AD-769 595

AN ENGINEERING AND PROGRAMMING GUIDE FOR A SIX DEGREE OF FREEDOM, TERMINAL HOMING SIMULATION PROGRAM

R. F. Ball, et al

Army Missile Command
Redstone Arsenal, Alabama

10 October 1973

UNCLASSIFIED
Security Classification

AD769595

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)* US Army Missile Research, Development and Engineering Laboratory, US Army Missile Command, Redstone Arsenal, Alabama 35809	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP

3. REPORT TITLE

AN ENGINEERING AND PROGRAMMING GUIDE FOR A SIX DEGREE OF FREEDOM, TERMINAL HOMING SIMULATION PROGRAM

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*

Technical Report

5. AUTHOR(S) *(First name, middle initial, last name)*

R. F. Ball, W. W. Lee, Jr., and C. L. Lewis

6. REPORT DATE 10 October 1973	7a. TOTAL NO. OF PAGES ~~466~~ 471	7b. NO. OF REFS 2
8a. CONTRACT OR GRANT NO. b. PROJECT NO. (DA) 1M262303A214 c. AMC Management Structure Code No. 632303.11.21403 d.	9a. ORIGINATOR'S REPORT NUMBER(S) Technical Report RG-73-22 9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*	

10. DISTRIBUTION STATEMENT

Approved for public release: distribution unlimited

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY Same as No. 1

13. ABSTRACT

This report documents an all-digital, six degree of freedom (6DOF) simulation program which provides a realistic method for the evaluation and analysis of a terminal homing weapons system. Missile hardware components, vehicle dynamics, environment descriptions, and target motion are simulated by identifiable program modules. In some cases (e.g., the Actuator and Autopilot modules), both high and low frequency models are available, allowing simulations to be tailored to user needs. Two gimballed seekers are modeled which null on the LOS vector to provide an initial signal for proportional navigation guidance. A number of print and plot output options provide versatility of data presentation.

1a

DD FORM 1 NOV 65 1473 REPLACES DD FORM 1473, 1 JAN 64, WHICH IS OBSOLETE FOR ARMY USE.

UNCLASSIFIED
Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Six degree of freedom simulation program						
Terminal homing weapons system						
Proportional navigation guidance						
Gimballed seekers						
Four-quadrant seeker						
Vidicon seeker						

DISPOSITION INSTRUCTIONS

DISCLAIMER

TRADE NAMES

ic

TABLE OF CONTENTS

TABLE OF CONTENTS (Cont.)

TABLE OF CONTENTS (Cont.)

1. INTRODUCTION

The all-digital, six degree of freedom (6DOF) simulation program documented in this report provides a realistic method for the evaluation and analysis of a terminal homing weapons system. Missile hardware components, vehicle dynamics, environment descriptions, and target motion are simulated by identifiable program modules. In some cases (e. g., the Actuator and Autopilot modules), both high and low frequency models are available, allowing simulations to be tailored to user needs. Two gimballed seekers are modeled which null on the LOS vector to provide an initial signal for proportional navigation guidance. Either model, one simulating a four-quadrant (laser) seeker, and the other simulating a vidicon (Optical Contrast) seeker, can be optionally selected. Another input option permits simulation of either a direct fire (pre-launch target acquisition) mode, or an indirect fire (target acquisition after launch) mode. Pre-acquisition guidance during an indirect fire mode simulation is provided by a pre-programmed trajectory, where trajectory shaping can be accomplished by an input parameter. A number of print and plot output options provide versatility of data presentation. This report presents a functional overview of the program, a detailed description of the individual modules and subroutines, a brief description (where appropriate) of the components modeled, and a guide to the use of the program.

2. FUNCTIONAL DESCRIPTION

The six degree of freedom Terminal Homing Simulation Program provides the capability of simulating flights of a given configuration of the terminal homing seeker-missile under varying input conditions.

Figure 2-1 is a functional block diagram of the simulation program, depicting both basic program flow and basic data flow. Table 2.1 defines the variable symbols used in Figure 2.1. The program is modularly designed, where the modules describe either hardware components (e.g., Seeker, Actuator, etc.), homogenous environment conditions (e.g., Winds and Air Data), or simulation-unique functions (e.g., Coordinate Conversion, Translational Dynamic, etc.). This modular approach facilitates adaptation of the program to reflect changing hardware components, different levels of environment modeling detail, etc.

The program is structured to accomplish multiple case simulations during a single run.

As illustrated in Figure 2.1, the system is closed loop. After initialization, program flow begins with the Winds module. The Winds module produces a constant velocity wind vector, $\overline{V}_W$, which depends only upon input data. The Air Data module uses $\overline{V}_W$, the missile velocity vector $\overline{V}_M$, and missile altitude (not shown) inputs to generate the missile Mach number,

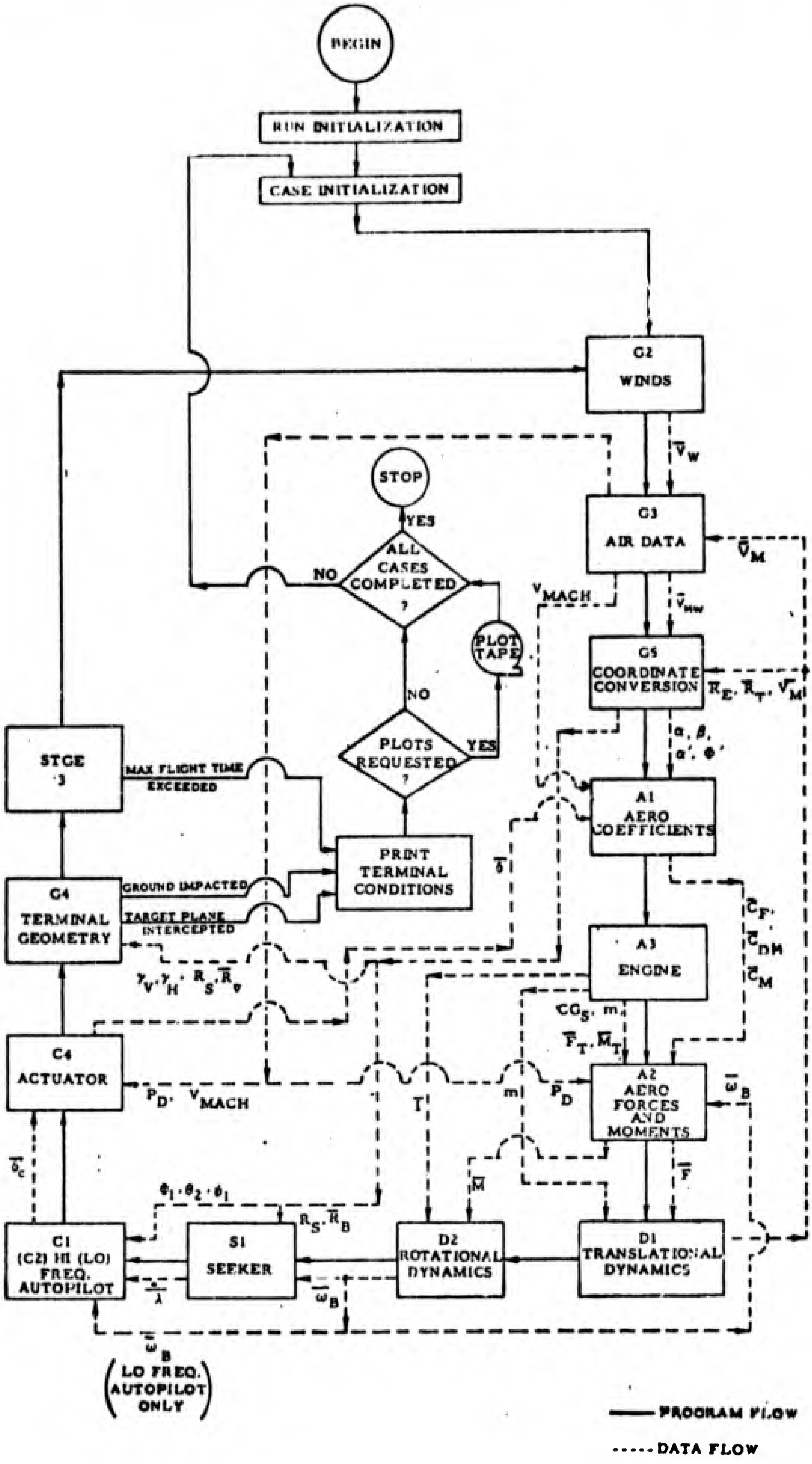


Figure 2.1 - Simulation Program Functional Diagram

Table 2.1. Functional Diagram Variable Definitions

$\bar{C}_{DM}$ - Aerodynamic damping coefficient vector, $(C_{\ell p}, C_{mq}, C_{mr})$

$\bar{C}_F$ - Aerodynamic force coefficient vector, (C_X, C_Y, C_Z)

$\bar{C}_M$ - Aerodynamic moment coefficient vector, (C_ℓ, C_m, C_n)

CG_s - Center of gravity shift due to burned propellant

$\bar{F}$ - Total force vector, (F_X, F_Y, F_Z), in body axes

$\bar{F}_T$ - Engine thrust force vector, (F_{TX}, F_{TY}, F_{TZ}), in body axes

$\bar{I}$ - Moment of inertia vector, (I_{XX}, I_{YY}, I_{ZZ}). (The products of inertia, I_{XY}, I_{XZ}, etc., are assumed to be zero.)

m - Vehicle mass (time dependent, due to burning propellant)

$\bar{M}$ - Total moment vector, (M_X, M_Y, M_Z)

$\bar{M}_T$ - Engine thrust moment vector, (M_{TX}, M_{TY}, M_{TZ})

P_D - Dynamic pressure

$\bar{R}_B$ - Line-of-sight vector (LOS), (R_{BX}, R_{BY}, R_{BZ}), in body axes

$\bar{R}_E$ - Missile position vector, (X_E, Y_E, Z_E), in earth axes

R_S - Missile to target slant range (magnitude of LOS vector)

$\bar{R}_T$ - Target position vector, (X_T, Y_T, Z_T) in earth axes

$\bar{R}_\nabla$ - LOS vector, $(X_\nabla, Y_\nabla, Z_\nabla)$, in earth axes

$\bar{V}_M$ - Missile velocity vector, (V_X, V_Y, V_Z), in earth axes

V_{MACH} - Missile Mach number

$\bar{V}_{MW}$ - Missile velocity vector, (V_{WX}, V_{WY}, V_{WZ}), relative to the wind, in earth axes.

Table 2.1. Functional Diagram Variable Definitions (continued)

$\bar{V}_W$ - Wind velocity vector in earth axes

α - Missile angle of attack (pitch)

β - Missile angle of attack (yaw)

α' - Missile angle of attack (total)

Φ_1 - Euler roll angle

Φ_2 - Euler pitch angle

Ψ_1 - Euler yaw angle

Measured between missile platform and body axes

ϕ' - Aerodynamic roll angle

$\bar{\delta}$ - Fin position vector, $(\delta_1, \delta_2, \delta_3, \delta_4)$, for fins 1, 2, 3 and 4

$\bar{\delta}c$ - Commanded fin position vector, $(\delta_{c1}, \delta_{c2}, \delta_{c3}, \delta_{c4})$

$\dot{\bar{\lambda}}$ - Time rate of change vector of the pitch and yaw angles between the LOS and velocity vectors

$\bar{\omega}_B$ - Body angular rate vector, $(\omega_P, \omega_Q, \omega_R)$

γ_H - Azimuth angle in earth axes

γ_V - Vertical flight path angle in earth axes

V_{MACH}, dynamic pressure, P_D, and the missile velocity in the wind vector, $\overline{V}_{MW}$. The Coordinate Conversion module uses $\overline{V}_{MW}$, the missile position and velocity vectors, $\overline{R}_E$, and $\overline{V}_M$, and the target position vector, $\overline{R}_T$, to produce the following outputs:

α	-	angle of attack in pitch	Measured from missile longitudinal axis to the relative wind axes
β	-	angle of attack in yaw	
α'	-	total angle of attack	
ϕ'	-	aerodynamic roll angle	
R_S	-	slant range	
$\overline{R}_{\nabla}$	-	missile to target (LOS) vector in earth axes	
$\overline{R}_B$	-	LOS vector in body axes	
ϕ_1	-	Euler roll angle	Measured between missile inertial platform and body axes.
θ_2	-	Euler pitch angle	
ψ_1	-	Euler yaw angle	

The inertial platform/body axes Euler angles are obtained indirectly by integrating the time derivatives of the elements of the earth-to-body transformation matrix (not shown).

