AS54,5H  AS55,5H  AS56,
5         5H  AS61,5H  AS62,5H  AS63,5H  AS64,5H  AS65,5H  AS66,
6         5H  DC11,5H  DC21,5H  DC31,5H  DC41,5H  DC51,5H  DC61,
7         5H  AR11,5H  AR21,5H  AR31,5H  AR41,5H  AR51,5H  AR61,
8         5H  AE11,5H  AE21,5H  AE31,5H  AE41,5H  AE51,5H  AE61/
 EQUIVALENCE (HANK(1),XPDDTC),(HANK(2),YPDDTC),(HANK(3),ZPDDTC)
      IF(IPRNT.EQ.1) RETURN
C
C     INITIALIZATION DONE ONLY ONCE-INTERVAL
C
      IF(JJJ.EQ.1) GO TO 5
      KKK=0
    5 CONTINUE
      KKK=KKK+1
C
C     NO. EXPERIMENTAL DATA POINTS-INPUT
C
      N=CIN(399)
```

TABLE A-1. CODE LISTING FOR THE DATACORR SUBROUTINE (Continued)

```
      WRITE(6,111) KKK,N
  111 FORMAT(/55X,8HINTERVAL,3X,I3,1H-,I3,/)
      WRITE(6,1112) TIME
 1112 FORMAT(56X,'TIME',G15.7)
C
C     COVARIANCE MATRIX,INITIAL-INPUT
C
      IF(KKK.EQ.1) GO TO 15
      DO 15 I=1,6
      DO 15 J=1,6
      AS(I,J)=SAS(I,J)
   15 CONTINUE
      IF(KKK.GT.1) GO TO 10
      DO 40 I=1,6
      DO 40 J=1,6
      AS(I,J)=0.
   40 CONTINUE
      AS(1,1)=CIN(398)
      AS(2,2)=CIN(397)
      AS(3,3)=CIN(396)
      AS(4,4)=CIN(395)
      AS(5,5)=CIN(394)
      AS(6,6)=CIN(393)
   10 CONTINUE
      JJJ=1
C
C     U70 VALUES TAKEN AS COMPUTED
C     AND EXPERIMENTAL DATA-INPUT
C
      THE=THETA
      PHE=PHI
      PSE=PSI
      THE=THE+CIN(392)
      PHE=PHE+CIN(391)
      PSE=PSE+CIN(390)
      THC=THETA
      PHC=PHI
      PSC=PSI
      THDC=THETAD
      PHDC=PHID
      PSDC=PSID
      THDDC=QDC
      PHDDC=PDC
      PSDDC=RDC
      IMASS=W
      GST=CIN( 21)
      IXX=XIXX*GST
      IYY=XIYY*GST
      IZZ=XIYY*GST
      XE=ATC+CIN(389)
      YE=BTC+CIN(388)
```

TABLE A-1. CODE LISTING FOR THE DATACORR SUBROUTINE (Continued)

```
      ZE=GTC+CIN(387)
      XCC=ATC
      YCC=BTC
      ZCC=GTC
      XDCC=A11*VX+A12*VY+A13*VZ
      YDCC=A21*VX+A22*VY+A23*VZ
      ZDCC=A31*VX+A32*VY+A33*VZ
      XDDCC=XPDDTC
      YDDCC=YPDDTC
      ZDDCC=ZPDDTC
C
C     ERORR MATRIX
C
      AE(1,1)=THE-THC
      AE(2,1)=PHE-PHC
      AE(3,1)=PSE-PSC
      AE(4,1)=XE-XCC
      AE(5,1)=YE-YCC
      AE(6,1)=ZE-ZCC
C
C     INERTIAL TO MISSILE MATRIX-AB
C
      CTHC=COS(THC)
      STHC=SIN(THC)
      CPHC=COS(PHC)
      SPHC=SIN(PHC)
      CPSC=COS(PSC)
      SPSC=SIN(PSC)
      AB11=CPSC*CTHC-SPSC*SPHC*STHC
      AB12=CTHC*SPSC+STHC*SPHC*CPSC
      AB13=-STHC*CPHC
      AB21=-SPSC*CPHC
      AB22=CPHC*CPSC
      AB23=SPHC
      AB31=CPSC*STHC+CTHC*SPHC*SPSC
      AB32=STHC*SPSC-CTHC*SPHC*CPSC
      AB33=CTHC*CPHC
C
C     SENSITIVITY MATRIX
C
      AA(1,1)=
      AA(1,2)=
      AA(1,3)=
      AA(1,4)=
      AA(1,5)=
      AA(1,6)=
      AA(2,1)=
      AA(2,2)=
      AA(2,3)=
      AA(2,4)=
      AA(2,5)=
```

TABLE A-1. CODE LISTING FOR THE DATACORR SUBROUTINE (Continued)

```
      AA(2,6)=
      AA(3,1)=
      AA(3,2)=
      AA(3,3)=
      AA(3,4)=
      AA(3,5)=
      AA(3,6)=
      AA(4,1)=
      AA(4,2)=
      AA(4,3)=
      AA(4,4)=
      AA(4,5)=
      AA(4,6)=
      AA(5,1)=
      AA(5,2)=
      AA(5,3)=
      AA(5,4)=
      AA(5,5)=
      AA(5,6)=
      AA(6,1)=
      AA(6,2)=
      AA(6,3)=
      AA(6,4)=
      AA(6,5)=
      AA(6,6)=
C
C     MAIN GROUP PRINT ORDER
C
      PMG( 1)=THE
      PMG( 2)=PHE
      PMG( 3)=PSE
      PMG( 4)=XE
      PMG( 5)=YE
      PMG( 6)=ZE
      PMG( 7)=THC
      PMG( 8)=PHC
      PMG( 9)=PSC
      PMG(10)=XCC
      PMG(11)=YCC
      PMG(12)=ZCC
      PMG(13)=THDC
      PMG(14)=PHDC
      PMG(15)=PSDC
      PMG(16)=XDCC
      PMG(17)=YDCC
      PMG(18)=ZDCC
      PMG(19)=THDDC
      PMG(20)=PHDDC
      PMG(21)=PSDDC
      PMG(22)=XDDCC
      PMG(23)=YDDCC
```

TABLE A-1. CODE LISTING FOR THE DATACORR SUBROUTINE (Continued)

```
      PMG(24)=ZDDCC
      PMG(25)=AA(1,1)
      PMG(26)=AA(1,2)
      PMG(27)=AA(1,3)
      PMG(28)=AA(1,4)
      PMG(29)=AA(1,5)
      PMG(30)=AA(1,6)
      PMG(31)=AA(2,1)
      PMG(32)=AA(2,2)
      PMG(33)=AA(2,3)
      PMG(34)=AA(2,4)
      PMG(35)=AA(2,5)
      PMG(36)=AA(2,6)
      PMG(37)=AA(3,1)
      PMG(38)=AA(3,2)
      PMG(39)=AA(3,3)
      PMG(40)=AA(3,4)
      PMG(41)=AA(3,5)
      PMG(42)=AA(3,6)
      PMG(43)=AA(4,1)
      PMG(44)=AA(4,2)
      PMG(45)=AA(4,3)
      PMG(46)=AA(4,4)
      PMG(47)=AA(4,5)
      PMG(48)=AA(4,6)
      PMG(49)=AA(5,1)
      PMG(50)=AA(5,2)
      PMG(51)=AA(5,3)
      PMG(52)=AA(5,4)
      PMG(53)=AA(5,5)
      PMG(54)=AA(5,6)
      PMG(55)=AA(6,1)
      PMG(56)=AA(6,2)
      PMG(57)=AA(6,3)
      PMG(58)=AA(6,4)
      PMG(59)=AA(6,5)
      PMG(60)=AA(6,6)
      PMG(61)=IXX
      PMG(62)=IYY
      PMG(63)=IZZ
      PMG(64)=IMASS
C
  225 FORMAT((1H ,6(1X,A5,G15.7)))
      WRITE(6,225) (DMG(I),PMG(I),I=1,64)
C
C     COMPUTE AA TRANSPOSE=AAT
C
      IF(ICON(70).EQ.0) GO TO 30
      N=6
      NM1=N-1
      DO 30 I=1,NM1
```

```
      IP1=I+1
      DO 20 J=IP1,N
      TMP1=AA(I,J)
      AA(I,J)=AAT(J,I)
   20 AAT(J,I)=TMP1
      WRITE(6,1002) AAT(3,1)
 1002 FORMAT(1X,G15.7)
   30 CONTINUE
      IF(KKK.EQ.1) GO TO 35
C
C     COMPUTE INVERSE AS =AS1
C
      DO 50 I=1,6
      DO 50 J=1,6
      X(I,J) = AS(I,J)
   50 CONTINUE
      MN1=6
      N=6
      NB=N
      MS=1
      IC=1
      ID=0
      IS=1
      CALL SESOMI(X,N,NB,MS,MN1,D,R,E,WORK,IHLD,IC,ID,IS)
      DO 60 J=1,6
      DO 60 K=1,6
      AS1(J,K)=X(J,K)
   60 CONTINUE
   35 CONTINUE
      IF(KKK.GT.1) GO TO 55
      DO 55 I=1,6
      DO 55 J=1,6
      AS1(I,J)=0.
      AS1(1,1)=1./CIN(398)
      AS1(2,2)=1./CIN(397)
      AS1(3,3)=1./CIN(396)
      AS1(4,4)=1./CIN(395)
      AS1(5,5)=1./CIN(394)
      AS1(6,6)=1./CIN(393)
   55 CONTINUE
      WRITE(14,400)
  400 FORMAT(1X,'55')
C
C     SENSITIVITY-T*COVARIANC-1 MATRIX=AL
C
      M=6
      N=6
      P=6
      DO 70 I=1,M
      DO 70 J=1,P
      AL(I,J)=0.
```

TABLE A-1. CODE LISTING FOR THE DATACORR SUBROUTINE (Continued)

```
      DO 70 K=1,N
   70 AL(I,J)=AL(I,J) + AA(K,I) * AS1(K,J)
      WRITE(14,401)
  401 FORMAT(1X,'70')
C
C     AL*SENSITIVITY MATRIX=AM
C
      M=6
      N=6
      P=6
      DO 80 I=1,M
      DO 80 J=1,P
      AM(I,J)=0.
      DO 80 K=1,N
   80 AM(I,J)=AM(I,J) + AL(I,K) * AA(K,J)
      WRITE(14,402)
  402 FORMAT(1X,'80')
C
C     AL*AE MATRIX=AN
      N=6
      P=1
      DO 90 I=1,M
      DO 90 J=1,P
      AN(I,J)=0.
      DO 90 K=1,N
   90 AN(I,J)=AN(I,J) + AL(I,K) * AE(K,J)
      WRITE(14,403)
  403 FORMAT(1X,'90')
C
C     COMPUTE INVERSE AM =AM1
C
      DO 100 I=1,6
      DO 100 J=1,6
      X(I,J)=AM(I,J)
  100 CONTINUE
      WRITE(14,404)
  404 FORMAT(1X,'100')
      MN1=6
      N=6
      NB=N
      MS=1
      IC=1
      ID=0
      IS=1
      CALL SESDMI(X,N,NB,MS,MN1,D,R,E,WORK,IHLD,IC,ID,IS)
      DO 110 J=1,6
      DO 110 K=1,6
  110 AM1(J,K) = X(J,K)
      WRITE(14,405)
  405 FORMAT(1X,'110')
C
```

TABLE A-1. CODE LISTING FOR THE DATACORR SUBROUTINE (Continued)

```
C       AM1*AN MATRIX=DC
C
        M=6
        N=6
        P=1
        DO 120 I=1,M
        DO 120 J=1,P
        DC(I,J)=0.
        DO 120 K=1,N
  120   DC(I,J)=DC(I,J) + AM1(I,K) * AN(K,J)
C
C       SENSITIVITY*DC=AP
C
        M=6
        N=M
        P=1
        DO 130 I=1,M
        DO 130 J=1,P
        AP(I,J)=0.
        DO 130 K=1,N
  130   AP(I,J)=AP(I,J) + AA(I,K) * DC(K,J)
C
C       AE-AP MATRIX=G
C
        NR=6
        NC=1
        DO 140 J=1,NC
        DO 140 I=1,NR
  140   AAR(I,J)= AE(I,J) - AP(I,J)
        N=CIN(399)
C
C       G*G -T=AS
C
        AS(1,1)=AAR(1,1)*AAR(1,1)/N
        AS(1,2)=AAR(1,1)*AAR(2,1)/N
        AS(1,3)=AAR(1,1)*AAR(3,1)/N
        AS(1,4)=AAR(1,1)*AAR(4,1)/N
        AS(1,5)=AAR(1,1)*AAR(5,1)/N
        AS(1,6)=AAR(1,1)*AAR(6,1)/N
        AS(2,1)=AAR(2,1)*AAR(1,1)/N
        AS(2,2)=AAR(2,1)*AAR(2,1)/N
        AS(2,3)=AAR(2,1)*AAR(3,1)/N
        AS(2,4)=AAR(2,1)*AAR(4,1)/N
        AS(2,5)=AAR(2,1)*AAR(5,1)/N
        AS(2,6)=AAR(2,1)*AAR(6,1)/N
        AS(3,1)=AAR(3,1)*AAR(1,1)/N
        AS(3,2)=AAR(3,1)*AAR(2,1)/N
        AS(3,3)=AAR(3,1)*AAR(3,1)/N
        AS(3,4)=AAR(3,1)*AAR(4,1)/N
        AS(3,5)=AAR(3,1)*AAR(5,1)/N
        AS(3,6)=AAR(3,1)*AAR(6,1)/N
```

TABLE A-1. CODE LISTING FOR THE DATACORR SUBROUTINE (Continued)

```
      AS(4,1)=AAR(4,1)*AAR(1,1)/N
      AS(4,2)=AAR(4,1)*AAR(2,1)/N
      AS(4,3)=AAR(4,1)*AAR(3,1)/N
      AS(4,4)=AAR(4,1)*AAR(4,1)/N
      AS(4,5)=AAR(4,1)*AAR(5,1)/N
      AS(4,6)=AAR(4,1)*AAR(6,1)/N
      AS(5,1)=AAR(5,1)*AAR(1,1)/N
      AS(5,2)=AAR(5,1)*AAR(2,1)/N
      AS(5,3)=AAR(5,1)*AAR(3,1)/N
      AS(5,4)=AAR(5,1)*AAR(4,1)/N
      AS(5,5)=AAR(5,1)*AAR(5,1)/N
      AS(5,6)=AAR(5,1)*AAR(6,1)/N
      AS(6,1)=AAR(6,1)*AAR(1,1)/N
      AS(6,2)=AAR(6,1)*AAR(2,1)/N
      AS(6,3)=AAR(6,1)*AAR(3,1)/N
      AS(6,4)=AAR(6,1)*AAR(4,1)/N
      AS(6,5)=AAR(6,1)*AAR(5,1)/N
      AS(6,6)=AAR(6,1)*AAR(6,1)/N
  500 FORMAT(57X,17HCONVERGENCE GROUP)
C
C     RESIDUAL MATRIX-ONE ITERATION AHEAD ERROR MATRIX
C
      AR(1,1)=AAR(1,1)
      AR(2,1)=AAR(2,1)
      AR(3,1)=AAR(3,1)
      AR(4,1)=AAR(4,1)
      AR(5,1)=AAR(5,1)
      AR(6,1)=AAR(6,1)
C
C     CONVERGENCE GROUPS PRINT ORDER
C
      PCG( 1)=AS(1,1)
      PCG( 2)=AS(1,2)
      PCG( 3)=AS(1,3)
      PCG( 4)=AS(1,4)
      PCG( 5)=AS(1,5)
      PCG( 6)=AS(1,6)
      PCG( 7)=AS(2,1)
      PCG( 8)=AS(2,2)
      PCG( 9)=AS(2,3)
      PCG(10)=AS(2,4)
      PCG(11)=AS(2,5)
      PCG(12)=AS(2,6)
      PCG(13)=AS(3,1)
      PCG(14)=AS(3,2)
      PCG(15)=AS(3,3)
      PCG(16)=AS(3,4)
      PCG(17)=AS(3,5)
      PCG(18)=AS(3,6)
      PCG(19)=AS(4,1)
      PCG(20)=AS(4,2)
```

```
      PCG(21)=AS(4,3)
      PCG(22)=AS(4,4)
      PCG(23)=AS(4,5)
      PCG(24)=AS(4,6)
      PCG(25)=AS(5,1)
      PCG(26)=AS(5,2)
      PCG(27)=AS(5,3)
      PCG(28)=AS(5,4)
      PCG(29)=AS(5,5)
      PCG(30)=AS(5,6)
      PCG(31)=AS(6,1)
      PCG(32)=AS(6,2)
      PCG(33)=AS(6,3)
      PCG(34)=AS(6,4)
      PCG(35)=AS(6,5)
      PCG(36)=AS(6,6)
      PCG(37)=DC(1,1)
      PCG(38)=DC(2,1)
      PCG(39)=DC(3,1)
      PCG(40)=DC(4,1)
      PCG(41)=DC(5,1)
      PCG(42)=DC(6,1)
      PCG(43)=AR(1,1)
      PCG(44)=AR(2,1)
      PCG(45)=AR(3,1)
      PCG(46)=AR(4,1)
      PCG(47)=AR(5,1)
      PCG(48)=AR(6,1)
      PCG(49)=AE(1,1)
      PCG(50)=AE(2,1)
      PCG(51)=AE(3,1)
      PCG(52)=AE(4,1)
      PCG(53)=AE(5,1)
      PCG(54)=AE(6,1)
C
C     WRITE COVARIANC,DC,RESIDUAL,ERROR
C
      WRITE(6,500)
      WRITE(6,225) (DCG(I),PCG(I),I=1,54)
      DO 150 I=1,6
      DO 150 J=1,6
  150 SAS(I,J) = AS(I,J)
      RETURN
      END
```

APPENDIX B
MODIFICATIONS TO BOUT (Boost Output)
AND RVOUT (Re-entry Vehicle Output)

APPENDIX B

MODIFICATIONS TO BOUT (Boost Output) AND RVOUT (Re-entry Vehicle Output)

The changes to the BOUT and RVOUT subroutines of the U70 simulator needed for the Tactical Ballistic Missile Trajectory Study are listed in Table B-1.

TABLE B-1. MODIFICATIONS TO BOUT AND RVOUT

PRM(70) changed to PRM (72)
RPG (162) added

```
      DIMENSION PR(30),PRE(18), PRM(72),PRG(78),RPG(162)
```

Both DO loops changed from '1,70' to '1,72'

```
      725 DO 730 IMX = 1,72
      290 CONTINUE
          IF(IMISS.EQ.0)GO TO 310
```

Added after PRM(70) = WACB

```
      PRM(71) = 0
      PRM(72) = 0

      551 CONTINUE
          PRG(22)= PRG(22)*RTOD
          PRG(23)= PRG(23)*RTOD
          PRG(28)= PRG(28)*RTOD
          PRG(29)= PRG(29)*RTOD
          PRG(34)= PRG(34)*RTOD
          PRG(35)= PRG(35)*RTOD
          PRG(41)= PRG(41)*RTOD
          PRG(64)= PRG(64)*RTOD
          PRG(65)= PRG(65)*RTOD
          PRG(75)= PRG(75)*RTOD
```

RPT(I) = PRG(I) added after PRG(75) = PRG(75)*RTOD

```
          RPG(4)=PRG(22)
          RPG(52)=PRG(23)
          RPG(10)=PRG(28)
          RPG(58)=PRG(29)
          RPG(16)=PRG(34)
          RPG(64)=PRG(35)
          RPG(137)=PRG(41)
          RPG(130)=PRG(64)
          RPG(151)=PRG(65)
          RPG(129)=PRG(75)
```

TABLE B-1. MODIFICATIONS TO BOUT AND RVOUT (Continued)

WRITE (14,26) statements added before the WRITE (6,2) statements

```
      IF(ICON(52).EQ.0) GO TO 515
      WRITE(14,25) PTIM
  515 CONTINUE
   25 FORMAT(615.7)
C     WRITE(6,2) PTIM,TSTT,DT,IRJU,IFAIL

      IF(ICON(52).EQ.0) GO TO 516
      WRITE(14,25) TIMC
  516 CONTINUE
C     WRITE(6,2) TIMC,TST,DT,IRJU,IFAIL

      IF(ICON(52).EQ.0) GO TO 533
      WRITE(14,26) (PR(I),I=1,30)
  533 CONTINUE
      WRITE(6,5) (DR(I),PR(I),I=1,30)

      IF(ICON(52).EQ.0) GO TO 534
      WRITE(14,26) (PRE(I),I=1,18)
  515 CONTINUE
   26 FORMAT(6(E15.7))
      WRITE(6,2) DRE(I),PRE(I),I=1,18)

      IF(ICON(52).EQ.0) GO TO 546
      WRITE(14,26) (PRM(I),I=72)
  546 CONTINUE
      WRITE(6,5) (DRM(I),PRM(I),I=1,70)

      IF(ICON(52).EQ.0) GO TO 553
      WRITE(14,26) (RPG(I),I=1,162)
  515 CONTINUE
      WRITE(6,5) (DRG(I),PRG(I),I=1,NN)
```

TABLE B-1. MODIFICATIONS TO BOUT AND RVOUT (Continued)

```
PRG(78) = TFF
IF(ICON(52).EQ.0) GO TO 220
RRG( 1) = PRG(1)
RRG( 2) = PRG(2)
RRG( 3) = PRG(3)
RRG( 4) = PRG(22)
RRG( 5) = PRG(5)
RRG( 6) = PRG(6)
RRG( 7) = PRG(7)
RRG( 8) = PRG(8)
RRG( 9) = PRG(9)
RRG(10) = PRG(28)
RRG(11) = PRG(11)
RRG(12) = PRG(12)
RRG(13) = PRG(13)
RRG(14) = PRG(14)
RRG(15) = PRG(15)
RRG(16) = PRG(34)
RRG(17) = PRG(17)
RRG(18) = PRG(18)
RRG(19) = 0
RRG(20) = 0
RRG(21) = 0
RRG(22) = 0
RRG(23) = 0
RRG(24) = 0
RRG(25) = 0
RRG(26) = 0
RRG(27) = 0
RRG(28) = 0
RRG(29) = PRG(40)
RRG(30) = 0
RRG(31) = 0
RRG(32) = 0
RRG(33) = 0
RRG(34) = 0
RRG(35) = 0
RRG(36) = 0
RRG(37) = 0
RRG(38) = 0
RRG(39) = 0
RRG(40) = 0
RRG(41) = 0
RRG(42) = 0
RRG(43) = 0
RRG(44) = 0
RRG(45) = 0
INDGC = IAPFLG + 10*IGAFLG + 100*IACFLG + 1000*ISEQ
RINDGC = INDGC + 99.05 + 0.000005
RRG(46) = PRG(76)
```

TABLE B-1. MODIFICATIONS TO BOUT AND RVOUT (Continued)

```
      RRG(47) = 0
      RRG(48) = 0
      RRG(49) = 0
      RRG(50) = 0
      RRG(51) = 0
      RRG(52) = PRG(23)
      RRG(53) = 0
      RRG(54) = 0
      RRG(55) = 0
      RRG(56) = 0
      RRG(57) = 0
      RRG(58) = PRG(29)
      RRG(59) = PRG(53)
      RRG(60) = 0
      RRG(61) = 0
      RRG(62) = 0
      RRG(63) = 0
      RRG(64) = PRG(35)
      RRG(65) = PRG(54)
      IFTEMP = IALTU+10*(IPUDFL+1)+100*(IUPDFL+1)+1000*JBSW
     1 + 10000*(IRCHFG+1)
      RRG(66) = 0
      RRG(67) = PRG(19)
      RRG(68) = PRG(21)
      RRG(69) = 0
      RRG(70) = 0
      RRG(71) = 0
      RRG(72) = 0
      RRG(73) = PRG(25)
      RRG(74) = PRG(27)
      RRG(75) = 0
      RRG(76) = 0
      RRG(77) = 0
      RRG(78) = 0
      RRG(79) = PRG(31)
      RRG(80) = PRG(33)
      RRG(81) = 0
      RRG(82) = 0
      RRG(83) = 0
      RRG(84) = 0
      RRG(85) = 0
      RRG(86) = 0
      RRG(87) = PRG(57)
      RRG(88) = 0
      RRG(39) = 0
      RRG(90) = 0
      RRG(91) = 0
      RRG(92) = 0
      RRG(93) = PRG(63)
      RRG(94) = 0
      RRG(95) = 0
```

TABLE B-1. MODIFICATIONS TO BOUT AND RVOUT (Continued)

RRG(96) = 0
RRG(97) = 0
RRG(98) = 0
RRG(99) = PRG(69)
RRG(100)= 0
RRG(101)= 0
RRG(102)= 0
~~RRG(103)= 0~~
RRG(104)= PRG(4)
RRG(105)= 0
RRG(106)= 0
RRG(107)= 0
RRG(108)= 0
~~RRG(109)= 0~~
RRG(110)= PRG(10)
RRG(111)= 0
RRG(112)= 0
RRG(113)= 0
RRG(114)= 0
~~RRG(115)= 0~~
RRG(116)= PRG(16)
RRG(117)= 0
RRG(118)= 0
RRG(119)= 0
RRG(120)= 0
~~RRG(121)= PRG(20)~~
RRG(122)= PRG(26)
RRG(123)= PRG(32)
RRG(124)= PRG(47)
RRG(125)= PRG(48)
RRG(126)= PRG(77)
~~RRG(127)= PRG(59)~~
RRG(128)= PRG(60)
RRG(129)= PRG(75)
RRG(130)= PRG(64)
~~RRG(131)= PRG(24)~~
RRG(132)= PRG(30)
~~RRG(133)= PRG(36)~~
RRG(134)= PRG(37)
~~RRG(135)= PRG(38)~~
RRG(136)= PRG(39)
~~RRG(137)= PRG(41)~~
RRG(138)= PRG(42)
RRG(139)= PRG(44)
RRG(140)= PRG(45)
~~RRG(141)= PRG(49)~~
RRG(142)= PRG(43)
~~RRG(143)= PRG(50)~~
RRG(144)= PRG(51)
~~RRG(145)= PRG(52)~~
RRG(146)= PRG(55)

~~RRG(147)= PRG(56)~~
RRG(148)= PRG(58)
~~RRG(149)= PRG(61)~~
RRG(150)= PRG(62)
~~RRG(151)= PRG(65)~~
RRG(152)= PRG(66)
~~RRG(153)= PRG(67)~~
RRG(154)= PRG(68)
~~RRG(155)= PRG(70)~~
RRG(156)= PRG(71)
~~RRG(157)= PRG(72)~~
RRG(158)= PRG(73)
~~RRG(159)= PRG(74)~~
RRG(160)= PRG(46)
~~RRG(161)= PRG(78)~~
RRG(162)= 0

PRG(22)= PRG(22)*RTOD
~~PRG(23)= PRG(23)*RTOD~~
PRG(28)= PRG(28)*RTOD
PRG(29)= PRG(29)*RTOD
PRG(34)= PRG(34)*RTOD
PRG(35)= PRG(35)*RTOD
PRG(41)= PRG(41)*RTOD
~~PRG(64)=PRG(64)*RTOD~~
PRG(65)= PRG(65)*RTOD
~~PRG(75) = PRG(75)*RTOD~~
RRG(4) = PRG(22)
RRG(52)= PRG(23)
RRG(10)= PRG(28)
~~RRG(58)= PRG(29)~~
RRG(16)= PRG(34)
RRG(64)= PRG(35)
RRG(137)=PRG(41)
RRG(130)=PRG(64)
RRG(151)=PRG(65)
RRG(129)= PRG(75)

TABLE B-1. MODIFICATIONS TO BOUT AND RVOUT (Continued)

```
      IF(ICON(52).EQ.0) GO TO 515
      WRITE(14,25) PTIM
  515 CONTINUE
   25 FORMAT(615.7)
C     WRITE(6,2) PTIM,TSTT,DT,IRJU,IFAIL

      IF(ICON(52).EQ.0) GO TO 516
      WRITE(14,25) TIMC
  516 CONTINUE
C     WRITE(6,2) TIMC,TST,DT,IRJU,IFAIL

      IF(ICON(52).EQ.0) GO TO 533
      WRITE(14,26) (PR(I),I=1,30)
  533 CONTINUE
   26 FORMAT(6,(615.7)
      WRITE(6,5) (BT(I),PT(I),I=1,30)

      IF(ICON(52).EQ.0) GO TO 534
      WRITE(14,26) (PV(I),I=1,18)
  534 CONTINUE
      WRITE(6,5) (DV(I),PV(I),I=1,18)

      IF(ICON(15).GT.0) GO TO 544
      LINC = LINC + 14
      WRITE(6,4)
      IF(ICON(52).EQ.0) GO TO 541
      WRITE(14,26) (RM(I),I=1,72)
  541 CONTINUE

      IF(ICON(52).EQ.0) GO TO 554
      WRITE(14,26) (RG(I),I=1,162)
  554 CONTINUE
      WRITE(6,5) (DG(I),PG(I),I=1,NG)

      DIMENSION PT(30),PV(18),PM(54),PG(120),RG(162),RM(72)
```

RG(162),RM(72) added

TABLE B-1. MODIFICATIONS TO BOUT AND RVOUT (Continued)

```
PM( 1) = PM( 1)*GST          RM( 1) = PM( 1)
PM( 4) = PM( 4)*GST          RM( 4) = PM( 4)
PM( 6) = PM( 6)*GST          RM( 7) = PM( 6)
PM( 9) = PM( 9)*GST          RM(10) = PM( 9)
PM(11) = PM(11)*GST          RM(13) = PM(11)
PM(16) = PM(16)*GST          RM(19) = PM(16)
PM(18) = PM(18)*GST          RM(23) = PM(18)
PM(19) = PM(19)*GST          RM(24) = PM(19)
PM(20) = PM(20)*GST          RM(25) = PM(20)
PM(22) = PM(22)*GST          RM(29) = PM(22)
PM(23) = PM(23)*GST          RM(30) = PM(23)
PM(24) = PM(24)*GST          RM(31) = PM(24)
PM(26) = PM(26)*GST          RM(35) = PM(26)
PM(27) = PM(27)*GST          RM(36) = PM(27)
PM(28) = PM(28)*RTOD         RM(37) = PM(28)
PM(29) = PM(29)*RTOD         RM(38) = PM(29)
PM(30) = PM(30)*RTOD         RM(39) = PM(30)
PM(31)= PM(31)*RTOD          RM(40) = PM(31)
PM(32) = PM(32)*GST          RM(41) = PM(32)
PM(33) = PM(33)*RTOD         RM(43) = PM(33)
PM(34) = PM(34)*RTOD         RM(44) = PM(34)
PM(35)= PM(35)*RTOD          RM(45) = PM(35)
PM(36) = PM(36)*RTOD         RM(46) = PM(36)
PM(37) = PM(37)*GST          RM(47) = PM(37)
PM(38)= PM(38)*RTOD          RM(49) = PM(38)
PM(39)= PM(39)*RTOD          RM(50) = PM(39)
PM(40)= PM(40)*RTOD          RM(51) = PM(40)
PM(41)= PM(41)*RTOD          RM(52) = PM(41)
PM(42) = PM(42)*GST          RM(53) = PM(42)
PM(46)= PM(46)*RTOD          RM(53) = PM(46)
PM(52) = PM(52)*RTOD         RM(67) = PM(52)
```

TABLE B-1. MODIFICATIONS TO BOUT AND RVOUT (Continued)

```
PG(120)=DVD-ZMTD
RG( 1) = PG(1)
RG( 2) = PG(2)
RG( 3) = PG(3)
RG( 4) = PG(4)
RG( 5) = PG(5)
RG( 6) = PG(6)
RG( 7) = PG(7)
RG( 8) = PG(8)
RG( 9) = PG(9)
RG(10) = PG(10)
RG(11) = PG(11)
RG(12) = PG(12)
RG(13) = PG(13)
RG(14) = PG(14)
RG(15) = PG(15)
RG(16) = PG(16)
RG(17) = PG(17)
RG(18) = PG(18)
RG(19) = PG(19)
RG(20) = PG(20)
RG(21) = PG(21)
RG(22) = PG(22)
RG(23) = PG(23)
RG(24) = PG(24)
RG(25) = PG(25)
RG(26) = PG(26)
RG(27) = PG(28)
RG(28) = PG(28)
RG(29) = PG(29)
RG(30) = PG(30)
RG(31) = PG(31)
RG(32) = PG(32)
RG(33) = PG(33)
RG(34) = PG(34)
RG(35) = PG(35)
RG(36) = PG(36)
RG(37) = PG(37)
RG(38) = PG(38)
RG(39) = PG(39)
RG(40) = PG(40)
RG(41) = PG(41)
RG(42) = PG(42)
RG(43) = PG(43)
RG(44) = PG(44)
RG(45) = PG(45)
INDGC = IAPFLG + 10*IGAFLG + 100*IACFLG + 1000*ISEQ
RINDGC = INDGC + 99.D5 + 0.000005
RG(46) = PG(46)
RG(47) = PG(47)
RG(48) = PG(48)
```

TABLE B-1. MODIFICATIONS TO BOUT AND RVOUT (Continued)

```
      RG(49) = PG(49)
      RG(50) = PG(50)
      RG(51) = PG(51)
      RG(52) = PG(52)
      RG(53) = PG(53)
      RG(54) = PG(54)
      RG(55) = PG(55)
      RG(56) = PG(56)
      RG(57) = PG(57)
      RG(58) = PG(58)
      RG(59) = PG(59)
      RG(60) = PG(60)
      RG(61) = PG(61)
      RG(62) = PG(62)
      RG(63) = PG(63)
      RG(64) = PG(64)
      RG(65) = PG(65)
      IFTEMP = IALTU + 10*(IPUDFL+1) + 100*(IUPDFL+1) + 1000*
     1 JBSW + 10000*(IRCHFG+1)
      RG(66) = PG(66)
      RG(67) = PG(67)
      RG(68) = PG(68)
      RG(69) = PG(69)
      RG(70) = PG(70)
      RG(71) = PG(71)
      RG(72) = PG(72)
      RG(73) = PG(73)
      RG(74) = PG(74)
      RG(75) = PG(75)
      RG(76) = PG(76)
      RG(77) = PG(77)
      RG(78) = PG(78)
      RG(79) = PG(79)
      RG(80) = PG(80)
      RG(81) = PG(81)
      RG(82) = PG(82)
      RG(83) = PG(83)
      RG(84) = PG(84)
      RG(85) = PG(85)
      RG(86) = PG(86)
      RG(87) = PG(87)
      RG(88) = PG(88)
      RG(89) = PG(89)
      RG(90) = PG(90)
      RG(91) = PG(91)
      RG(92) = PG(92)
      RG(93) = PG(93)
      RG(94) = PG(94)
      RG(95) = PG(95)
      RG(96) = PG(96)
      RG(97) = PG(97)
```

TABLE B-1. MODIFICATIONS TO BOUT AND RVOUT (Continued)

```
RG(98) = PG(98)
RG(99) = PG(99)
RG(100)= PG(100)
RG(101)= PG(101)
RG(102)= PG(102)
RG(103)= PG(103)
RG(104)= PG(104)
RG(105)= PG(105)
RG(106)= PG(106)
RG(107)= PG(107)
RG(108)= PG(108)
RG(109)= PG(109)
RG(110)= PG(110)
RG(111)= PG(111)
RG(112)= PG(112)
RG(113)= PG(113)
RG(114)= PG(114)
RG(115)= PG(115)
RG(116)= PG(116)
RG(117)= PG(117)
RG(118)= PG(118)
RG(119)= PG(119)
RG(120)= PG(120)
RG(121)= 0
RG(122)= 0
RG(123)= 0
RG(124)= 0
RG(125)= 0
RG(126)= 0
RG(127)= 0
RG(128)= 0
RG(129)= 0
RG(130)= 0
RG(131)= 0
RG(132)= 0
RG(133)= 0
RG(134)= 0
RG(135)= 0
RG(136)= 0
RG(137)= 0
RG(138)= 0
RG(139)= 0
RG(140)= 0
RG(141)= 0
RG(142)= 0
RG(143)= 0
RG(144)= 0
RG(145)= 0
RG(146)= 0
RG(147)= 0
RG(148)= 0
RG(149)= 0
RG(150)= 0
RG(151)= 0
RG(152)= 0
RG(153)= 0
RG(154)= 0
RG(155)= 0
RG(156)= 0
RG(157)= 0
RG(158)= 0
RG(159)= 0
RG(160)= 0
RG(161)= 0
RG(162)= 0
```

TABLE B-1. MODIFICATIONS TO BOUT AND RVOUT (Continued)