The Aero Coefficients module uses the angles α, β, α', and ϕ', the missile Mach number, V_{MACH}, and the fin deflection $\overline{\delta}$ (from the Actuator) to generate the force, moment, and damping moment coefficients $\overline{C}_F$, $\overline{C}_M$, and $\overline{C}_{DM}$. Program control then passes to the Engine module, which uses only flight time and initial values as input to compute thrust, $\bar{F}_T$, moments due to thrust misalignment, $\overline{M}_T$, mass, m, center of gravity shift, CG_s, and moments of inertia, $\bar{I}$.

The Aero Forces module computes the total translational forces $\bar{F}$, and total moments, $\bar{M}$, using the following input data:

$$\left.\begin{array}{l}\bar{C}_F\\ \bar{C}_M\\ \bar{C}_{DM}\\ \bar{F}_T\\ \overline{M}_T\\ m\\ CG_s\\ P_D\end{array}\right\}\quad\text{Previously defined}$$

$\bar{\omega}_B$ - Body angular rates (ω_P, ω_Q, ω_R)

The Translational Dynamics module computes the missile position and velocity vectors, $\bar{R}_E$, and $\bar{V}_M$, based upon the inputs of force, $\bar{F}$, and mass, m. Target position, $\bar{R}_T$, is computed, when the target motion option is exercised, using constant input parameters.

The Rotational Dynamics module uses moments of inertia, $\bar{I}$, and moments, $\overline{M}$, to calculate body angular rates, $\bar{\omega}_B$.

The Seeker module generates the proportional navigation error signal,

$\bar{\dot{\lambda}}$, (the angular rates of the LOS in pitch and yaw) from the input LOS vector, $\bar{R}_B$, (expressed in the body axes) and the seeker platform orientation. The slant range, R_S, is used to adjust seeker deadband as a function of range.

The Autopilot uses the $\bar{\dot{\lambda}}$ error signal to develop attitude command signals in pitch and yaw. Roll error signals are generated by the roll stabilization channel. These are differenced with appropriate components of the missile attitudes (Φ_1, θ_2, ψ_1), measured from the stable platform, to form fin position command signal, $\bar{\delta}_c$. The Actuator determines the actual fin position, $\bar{\delta}$, based upon the error signal ($\bar{\delta}_c - \bar{\delta}$), and the differential equations describing the actuator dynamics.

At this point in the program logic flow, the STGE3 and Terminal Geometry subroutines determine if the simulation of the current case should be terminated. If so, trajectory end conditions (miss distance, etc.) are printed and, if requested by input option, plotted. The program then begins simulation of the next case, or terminates, if no additional cases are to be run.

If STGE3 and the Terminal Geometry subroutines do not terminate simulation of the current case, control passes to the Winds module, and the previously described process continues until terminal conditions are met.

3. SUBROUTINE DESCRIPTION

3.1 MAIN PROGRAM

MAIN is the primary executive routine of the simulation program. All subroutines and modules executed during a simulation run are either called from MAIN, or are called by subroutines or modules which are called by MAIN.

The initialization routines are called first by MAIN, after which the integration loop is entered and repeated until the target intercept plane is passed, the missile impacts the ground, or the simulation flight time exceeds an input maximum value. At this point, variables requiring it are reinitialized for possible subsequent runs, the plotting subroutines (PLOT4, PLOT2, or PLOTN) are called, if any plot options were exercised, and the process either repeated (if there are additional cases to be run), or terminated.

3.2 MODULES

3.2.1 A1-Aerodynamic Coefficient Table Look-Up Subroutine

3.2.1.1 Function Description

This subroutine utilizes the executive subroutines TABLE, TABL2, and TABL3 to retrieve the aerodynamic coefficients from data arrays stored by BLOCK DATA. The aerodynamic coefficients stored by BLOCK DATA are referenced to the wind axes coordinate system shown in Figure 3.2.1. For this coordinate system, a normal and axial force convention is used as opposed to a lift and drag force convention. The aerodynamic coefficient components and the independent variables of which they are a function that are stored by BLOCK DATA are as follows:*

Axial force coefficient at zero angle of attack

$$C_A = f_1 \text{ (MACH)}$$

Pitching moment coefficient at zero angle of attack

$$C_{mo} = f_2 \text{ (MACH)}$$

Normal force coefficient

$$C_{N'}(\alpha') = f_1 \text{ (MACH, } \alpha' \text{)}$$

Pitching moment coefficient

$$C_{m'}(\alpha') = f_2 \text{ (MACH, } \alpha' \text{)}$$

Incremental pitching moment coefficient due to aerodynamic roll angle (ϕ')

$$C_{m'} = f_3 \text{ (MACH, } \alpha' \text{)}$$

---

* The function notation f_X (Y) indicates the variables of which the dependent variable is a function.

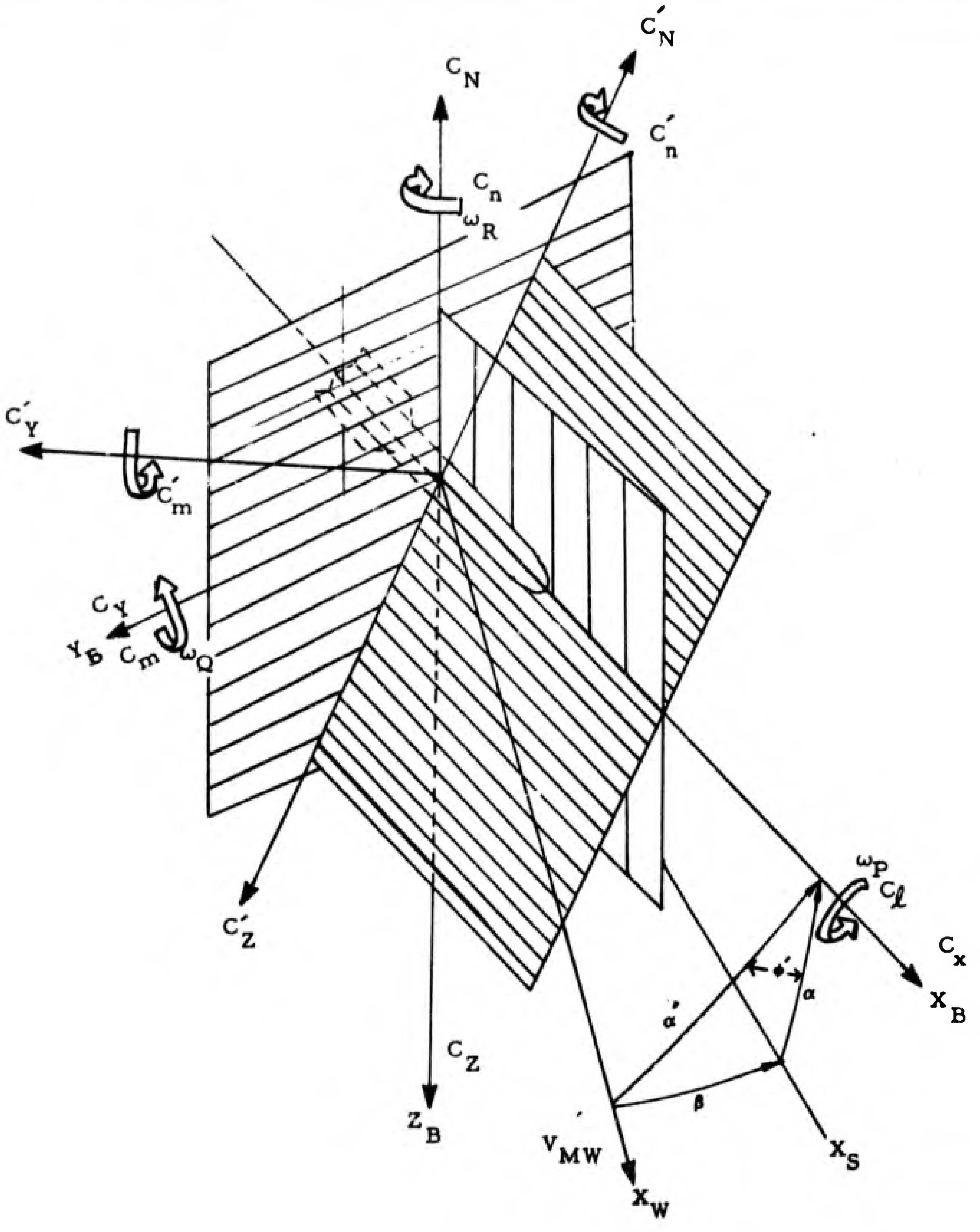


Figure 3. 2. 1 - Wind Axes Coordinate System

Pitch control moment coefficient due to fins

$$\left(\frac{C_{m'}}{\delta q}\right)_{\phi'=0} = f_4 \text{ (MACH, } \alpha' \text{)}$$

$$\left(\frac{C_{m'}}{\delta r}\right)_{\phi'=0} = f_4 \text{ (MACH, } \alpha' \text{)}$$

Incremental normal force coefficient due to aerodynamic roll angle (ϕ')

$$C_{N'} = f_5 \text{ (MACH, } \alpha' \text{)}$$

Normal force coefficient due to fin pitch position

$$\left(\frac{C_{N'}}{\delta q}\right)_{\phi'=0} = f_6 \text{ (MACH, } \alpha' \text{)}$$

Normal force coefficient due to fin yaw position

$$\left(\frac{C_{N'}}{\delta r}\right)_{\phi'=0} = f_6 \text{ (MACH, } \alpha' \text{)}$$

Incremental side force coefficient due to aerodynamic roll angle (ϕ')

$$\Delta C_{Y'} = f_7 \text{ (MACH, } \alpha' \text{)}$$

Incremental yawing moment due to aerodynamic roll angle (ϕ')

$$\Delta C_{N'} = f_8 \text{ (MACH, } \alpha' \text{)}$$

Incremental rolling moment due to aerodynamic roll angle

$$\Delta C_{\ell'_{Lugs}} = f_9 \text{ (MACH, } \alpha' \text{)}$$

$$\Delta C_{\ell'} = f_{10} \text{ (MACH,} \alpha' \text{)}$$

Rolling moment coefficient due to fin roll control

$$\left(\frac{C_{\ell'}}{\delta p}\right)_{\Phi' = 0} = f_{11} \text{ (MACH, } \alpha' \text{)}$$

Pitch, yaw, and roll damping coefficients

$$C_{mq} = f_{12} \text{ (MACH, } \alpha \text{)}$$

$$C_{nr} = f_{12} \text{ (MACH, } \alpha \text{)}$$

$$C_{\ell p} = f_{13} \text{ (MACH } \alpha \text{)}$$

The dependent variable and the independent variable(s) shown above are stored by BLOCK DATA and each dependent and each independent variable array is assigned to a COMMON BLOCK by BLOCK DATA. When more than one independent variable is involved, the arrays containing the independent variables are located in a single COMMON BLOCK. For example, $C_{m'}(\alpha')$ is a function of two independent variables, MACH and α'. COMMON BLOCK assignments are made thus by,

Variable Name	Array Name	Common Block	Common Block Order
$C_m'(\alpha')$	CMP	CMPF	/CMPF/CMP(35)
α'	BA3	CA3	/CA3/BA3(7), VM3(5)
MACH	VM3	CA3	

The order in which the array name appears in the COMMON BLOCK is based upon which independent variable is held constant while the other is varied through its numerical range. The array name appearing first in a COMMON BLOCK is varied through its range while the second is held constant.