```
PG( 4)= PG( 4)*RTOD
PG(10)= PG(10)*RTOD
PG(16)= PG(16)*RTOD
PG(43)= PG(43)*RTOD
PG(49)= PG(49)*RTOD
PG(52) = PG(52)*RTOD
PG(55) = PG(55)*RTOD
PG(58) = PG(58)*RTOD
PG(64) = PG(64)*RTOD
PG(71)= PG(71)*RTOD
PG(77)= PG(77)*RTOD
PG(83)= PG(83)*RTOD
PG(88)= PG(88)*RTOD
PG(89)= PG(89)*RTOD
PG(90)= PG(90)*RTOD
PG(94)= PG(94)*RTOD
PG(95)= PG(95)*RTOD
PG(96)= PG(96)*RTOD
PG(97)= PG(97)*RTOD
PG(100)=PG(100)*RTOD
PG(101)=PG(101)*RTOD
PG(102)=PG(102)*RTOD
RG( 4) = PG( 4)
RG(10) = PG(10)
RG(16) = PG(16)
RG(43) = PG(43)
RG(49) = PG(49)
RG(52) = PG(52)
RG(55) = PG(55)
RG(58) = PG(58)
RG(64) = PG(64)
RG(71) = PG(71)
RG(77) = PG(77)
RG(83) = PG(83)
RG(88) = PG(88)
RG(89) = PG(89)
RG(90) = PG(90)
RG(94) = PG(94)
RG(95) = PG(95)
RG(96) = PG(96)
RG(97) = PG(97)
RG(100)= PG(100)
RG(101)= PG(101)
RG(102)= PG(102)
```

TABLE B-1. MODIFICATIONS TO BOUT AND RVOUT (Continued)

PM(54) = WDACB
RM(1) = PM(1)
RM(2) = PM(2)
~~RM(3) = PM(3)~~
RM(4) = PM(4)
~~RM(5) = PM(5)~~
RM(6) = 0
~~RM(7) = PM(6)~~
RM(8) = PM(7)
~~RM(9) = PM(8)~~
RM(10) = PM(9)
~~RM(11) = PM(10)~~
RM(12) = 0
~~RM(13) = PM(11)~~
RM(14) = PM(12)
~~RM(15) = PM(13)~~
RM(16) = PM(14)
RM(17) = PM(15)
RM(13) = 0
RM(19) = PM(16)
RM(20) = 0
~~RM(21) = 0~~
RM(22) = PM(17)
RM(23) = PM(18)
RM(24) = PM(19)
RM(25) = PM(20)
RM(26) = 0
RM(27) = 0
RM(28) = PM(21)
RM(29) = PM(22)
RM(30) = PM(23)
RM(31) = PM(24)
RM(32) = 0
~~RM(33) = 0~~
RM(34) = PM(25)
RM(35) = PM(26)
RM(36) = PM(27)
RM(37) = PM(28)
RM(38) = PM(29)
RM(39) = PM(30)
RM(40) = PM(31)
RM(41) = PM(32)
RM(42) = 0
RM(43) = PM(33)
RM(44) = PM(34)
~~RM(45) = PM(35)~~
RM(46) = PM(36)
RM(47) = PM(37)
RM(48) = 0
RM(49) = PM(38)
RM(50) = PM(39)
~~RM(51) = PM(40)~~
RM(52) = PM(41)
RM(53) = PM(42)
RM(54) = 0
~~RM(55) = PM(43)~~
RM(56) = PM(44)
~~RM(57) = PM(45)~~
RM(53) = PM(46)
~~RM(59) = PM(47)~~
RM(60) = 0
~~RM(61) = PM(48)~~
RM(62) = PM(49)
~~RM(63) = 0~~
RM(64) = 0
~~RM(65) = PM(50)~~
RM(66) = PM(51)
~~RM(67) = PM(52)~~
RM(68) = PM(53)
~~RM(69) = PM(54)~~
RM(70) = 0
RM(71) = 0
RM(72) = 0

APPENDIX C
U70 PROGRAM .CSS (COMMAND SUBSTITUTE SYSTEM)
SOURCE FILES

APPENDIX C

U70 PROGRAM .CSS (Command Substitute System) SOURCE FILES

The Perkin-Elmer 3220 computer on which the U70 program was run, has a development compiler which compiles rapidly but produces slow-running object code, and an optimizing compiler which compiles slowly but produces fast running code. The development compiler was used to compile the U70 program, since compile times rather than execution times were at a premium during the period of performance of this task. As stated in Section III,B, in order to further reduce compile times, each subroutine was assigned its own data file so that one could recompile one subroutine at a time rather than having to recompile the whole program each time a change was made in one of the subroutines.

There are two .CSS commands that can be used to actually carry out a compilation: U70CA and U70C. The U70CA command compiles all the Pershing II source files, while the U70C command is used to compile an individual subroutine. The specific job control sequence which initiates the U70 compilation is as follows:

U70 @1, @2, @3, @4, @5

- @1 Input source file
- @2 Object file name (defaults to @1.OBJ)
- @3 List file name (defaults to @1.LST)
- @4 Start option(s) separated with spaces (defaults to no options)
- @5 Load size for F7D (defaults to 50K bytes).

A cross-reference listing can be obtained by including the command XREF in the @4 list of start options.

A preprocessing program called TRAJ must be run before the U70 program is ready to run. The TRAJ program is derived from equations resident in the Pershing Launch Computer, which is located on board the missile. Given the launch and target latitude, longitude, altitude, and the number of stages (Table C-1), it computes launch presets and transmits them to the portion of the program which simulates boost and terminal guidance. This, in turn, generates a data file containing estimates of the desired velocities and cutoff angles, estimates of flight and pitch-over times, and target offsets (Table C-2). The U70 program uses these outputs to make its own calculations, interpolating the TRAJ output tables.

Execution of the U70 program is initiated using the routines UST2 or UST3. These commands invoke .CSS files containing other commands to load the task, make logical unit assignments, and start the task.

The UST2 routine is used in order to make a full run (flight from launch to impact). The inputs to UST2 are obtained from the TRAJ programs and are used to build a new data file (logical unit 11). This file is used to create a new input tape in the U70 program, TAPE 11. The output from UST2.CSS is printed at time intervals whose value is input in Constant 271, or is printed at any point along the flight path where a discontinuity appears (such as the end of the boost phase, the apogee, etc.,).

The subroutines in the U70 program are linked using the .CSS command U70LINK. This command builds an overlaid executable task. For example, it ensures that copies of all of the subroutines called within a given subroutine are present within that subroutine.

TABLE C-1. TARGETING PROGRAM INPUTS

ITRS	=	210	Single Stage
		220	Two Stage
LONT	=		Longitude of the Target
LATDT	=		Latitude of the Target
HT	=		Altitude of the Target
LONL	=		Longitude of the Launch Site
LATDL	=		Latitude of the Launch Site
HL	=		Altitude of the Launch Site

TABLE C-2. GLOSSARY OF TARGETING PROGRAM OUTPUT VARIABLES

U70	TRAJECTORY OUTPUT	DEFINITION
140	GAMDE + DELGAM	Flight path angle desired before pitch-over
142	AZD	Flight azimuth measured from north
115	AZD	* Same value
114	LATDL	Launch latitude
116	LONL	Launch longitude
117	HL	Launch altitude
203	DLAT	Aimpoint offset latitude
205	(Unnamed)	Aimpoint offset altitude
206	LATDT	Target latitude
207	LONT	Target longitude
208	HT	Target altitude
219	GAMDE	GAMME desired at cut-off
220	TCO	Time of booster cut-off
302	TFF	Elasped time of re-entry vehicle freefall
310	TGAO	Elasped time before pitch-over
312	VRE	Required velocity for terminal phase
314	TF	Boost guidance flight time estimate
411	CHI	Normalized velocity loss
420	TAU	Initial delta value time impact
421	AXPU	Acceleration begins pull-up
412	DVP	Velocity control bias

U70LINK is structured so that it contains four overlays. The first overlay includes a subroutine called MINIT.OBJ, which presets all variables to zero. The second overlay includes ENPUT.OBJ, which creates a new input data tape. The third overlay includes all binary files appropriate to the boost phase of simulation. The fourth overlay includes all re-entry-vehicle binary code.

In addition, U70LINK creates a load map, U70-U70LINK.MAP, that is useful for debugging purposes. Control indicators are input to set flight options, and program constants are entered to control the initial conditions, multipliers, limits, errors, print controls, etc. Aerodynamic drag and atmospheric data are input through the tabular data portion of the input file. A list and description of these data inputs is given in Reference 1.

After a full run has been made, a restart can be created using the output from UST2.CSS and the UST3.CSS command. The restart procedure is a very useful tool in a study where new conditions are imposed on a missile at a certain point in flight. However, the restart could not be used for the nozzle deflection study, since a restart could not be made during the boost phase. Certain values which appeared in THRUST were functions of time and could not be re-initialized at the arbitrary time desired for a restart.

The values which are required to be input into the UST3.CSS data file of UST3.CSS are listed in Table C-3. Tables C-4, C-5, and C-6 list other U70 files used in the performance of this task for the benefit of anyone who might want to initiate U70 runs.

The procedure of outputting U70 flight data to tapes required that modifications be made to the subroutines BOUT (Boost Output) and RVOUT (Re-entry Vehicle Output). The desired output was written to logical unit 14 and a .CSS file was created, with logical unit 14 set to MAGØ, to write the data to the tapes.

TABLE C-3. GLOSSARY OF RESTART INPUT VARIABLES

U70	UST3, .CSS Input		Definition
101	TIME		Time desired for restart
102	XL		X component position in launch coordinate
103	YL	at time of restart	Y component position in launch coordinate
104	ZL		Z component position in launch coordinate
105	XLD		X component velocity in launch coordinate
106	YLD	at time of restart	Y component velocity in launch coordinate
107	ZLD		Z component velocity in launch coordinate
111	PHI		Euler angle
112	THETA	at time of restart	Euler angle
113	PSI		Euler angle
114	LATDL		Launch latitude
116	LONL		Launch longitude
117	HL		Launch altitude
140	GAMDE + DELGAM		Flight path angle desired at pitch-over
142	AZD		Azimuth
203	DLAT		Aimpoint offset in latitude
205	DALT		Aimpoint offset in altitude
206	LATDT		Target latitude
207	LONT		Target longitude
208	HT		Target altitude
219	GAMDE		Gamma desired at cut-off
220	TCO		Time of cut-off
314	TF		Boost guidance flight time estimate
420	TAU		Initial delta valve time impact
421	AXPU		Acceleration begins pull-up

TABLE C-4. U70 SOURCE FILES

ARD59D.FTN	QUAD.FTN
BEE.FTN	RVINIT.FTN
CONV.FTN	RVAPLT.FTN
FASTD.FTN	RVAERO.FTN
FASTC.FTN	SCS.FTN
SORTIT.FTN	GAUSS.FTN
SPEEDY.FTN	RANDU.FTN
WRTAPE.FTN	BEXEC.FTN
BINIT.FTN	THRUST.FTN
BGUID.FTN	TRIMA.FTN
FUNC.FTN	BOUT.FTN
VACRAN.FTN	BAPLT.FTN
FOLD.FTN	RATLIM.FTN
GRAPH.FTN	BAERO.FTN
NINIT.FTN	UST2.CSS
ENPUT.FTN	UST3.CSS
RVEXEC.FTN	UST4.CSS
STAT.FTN	U70LINK.CSS
RVOUT.FTN	U70C.CSS
RVGUID.FTN	

TABLE C-5. FILES USED FOR DEFLECTION STUDY

COB351.DAT	-	30 seconds, .5 degree pitch
COB352.DAT	-	30 seconds, 2.0 degree pitch
COB353.DAT	-	30 seconds, 7.6 degree pitch
COB354.DAT	-	49 seconds, .5 degree pitch
COB355.DAT	-	49 seconds, 2.0 degree pitch
COB356.DAT	-	49 seconds, 7.6 degree pitch
COB357.DAT	-	30 seconds, .5 degree yaw
COB358.DAT	-	30 seconds, 2.0 degree yaw
COB359.DAT	-	30 seconds, 7.6 degree yaw
COB360.DAT	-	49 seconds, .5 degree yaw
COB361.DAT	-	49 seconds, 2.0 degree yaw
COB362.DAT	-	49 seconds, 7.6 degree yaw

TABLE C-6. FILES USED FOR TRAJECTORY PROFILE

COB370.DAT
COB371.DAT
COB372.DAT
COB373.DAT
COB374.DAT
COB375.DAT
COB376.DAT
COB377.DAT
COB378.DAT

APPENDIX D
TRW PROGRAM FILE NAMES STORED ON DISK

TABLE D-1. SUBROUTINE FILE NAMES STORED ON DISK

TWAERO.FTN	TWMATRX1.FTN
TWAFC.FTN	TWMATRX2.FTN
TWALNON.FTN	TWMAXBT.FTN
TWAMC.FTN	TWNAVON.FTN
TWATMENT.FTN	TWNESTG.FTN
TWATMEXT.FTN	TWPWRON.FTN
TWATMOS.FTN	TWRADAR.FTN
TWAUX1.FTN	TWRCS.FTN
TWAUX2.FTN	TWRCSFM.FTN
TWBATT.FTN	TWRGU.FTN
TWBURST.FTN	TWRKUTTA.FTN
TWCMDPRC.FTN	TWRMMUL.FTN
TWCROSS.FTN	TWROTATE.FTN
TWDCU.FTN	TWRVAERO.FTN
TWDCUERR.FTN	TWRVBAS.FTN
TWDERIV.FTN	TWRVPTCH.FTN
TWDMNDC.FTN	TWRVROLL.FTN
TWDOT.FTN	TWRVYAW.FTN
TWDRAND.FTN	TWSAF.FTN
TWEBITS.FTN	TWSBITS.FTN
TWFLIGHT.FTN	TWSDPROC.FTN
TWFSICFU.FTN	TWSSICFU.FTN
TWFSISA.FTN	TWSSSCFU.FTN
TWFSSCFO.FTN	TWSTINIT.FTN
TWGRAV.FTN	TWTHRUST.FTN
TWGRN.FTN	TWTRNSF2.FTN
TWIMSALN.FTN	TWTRNSF3.FTN
TWIMSPLS.FTN	TWTRNSPO.FTN
TSIMSRSV.FTN	TWURN.FTN
TWIMSTST.FTN	TWVANES.FTN
TWINTERP.FTN	TWVMP.FTN
TWJDM.FTN3	TWWIND.FTN
TWMAGN.FTN	

TABLE D-2. FILES CONTAINED ON DISC FOR USER PROGRAM HANDLING

The following files are used:

DAVEF7.CSS - Used to compile each subroutine file.
TRWLINK.CSS - Used to link the TRW 6DOF simulator.
DCOMMENT.FTN - Contains comments which describe the function of each subroutine.
TRWRV.DAT - Contains re-entry aerodynamic data.
TRWSS.DAT - Contains single stage aerodynamic data.
TRWTAB.FTN - Used to read TRWRV.DAT and TRWSS.DAT and string them into a long one dimensional array. TRWTAB.CSS is used to run this program and TRWTAB.DAT contains the output of the TRWTAB.FTN.
BIG01R.DAT, BIG01I, BIG01D - Contain variables defined in the program and their storage locations in the BIG01 arrays, as specified by EQUIVALENCE statements.

The following files are used in the $INCLUDE statements to handle common blocks and equivalence statements.

TRW001.DAT
TRW002.DAT
TRW003.DAT
TRW004.DAT
TRW005.DAT
TRW006.DAT
TRW007.DAT
TRW008.DAT
TRW009.DAT
TRW0010.DAT
TRW0011.DAT
TRW0012.DAT
TRWCON.DAT
TRWCONST.DAT

APPENDIX E
TRW PROGRAM SUBROUTINE NAMES AND CALLING SEQUENCES

TABLE E-1. SUBROUTINE NAMES AND CALLING SEQUENCES

AERO	DCU	FSISA	NEWSTG	SBITS
RVAERO	INTERP	SBITS	TRNSF2	
AFC	SBITS			SDPROC
AMC	DCUERR	FSSCFU	PWRON	SBITS
		SBITS	TRNMAT	
AFC	DCUERR		TRNSPO	SSICFU
INTERP	INTERP	GRAV	STINIT	SBITS
			GRAV	
ALNON	DERIV	GRN	RMMUL	SSSCFU
MATRX1	VMP			SBITS
TRNMAT	AUX1	IMSALN	RADAR	
TRNSPO	NOZZLE	SBITS	TRNSF3	STINIT
RMMUL	THRUST	RMMUL		RMMUL
	JDM	TRNMAT	RCS	
	AUX1	TRNSPO		THRUST
AMC	AERO		RCSFM	INTERP
INTERP		IMSPLS	CROSS	
	DMNDC	RMMUL		TRNSF2
ATMENT	SAF		RGU	
	SDPROC	IMSRSV		TRNSF3
ATMEXT	BATT		RKUTTA	
	FSICFU	IMSTST	DERIV	TRNSPO
ATMOS	FSISA	SBITS		
	FSSCFU		RMMUL	URN
AUX1	SSICFU	INTERP		
MATRX1	SSSCFU		ROTATE	VANES
GRAV	BURST	JDM		
RMMUL	SBITS		RAVERO	VMP
MAGN	CMDPRC	MAGN	RVBAS	INTERP
ATMOS			RVROLL	
	DOT	MATRX1	RVPTCH	WIND
AUX2	MAGN	TRNMAT	RVYAW	INTERP
MAGN		ROTATE		
MATRX2	DRAND	RMMUL	RVBAS	
RMMUL		TRNSPO	INTERP	
	EBITS			
BATT		MATRX2	RVPTCH	
SBITS	FLIGHT	TRNMAT	INTERP	
	RKUTTA			
BITS	AUX1	MAXBT	RVROLL	
	AUX2		INTERP	
BURST	IMSALN	NAVON		
	IMSPLS	TRNMAT	RVYAW	
CMDPRC	IMSRSV	TRNSPO	INTERP	
SBITS	RGU	MATRX1		
		STINIT	SAF	
CROSS	FSICFU	GRAV	SBITS	
UNITV	SBITS	RMMUL		

APPENDIX F
ORGANIZATION OF THE INTERP(INTERPOLATION) DATA TABLES

APPENDIX F

ORGANIZATION OF THE INTERP(INTERPOLATION) DATA TABLES

An interpolation routine called INTERP forms the backbone of the TRW program's aerodynamic simulator, and utilizes a large set of multidimensional tables to generate its results. The multidimensional interpolation process is straightforward and is described with the aid of the following example.

Suppose that we have a table of pressures:

TABLE F-1. AERODYNAMIC PRESSURES

0	0.1	0.3	0.5 (bars)

and a table of temperatures,

TABLE F-2. AERODYNAMIC TEMPERATURES

1000°	1300°	1600°

The pressure and temperature are considerd to be independent variables. Associated with these independent variables is a table (Table F-3) of aerodynamic coefficients, C:

TABLE F-3. AERODYNAMIC COEFFICIENTS

0.30021	0.30106	0.30028
0.30021	0.30176	0.30057
0.30021	0.30268	0.30081
0.30022	0.30345	0.30117

Now suppose that we are given pressure, P, = 0.3306 (bars) and temperature, T, = 1116°, and we want to interpolate to find the aerodynamic coefficients that correspond to these values of pressure and temperature. First, the interpolation routine tests the input pressure value P, against the table of pressures and determines that the input value P, = 0.3306 lies between 0.3 and 0.5. The location of the upper number - the "0.5" - is read out and stored in a variable which might be labeled "PINDEX" for Pressure INDEX. Since 0.5 is the fourth element in this independent variable table, the number stored in PINDEX is a "4", and it is a pointer to the interval in the independent variable table in which 0.3306 is located.

Next, the same operations are carried out for the temperature, using the temperature table. Since the temperature lies between 1000° and 1300°, the location of the 1300° value - location #2 - is stored in TINDEX (Temperature INDEX) to designate the proper interpolation interval.

Finally, the two index values, 4 and 2, are used to locate the interval around the 4th row and 2nd column of the dependent variable table of aerodynamic coefficients (Table F-3). Then the pressure and temperature to be evaluated, P = 0.3306 bars and T = 1116°, are used to carry out the actual interpolation, in order to obtain the interpolated values of the aerodynamic coefficients.

In the listing of the INTERP subroutine's data output, the following commands appear:

```
BEGIN
INDEPENDENT_TABLE OUTPUT VARIABLE/
                INPUT VARIABLE
DEPENDENT_TABLE OUTPUT VARIABLE/
            INPUT VARIABLE, INPUT VARIABLE
END.
```

At first, these were thought to be VAX 11/780 commands. Later, it was determined that the TRW program contains its own data input processor and logical analyzer, which reads and interprets input data tapes, including the above *statements*. Apparently, these table definitions are used only for printer output formatting. Using the example given above, these statements would take the following form:

```
BEGIN
INDEPENDENT_TABLE PINDEX/PRESUR
INDEPENDENT_TABLE TINDEX/TEMP
DEPENDENT_TABLE AEROCO/PINDEX, TINDEX
END
```

When the results are printed out, AEROCO, standing for "Aerodynamic Coefficients", would be printed at the top of the column listing the aerodynamic coefficient values which INTERP generates by table interpolation.

APPENDIX G
FLOWCHART OF AN EARLY MAIN PROGRAM

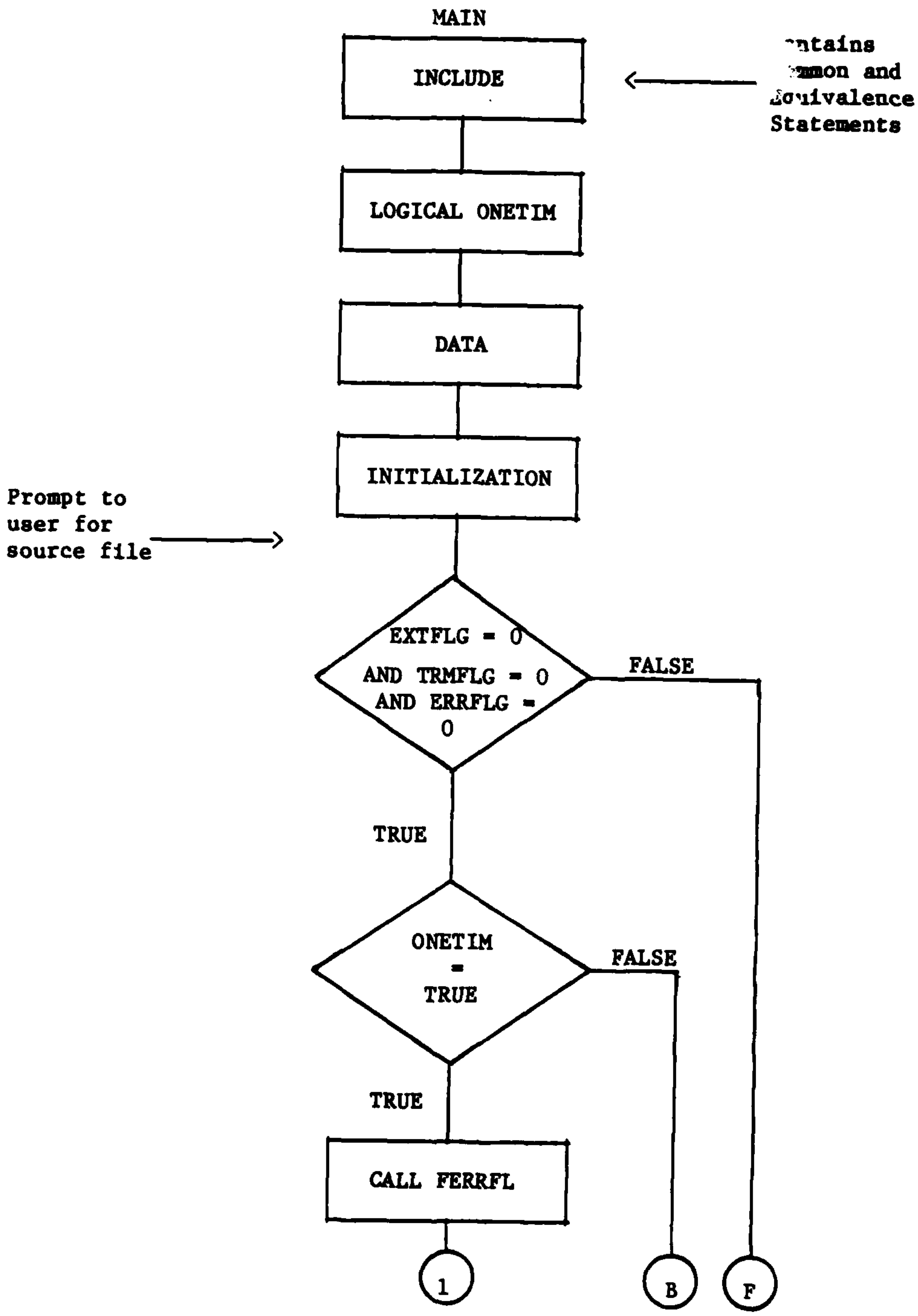

Figure G-1. FLowchart of Early MAIN Program.

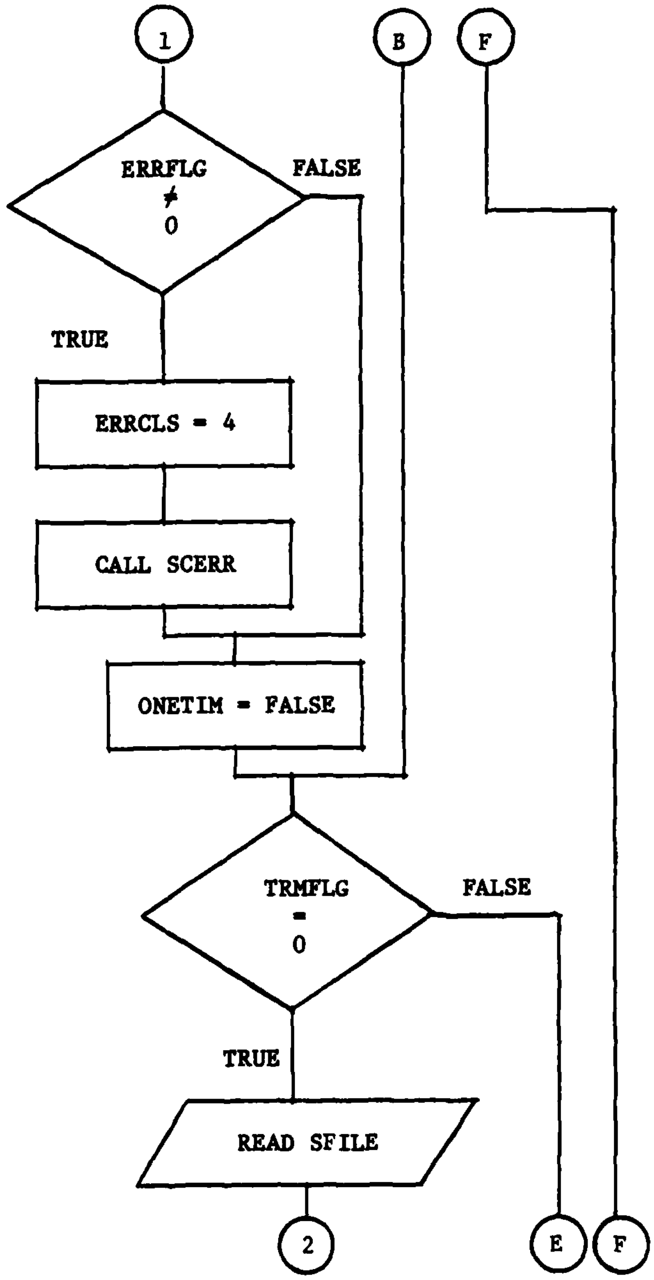

Figure G-1. FLowchart of Early MAIN Program (Continued).

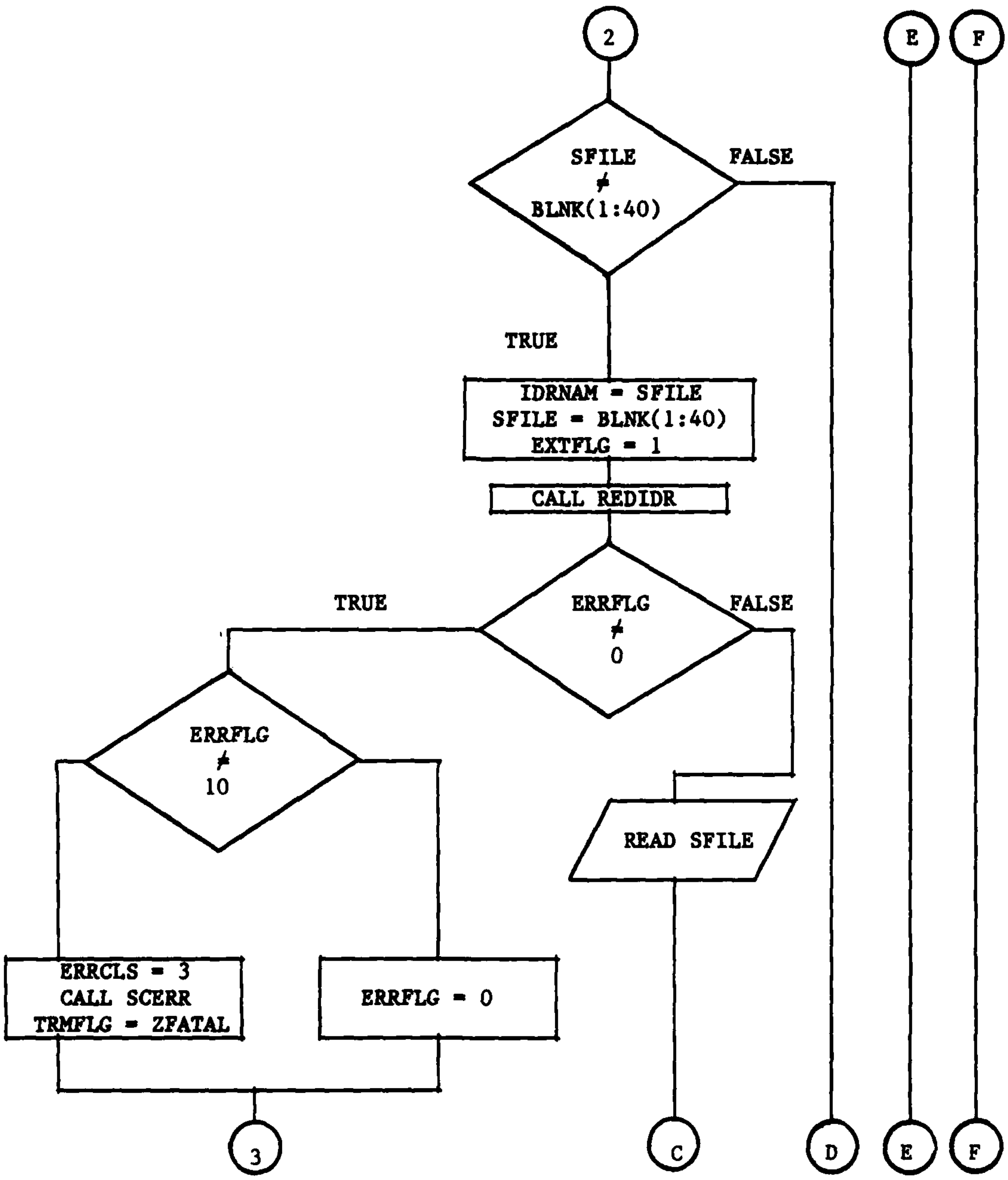

Figure G-1. Flowchart of Early MAIN Program (Continued).

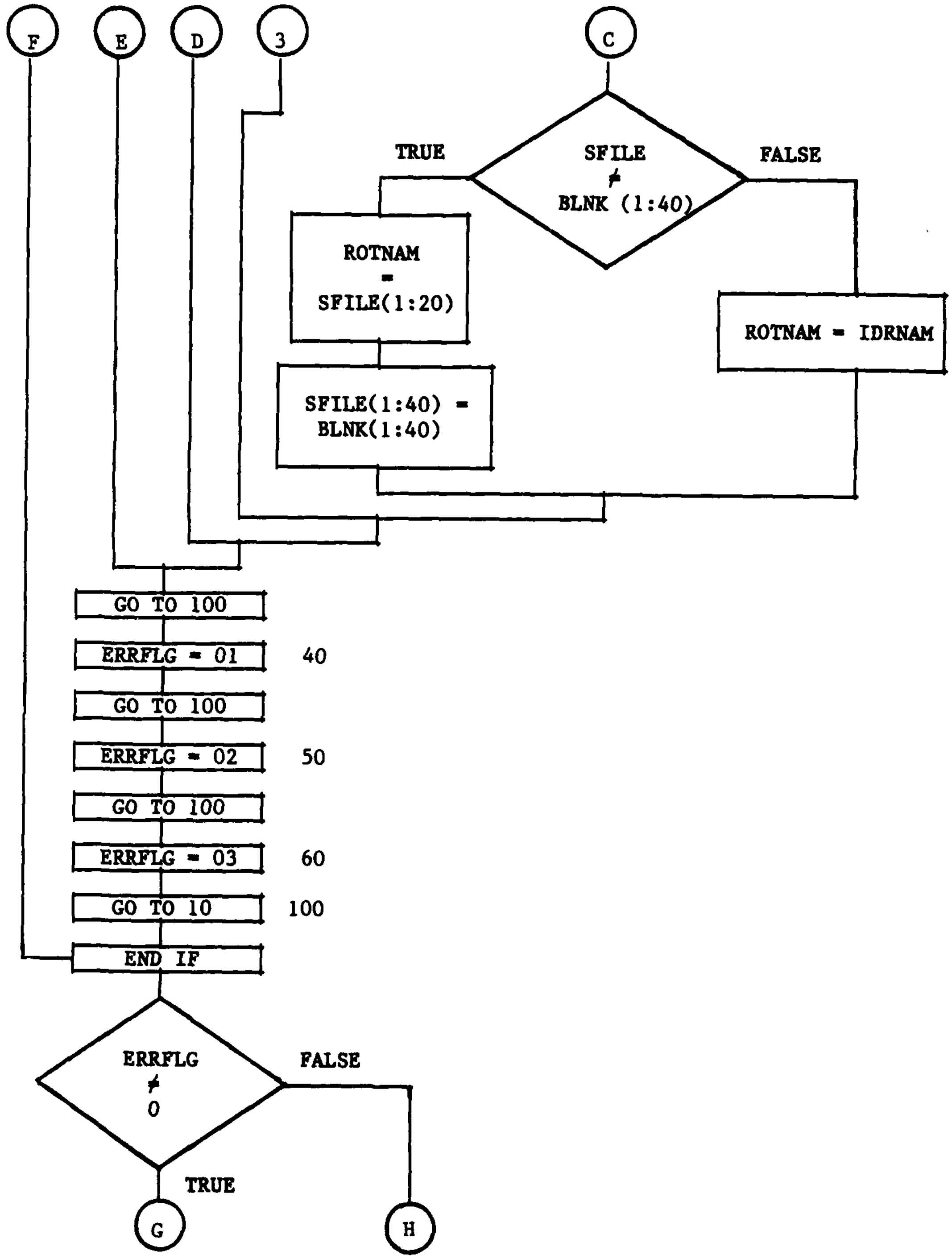

Figure G-1. Flowchart of Early MAIN Program (Continued).

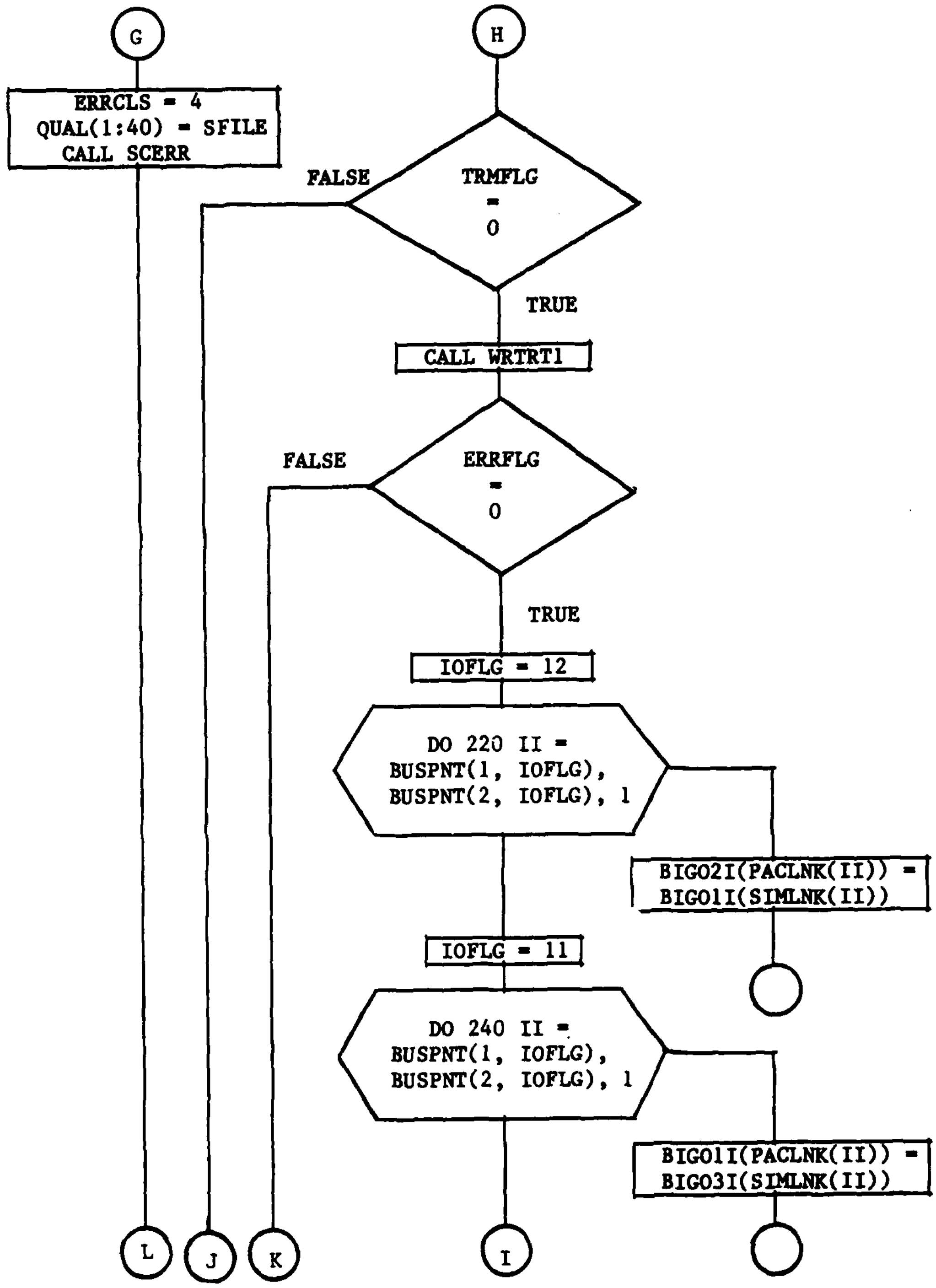

Figure G-1. Flowchart of Early MAIN Program (Continued).

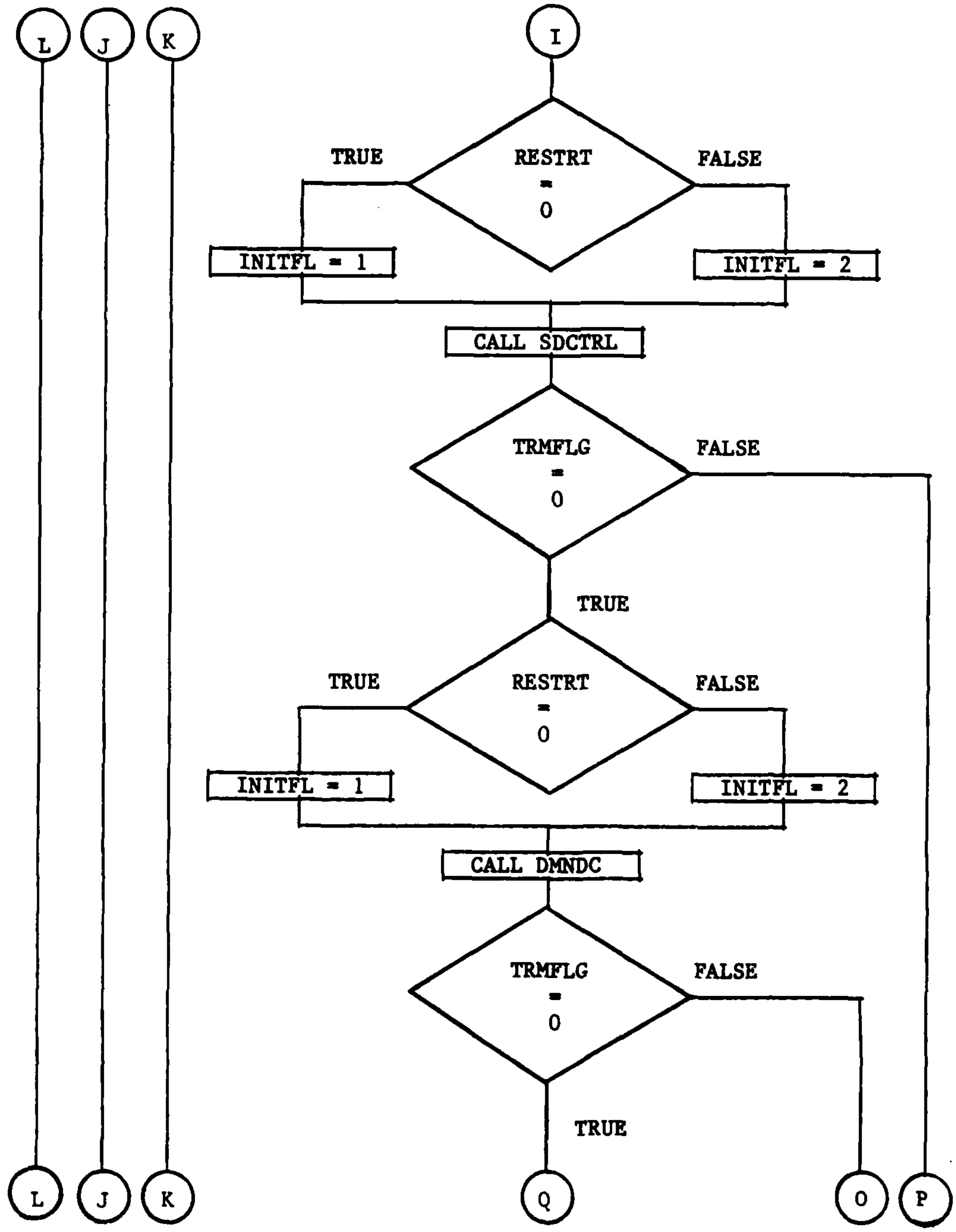

Figure G-1. Flowchart of Early MAIN Program (Continued).

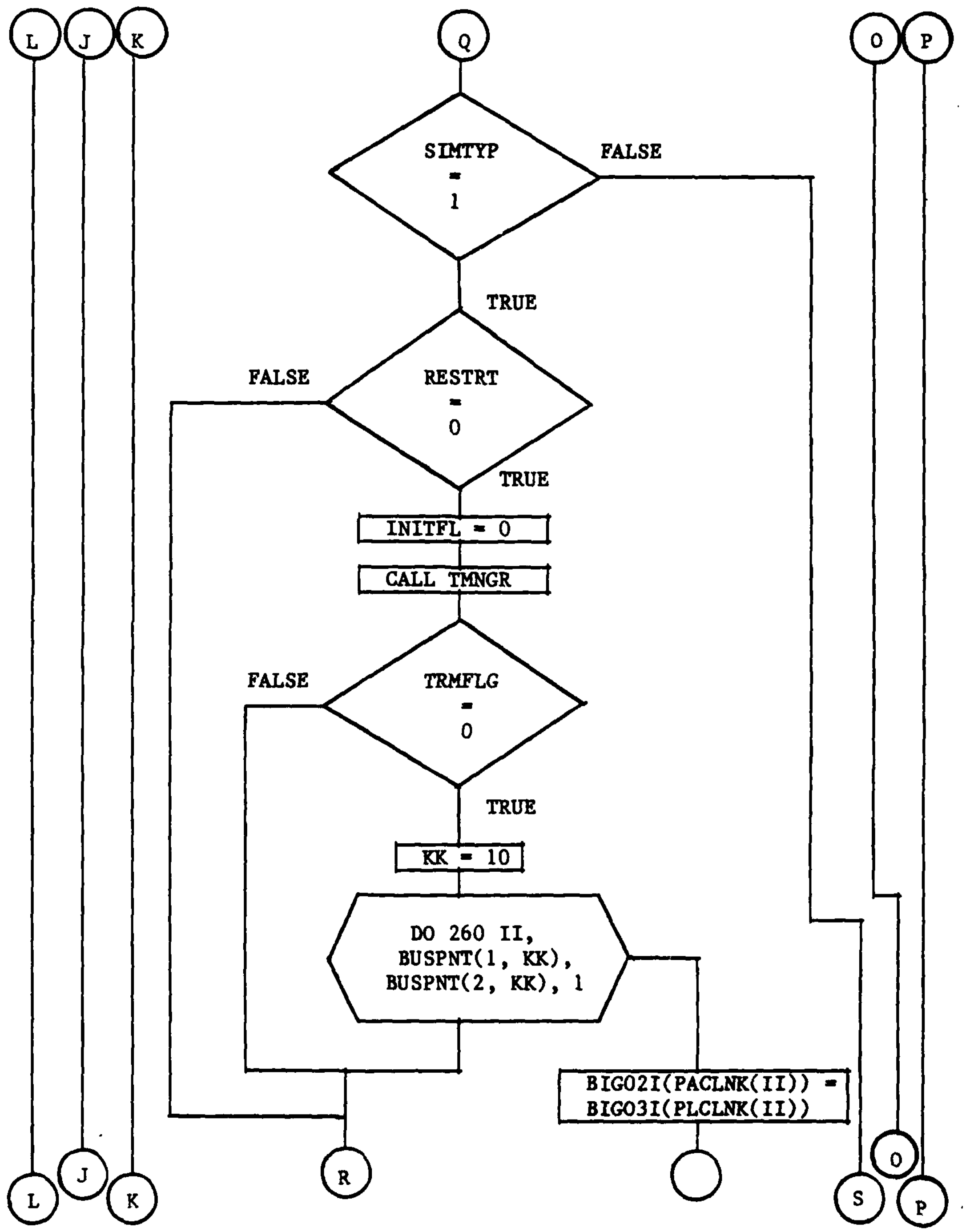

Figure G-1. Flowchart of Early MAIN Program (Continued).

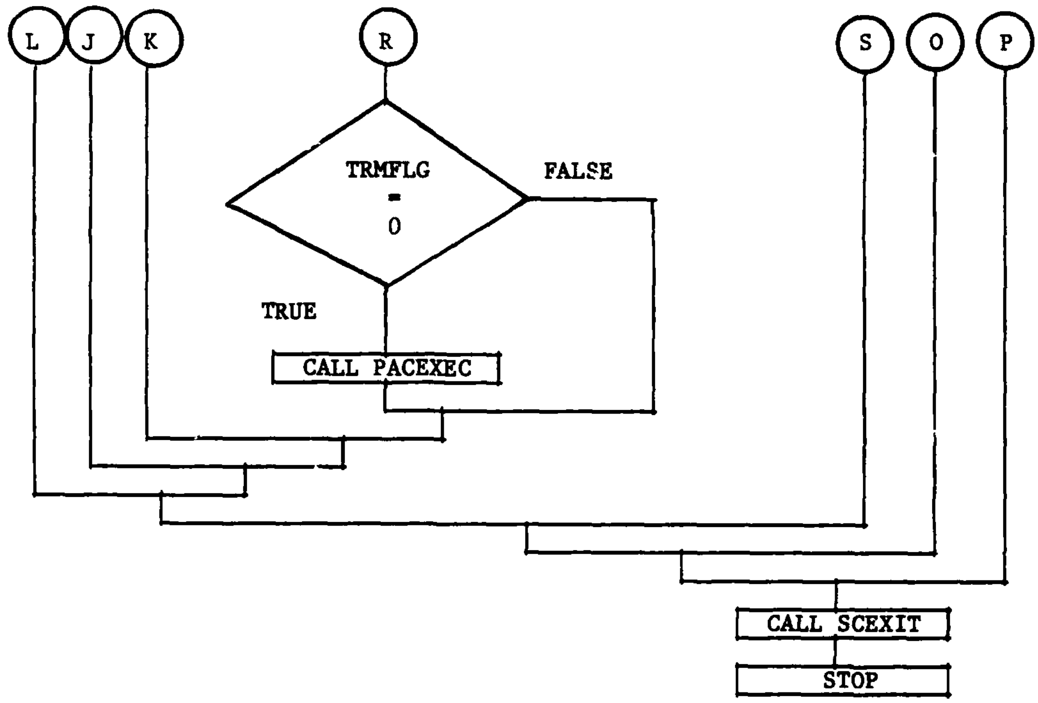

Figure G-1. Flowchart of Early MAIN Program (Continued).

APPENDIX H
FLOWCHART OF THE SUBROUTINE SDCTRL
(SIX-DEGREE-OF-FREEDOM CONTROL) PROGRAM

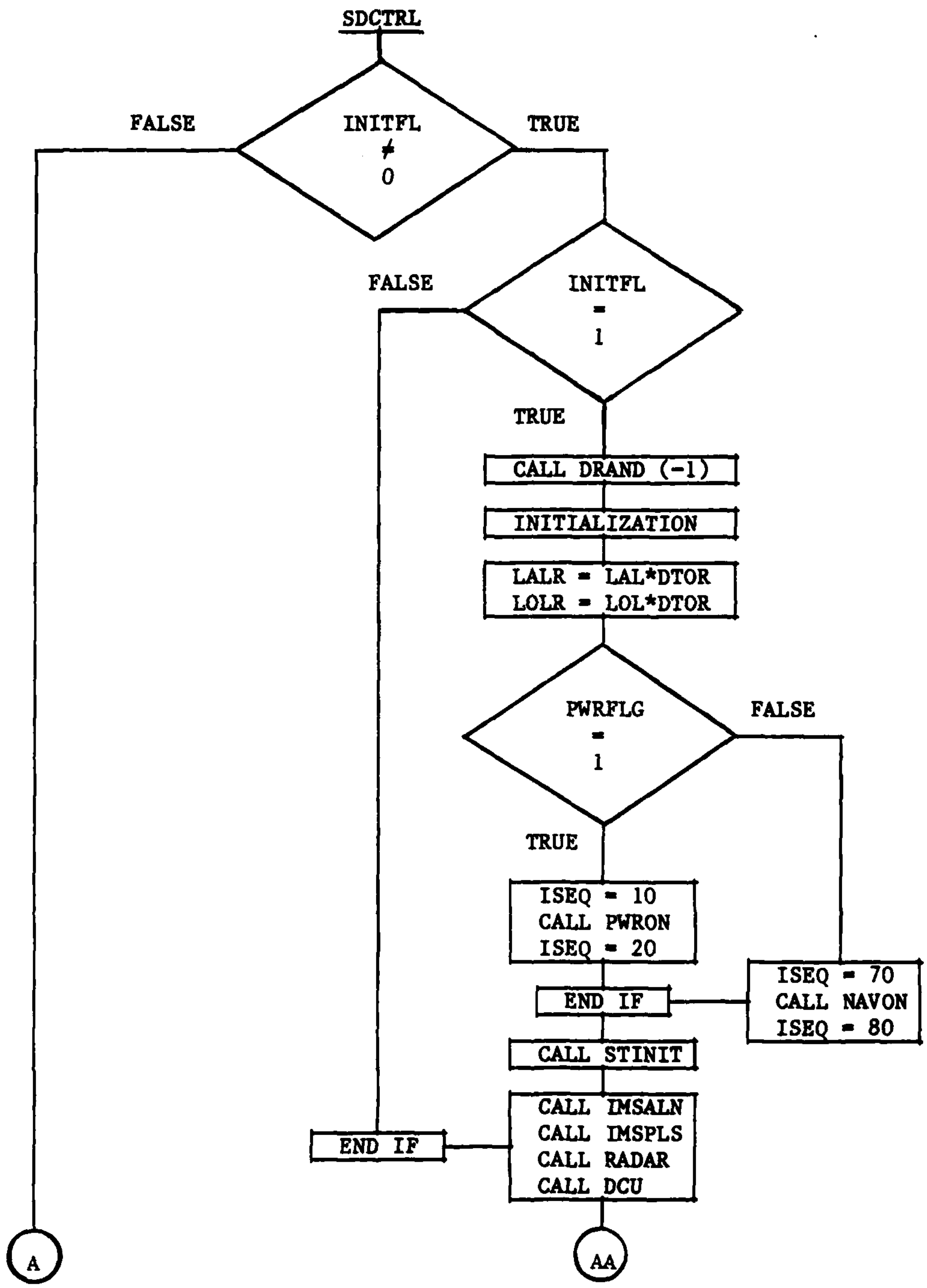

Figure H-1. Flowchart of SDCTRL Program.

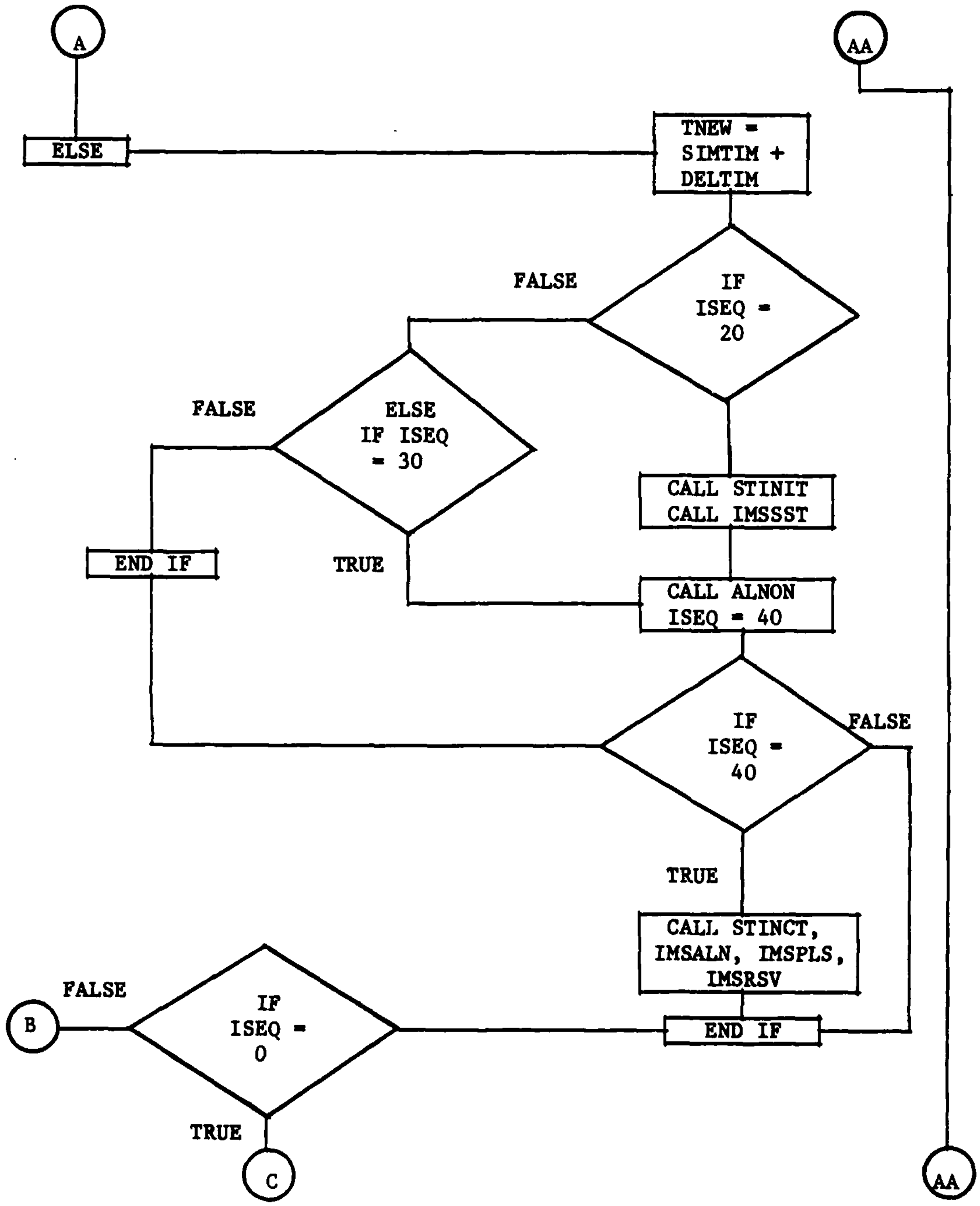

Figure H-1. Flowchart of SDCTRL Program (Continued).

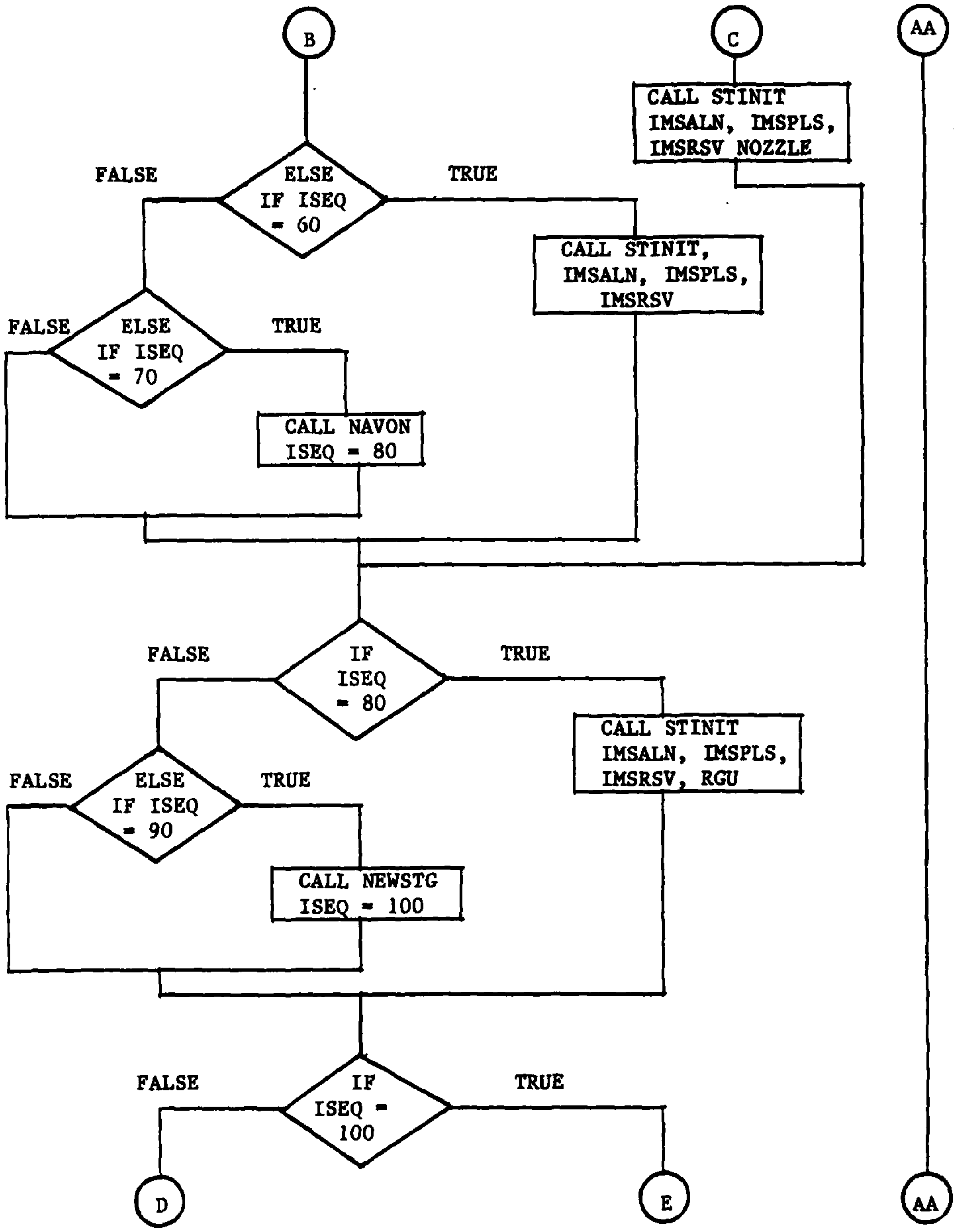

Figure H-1. Flowchart of SDCTRL Program (Continued).

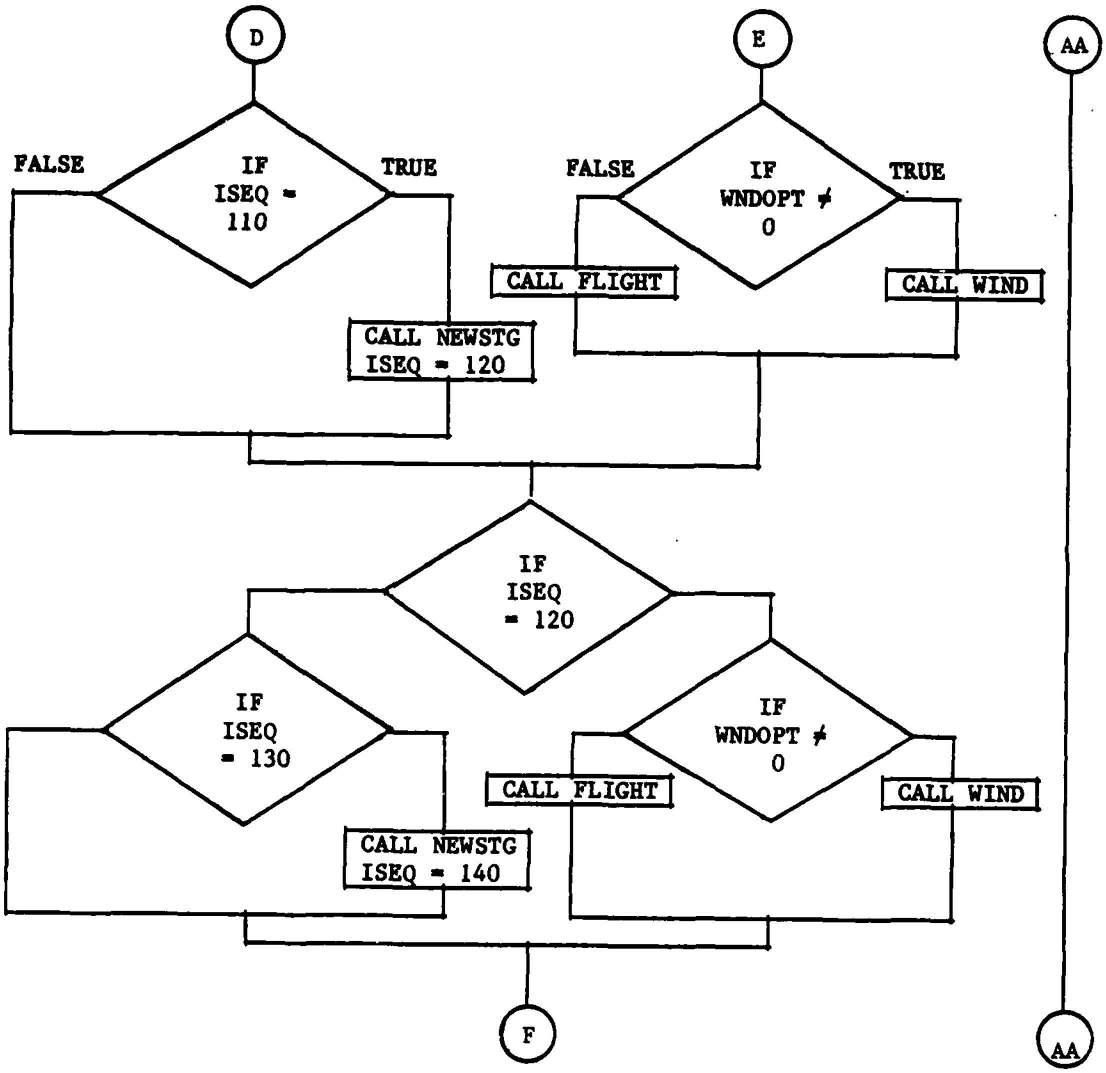

Figure H-1. Flowchart of SDCTRL Program (Continued).

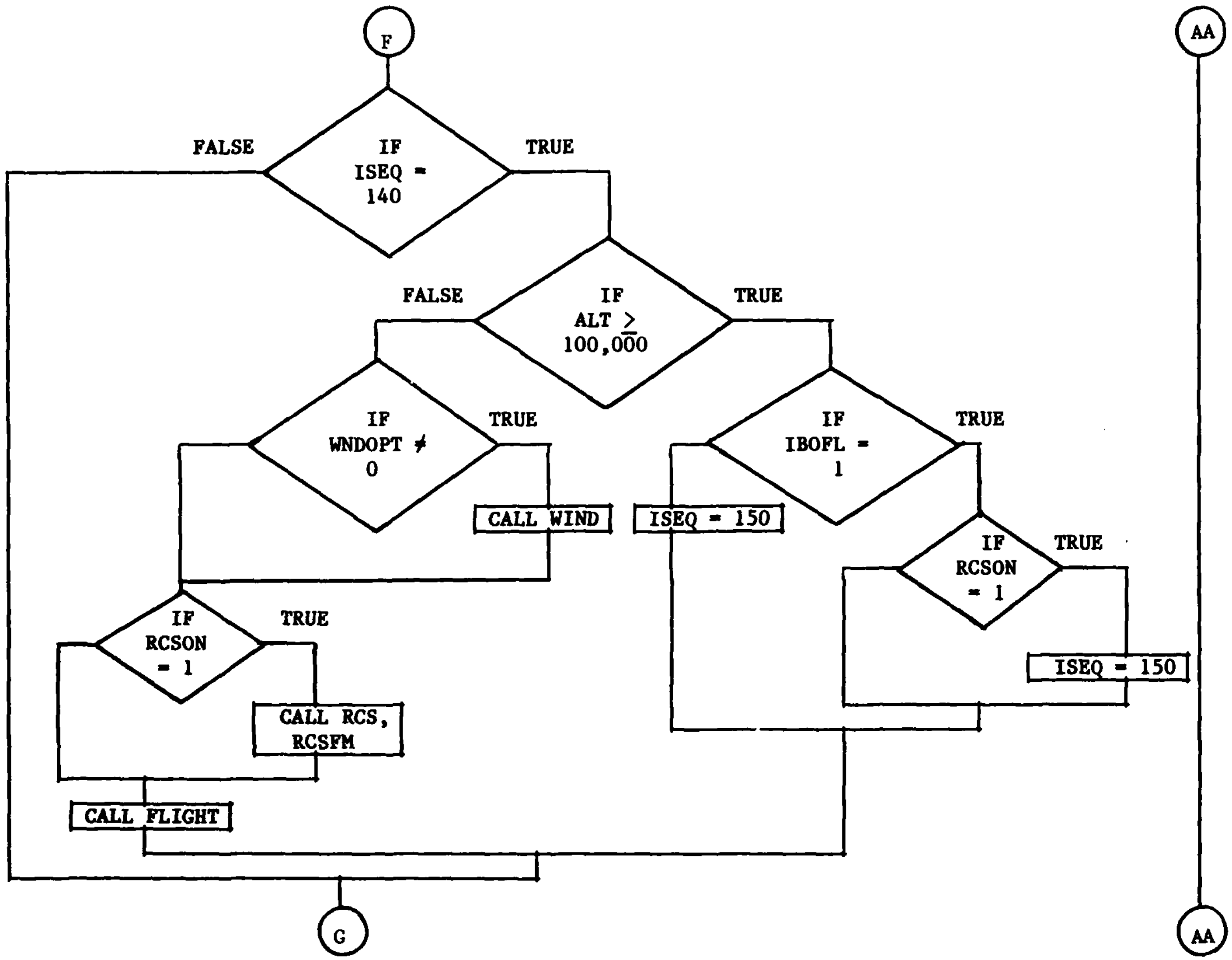

Figure H-1. Flowchart of SDCTRL Program (Continued).

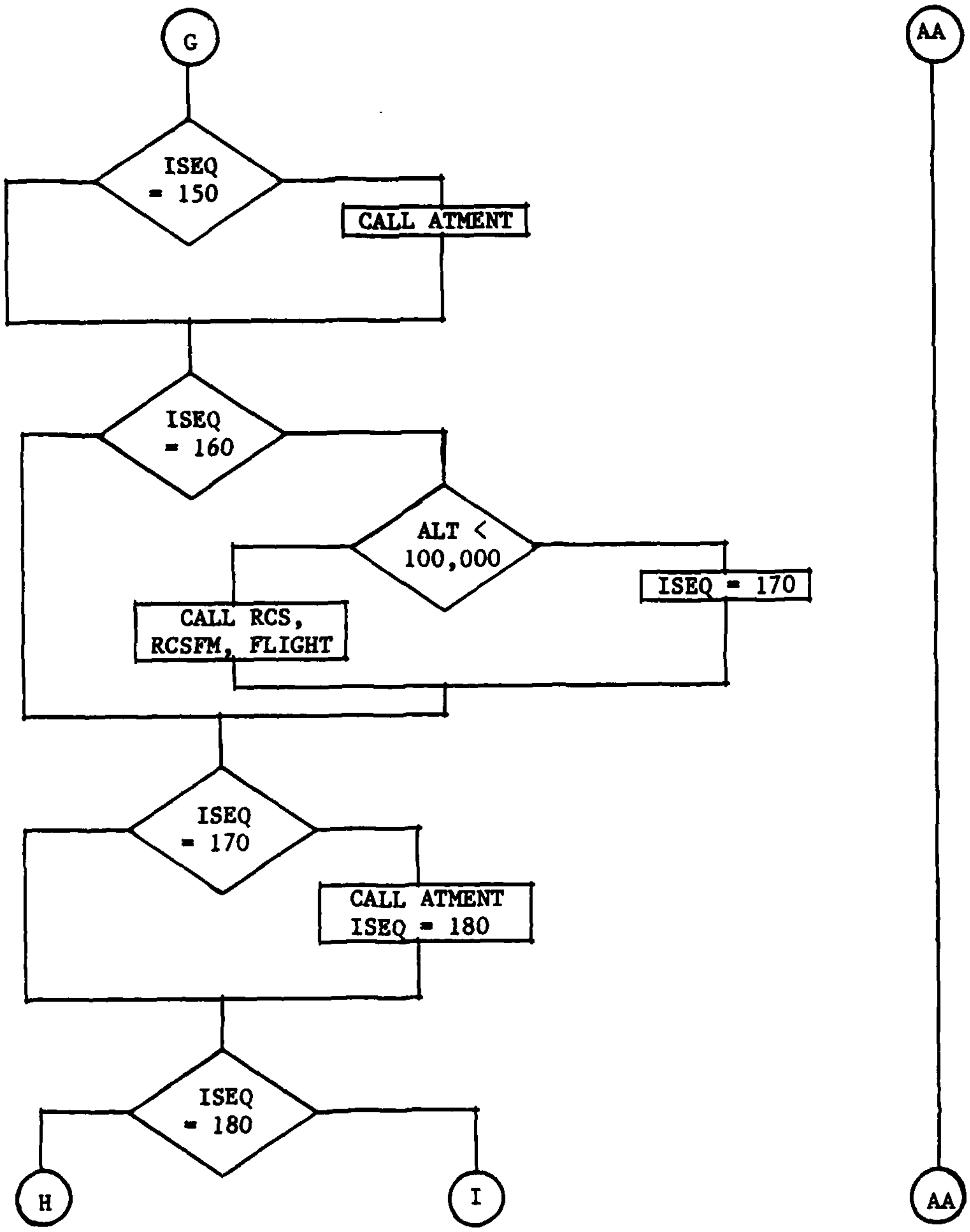

Figure H-1. Flowchart of SDCTRL Program (Continued).

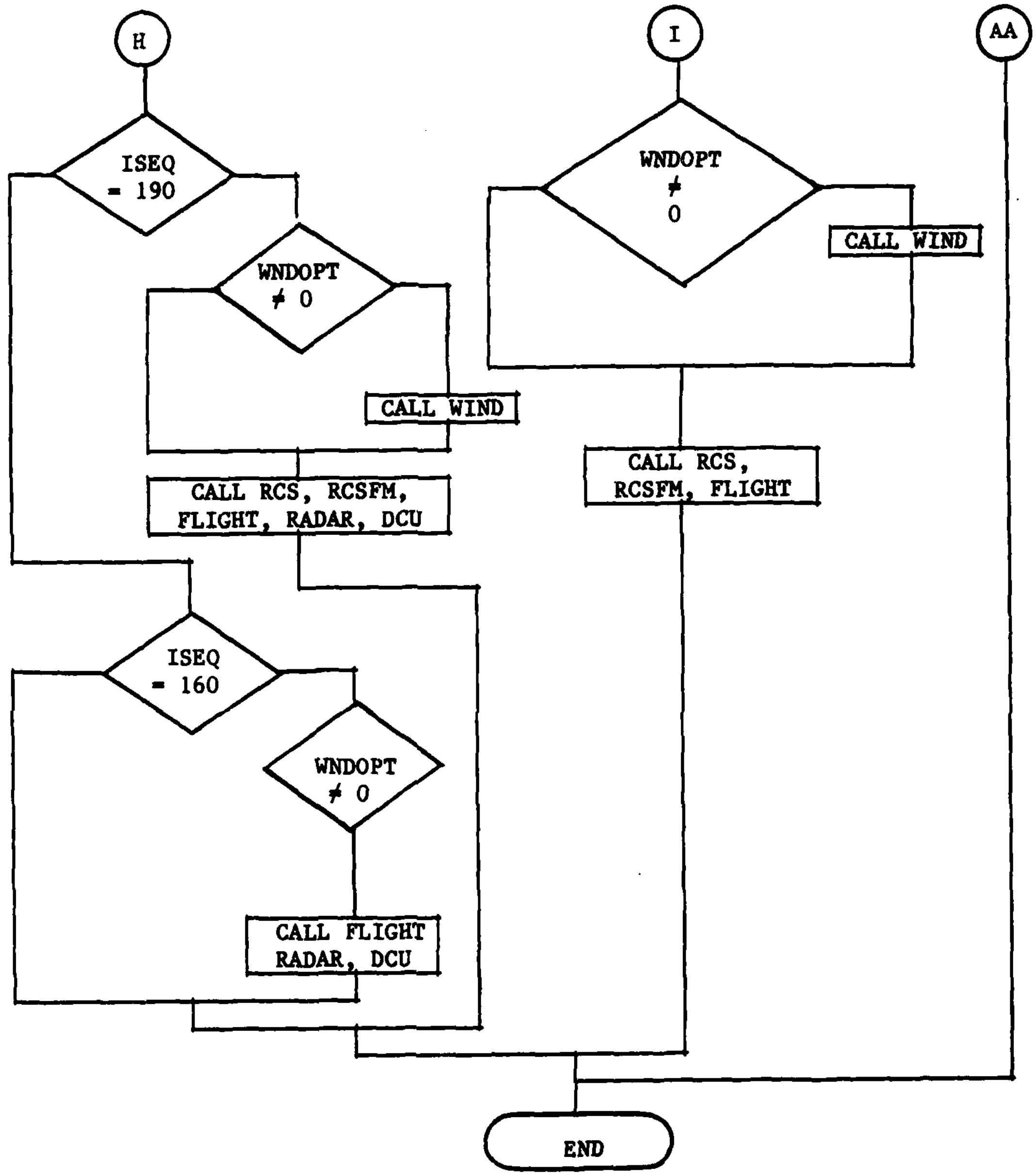

Figure H-1. Flowchart of SDCTRL Program (Continued).

APPENDIX I
TRW PROGRAM OUTPUT VARIABLES

TABLE I-1. TRW PROGRAM OUTPUT VARIABLES

```
AEARTH BIG01R(  1)EQUATORIAL EARTH RADIUS
ALT    BIG01R(  2)ALTITUDE
AOT    BIG01R(  3)ALTITUDE OF TARGET
DCM I  BIG01R(  4)INITIAL CM OFFSET
DELTIM BIG01R(  7)INCREMENT FROM LAST PASS TO BE ADDED SIMTIM
DER    BIG01R(  8)STATE VARIABLE DERIVATIVES
DLFSV  BIG01R( 20)FIRST STAGE VANE DEFL. ANGLES
DLRVV  BIG01R( 24)RV VANE DEFL. ANGLES
EFTODF BIG01R( 28)EARTH-FIXED TO DESIRED TRANSFORMATION MATRIX
EFTOMF BIG01R( 37)EARTH-FIXED TO MISSILE TRANSFORMATION MATRIX
EFTOND BIG01R( 46)EARTH-FIXED TO LOCAL NED TRANSFORMATION MATRIX
FRCSM  BIG01R( 55)RCS THRUST VECTOR IN MISSILE FRAME
GE     BIG01R( 60)GRAVITY VECTOR IN EARTH-FIXED FRAME
GM     BIG01R( 63)
HEAD   BIG01R( 66)AZIMUTH OF MISSILE LONGITUDINAL AXIS ON EL
LAL    BIG01R( 74)LATITUDE OF LAUNCH
LALR   BIG01R( 75)LATITUDE OF LAUNCH IN RADIANS
LOL    BIG01R( 76)LONGITUDE OF LAUNCH
LOLR   BIG01R( 77)LONGITUDE OF LAUNCH IN RADIANS
MDOT   BIG01R( 78)MAIN ENGINE MASS FLOW RATE
MFTOEF BIG01R( 79)MISSILE TO EARTH FIXED TRANSFORMATION MATRIX
MJDM   BIG01R( 88)JET DAMPING MOMENT VECTOR IN MISSILE FRAME
MRCSM  BIG01R( 91)RCS MOMENT VECTOR IN MISSILE FRAME
MTM    BIG01R( 94)PROPULSIVE THRUST MOMENT VECTOR MISSILE FRAME
NOZSTA BIG01R( 97)NOZZLE PIVOT POINT LOCATION-X COMPONENT
P      BIG01R( 99)ROLL, PITCH, AND YAW BODY RATES-P, Q, AND R
PR     BIG01R(102)ATMOSPHERIC PRESSURE
PSI    BIG01R(103)EULER ANGLES-PLATFORM TO MISSILE FRAME
RE     BIG01R(109)OBLATE EARTH RADIUS
TFRVS  BIG01R(117)TIME FROM RV SEPARATION
TM     BIG01R(119)BASIC PROPULSIVE THRUST VECTOR IN MISSILE FRAME
TNEW   BIG01R(122)TIME AT END OF CURRENT STEP (SIMTIM + DELTIM)
TOL    BIG01R(123)TIME OF LIFT-OFF
TORVS  BIG01R(124)TIME OF RV SEPARATION
TOSI   BIG01R(125)TIME OF SECOND STAGE IGNITION
U      BIG01R(126)VELOCITY IN MISSILE FRAME
WE     BIG01R(129)EARTH ROTATION RATE
XCM STA BIG01R(130)MISSILE TRANSIENT CM LOCATION
XE     BIG01R(131)POSITION IN EARTH-FIXED FRAME
XLE    BIG01R(134)LAUNCH POSITION VECTOR IN EARTH-FIXED FRAME
XTE    BIG01R(137)TARGET POSITION VECTOR IN EARTH-FIXED FRAME
ALFDOT BIG01R(140)TIME DERIVATIVE OF ALPHAM
ALPHAM BIG01R(141)PITCH PLANE COMPONENT OF ANGLE OF ATTACK
ALPHAT BIG01R(142)TOTAL ANGLE OF ATTACK
AMCDLX BIG01R(143)AERO. MOMENT COEFFICIENT TRANSFER DISTANCE
AMCSTA BIG01R(144)AERO. MOMENT COEFFICIENT REFERENCE LOCATION
AOL    BIG01R(145)ALTITUDE OF LAUNCH
AOL    BIG01R(145)ALTITUDE OF LAUNCH
AREF   BIG01R(146)AERO. REFERENCE AREA
BETAM  BIG01R(147)YAW PLANE COMPONENT OF ANGLE OF ATTACK
BETDOT BIG01R(148)TIME DERIVATIVE OF BETAM
```

TABLE I-1. TRW PROGRAM OUTPUT VARIABLES (Continued)