The instantaneous value of a coefficient is obtained from the data arrays by use of the executive subroutines (look-up subroutines) TABLE, TABL2, and TABL3. The look-up subroutines used depends upon the number of independent variables, e.g.:

1 independent variable - use TABLE

2 independent variables - use TABL2

3 independent variables - use TABL3

The instantaneous value of the independent variable is sent into the appropriate look-up subroutine, along with the COMMON BLOCKS containing the dependent and independent variable arrays. A linear interpolation is performed between appropriate points and the instantaneous value of the dependent variable is returned.

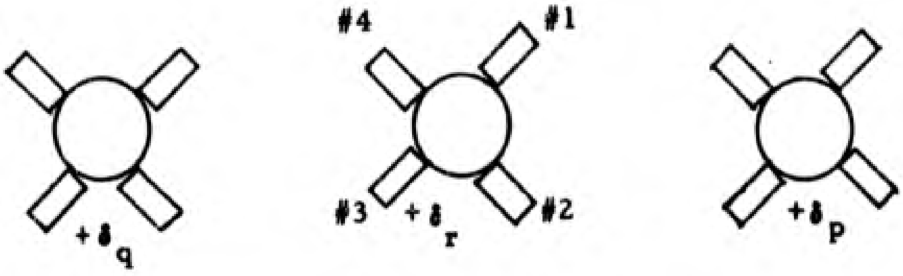
The force and moment coefficients of the control surfaces (fins) are given as a function of the ratio of the control surface coefficient to the average value of the combined angular displacement of the four control surfaces. The average value of the combined fin settings are computed in this module based upon the individual fin settings computed in the actuator module, C4, if the input flag OPTM is less than or equal to zero. (If the flag OPTM is greater than zero, the average value of the combined fin settings are computed from individual fin setting input data. Under this option, the fin settings remain constant throughout the flight as far as computed aerodynamic coefficients are concerned. This option overrides all autopilot commands to the fins, thus causing an unguided flight since all directional control is performed by the fins.)

3.2.1.2 Equations

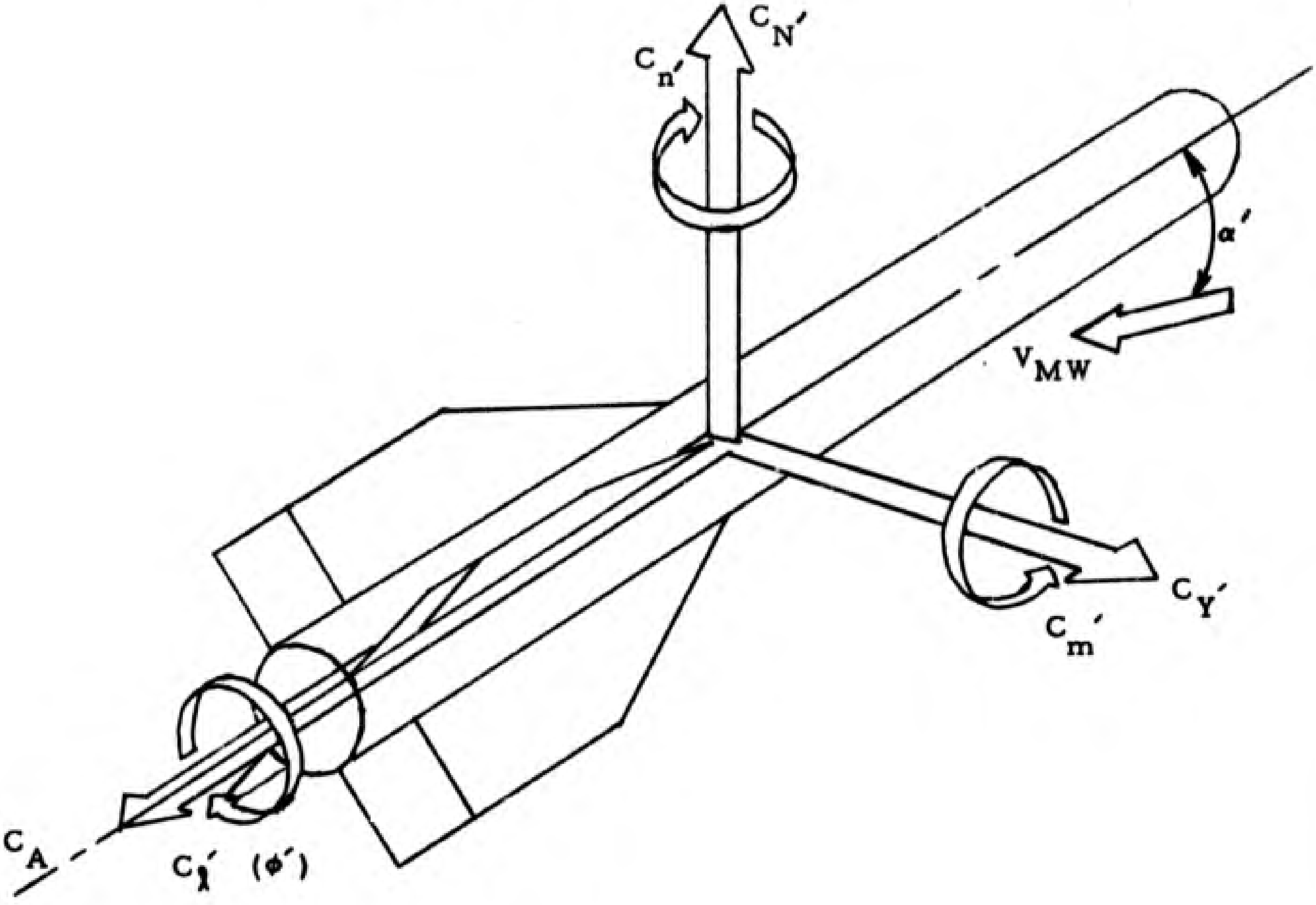
1. Fin Deflections

The average value of the combined fin settings based on individual fin settings are computed following the sign convention and fin setting configurations given in Figure 3.2.2. This figure illustrates the combined settings that produce pure attitudes of pitch, yaw, and roll.

The sign convention is based on the fin angular rotations that produce pure pitch. Thus, fin positive rotations - looking down the fin hinge toward the fin base - are:



Looking Forward, $\phi' = 0^{o}$.



NOTES:

i. C_{N}' is always in the plane of α'

2. Tail deflections shown are trailing edge displacements.

3. Tail hinge moments have the same direction as deflections δq.

Figure 3.2.2 - Aero Sign Conventions

Fin #1 - positive counter - clockwise (ΔF_1)

Fin #2 - positive counter - clockwise (ΔF_2)

Fin #3 - positive clockwise (ΔF_3)

Fin #4 - positive clockwise (ΔF_4)

Thus, the average value of the combined set is:

$$\delta_p = (-\Delta F_1 - \Delta F_2 + \Delta F_3 + \Delta F_4)/4. \qquad \text{(roll)}$$

$$\delta_q = (\Delta F_1 + \Delta F_2 + \Delta F_3 + \Delta F_4)/4. \qquad \text{(pitch)}$$

$$\delta_r = (-\Delta F_1 + \Delta F_2 - \Delta F_3 + \Delta F_4)/4. \qquad \text{(yaw)}$$

2. Total Force and Moment Coefficient Wind Axes

The total force and moment coefficients in the wind axes system are computed from the components listed in 3. 2. 1. 1, above according to the following:

Normal force coefficient

$$C_N' = C_N'(\alpha') + \Delta C_N' \text{SIN}^2 2\Phi' + \left(\frac{C_N'}{\delta q}\right)_{\Phi'=0} (\delta_q) \text{ COS } \Phi'$$
$$- \left(\frac{C_N'}{\delta r}\right)_{\Phi'=0} (\delta_r) \text{ SIN } \Phi'$$

Pitching moment coefficient

$$C_m' = C_m'(\alpha') + \Delta C_m' \text{ SIN}^2 2\,\Phi' + \left(\frac{C_m'}{\delta q}\right)_{\Phi'=0} (\delta_q) \text{ COS } \Phi'$$
$$- \left(\frac{C_m'}{\delta r}\right)_{\Phi'=0} (\delta_r) \text{ SIN } \Phi'$$

Yawing moment coefficient

$$C_{n'} = \Delta C_{n'} \ \mathrm{SIN4}\,\Phi' + \left(\frac{C_{n'}}{\delta\, r}\right)_{\Phi'=0} (\delta_r) \ \mathrm{COS}\ \Phi'$$

$$+ \left(\frac{C_{n'}}{\delta\, q}\right)_{\Phi'=0} (\delta_q) \ \mathrm{SIN}\ \Phi'$$

Side force coefficient

$$C_{Y'} = \Delta C_{Y'} \ \mathrm{SIN4}\ \Phi' + \left(\frac{C_{Y'}}{\delta\, r}\right)_{\Phi'=0} (\delta_r) \ \mathrm{COS}\ \Phi'$$

$$+ \left(\frac{C_{Y'}}{\delta\, q}\right)_{\Phi'=0} (\delta_q) \ \mathrm{SIN}\ \Phi'$$

Rolling moment coefficient

$$C_{\ell'} = \Delta C_{\ell'_{LUGS}} \ \mathrm{SIN2}\,\Phi' + \Delta C_{\ell'} \ \mathrm{SIN4}\,\Phi' + \left(\frac{C_{\ell'}}{\delta\, p}\right)_{\Phi'=0} (\delta_p)$$

3. Total Force and Moment Aerodynamic Coefficients - Body Axes

The aerodynamic forces and moments are computed in module A2 (forces and moments module) using the coefficients computed in this module. These forces and moments are computed in the missile body axes system; therefore, the coefficients computed above must be transformed into the body system. The transformation is made based on the geometry given in Figure 3.2.1. Thus, Force Coefficients

$$C_X = C_A{}^*$$

$$C_Y = C_Y{}' \cos\Phi' - C_N{}' \sin\Phi'$$

$$C_Z = -C_Y{}' \sin\Phi' - C_N{}' \cos\Phi'$$

Moment Coefficients

$$C_\ell = C_\ell{}'$$

$$C_m = C_n{}' \sin\Phi' + C_m{}' \cos\Phi' + C_{mo}$$

$$C_n = C_n{}' \cos\Phi' - C_m{}' \sin\Phi'$$

3.2.1.3 Initialization Subroutine

Entry point A1I in module C4I initializes the individual fin settings at time = 0 from data input on Type 3 input cards (see Section 3.5.1). Initial values of the fin settings are computed from the combined fin settings that produce pitch (δq), yaw (δr), and/or roll (δp). The individual fin setting computed are:

$$\Delta F_1 = -\delta p - \delta q + \delta r$$

$$\Delta F_2 = -\delta p + \delta q + \delta r$$

$$\Delta F_3 = \delta p + \delta q - \delta r$$

$$\Delta F_4 = \delta p + \delta q - \delta r$$

---

* Axial force coefficient C_A is actually opposite in sign of C_X. This sign difference is accounted for in subroutine A2 when the forces are computed.

The initialization equations for the individual fin settings are derived from the combined fin setting parameters under the constraint that:

$$\Delta F_1 - \Delta F_2 - \Delta F_3 + \Delta F_4 = 0$$

3.2.1.4 Assumptions and Limitations

1. The combined fin settings that produce pure attitudes of pitch, yaw and roll are given in Figure 3.2.2. The sign convention is based on the fin angular rotations that produce pure pitch.

   a. A positive fin setting for pitch (δ_q) gives a negative pitching moment.

   b. A positive fin setting for yaw (δ_r) gives a negative yawing moment.

   c. A positive fin setting for roll (δ_p) gives a positive rolling moment.

2. Due to vehicle fin symmetry, the fin pitch and yaw moment coefficients are equal and the fin normal and side force coefficients are equal.

   a. $$\frac{C_{m'}}{\delta r} = \frac{C_{m'}}{\delta q}$$

   b. $$\frac{C_{n'}}{\delta q} = \frac{C_{m'}}{\delta q}$$

   c. $$\frac{C_{n'}}{\delta r} = \frac{C_{m'}}{\delta q}$$

d. $$\frac{C_Y'}{\delta r} = \frac{C_Y'}{\delta q}$$

e. $$\frac{C_Y'}{\delta q} = \frac{C_N'}{\delta q}$$

f. $$\frac{C_N'}{\delta r} = \frac{C_N'}{\delta q}$$

3. In subroutine A1, the symbol for normal force coefficient C_N' was changed to C_Z'. According to Figure 3.2.1, the force coefficient C_Z' has opposite sign of C_N'. However, this sign change was not observed in programming the equation. Thus, the equation for normal force coefficient uses the symbol C_Z' but has the sign convention of C_N'.

4. Limitations imposed upon the angle of attack and on Mach number for the purpose of interpolating in the aero coefficient tables are as follows.

a. $\alpha' \leq 20^\circ$

$\alpha \leq 20^\circ$

$\beta \leq 20^\circ$

b. $.5 \leq \text{MACH} \leq 1.25$

3.2.1.5 Input/Output and Cross Reference of C-Array

1. Input from data cards

FORTRAN SYMBOL	SYMBOL USED IN TEXT	C INDEX	DEFINITION
OPTM	OPTM	1551	Option Flag 0 - guided missile 1 - unguided missile
UDL1	ΔF_1	1555	Angular setting of fin #1, measured positive as a counterclockwise rotation (DEG)
UDL2	ΔF_2	1556	Angular setting of fin #2, measured positive as a counterclockwise rotation (DEG)
UDL3	ΔF_3	1557	Angular setting of fin #3, measured positive as a clockwise rotation (DEG)
UDL4	ΔF_4	1558	Angular setting of fin #4, measured positive as a clockwise rotation (DEG)

2. Input from other Modules

FORTRAN SYMBOL	SYMBOL USED IN TEXT	C INDEX	DEFINITION
VMACH	MACH	204	Instantaneous value of mach number
BALPHA	α	367	Instantaneous value of the vertical angle of attack, measured from the projection of the missile velocity with respect to the wind into the X_B-Z_B plane and the X_B axis (DEG)
BALPHY	β	368	Instantaneous value of the horizontal angle of attack, measured from the projection of the missile velocity relative to the wind into the X_B-Z_B plane and the velocity relative to the wind (DEG)
BALPHP	α'	369	Instantaneous value of the total angle of attack between vehicle X_B axis and the relative wind velocity (DEG)
BPHIP	Φ'	370	Instantaneous value of the aerodynamic roll angle between the plane containing the Z_B-X_B axes and the plane containing the X_B-velocity vector relative to the wind (DEG)
BDELT1	ΔF_1	1103	Angular fin setting of Fin #i, measured positive, counterclockwise (DEG)
BDELT2	ΔF_2	1107	Angular fin setting of fin #2, measured positive counterclockwise (DEG)
BDELT3	ΔF_3	1111	Angular fin setting of fin #3, measured positive clockwise (DEG)
BDELT4	ΔF_4	1115	Angular fin setting of fin #4, measured positive clockwise (DEG)

3. Input from BLOCK DATA through COMMON BLOCKS and Output from Subroutines TABLE, TABL2, and TABL3

The following tables list the input arrays passed into Module A1 through COMMON BLOCKS from BLOCK DATA. These arrays contain the aerodynamic coefficient tables and their independent variable tables that are passed into the table look-up subroutines TABLE, TABL2 and TABL3 for the purpose of interpolating for instantaneous values of the coefficients. The variable symbol appearing in column 1 is the instantaneous value of that particular coefficient output from the table look-up subroutine. (These coefficients are wind axes coefficients.) Column 4 lists the COMMON BLOCK that contains the tables identified in column 5.

1 FORTRAN SYMBOL	2 SYMBOL USED IN TEXT	3 C INDEX	4 COMMON BLOCK	5 DEFINITION
NC1			NC1	Number of data points in the axial force coefficient table (CXOF) and the pitching moment coefficient table (CMOF). NC1(1) - specifies number of data points in the table NC1(2) - not used.
CA1	MACH		CA1	Mach number table containing mach number values corresponding to the axial force coefficient table CXOF
CXO	C_A	1212	CXOF	Axial force coefficient table containing axial force coefficient values corresponding to the mach number values in CA1
CA4	MACH		CA4	Mach number table containing mach number values corresponding to the pitching moment coefficient table CMOF
CM0	C_{m_0}		CMOF	Pitching moment coefficient at zero angle of attack table containing pitching moment coefficient values corresponding to the mach number table values in CA4
NC3(4)			NC3	Number of data points in the following tables: (1) Normal force – CZPF (2) Pitching moment - CMPF (3) Incremental pitching moment due to aerodynamic roll angle - CM2F (4) Normal force due to fin pitch position - CZDF (5) Normal force due to fin yaw position - CZDF
DCZ2	$C_{N'}$	1214	CZ2F	Incremental normal force coefficient due to aerodynamic roll angle (Φ')

1 FORTRAN SYMBOL	2 SYMBOL USED IN TEXT	3 C INDEX	4 COMMON BLOCK	5 DEFINITION
				(7) Pitching moment due to fin yaw position-CMDF
				NC3(1)-Number of points in the dependent variable table (ordinate axis-coefficient values) and the first independent variable table (abscissa axis, α') NC3(2)-Number of data points in the second independent variable (MACH) table NC3(3)-Not Used NC3(4)-Not Used
CA3	α'/MACH		CA3	This common block contains both the total angle of attack (α') table and the mach number (MACH) table. The angle of attack table containing NC3(1) points, appears first followed by the mach number table with NC3(2) points.
CZ0	$C_N(\alpha')$	1213	CZPF	Normal force coefficient table containing the number of points specified by the product of NC3(1) and NC3(2).
CMO	$C_m(\alpha')$	1218	CMPF	Pitching moment coefficient table containing the number of points specified by the product of NC3(1) and NC3(2).
DCM2	$\Delta C_{m'}$	1219	CM2F	Incremental pitching moment, table containing the number of points specified by the product of NC3(1) and NC3(2).
CZDQ	$\left(\frac{C_N}{\delta q}\right)$	1215	CZDF	Normal force coefficient (due to fin pitch position)table containing the number of points specified by the product of NC3(1) and NC3(2).

1 FORTRAN SYMBOL	2 SYMBOL USED IN TEXT	3 C INDEX	4 COMMON BLOCK	5 DEFINITION
CZDR	$\left(\frac{C_{N'}}{\delta r}\right)$	1216	CZDF	Normal force coefficient (due to fin yaw position) table containing the number of points specified by the product of NC3(1) and NC3(2)
CMDQ	$\left(\frac{C_{m'}}{\delta q}\right)$	1220	CMDF	Pitch control moment coefficient (due to fin pitch position) table containing the number of points specified by the product of NC3(1) and NC3(2).
CMDR	$\left(\frac{C_{m'}}{\delta r}\right)$	1221	CMDF	Pitch control moment coefficient (due to fin yaw position) table containing the number of points specified by the product of NC3(1) and NC3(2).
NC2(4)			NC2	Number of data points in the following tables: (1) Incremental side force coefficient due to fin yaw position-CY4F (2) Incremental yawing moment due to aerodynamic roll angle (ϕ') (3) Pitch damping coefficient-CMQF (4) Yaw damping coefficient-CMQF NC2(1) -number of points in the dependent variable table (ordinate axis, coefficient value) and the first independent variable table (abscissa, α) NC2(2)-number of data points in the second independent variable table (MACH) NC2(3) - Not Used NC2(4) - Not Used

1 FORTRAN SYMBOL	2 SYMBOL USED IN TEXT	3 C INDEX	4 COMMON BLOCK	5 DEFINITION
CA2	α/MACH		CA2	This common block contains both the vertical angle of attack (α) table and the mach number (MACH) table. The angle of attack table, containing NC2(1) points, appears first, followed by the mach number table with NC2(2) points
DCY4	$\Delta C_Y'$	1217	CY4F	Incremental side force coefficient table (due to aerodynamic roll angle, Φ') containing the number of points specified by the product of NC2(1) and NC2(3).
DCN4	$\Delta C_n'$	1222	CN4F	Incremental yawing moment coefficient table (due to aerodynamic roll angle, Φ') containing the number of points specified by the produce of NC2(1) and NC2(3).
CMQ	C_{mq}	1207	CMQF	Pitch damping moment coefficient table containing the number of points specified by the product of NC2(1) and NC2(3).
CNR	C_{nr}	1208	CMQF	Yaw damping moment coefficient table containing the number of points specified by the product of NC2(1) and NC2(3).
NC5(4)			NC5	Number of data points in the following tables: (1) Rolling moment coefficient due to fin roll control - CLDF (2) Incremental rolling moment due to aerodynamic roll angle (Φ') - CL4F (3) Incremental rolling moment coefficient due to launch lugs - CL2F

1 FORTRAN SYMBOL	2 SYMBOL USED IN TEXT	3 C INDEX	4 COMMON BLOCK	5 DEFINITION
				NC5(1)-Number of points in the dependent variable table (ordinate axis-coefficient table) and the first independent variable table (abscissa-α') NC5(2)-Number of data points in the second independent variable table (MACH) NC5(3)-Not Used NC5(4)-Not Used
CA5	α'/MACH		CA5	This common block contains both the total angle of attack (α') table and the mach number table (MACH). The angle of attack table, containing NC5(1) points, appears first, followed by the mach number table with NC5(2) points.
CLDP	$\left(\frac{C_\ell'}{\delta_p}\right)$	1225	CLDF	Rolling moment coefficient table containing the product of NC5(1) and NC5(2) points.
DCL4	$\left(\Delta C_\ell'\right)$	1224	CL4F	Incremental rolling moment (due to aerodynamic roll angle) table containing the product of NC5(1) and NC5(2) points.
DCL2	$\left(\Delta C'_{\ell_{LUGS}}\right)$		CL2F	Incremental rolling moment (due to launch lugs) table containing the product of NC5(1) and NC5(2) points.