```
CAT     BIG01R(149)TOTAL AERO. AXIAL FORCE COEFFICIENT
CN      BIG01R(150)BASIC NORMAL FORCE COEFFICIENT
CN      BIG01R(150)PARAMETER FOR ACCELERATION LIMIT
CNSTAT  BIG01R(151)TOTAL STATIC NORMAL FORCE COEFFICIENT
CNT     BIG01R(152)TOTAL AERO. NORMAL FORCE COEFFICIENT
CPMT    BIG01R(153)TOTAL AERO. PITCHING MOMENT COEFFICIENT
CRMT    BIG01R(154)TOTAL AERO. ROLLING MOMENT COEFFICIENT
CS      BIG01R(155)SPEED OF SOUND
CY      BIG01R(156)BASIC SIDE FORCE COEFFICIENT
CYMT    BIG01R(157)TOTAL AERO. YAWING MOMENT COEFFICIENT
CYSTAT  BIG01R(158)TOTAL STATIC SIDE FORCE COEFFICIENT
CYT     BIG01R(159)TOTAL AERO. SIDE FORCE COEFFICIENT
DELRV   BIG01R(160)RV VANE DELFECTION USED IN INTERPOLATION
DELTAP  BIG01R(161)EQUIVALENT ROLL CONTROL VANE DEFL.
DELTAQ  BIG01R(162)EQUIVALENT PTICH CONTROL VANE DEFL.
DELTAR  BIG01R(163)EQUIVALENT YAW CONTROL VANE DEFL.
DLFSVP  BIG01R(164)FIRST STAGE ROLL CONTROL VANE DEFL.
DXDRL   BIG01R(165)TOTAL AERO. MOMENT COEF. TRANSFER DISTANCE
DYNPR   BIG01R(166)DYNAMIC PRESSURE
ELIPTE  BIG01R(167)ELLIPICITY OF EARTH
EPSAB   BIG01R(168)BIAS ERRORS IN TOTAL AERO. COEFFICIENTS
EPSAF   BIG01R(174)FRACTIONAL ERRORS IN TOTAL AERO. COEFFICIENTS
EPSCMB  BIG01R(180)BIAS ERROR IN LONGITUDINAL CM POSITION
EPSIB   BIG01R(181)BIAS ERRORS IN MOMENTS AND PRODUCTS OF INERTIA
EPSIF   BIG01R(187) ERRORS IN MOMENTS AND PRODUCTS OF INERTIA
EPSMB   BIG01R(193)MASS BIAS ERROR
EPSVEB  BIG01R(194)BIAS ERROR IN MOMENT COEF. DUE TO VANE DEFL.
EPSVEF  BIG01R(197)ERROR IN MOMENT COEF. DUE TO VANE DEFL.
FAM     BIG01R(200)AERO. FORCE VECTOR IN MISSILE FRAME
FWM     BIG01R(203)MISSILE WEIGHT VECTOR IN MISSILE FRAME
ITEN    BIG01R(206)INERTIA TENSOR
ITENDT  BIG01R(215)TIME RATE OF CHANGE OF MOMENTS OF INERTIA
LREF    BIG01R(219)AERO. REFERENCE LENGTH
MACH    BIG01R(220)MACH NUMBER
MAM     BIG01R(221)AERO. MOMENTS ABOUT MISSILE AXES
MASS    BIG01R(224)BOOST MASS ESTIMATE
MASS    BIG01R(224)VEHICLE MASS
MASSGG  BIG01R(225)MASS EXPELLED BY GAS GENERATOR
MDGG    BIG01R(226)GAS GENERATOR MASS FLOW RATE
PHIAP   BIG01R(228)AERO. ROLL ANGLE MEASURED IN A/P FRAME
PHIPR   BIG01R(229)AERO. ROLL ANGLE
PHIRV   BIG01R(230)MODIFIED AERO. ROLL ANGLE FOR INTERPOLATION
RHO     BIG01R(231)ATMOSPHERIC DENSITY
RLOIV   BIG01R(232)LREF / (2.0 * VAT)
RN      BIG01R(233)REYNOLDS NUMBER
RNTRNS  BIG01R(234)BOUNDARY LAYER TRANSITION REYNOLDS NUMBER
SUMFM   BIG01R(235)SUMMATION OF FORCES IN MISSILE FRAME
SUMMM   BIG01R(238)SUMMATION OF MOMENTS IN MISSILE FRAME
TFL     BIG01R(241)TIME FROM LIFT-OFF
TFSI    BIG01R(242)TIME FROM SECOND STAGE IGNITION
TOLD    BIG01R(243)
```

TABLE I-1. TRW PROGRAM OUTPUT VARIABLES (Continued)