FORTRAN SYMBOL	SYMBOL USED IN TEXT	C INDEX	DEFINITION
CH1		1236	Fin #1 hinge moment coefficient
CH2		1237	Fin #2 hinge moment coefficient
CH3		1238	Fin #3 hinge moment coefficient
CH4		1239	Fin #4 hinge moment coefficient
CH11		1240	Not Used
CH21		1241	
CH31		1242	
CH41		1243	

4. Output

FORTRAN SYMBOL	SYMBOL USED IN TEXT	C INDEX	DEFINITION
OPTHNG		1200	Not used in this module
CX	C_X	1203	Total axial force coefficient value in body axis system
CY	C_Y	1204	Total side force coefficient value in body axis system
CZ	C_Z	1205	Total normal force coefficient value in body axis system
CLP	$C_{\ell p}$	1206	Roll moment damping coefficient
CMQ	C_{mq}	1207	Pitch moment damping coefficient
CNR	C_{nr}	1208	Yaw moment damping coefficient
CL	C_{ℓ}	1209	Total rolling moment coefficient value in body axis system
CM	C_m	1210	Total pitching moment coefficient value in body axis system
CN	C_n	1211	Total yawing moment coefficient value in body axis system
VM	MACH	1226	Mach number
BAP	α'	1227	Total angle of attack (DEG)
BDL		1228	Absolute value of the four fin combinations for roll control (BDP) with a 10° limiting value (DEG)
BDM		1229	Absolute value of the four fin combinations for pitch control (BDQ) with a 10° limiting value (DEG)
BDN		1230	Absolute value of the four fin combinations for yaw control (BDR) with a 10° limiting value (DEG)
BDP	δp	1231	Average value of the combined fin settings for roll control (DEG)
BDQ	δq	1232	Average value of the combined fin settings for pitch control (DEG)
BDR	δr	1233	Average value of the combined fin settings for yaw control (DEG)

3.2.2 A2 - Forces and Moments

3.2.2.1 Function Description

This module computes aerodynamic forces and moments acting on the vehicle; computes forces and moments acting on the missile lugs due to launch rail motion and missile motion; and sums up all forces and moments acting on the vehicle computed in this module and those from other sources (such as thrust which is computed in A3). The resulting forces and moments output from this module are the total external forces and moments.

The aerodynamic forces and moments consist of those forces and moments acting on the vehicle and the moments on the movable control surface hinge points. The hinge moments are computed in this module to be used in the actuator module, C4. However, these moments are zeroed out in this module by allowing the hinge moment coefficients (CH1, CH2, CH3, and CH4) to default to zero (by virtue of the fact they are undefined) in module A1. C4 recomputes the hinge moments from a set of equations in C4 using hinge moment coefficients defined in a C4 DATA statement. The aerodynamic forces and moments acting on the vehicle are computed using the aerodynamic coefficients determined by table lookup in subroutine A1.

Lug forces and moments are computed for missile motion along the launch rail if the input flag OPTN4 is greater than zero. The lug forces and moments are computed for two flight phases. They are:

1) Both lugs on the rail
2) Rear lug only on the rail

Printed output concerning the events along the rail are controlled by this module. The events are: (1) front lug rail clearance, and (2) rear lug rail clearance. When the front lug clears the rail, the output is:

> "FRONT LUG CLEARS RAIL, T = (time),
> REL VEL = (airspeed), PITCH MOMENT = (pitching
> moment due to rear lug)"

The output pitching moment is the pitching moment computed at the first time point after front lug clearance. No interpolation is made to determine the exact time of front rail clearance.

When the rear lug clears the rail, the output is:

> "REAR LUG CLEARS RAIL, T = (time),
> REL VEL = (airspeed), RAIL FORCE = (the
> z-component of the rail force)"

The output rail force is the force exerted on both the front and rear lug at the time point just prior to front lug rail clearance. No interpolation is made to determine the exact time of rear lug rail clearance.

3.2.2.2 Equations

Moments and Forces Due to Aerodynamics and Thrust

Conventional aerodynamic force and moment equations are used to compute the forces and moments in the missile body axis system. They are:

1. Vehicle aero forces

$$F_{AXB} = QS_A\,(-C_X)^*$$

$$F_{AYB} = QS_A\,(C_Y)$$

$$F_{AZB} = QS_A\,(C_Z)$$

where Q is dynamic pressure, S_A is the vehicle reference area and the $C_{(subscripts)}$ are the aerodynamic coefficients

2. Vehicle aero moments

$$M_{AXB} = (C_\ell + C_{\ell_p}\left(\frac{S_L}{2V_{MW}}\right)\omega_P)\,QS_AS_L$$

$$M_{AYB} = (C_m + C_{mq}\left(\frac{S_L}{2V_{MW}}\right)\omega_Q)\,QS_AS_L$$

$$M_{AZB} = (C_n + C_{nr}\left(\frac{S_L}{2V_{MW}}\right)\omega_R)\,QS_AS_L$$

---

*C_X is sent into this module with the sign convention of C_A. Therefore, the sign change is made here to account for the sign convention used in this program, namely $C_X = -C_A$.

where S_L is the vehicle reference length; C_ℓ , C_m, and C_n are the aero moment coefficients; $C_{\ell p}$, C_{mq}, and C_{nr} are the damping moment coefficients; V_{MW} is the missile velocity relative to the wind and the $\omega_{(\text{subscripts})}$ are the vehicle rotation rates about their respective axes.

3. Control surface hinge moments

$$M_{H1} = C_{H1} Q S_A S_L$$

$$M_{H2} = C_{H2} Q S_A S_L$$

$$M_{H3} = C_{H3} Q S_A S_L$$

$$M_{H4} = C_{H4} Q S_A S_L$$

where C_{H1}, C_{H2}, C_{H3}, and C_{H4} are the aero coefficients of the respective control surface hinges.

Due to a C. G. shift from the vehicle reference axes, delta moments due to aerodynamic forces and thrust are computed. The assumption that the shift occurs only along the X-axis is made. Thus,

$$\Delta Y_{CG} = 0$$

and

$$\Delta Z_{CG} = 0 \; .$$

Therefore, delta body moments due to the C.G. shift are:

$$\Delta M_X = 0$$

$$\Delta M_Y = (F_{AZB} + F_{TZ}) \Delta X_{CG}$$

$$\Delta M_Z = (F_{AYB} + F_{TY}) \Delta X_{CG}$$

where F_{TY} and F_{TZ} are the thrust misalignment forces along the Y and Z body axes (computed in A3) and ΔX_{CG} is the CG shift as shown in Figure 3.2.3.

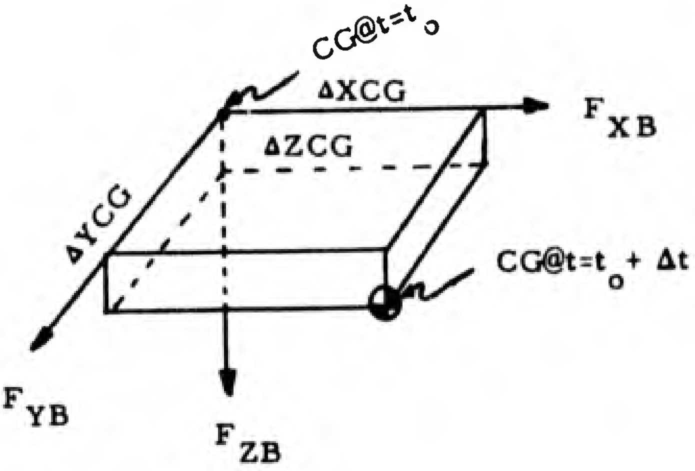


Figure 3.2.3

The sum total of the external body forces and moments computed in this module and input from the engine module (A3) are

$$\left.\begin{aligned} F_{XB} &= F_{AXB} + F_{TX} \\ F_{YB} &= F_{AYB} + F_{TY} \\ F_{ZB} &= F_{AZB} + F_{TZ} \end{aligned}\right\} \quad \text{(Components of total body forces)*}$$

$$\left.\begin{aligned} M_{XB} &= M_{AXB} + M_{XTH} \\ M_{YB} &= M_{AYB} + \Delta M_{Y} + M_{YTH} \\ M_{ZB} &= M_{AZB} + \Delta M_{Z} + M_{ZTH} \end{aligned}\right\} \quad \text{(Components of total body forces)*}$$

where F_{TX} is the thrust misalignment along the X-axis; and M_{XTH}, M_{YTH} and M_{ZTH} are the body moments due to the initial thrust misalignment (at $t = 0$) computed in the engine module (A3).

Moments and Forces Due to Lugs

The moments and forces acting on the missile while moving along the launch rail are shown in Figure 3. 2. 4 below. The variable symbols on the figure are defined as:

---

* Missile lug forces and moments are added while the missile is still on the launch rail if the flag OPTN4 is set to non-zero.

$\overline{F}_{TH}$ - thrust vector composed of (F_{TX}, F_{TY}, F_{TZ})

$\overline{F}_{AERO}$ - aerodynamic force vector composed of $(F_{AXB}, F_{AYB}, F_{AZB}$

$\overline{F}_{L1}$ - force acting on front lug

$\overline{F}_{L2}$ - force acting on rear lug

$\overline{W}_B$ - weight vector

$\overline{M}$ - total moment vector

$\overline{d}_1$ - front lug moment arm

$\overline{d}_2$ - rear lug moment arm

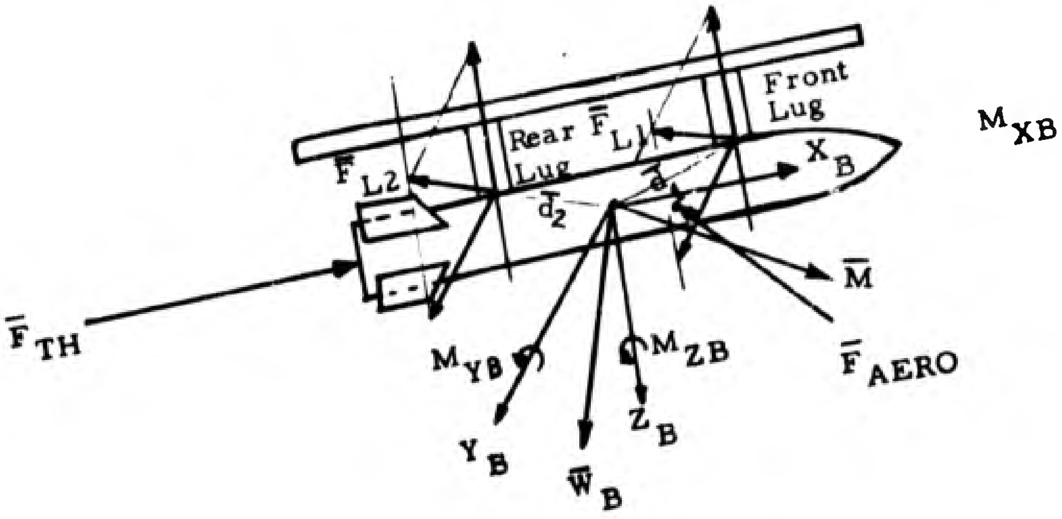


Figure 3.2.4

The forces and moments are due to the missile motion and the rail motion. Rail motion is described in Section 3. 2. 12 (G5 - Coordinate Conversion) and is given below in Figure 3. 2. 5.

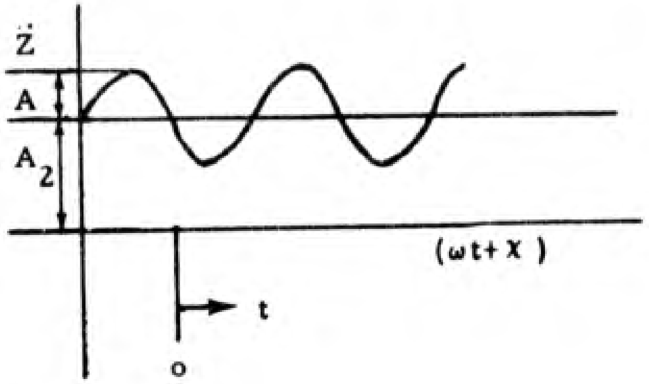


Figure 3. 2. 5 - Rail Launcher Motion

The motion is assumed to be along the Z-body axis and is of the form

$$\ddot{Z}_{RAIL} = A \sin (\omega t + X) + A_2 .$$

The forces and moments acting on the missile due to the lugs for the two flight phases are given below.