```
VEF    BIG01R(244)MISSILE INERTIAL VEL. W.R.T. EARTH-FIXED FRAME
VAT    BIG01R(247)MAGNITUDE OF MISSILE VELOCITY W.R.T. AIR
VE     BIG01R(243)MISSILE EARTH-RELATIVE VELOCITY
VE     BIG01R(243)EARTH RELATIVE VELOCITY VECTOR
VI     BIG01R(249)MISSILE INERTIAL VELOCITY
VISC   BIG01R(250)COEFFICIENT OF VISCOSITY
VLAT   BIG01R(251)VEHICLE LATITUDE
VLON   BIG01R(252)VEHICLE LONGITUDE
WINDE  BIG01R(253)WIND VECTOR IN EARTH-FIXED FRAME
XEWE   BIG01R(231)EARTH RELATIVE VELOCITY COMPONENTS
AEXIT  BIG01R(309)NOZZLE EXIT AREA
D1P    BIG01R(310)FIRST STAGE NOZZLE PITCH COMMAND
D1Y    BIG01R(311)FIRST STAGE NOZZLE YAW COMMAND
D2P    BIG01R(312)SECOND STAGE NOZZLE PITCH COMMAND
D2Y    BIG01R(313)SECOND STAGE NOZZLE YAW COMMAND
D31    BIG01R(314)RV VANE 1 COMMAND
D32    BIG01R(315)RV VANE 2 COMMAND
D33    BIG01R(316)RV VANE 3 COMMAND
D34    BIG01R(317)RV VANE 4 COMMAND
DCJ    BIG01R(318)DIRECTION COSINES OF THE RCS THRUST VECTORS
DLFSVD BIG01R(342)FIRST STAGE VANE ANGULAR RATES
DLNP   BIG01R(346)NOZZLE PITCH ANGLE
DLNPD  BIG01R(347)NOZZLE PITCH ANGULAR RATE
DLNY   BIG01R(348)NOZZLE YAW ANGLE
DLNYD  BIG01R(349)NOZZLE YAW ANGULAR RATE
DLRVVD BIG01R(350)RV VANE ANGULAR RATES
DRC    BIG01R(354)FIRST STAGE VANE ROLL COMMAND
DTRCS  BIG01R(355)RCS JET TIME DELAY
EPSTBS BIG01R(356)THRUST BIAS ERROR
EPSTRC BIG01R(357)FRACTIONAL ERRORS IN RCS THRUST MAGNITUDE
EPSTSF BIG01R(365)FRACTIONAL ERROR IN THRUST
FRCS   BIG01R(366)NOMINAL MAGNITUDE OF RCS THRUST FOR ITH JET
GMU    BIG01R(374)GRAVITY PARAMETER
J2     BIG01R(375)SECOND ORDER ZONAL HARMONIC
J3     BIG01R(376)THIRD ORDER ZONAL HARMONIC
J4     BIG01R(377)FOURTH ORDER ZONAL HARMONIC
LAMNZ  BIG01R(378)
LJD    BIG01R(379)CHARACTERISTIC JET DAMPING LENGTH POSITION
MAJ    BIG01R(382)MOMENT ARM VECTORS OF THE RCS JETS (8 JETS)
NLIMFS BIG01R(406)FIRST STAGE NOZZLE DEFL. ANGLE LIMIT
NLIMSS BIG01R(407)SECOND STAGE NOZZLE DEFL. ANGLE LIMIT
PHINZ  BIG01R(408)
PSLV   BIG01R(409)SEA LEVEL PRESSURE
RCSDLX BIG01R(410)RCS MOMENT ARM COMPONENT ALONG X AXIS
RCSSTA BIG01R(411)RCS LOCATION-X COMPONENT
TNOZ   BIG01R(420)TIME AT WHICH LAST NOZZLE ANGLES COMPUTED
TVANES BIG01R(421)TIME AT WHICH LAST VANE ANGLES COMPUTED
VLIMFS BIG01R(422)FIRST STAGE VANE DEFL. ANGLE LIMIT
```

TABLE I-1. TRW PROGRAM OUTPUT VARIABLES (Continued)

```
VLIMRV BIG01R(423)RV VANE DEFL. ANGLE LIMIT
WFSV   BIG01R(424)FIRST STAGE VANE NATURAL FREQUENCY
WNP    BIG01R(425)NOZZLE PITCH NATURAL FREQUENCY
WNY    BIG01R(426)NOZZLE YAW NATURAL FREQUENCY
WRVV   BIG01R(427)RV VANE NATURAL FREQUENCY
XNOZ   BIG01R(428)INITIAL NOZZLE POSITION WITH RESPECT TO CM
ZETANP BIG01R(431)NOZZLE PITCH DAMPING FACTOR
ZETANY BIG01R(432)NOZZLE YAW DAMPING FACTOR
ZETFSV BIG01R(433)FIRST STAGE VANE DAMPING FACTOR
ZETRVV BIG01R(434)RV VANE DAMPING FACTOR
ABIAS  BIG01R(453)ACCELEROMETER BIASES
AGP    BIG01R(456)GRAVITATIONAL ACCELERATION VECTOR
AGPOLD BIG01R(459)PREVIOUS GRAVITATIONAL ACCELERATION VECTOR
ARO    BIG01R(462)AZIMUTH (YAW)RESOLVER OFFSET
ARO    BIG01R(462)AZIMUTH (YAW)RESOLVER OFFSET
DDYX   BIG01R(465)Y TO X NONORTHOGONALITY PRESET
DDYX   BIG01R(465)Y TO X NONORTHOGONALITY PRESET
DELYX  BIG01R(466)NONORTHOGONALITY BETWEEN Y AND X ACCELEROMETERS
DELZX  BIG01R(467)NONORTHOGONALITY BETWEEN Z AND X ACCELEROMETERS
DELZY  BIG01R(468)NONORTHOGONALITY BETWEEN Z AND Y ACCELEROMETERS
DF     BIG01R(469)GYRO RESTRAINT DRIFTS-X, Y, AND Z
DI     BIG01R(472)INPUT MASS UNBALANCE DRIFTS-X, Y, AND Z
DOZ    BIG01R(475)Z GYRO ACCELERATION ALONG X CLUSTER
DO     BIG01R(476)OUTPUT MASS UNBALANCE DRIFTS-X, Y, AND Z
EIMS   BIG01R(479)EULER ANGLES-TRUE IMS PLATFORM TO PLATFORM
EPSBS  BIG01R(482)FRACTIONAL ERROR ACCELEROMETER BIAS ESTIMATE
EPSCBG BIG01R(485)GIMBAL ANGLE BIASES FOR COSINE RESOLVERS
EPSCFG BIG01R(488)FRACTIONAL ERROR IN IMS COSINE RESOLVER MEASURE
EPSFD  BIG01R(491)ERROR IN PLATFORM FIXED DRIFT ESTIMATES
EPSGD  BIG01R(494)FRACTIONAL ERROR IN G-SENSITIVE DRIFT PARTIALS
EPSNSF BIG01R(497)ERROR IN NEG. ACCELEROMETER SCALE FACTOR
EPSORT BIG01R(500)ERROR IN ACCELEROMETER NONORTHOGONALTIY
EPSPSF BIG01R(503)ERROR IN POS. ACCELEROMETER SCALE FACTOR
EPSSBG BIG01R(506)GIMBAL ANGLE BIASES FOR SINE RESOLVERS
EPSSFG BIG01R(509)FRACTIONAL ERROR IN IMS SINE RESOLVER MEASURE
EPSTD  BIG01R(512)ERROR IN TORQUER SLEW RATES
IMSTIM BIG01R(516)IMS TEST TIMER
IRTOIF BIG01R(517)INERTIAL REFERENCE TO INERTIAL EARTH TRANS.
IRTOPF BIG01R(526)INERTIAL REFERENCE TO PLATFORM TRANS. MATRIX
KI     BIG01R(535)ACCELEROMETER LOW GAIN SCALE ERRORS-XYZ
KIH    BIG01R(538)ACCELEROMETER HIGH GAIN SCALE ERRORS-XYZ
KNHSF  BIG01R(541)NOMINAL NEGATIVE HIGH GAIN ACCELEROMETER SCALE
KNLSF  BIG01R(542)NOMINAL NEGATIVE LOW GAIN ACCELEROMETER SCALE
KO     BIG01R(543)ACCELEROMETER LOW GAIN BIASES-X,Y,AND Z
KOH    BIG01R(546)ACCELEROMETER HIGH GAIN BIASES-X, Y, AND Z
KPHSF  BIG01R(549)NOMINAL POSITIVE HIGH GAIN ACCELEROMETER SCALE
KPLSF  BIG01R(550)NOMINAL POSITIVE LOW GAIN ACC. SCALE FACTOR
KS     BIG01R(551)ACCELEROMETER LOW GAIN SYMMETRY ERRORS-XYZ
KSH    BIG01R(554)ACCELEROMETER HIGH GAIN SYMMETRY ERRORS-XYZ
KT     BIG01R(557)GYRO TORQUER SCALE FACTOR RATIO
LAT    BIG01R(560)LATITUDE OF TARGET
```

TABLE I-1. TRW PROGRAM OUTPUT VARIABLES (Continued)

LOT	BIG01R(561)LONGITUDE OF TARGET
MFTOIR	BIG01R(562)MISSILE TO INERTIAL REFERENCE TRANS. MATRIX
MFTOND	BIG01R(571)MISSILE TO LOCAL NED TRANS. MATRIX
MFTOPF	BIG01R(580)MISSILE TO PLATFORM TRANS. MATRIX
PFTOIR	BIG01R(596)PLATFORM TO INERTIAL REFERENCE TRANS. MATRIX
PRO	BIG01R(605)PITCH RESOLVER OFFSET
PSIDTP	BIG01R(606)PREVIOUS VALUE OF PRE-LAUNCH EULER ANGLE RATES
RCPCH	BIG01R(609)COSINE IMS PITCH
RCROL	BIG01R(610)COSINE IMS ROLL
RCYAW	BIG01R(611)COSINE IMS YAW
REL	BIG01R(612)LAUNCH LOCAL EARTH RADIUS
RET	BIG01R(613)TARGET LOCAL EARTH RADIUS
RHG	BIG01R(614)HIGH GAIN ACCELEROMETER DRIFT-X AND Y
RLG	BIG01R(616)LOW GAIN ACCELEROMETER DRIFT-X AND Y
RRO	BIG01R(618)ROLL RESOLVER OFFSET
RSPCH	BIG01R(619)SINE IMS PITCH
RSROL	BIG01R(620)SINE IMS ROLL
RSYAW	BIG01R(621)SINE IMS YAW
SIGDLV	BIG01R(622)STANDARD DEVIATION OF DELTA V SENSED BY IMS
SIGRD	BIG01R(623)DEVIATING PLATFORM DRIFT FROM RANDOM ERRORS
SIGRGA	BIG01R(624)STANDARD DEVIATION OF IMS RESOLVER MEASURE
VIR	BIG01R(626)MISSILE VELOCITY IN INERTIAL REFERENCE FRAME
VP	BIG01R(629)MISSILE VELOCITY IN PLATFORM FRAME
VPOLD	BIG01R(632)PREVIOUS VELOCITY VECTOR SENSED BY IMS
ALTP	BIG01R(635)PREVIOUS ALT FROM DCU ERROR MODEL
APHA	BIG01R(636)TRANSFER FUNCTION -ROLL ANTENNA ANGLE
APSA	BIG01R(637)TRANSFER FUNCTION -YAW ANTENNA ANGLE
ATHA	BIG01R(638)TRANS. FUNCTION -PITCH ANTENNA ANGLE
BPE	BIG01R(639)EAST ERROR PARAMETER FROM DCU ERROR MODEL
BPHA	BIG01R(640)TRANS. FUNCTION -ROLL ANTENNA ANGLE
BPN	BIG01R(641)NORTH ERROR PARAMETER FROM DCU ERROR MODEL
BPSA	BIG01R(642)TRANS. FUNCTION -YAW ANTENNA ANGLE
BTHA	BIG01R(643)TRANS. FUNCTION -PITCH ANTENNA ANGLE
CP	BIG01R(644)UPDATED INERTIAL POSITION-N, E, AND D(PZ)
D	BIG01R(647)ERROR IN POSITION-NORTH, EAST, AND DOWN
DTACQ	BIG01R(650)TIME SINCE START OF DCU ACQUISITION ATTEMPT
EPSBGR	BIG01R(651)MISSILE BODY RATE MEASUREMENT BIAS ERROR
EPSCER	BIG01R(654)RU GIMBAL BIASES FOR COSINE RESOLVERS
EPSCFR	BIG01R(657)FRACTIONAL ERROR IN RU COSINE RESOLVER MEASURE
EPSFGR	BIG01R(660)FRACTIONAL ERROR IN MISSILE BODY RATE MEASURE
EPSSER	BIG01R(663)RU GIMBAL ANGLE BIASES FOR SINE RESOLVERS
EPSSFR	BIG01R(666)FRACTIONAL ERROR IN RU SINE RESOLVER MEASURE
ERE	BIG01R(669)DCU ERROR EAST
ERN	BIG01R(670)DCU ERROR NORTH
KPHA	BIG01R(674)TRANSFER FUNCTION CONSTANT-ROLL ANT. ANGLE
KPSA	BIG01R(675)TRANSFER FUNCTION CONSTANT-YAW ANT. ANGLE
KTHA	BIG01R(676)TRANSFER FUNCTION CONSTANT-PITCH ANT. ANGLE
MQ	BIG01R(673)MATCH QUALITY
R1	BIG01R(680)FIRST STAGE ROLL RATE
R2	BIG01R(681)SECOND STAGE ROLL RATE
R3	BIG01R(682)SINGLE STAGE ROLL RATE

TABLE I-1. TRW PROGRAM OUTPUT VARIABLES (Continued)

```
P4      BIG01R(683)RV EP/AB/SB ROLL RATE
P5      BIG01R(684)RV EP ROLL RATE
P6      BIG01R(685)RV AB/SB ROLL RATE
PHC     BIG01R(686)SAU ROLL DRIVE
PSC     BIG01R(687)SAU YAW DRIVE
Q1      BIG01R(688)FIRST STAGE PITCH RATE
Q2      BIG01R(689)SECOND STAGE PITCH RATE
Q3      BIG01R(690)SINGLE STAGE PITCH RATE
Q4      BIG01R(691)RV EP/AB/SB PITCH RATE
Q5      BIG01R(692)RV EP PITCH RATE
Q6      BIG01R(693)RV AB/SB PITCH RATE
R1      BIG01R(694)FIRST STAGE YAW RATE
R2      BIG01R(695)SECOND STAGE YAW RATE
R3      BIG01R(696)SINGLE STAGE YAW RATE
R4      BIG01R(697)RV EP/AB/SB YAW RATE
R5      BIG01R(698)RV EP YAW RATE
R6      BIG01R(699)RV AB/SB YAW RATE
RA      BIG01R(700)EULER ANGLES-ANTENNA TO MISSILE FRAME
RCAPH   BIG01R(703)COSINE ANTENNA PITCH
RCARL   BIG01R(704)COSINE ANTENNA ROLL
RCAYW   BIG01R(705)COSINE ANTENNA YAW
RSAPH   BIG01R(707)SINE ANTENNA PITCH
RSARL   BIG01R(708)SINE ANTENNA ROLL
RSAYW   BIG01R(709)SINE ANTENNA YAW
SIGRGR  BIG01R(710)STANDARD DEVIATION OF MISSILE BODY RATE MEASURE
SIGRRA  BIG01R(711)STANDARD DEVIATION OF RJ RESOLVER MEASUREMENTS
THC     BIG01R(712)SAU PITCH DRIVE
TMXACQ  BIG01R(713)MAXIMUM TIME FOR ACQUISITION BY DCU IN ALTITUDE
TRUON   BIG01R(714)TIME AT WHICH RADAR UNIT TURNED ON
CAPON   BIG01R(715)BASIC AXIAL FORCE COEFFICIENT (POWER ON)
CPM     BIG01R(716)BASIC PITCHING MOMENT COEFFICIENT
CPMQ    BIG01R(717)PITCHING MOMENT Q DAMPING DERIVATIVE
CRM     BIG01R(718)BASIC ROLLING MOMENT COEFFICIENT
CRMP    BIG01R(719)BASIC ROLLING MOMENT DAMPING DERIVATIVE
CYM     BIG01R(720)BASIC YAWING MOMENT COEFFICIENT
CYMR    BIG01R(721)YAWING MOMENT R DAMPING DERIVATIVE
DCAAE   BIG01R(722)INCREMENTAL CA DUE TO AEROELAS. EFFECTS
DCADP   BIG01R(723)INCREMENTAL CA DUE TO DELTAP VANE DEFL.
DCADQ   BIG01R(724)INCREMENTAL CA DUE TO DELTAQ VANE DEFL.
DCADR   BIG01R(725)INCREMENTAL CA DUE TO DELTAR VANE DEFL.
DCAPCF  BIG01R(726)ADDITIVE BASE PRESSURE AXIAL FORCE COEF.
DCASF   BIG01R(727)SKIN FRICTION AXIAL FORCE COEFFICIENT
DCAVM   BIG01R(728)INCREMENTAL CA DUE TO VANE MISALIGNMENT
DCNAE   BIG01R(729)INCREMENTAL CN DUE TO AEROELAS. EFFECTS
DCNDP   BIG01R(730)INCREMENTAL CN DUE TO DELTAP VANE DEFL.
DCNDQ   BIG01R(731)INCREMENTAL CN DUE TO DELTAQ VANE DEFL.
DCNDR   BIG01R(732)INCREMENTAL CN DUE TO DELTAR VANE DEFL.
DCNVM   BIG01R(733)INCREMENTAL CN DUE TO VANE MISALIGNMENT
DCPMAE  BIG01R(734)INCREMENTAL CPM DUE TO AEROELAS. EFFECTS
DCPMDP  BIG01R(735)INCREMENTAL CPM DUE TO DELTAP VANE DEFL.
DCPMDQ  BIG01R(736)INCREMENTAL CPM DUE TO DELTAQ VANE DEFL.
```

TABLE I-1. TRW PROGRAM OUTPUT VARIABLES (Continued)

DCPMDR	BIG01R(737)	INCREMENTAL CPM DUE TO DELTAR VANE DEFL.
DCPMVM	BIG01R(738)	INCREMENTAL CPM DUE TO VANE MISALIGNMENT
DCRMAE	BIG01R(739)	INCREMENTAL CRM DUE TO AEROELAS. EFFECTS
DCRMDP	BIG01R(740)	INCREMENTAL CRM DUE TO DELTAP VANE DEFL.
DCRMDQ	BIG01R(741)	INCREMENTAL CRM DUE TO DELTAQ VANE DEFL.
DCRMDR	BIG01R(742)	INCREMENTAL CRM DUE TO DELTAR VANE DEFL.
DCRMVM	BIG01R(743)	INCREMENTAL CRM DUE TO VANE MISALIGNMENT
DCYAE	BIG01R(744)	INCREMENTAL CY DUE TO AEROELAS. EFFECTS
DCYDP	BIG01R(745)	INCREMENTAL CY DUE TO DELTAP VANE DEFL.
DCYDQ	BIG01R(746)	INCREMENTAL CY DUE TO DELTAQ VANE DEFL.
DCYDR	BIG01R(747)	INCREMENTAL CY DUE TO DELTAR VANE DEFL.
DCYMAE	BIG01R(748)	INCREMENTAL CYM DUE TO AEROELAS. EFFECTS
DCYMDP	BIG01R(749)	INCREMENTAL CYM DUE TO DELTAP VANE DEFL.
DCYMDQ	BIG01R(750)	INCREMENTAL CYM DUE TO DELTAQ VANE DEFL.
DCYMDR	BIG01R(751)	INCREMENTAL CYM DUE TO DELTAR VANE DEFL.
DCYMVM	BIG01R(752)	INCREMENTAL CYM DUE TO VANE MISALIGNMENT
GAME	BIG01R(757)	EARTH-RELATIVE FLIGHT ANGLE WRT HORIZONTAL
GAMT	BIG01R(758)	ELEVATION OF LOS TO TARGET WRT LOCAL HOR.
GAMX	BIG01R(759)	ELEVATION MISSILE X-AXIS WRT LOCAL HOR.
GRL	BIG01R(760)	GROUND RANGE LAUNCH POINT TO CURRENT POS.
GRT	BIG01R(761)	GROUND RANGE FROM CURRENT POSITION TO TARGET
NEDDOT	BIG01R(763)	EARTH-RELATIVE VELOCITY IN LOCAL NED FRAME
SIGE	BIG01R(771)	EARTH-RELATIVE FLIGHT PATH ANGLE WRT NORTH
SIGT	BIG01R(772)	AZIMUTH OF LINE-OF-SIGHT TO TARGET
SIGX	BIG01R(773)	AZIMUTH OF MISSILE X AXIS
SRT	BIG01R(774)	SLANT RANGE TO TARGET
UREWM	BIG01R(775)	
VRELM	BIG01R(780)	VEL. OF MISSILE RELATIVE TO AIR IN MISSILE
XD	BIG01R(784)	MISSILE POS. VECTOR IN DESIRED (TARGET) FRAME
XEDOT	BIG01R(787)	MISSILE EARTH-RELATIVE VELOCITY IN EARTH FRAME
XTME	BIG01R(790)	POSITION OF MISSILE RELATIVE TO TARGET IN E
DC	BIG01R(818)	FIXED DRIFTS-X ,Y, AND Z
DLV	BIG01R(821)	DELTA MAGNITUDE OF INERTIAL VELOCITY
DSMAT	BIG01R(824)	G-SENSITIVE DRIFT PARTIAL DERIVATIVE MATRIX
FDIMS	BIG01R(833)	IMS FIXED DRIFT RATE
NSF	BIG01R(836)	IMS NEGATIVE SCALE FACTORS-X, Y, AND Z
PSF	BIG01R(839)	IMS POSITIVE SCALE FACTORS-X, Y, AND Z
RSDV	BIG01R(842)	IMS RESIDUAL DELTA V MEASUREMENT
RYX	BIG01R(845)	IMS AXIS NONORTHOGONALITY-Y TO X
RZX	BIG01R(846)	IMS AXIS NONORTHOGONALITY-Z TO X
RZY	BIG01R(847)	IMS AXIS NONORTHOGONALITY-Z TO Y
TQMAG	BIG01R(848)	MAGNITUDE OF IMS TORQUER RATE
IXX	BIG01R(849)	
IYY	BIG01R(850)	
IZZ	BIG01R(851)	
MOLD	BIG01R(852)	
NOZDLX	BIG01R(853)	NOZZLE MOMENT ARM COMPONENT ALONG X AXIS
TRCSON	BIG01R(855)	TIME AT WHICH RCS JET WAS TURNED ON
RAD	BIG01R(863)	EULER ANGLE RATES-ANTENNA TO MISSILE FRAME
RADD	BIG01R(866)	EULER ANGLE ACCS.-ANTENNA TO MISSILE FRAME
TACQ	BIG01R(870)	TIME REQUIRED FOR DCU ACQUISITION

TABLE I-1. TRW PROGRAM OUTPUT VARIABLES (Continued)

```
TOCPI  BIG01R(371)TIME AT WHICH CPI COMMAND WAS RECEIVED
FSIFLG BIG01I( 58)FIRST STAGE IGNITION FLAG
FSSFLG BIG01I( 59)FIRST STAGE SEPARATION FLAG
ID1    BIG01I( 67) 0   INPUT DISCRETE WORD 1
ID2    BIG01I( 68) 0   INPUT DISCRETE WORD 2
ID3    BIG01I( 69) 0   INPUT DISCRETE WORD 3
IDER   BIG01I( 70)DERIVATIVE COMPUTATION COUNTER
INITFL BIG01I( 71)INITIALIZATION FLAG
ISEQ   BIG01I( 72)FLIGHT SEQUENCE INDICATOR
ISTAGE BIG01I( 73)STAGE FLAG;=1 LAUNCH AND STAGE;=0 OTHERWISE
OD2    BIG01I( 98) 0   OUTPUT DISCRETE WORD 2
RVSFLG BIG01I(110)RV SEPARATION FLAG
SSIFLG BIG01I(113)SECOND STAGE IGNITION FLAG
SSSFLG BIG01I(114)SECOND STAGE SEPARATION FLAG
THROPT BIG01I(113)THRUST OPTION (1-USE TSLV;  2-USE TVAC)
JDFLG  BIG01I(213)JET FLAG (0=NO JD;  1-JD)
MSTOPT BIG01I(227)MASS TABLE OPTION FLAG
RVJ    BIG01I(412)REACTION JET-ON FLAGS ([illegible] JETS)
ARMCOD BIG01I(435)CHEFU ARM CODE
BACODE BIG01I(436)BATTERY ACTIVATE CODE
BAFLG  BIG01I(437)BATTERY ACTIVATED FLAG
COWRD  BIG01I(438)SAF REQUIRED CUTOFF WORD
EPWRD  BIG01I(439)SAF REQUIRED EP WORD
FIRCOD BIG01I(440)CHEFU FIRE CODE
FSAAFL BIG01I(441)FIRST STAGE IGNITION S AND A ARM FLAG
FSACOD BIG01I(442)FIRST STAGE IGNITION S AND A CODE
IBCFL  BIG01I(443)BOOST CONFIG. (1-SINGLE,  2-TWO STAGE)
IFAWRD BIG01I(444)SAF REQUIRED INFLIGHT ARMING WORD
LOWRD  BIG01I(445)SAF REQUIRED LIFT-OFF WORD
OD1    BIG01I(446) 0   OUTPUT DISCRETE WORD 1
PACNF  BIG01I(447)PAC NO FAULT FLAG
RVSWRD BIG01I(448)SAF REQUIRED RV SEPARATION WORD
TSIWRD BIG01I(449)SAF REQUIRED TERMINAL SEQUENCE INITIATE WORD
WHAFLG BIG01I(450)WARHEAD ARMED FLAG
WHBFLG BIG01I(451)WARHEAD BURST FLAG
WHT    BIG01I(452)WARHEAD OPTION
CLRDV  BIG01I(463)CLEAR IMS DELTA-V COUNTERS FLAG
DAYSC  BIG01I(464)NUMBER OF DAYS SINCE LAST IMS CALIBRATION
ID4    BIG01I(515) 0   INPUT DISCRETE WORD 4
NDLV   BIG01I(559)NEGATIVE PULSE COUNT-X, Y, AND Z
OD4    BIG01I(592) 0   OUTPUT DISCRETE WORD 4
PDLV   BIG01I(593)POSITIVE PULSE COUNT-X, Y, AND Z
TRQTST BIG01I(625)IMS TORQUER BITE TEST FLAG
ID5    BIG01I(671) 0   INPUT DISCRETE WORD 5
ID6    BIG01I(672) 0   INPUT DISCRETE WORD 6
INITDE BIG01I(673)DCU ERROR MODEL INITIALIZATION FLAG
MNFCNT BIG01I(677)MAJOR FRAME COUNTER
OD5    BIG01I(679) 0   OUTPUT DISCRETE WORD 5
*FDAFL BIG01I(705)FIX DATA READ BY PACP-FLAG
ANTON  BIG01I(754)ANTENNA ON FLAG-STABILIZE ANTENNA RECEIVED
DCUON  BIG01I(755)DCU ON FLAG
```

TABLE I-1. TRW PROGRAM OUTPUT VARIABLES (Continued)

```
FHYDON BIGO1I(758)FIRST STAGE HYDRAULIC SYSTEM ON FLAG
NAVFLG BIGO1I(762)BEGIN NAVIGATION FLAG
OD3    BIGO1I(766) O   OUTPUT DISCRETE WORD 3
PWRFLG BIGO1I(767)MISSILE POWER-ON FLAG
RCSON  BIGO1I(768)REACTION CONTROL SYSTEM ON FLAG
RUON   BIGO1I(769)RADAR UNIT ON FLAG
SHYDON BIGO1I(770)SECOND STAGE HYDRAULIC SYSTEM ON FLAG
VCSGG1 BIGO1I(778)VCS GAS GENERATOR NUMBER 1 INITIATE FLAG
VCSGG2 BIGO1I(779)VCS GAS GENERATOR NUMBER 2 INITIATE FLAG
WNDOPT BIGO1I(783)WIND OPTION; 0-NO WINDS, 1-NEV, 2-HAV
BAWRD  BIGO1I(793)BATTERY ACTIVATE WORD FROM PACP
COFLG  BIGO1I(794)SAF CUTOFF FLAG
EPFLG  BIGO1I(795)SAF EARTH PENETRATOR FLAG
FSAWRD BIGO1I(796)FIRST STAGE IGNITION S AND A WORD FROM PACP
FSIAFL BIGO1I(797)FIRST STAGE IGNITION ARMED FLAG
FSIWRD BIGO1I(798)FIRST STAGE IGNITION WORD FROM PACP
FSSAFL BIGO1I(799)FIRST STAGE SEPARATION ARMED FLAG
FSSWRD BIGO1I(800)FIRST STAGE SEPARATION WORD FROM PACP
IFAFLG BIGO1I(801)SAF INFLIGHT ARMING FLAG
LOFLG  BIGO1I(802)SAF LIFT-OFF FLAG
NB     BIGO1I(803)NUMBER OF BITS EXPECTED IN CURRENT SAF WORD
NBAB   BIGO1I(804)NUMBER OF BATTERY ACTIVATE BITS ACCUMULATED
NFSAB  BIGO1I(805)NUMBER OF FSI S AND A BITS ACCUMULATED
NFSIB  BIGO1I(806)NUMBER OF FIRST STAGE IGNITION BITS ACCUMULATED
NFSSB  BIGO1I(807)NUMBER OF FIRST STAGE SEP. BITS ACCUMULATED
NSAFB  BIGO1I(808)NUMBER OF SAF BITS ACCUMULATED
NSSIB  BIGO1I(809)NUMBER OF BITS ACCUMULATED IN SSI WORD
NSSSB  BIGO1I(810)NUMBER OF SECOND STAGE SEP. BITS ACCUMULATED
SAFRVS BIGO1I(811)SAF RV SEP. FLAG
SAFWRD BIGO1I(812)CURRENT SAF WORD FROM PACP
SSIAFL BIGO1I(813)SECOND STAGE IGNITION ARMED FLAG
SSIWRD BIGO1I(814)SECOND STAGE IGNITION WORD RECEIVED FROM PACP
SSSAFL BIGO1I(815)SECOND STAGE SEPARATION ARMED FLAG
SSSWRD BIGO1I(816)SECOND STAGE SEPARATION WORD FROM PACP
TSIFLG BIGO1I(817)SAF TERMINAL SEQUENCE INITIATE FLAG
OD2P   BIGO1I(854) O   PREVIOUS VALUE OF OUTPUT DISCRETE WORD 2
ACQFLG BIGO1I(851)DCL ACQUISITION FLAG
OD5P   BIGO1I(852) O   PREVIOUS VALUE OF OUTPUT DISCRETE WORD 5
SCNFLG BIGO1I(859)SCAN MODE FLAG
R      BIGO1D( 54)DISTANCE FROM EARTH'S CENTER
SIMTIM BIGO1D( 56)SIMULATION TIME FROM START OF NAVIGATION
TDER   BIGO1D( 58)TIME AT WHICH DERIVATIVES ARE COMPLTED
X      BIGO1D(129)STATE VECTOR-XE, PSI, U, AND P VECTORS
XP     BIGO1D(143)PSEUDO (INTERMEDIATE)STATE VECTOR
```

DISTRIBUTION

	No. of Copies
IIT Research Institute ATTN: GACIAC 10 West 35th Street Chicago, IL 60616	1
DRSMI-LP, Mr. Voigt	1
-RD	10
-RPR	15
-RPT, Record Copy	1