Phase 1 - Missile flight with both lugs on the rail.

Lug forces are:

$$F_{XLUG} = 0$$

$$F_{YLUG} = -(F_{YB} + mg\ C_{23})$$

$$F_{ZLUG} = mg(\ddot{Z}_{RAIL} - C_{33}) - F_{ZB} ,$$

where C_{23} and C_{33} are the (2, 3) and (3, 3) elements of the Earth-to-body transformation matrix, M, defined in Section 3. 2. 8.

Lug moments are:

$$M_{XLUG} = -M_{XB}$$

$$M_{YLUG} = -M_{YB}$$

$$M_{ZLUG} = -M_{ZB} .$$

The assumptions relating to flight phase 1 are:

1) No missile angular acceleration
2) The linear acceleration along the Y-body axis is zero
3) The only acceleration acting in the Z-body direction is due to launcher oscillation.

Phase 2 - Missile flight after the first lug clears rail

Lug forces are:

$$F_{XLUG} = 0$$

$$F_{YLUG} = \frac{m\, d_{x2} \frac{M_{ZB}}{I_Z} - (F_{YB} + mgC_{23})}{\left(1 + d_{x2}^2 \frac{m}{I_Z}\right)}$$

$$F_{ZLUG} = \frac{-m\left(\frac{d_{x2}}{I_Y} M_{YB} - \ddot{Z}_{RAIL}\right) - \left(F_{ZB} + mgC_{33}\right)}{\left(1 + d_{x2}^{2} \; \frac{m}{I_Y}\right)} \; .$$

Lug moments are:

$$M_{XLUG} = -M_{XB}$$

$$M_{YLUG} = d_{x2} \, F_{ZLUG}$$

$$M_{ZLUG} = -d_{x2} \, F_{ZLUG}$$

The assumptions relating to flight phase 2 are:

1) No rotation about the X-axis while the rear lug remains on the rail

2) The force on the front lug becomes zero

3) Launch rail acceleration is oscillatory and is along the Z-body axis.

Total external forces and Moments

The sum of all external forces and moments acting on the missile are:

Force Components

$$F_{XBA} = F_{AXB} + F_{TX} + F_{XLUG}$$

$$F_{YBA} = F_{AYB} + F_{TY} + F_{YLUG}$$

$$F_{ZBA} = F_{AZB} + F_{TZ} + F_{ZLUG}$$

Moment Components

$$M_{XBA} = M_{AXB} + M_{XTH} + M_{XLUG}$$

$$M_{YBA} = M_{AYB} + M_{YTH} + M_{YLUG} + \Delta M_{Y}$$

$$M_{ZBA} = M_{AZB} + M_{ZTH} + M_{ZLUG} + \Delta M_{Z}$$

3.2.2.3 Input/Output and Cross Reference of C-Array

1. Input from data cards

FORTRAN SYMBOL	SYMBOL USED IN TEXT	C INDEX	DEFINITION
RFAREA	S_A	1306	Aerodynamic cross section reference area (FT^2)
RFLGTH	S_L	1307	Aerodynamic reference length used to compute also moments (FT)
RLUG		1316	Spacing between the front and rear missile lugs (FT)
RAIL	RAIL	1317	Distance between rear of front lug and front of rail (FT)
AGV	A	1330	Amplitude of the oscillatory acceleration acting on the missile due to the launcher vibrating (G's)
CFREQ	ω	1331	Frequency of the oscillatory acceleration acting on the missile due to the launcher vibrating (CPS)
CPHAS	χ	1332	Phase relationship at time = 0 of the oscillatory acceleration acting on the missile due to the launcher vibrating (DEG
AGV2	A_2	1333	Magnitude of the linear component of the acceleration acting on the missile due to the launcher (G's)
AGRAV	g	1627	Acceleration due to gravity (FT/SEC)
OPTN4		3504	Rail launcher dynamics and fire selector option switch 0-no rail dynamics - direct fire 1-compute rail dynamics - direct fire 2-compute rail dynamics - indirect fire

2. Input from other modules

FORTRAN SYMBOL	SYMBOL USED IN TEXT	C INDEX	DEFINITION
PDYNMC	Q	203	Dynamic pressure (LBS/FT^2)
VMACH	M	204	Mach number
VAIRSP	V_{MW}	207	Magnitude of velocity relative to the wind (FT/SEC)
RANG0	$RANG_0$	380	Magnitude of separation distance between rail and missile (FT)
CX	C_X	1203	Aero axial force coefficient
CY	C_Y	1204	Aero side force coefficient
CZ	C_Z	1205	Aero normal force coefficient
CLP	$C_{\ell p}$	1206	Aero roll damping coefficient
CMQ	C_{mq}	1207	Aero pitch damping coefficient
CNR	C_{nr}	1208	Aero yaw damping coefficient
CL	C_ℓ	1209	Aero rolling moment coefficient
CM	C_m	1210	Aero pitching moment coefficient
CN	C_n	1211	Aero yawing moment coefficient
CH1	C_{H1}	1236	Hinge moment coefficients (currently defaults to zero)
CH2	C_{H2}	1237	
CH3	C_{H3}	1238	
CH4	C_{H4}	1239	
FMXTH	M_{XTH}	1320	Components of thrust misalignment moments about body axes (FT-LBS)
FMYTH	M_{YTH}	1321	
FMZTH	M_{ZTH}	1322	
RLCG	ΔX_{CG}	1422	C. G. shift along X_B-axis (FT)
FTHX	F_{TX}	1411	Thrust misalignment components along body axes (LBS)
FTHY	F_{TY}	1412	
FTHZ	F_{TZ}	1413	
CFA23	C_{23}	1723	Element (2, 3) of Earth-to-body transformation matrix, M
WP	ω_P	1739	Roll rate about X_B- axis (DEG/SEC)
WQ	ω_Q	1743	Pitch rate about Y_B- axis (DEG/SEC)
WR	ω_R	1747	Yaw rate about Z_B- axis (DEG/SEC)
T	t	2000	Flight time (SEC)
CFA33	C_{33}	1735	Elements (3, 3) of Earth-to-Body Transformation matrix, M

3. Output

FORTRAN SYMBOL	SYMBOL USED IN TEXT	C INDEX	DEFINITION
FXBA	F_{XB}	1300	X_B - component of all external forces acting on the missile (axial forces) (LBS)
FYBA	F_{YB}	1301	Y_B - component of all external forces acting on the missile (side force) (LBS)
FZBA	F_{ZB}	1302	Z_B - component of all external forces acting on the missile (normal force)(LBS)
FMXBA	M_{XB}	1303	X_B - component of all external moments acting on the missile (rolling moment) (FT-LBS)
FMYBA	M_{YB}	1304	Y_B - component of all external moments acting on the missile (pitching moment) (FT-LBS)
FMZBA	M_{ZB}	1305	Z_B - component of all external moments acting on the missile (yawing moment) (FT-LBS)
FMH1	M_{H1}	1309	Hinge moment about aerodynamics control surface, fin #1 (FT-LBS)
FMH2	M_{H2}	1310	Hinge moment about aerodynamics control surface, fin #2 (FT-LBS)
FMH3	M_{H3}	1311	Hinge moment about aerodynamics control surface, fin #3, (FT-LBS)
FMH4	M_{H4}	1312	Hinge moment about aerodynamics control surface, fin #4, (FT-LBS)
FMXLUG	M_{XLUG}	1323	(1) X_B - component of moment counteracted by the front and rear lugs when both lugs are on the rail. (2) X_B - component of moment acting on the rear lug when the front lug has cleared the rail. (Rolling moment, FT-LBS)
FMYLUG	M_{XLUG}	1324	(1) Y_B - component of moment counteracted by the front and rear lugs when both lugs are on the rail. (2) Y_B - component of moment acting on the rear lug when the front lug has cleared the rail. (Pitching moment - FT-LBS)
FMZLUG	M_{ZLUG}	1325	(1) Z_B- component of moment counteracted by the front and rear lug when the lugs are on the rail (2) Z_B - component of moment acting on the on the rear lug when the front lug has cleared the rail. (Yawing moment, FT-LBS

3.2.3 A3 - Engine Module

3.2.3.1 Function Description

The Engine Module calculates the total thrust (FTHRST) as a function of time, using a table look-up with linear interpolation between points. If the input engine misalignment switch, QNALGN, is greater than zero, the components of thrust (FTHX, FTHY, and FTHZ) along the body axes, X_B, Y_B, and Z_B, are calculated for use in determining translational accelerations; the corresponding moments (FMXTH, FMYTH, and FMZTH) are also calculated for use in determining rotational accelerations.

Regardless of the setting of the thrust misalignment switch, the Engine Module calculates instantaneous values for the time derivative of impulse (UIMPD), the weight of burned propellant (UDWP), the total vehicle mass (DMASS), the change in C. G. offset due to burned propellant (RDELCG), the moments of inertia (FMIX, FMIY, and FMIZ), and the distance between the C. G. and the rear Lug (RLCG).

A burnout switch (QBURN) is set to one when the first zero value is returned from the thrust table (indicating propellant burnout). After burnout, the Engine Module returns control to the calling routine without performing any calculations, and engine related variables retain their burnout values.

3.2.3.2 Equations

1. Calculated when Thrust Misalginment Option is selected (QNALGN > 0):

$$\left.\begin{aligned} FTHX &= (FTHRST)\ COS\ (\alpha_T) \\ FTHY &= -(FTHRST)\ SIN\ (\alpha_T)\ SIN\ (\Phi_T) \\ FTHZ &= (FTHRST)\ SIN\ (\alpha_T)\ \ COS\ (\Phi_T) \end{aligned}\right\} \text{Thrust components along body axes } (X_B, Y_B, Z_B)$$

$$\left.\begin{aligned} FMXTH &= -\ (FTHY)\ (RFZCG) + (FTHZ)(RFYCG) \\ FMYTH &= (FTHX)\ (RFZCG)+(FTHZ)(RFXCG) \\ FMZTH &= -(FTHX)(RFYCG)-(FTHY)(RFXCG) \end{aligned}\right\} \text{Moment components about } X_B,\ Y_B, \text{ and } Z_B \text{ axes due to thrust misalignment and thrust displacement from missile C.G.}$$

where the terms on the right hand side of the equations are as defined in Figure 3.2.6 below.

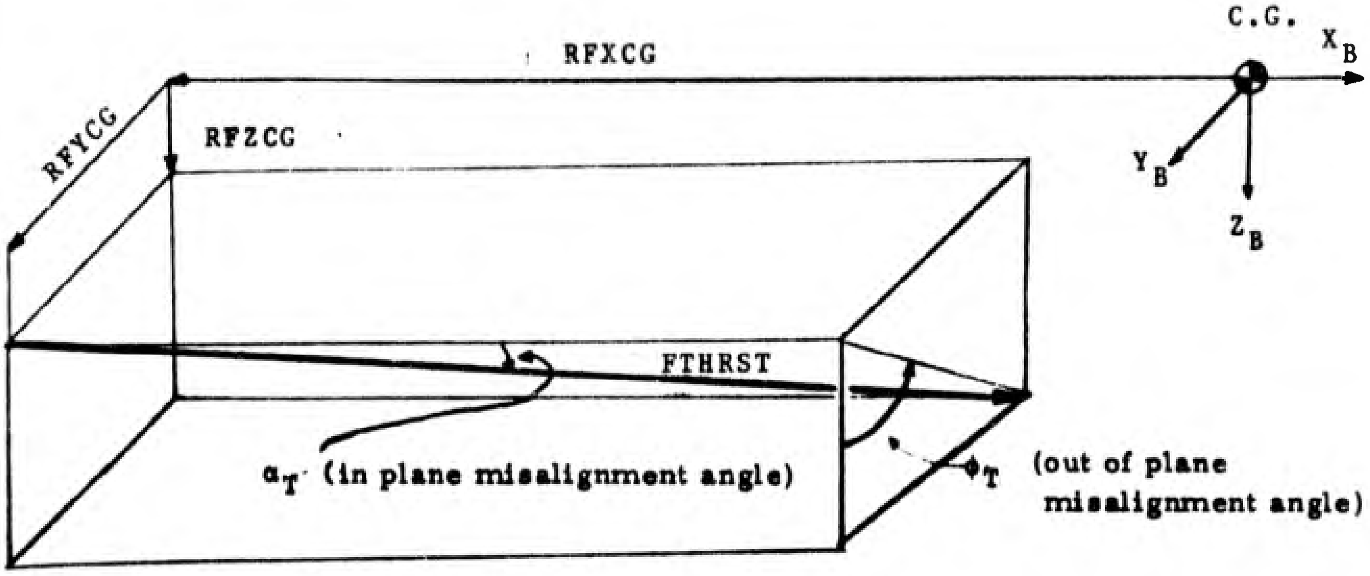


Figure 3.2.6

Thus, the magnitude of the thrust vector is FTHRST, its direction is specified by the angles α_T and Φ_T, and RFXCG, RFYCG, and RFZCG, are, respectively, the thrust application offsets from the missile C. G. along X_B, Y_B, and Z_B. Note the values of RFXCG, RFYCG, and RFZCG are to be input positive in the direction of the arrows shown in Figure 3. 2. 6 to obtain the correct sign for the moments in equations above.

2. Calculated regardless of the thrust misalignment option:

$$\text{UIMPD} = \text{FTHRST}$$

$$\text{UDWP} = \frac{\text{UIMP}}{\text{CISP}}$$

$$\text{DMASS} = \frac{(\text{DWT+DWP-UDWP})}{32.174}$$

$$\text{RDELCG} = \frac{-(\text{RDCGO-RDCGF})\ (\text{UDWP})}{\text{DWP}}$$

$$\text{FMIX} = \frac{\text{FMIXF}\ (\text{DWT+DWP-UDWP})}{\text{DWT}}$$

$$\text{FMIY} = \frac{\text{FMIYF}\ (\text{DWT+DWP-UDWP})}{\text{DWT}}$$

$$\text{FMIZ} = \text{FMIY}$$

$$\text{RLCG} = \text{RLCGO+RDELCG},$$

where the following terms appearing on the right hand side of the equation are inputs, and are defined as:

CISP = specific impulse

DWT = missile weight without propellant

RDCGO = launch value of C. G. displacement along X_B from Body Axes origin

RDCGF = Burnout value of C. G. displacement along X_B from Body Axes origin.

RLCGO = Distance between launch C. G. and rear lug

FMIXF
FMIYF } Burnout moments of inertia about X_B, Y_B, and Z_B axes, respectively.
FMIZF

3.2.3.3 Assumptions

Although the instantaneous C. G. displacement from the Body Axes origin (RDELCG) is calculated for use in the Forces and Moments Module (A2), the C. G. shift is not used in the calculations for the instantaneous value of moments of inertia; the moments of inertia are modified only for the changing mass.

3.2.3.4 Initialization Subroutine

The Engine Module Initialization Subroutine (A3I) performs functions unrelated to the engine when the trajectory simulation initialization time is prior to engine burnout. Specifically, A3I initializes the angular body rates (ω_P, ω_Q, and ω_R) and the attack angles (BALPHA, BALPHY, and BPHIP) to zero when simulation initiation is prior to burnout. The significance of these initialization actions is that angular rates and angles of attack will always be initially zero, regardless of the input values. This results from the fact that A3I is called after the input subroutine, OINPT1, and prior to the modules in the integration loop.

When simulation initiation is after engine burnout, A3I zeros all thrust components and sets moments of inertia, C.G. shifts, and vehicle mass to burnout values.

3.2.3.5 Input/Output and Cross Reference of C-Array

1. Input from Data Cards

FORTRAN SYMBOL	SYMBOL USED IN TEXT	C INDEX	DEFINITION
RFXCG	RFXCG	1313	Thrust application point offset along X_B,Y_B,and Z_BBody Axes, from C.G.
RFYCG	RFYCG	1314	
RFZCG	RFZCG	1315	
BALPHT	α_T	1401	Engine misalignment angles (see Figure 3.2.6)
BPHIT	Φ_T	1402	
QNALGN	QNALGN	1403	Engine misalignment option switch (QNALGN > 0 selects engine misalignment)
QBURN	QBURN	1405	Engine burnout switch (QBURN > 0) signals engine burnout at simulation initiation)
CISP	CISP	1414	Specific impulse (SEC)
DWT	DWT	1415	Missile weight without propellant (LBS)
DWP	DWP	1416	Propellant weight (LBS)
RDCGO	RDCGO	1417	Launch value of C.G. displacement along x-body axis (FT)
RDCGF	RDCGF	1418	Burnout value of C.G. displacement along x-body axis (FT)
FMIXF	FMIXF	1419	Burout moments of inertia about X and Y body axes. (Due to assumed missile axial symmetry, FMIZF is taken to be equal to FMIYF.) (SLUG-FT^2)
FMIYF	FMIYF	1420	
RLCGO	RLCGO	1421	Distance between launch C.G. and REAR lug. (FT)

2 Outputs

FORTRAN SYMBOL	SYMBOL USED IN TEXT	C INDEX	DEFINITION
RDELCG	RDELCG	1308	Change in C.G. shift due to burned propellant (FT)
FMXTH	FMXTH	1320	Moment of force components about X, Y, and Z body axes due to engine misalignment and displacement of thrust point application from C.G. (FT-LBS)
FMYTH	FMYTH	1321	
FMZTH	FMZTH	1322	
FTHX	FTHX	1411	Translational thrust components along X, Y, and Z body axes (LBS)
FTHY	FTHY	1412	
FTHZ	FTHZ	1413	
RLCG	RLCG	1422	Distance between the instantaneous C.G. and rear lug (FT)
DMASS	DMASS	1628	Instantaneous missile mass (SLUGS)
FMIX	FMIX	1748	Instantaneous values of moments of inertia about X, Y, and Z body axes. (SLUG-FT^2)
FMIY	FMIY	1749	
FMIZ	FMIZ	1750	

3.2.4 C1 - Autopilot Module (High Frequency Model

3.2.4.1 Function Description

The autopilot, illustrated in the block diagram of Figure 3.2.7, provides the pitch and yaw guidance signal shaping, short period rate damping, and navigation ratio mechanization required to implement the proportional navigation homing guidance technique. The compensation indicated in this figure has been modeled for the 6DOF simulation using he M method of programming to obtain the respective differential quations.[1]

3.2.4.2 Equations

The first function of the autopilot is to perform appropriate guidance switching and signal shaping. If the seeker is in the acquisition mode (TKRY=0, TKRZ=0), then a preprogrammed flight profile is executed by using the q_{bias} and r_{bias} terms to command pitch and yaw maneuvers. If the gyro has not been uncaged (CAGE = 0), the seeker signals directly drive the missile autopilot. Otherwise, the normal terminal homing is in effect and the seeker output signals are smoothed and shaped by a lead-lag filter with one zero at ω_L and a double pole at ω_{n2}. Gravity effects are continuously compensated for in all cases by addition of a g_{bias} term. At this point, additional shaping is performed by a first

---

[1] D.G. Schultz and J. L. Melsa, State Functions and Linear Control Systems, McGraw Hill Book Co., 1967, pp. 39-42.

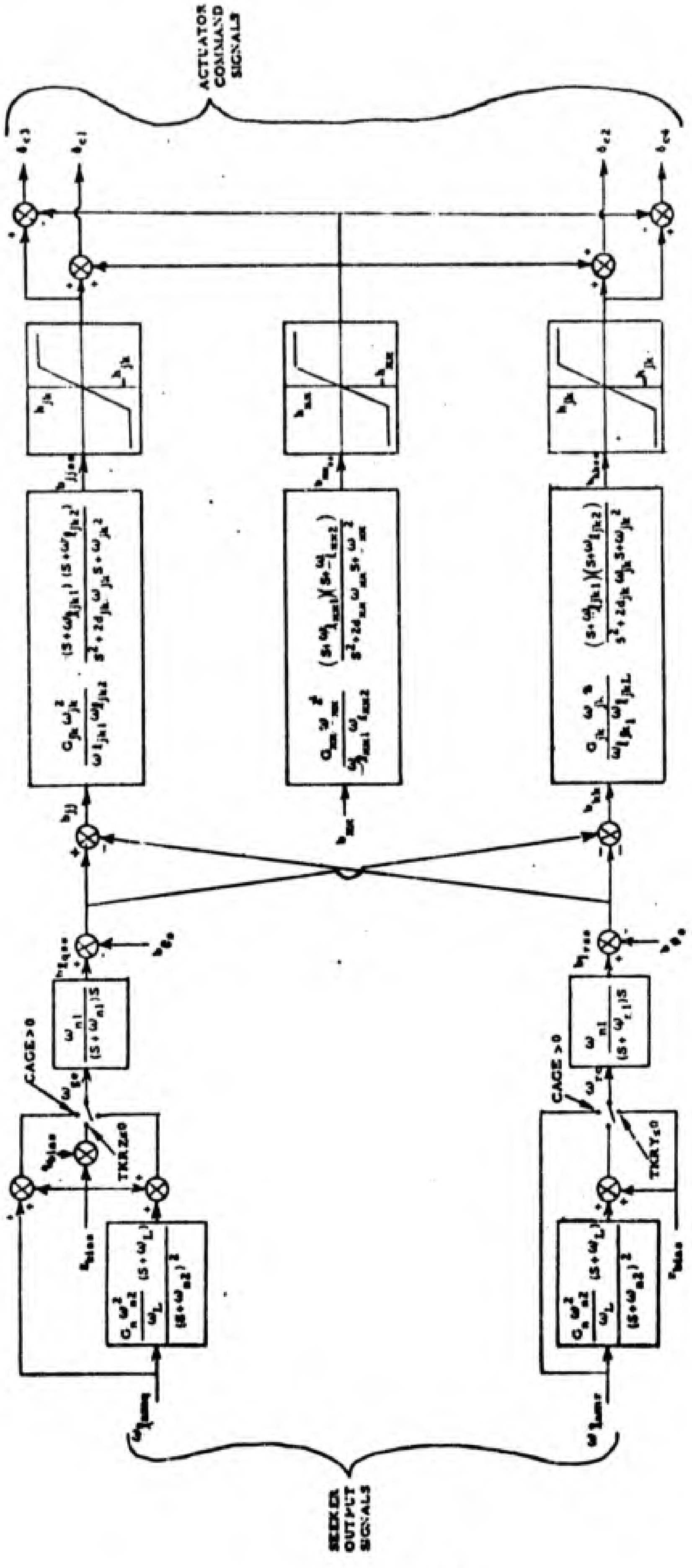
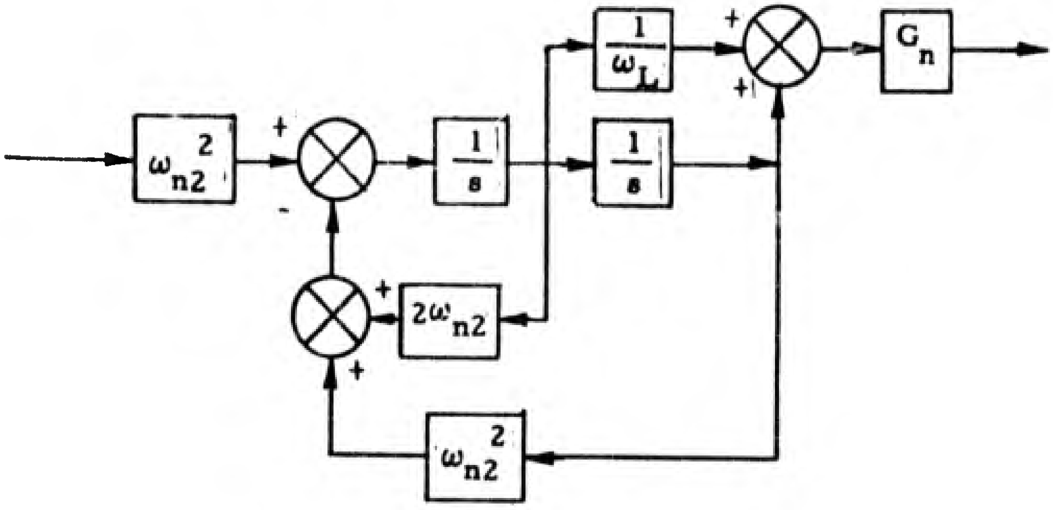


Figure 3.2.7 - Autopilot

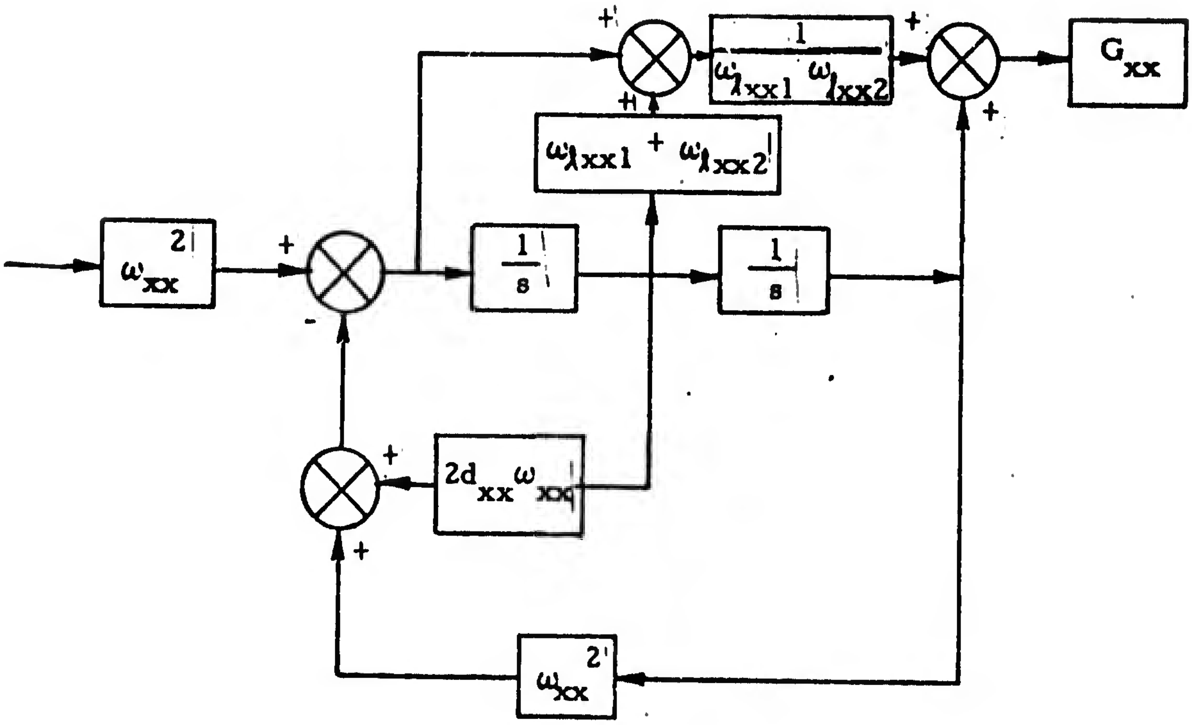
order lag at ω_{nl} and then the signal is integrated to provide an attitude command to the feedback controller.

The commanded attitude signals ($b_{\ell qss}$ and $b_{\ell rss}$) are differenced with the sensed attitude ($b_{\theta s}$ and $b_{\psi s}$), the feedback signal. The resulting error signals are appropriately summed and differenced to produce the components in the missile's pitch and yaw axes. These signals are compensated by a lead lag filter with real zeros at $\omega_{\ell jk1}$ and $\omega_{\ell jk2}$. and a complex pole pair with natural frequency ω_{jk} and damping ratio d_{jk}. Lastly, the signals are limited to keep the actuators from hitting their stops. The roll control system attempts to maintain a zero roll attitude via a similar filter to that above with zeros at $\omega_{\ell xx1}$ and $\omega_{\ell xx2}$ and poles at natural frequency $_{ww}$ and damping d_{xx}. The pitch-roll and yaw-roll signals are summed and differenced in order to produce the appropriate reactions with the four actuators.

Implementation of the two filter types by the M method is indicated in Figure 3.2.8.



(a) Implementation of $\frac{G_n\omega_{n2}^2}{\omega_L} \quad \frac{s+\omega_L}{s+\omega_{n2}^2}$



(b) Implementation of $\frac{G_{xx}\ \omega_{xx}^2}{\omega_{\lambda xx1}\ \omega_{\lambda xx2}} \quad \frac{s+\omega_{\lambda xx1} \quad s+\omega_{\lambda xx2}}{s^2+2d_{xx}\omega_{xx}s+\omega_{xx}^2}$

Figure 3.2.8 - M Method of Transfer Function Implementation

3.2.4.3 Input/Output Variables and Cross Reference of C-Array

1. Input from Data Cards

FORTRAN SYMBOL	SYMBOL USED IN TEXT	C INDEX	DEFINITION
TDY		860	Minimum time before trajectory pitch program will be switched to terminal homing guidance.
GBIAS	g_{bias}	861	Gravity bias program
GN	G_n	862	Seeker signal shaping gain
WN2	ω_{n2}	863	Second order pole location for seeker signal shaping filter
WN1	ω_{n1}	864	First order pole location for shaping filter
WL	ω_l	865	Zero location for seeker signal shaping filter
WLXX1	$\omega_{\ell xx1}$	866	First zero in roll stabilization filter
WLXX2	$\omega_{\ell xx2}$	867	Second zero in roll stabilization filter
WLJK1	$\omega_{\ell jk1}$	868	First zero in pitch-yaw stabilization filter
WLJK2	$\omega_{\ell jk2}$	869	Second zero in pitch-yaw stabilization filter
HJK	h_{jk}	870	Pitch-yaw signal limiter
WXX	ω_{xx}	871	Natural frequency of roll stabilization filter
DXX	d_{xx}	872	Damping ratio of roll stabilization filter
WJK	ω_{jk}	873	Natural frequency of pitch-yaw stabilization filter
DJK	d_{jk}	874	Damping ratio of pitch-yaw stabilization filter
GXX	G_{xx}	875	Roll compensation gain
GJK	G_{jk}	876	Pitch-yaw compensation gain
QBIAS	q_{bias}	878	Pitch trajectory shaping bias
RBIAS	r_{bias}	879	Yaw trajectory shaping bias
HXX	h_{xx}	890	Roll signal limiter

2. Inputs from Other Modules

FORTRAN SYMBOL	SYMBOL USED IN TEXT	C INDEX	DEFINITION
WLAMQ	ω_{lamq}	403	Seeker output - pitch axis
WLAMR	ω_{lamr}	407	Seeker output - yaw axis
CAGE	CAGE	461	Seeker gyro cage/uncage switch (CAGE=1 if uncaged).
TKRZ	TKRZ	462	Acquisition gain switches in pitch and yaw channel (Equal 1 if using acquisition gain)
TKRY	TKRY	463	

3. Outputs

FORTRAN SYMBOL	SYMBOL USED IN TEXT	C INDEX	DEFINITION
BDELTC(1)	δ_{c1}	856	Commanded position, fin #1
BDELTC(2)	δ_{c2}	857	Commanded position, fin #2
BDELTC(3)	δ_{c3}	858	Commanded position, fin #3
BDELTC(4)	δ_{c4}	859	Commanded position, fin #4

3.2.5 C2 - Autopilot Module (Low Frequency Model)

The low frequency model of the autopilot is identical to that used for high frequencies except that one simplifying assumption has been made. Since the pole of the roll and the pitch-yaw stabilization filters is very far in the left half s plane, the transfer function approximations below have been made.

$$\frac{G_{jk} \; \omega_{jk}^{2}}{\omega_{\ell jk1} \, \omega_{\ell jk2}} \; \frac{\left(S + \omega_{jk1}\right)\left(S + \omega_{jk2}\right)}{S^{2} + 2d_{jk} \, \omega_{jk} S + \omega_{jk}^{2}} \approx$$

$$\frac{G_{jk}}{\omega_{\ell jk1} \, \omega_{\ell jk2}} \left(S + \omega_{\ell jk1}\right) \left(S + \omega_{\ell jk2}\right)$$

and

$$\frac{G_{xx} \; \omega_{xx}^{2}}{\omega_{\ell xx1} \, \omega_{\ell xx2}} \; \frac{\left(S + \omega_{\ell xx1}\right)\left(S + \omega_{\ell xx2}\right)}{S^{2} + 2d_{xx} \omega_{xx} S + \omega_{xx}^{2}} \approx$$

$$\frac{G_{xx}}{\omega_{\ell xx1} \, \omega_{\ell xx2}} \left(S + \omega_{\ell xx1}\right) \left(S + \omega_{\ell xx2}\right)$$

As long as the actual frequencies encountered are significantly below the ignored poles, this approximation should have a negligible effect on the simulation accuracy.

NOTE: The low frequency autopilot initialization routine (C2I) sets the integration limits (HMIN, HMAX) and the initial stepsize (DER(1) equal to .005 seconds. Therefore, no inputs by Type "3" cards are accepted for these variables when using module C2.

3.2.6 C4 - Actuator Module

3.2.6.1 Function Description

The actuator system utilizes four independently controlled, pneumatically powered servos to position the four movable tail fins. Four commanded actuator fin position signals from the autopilot are summed with the negative feedback signals from the fin position potentiometers on the actuator mechanism. The resulting signals are used to pulse width modulate constant amplitude square wave signals. These signals control the on-off state of solenoid valves which are contained in the pneumatic system between the cold gas pressure source and the servos. An actuator position loop block diagram is depicted in Figure 3.2.9.

3.2.6.2 Equations

The 6DOF simulation program determines the derivatives of each of the four fin positions ($\dot{\delta}_i$, i = 1, 2, 3, 4). One of two actuator models is used, depending upon the value of an input switch, OPTACT. OPTACT ≤ 0 causes the low frequency actuator model to be used, while OPTACT >0 results in use of the high frequency actuator.

Regardless of the setting of OPTACT, however, certain gain, deadband, and limiting functions are performed for each of the four fins. The inputs to these functions are error signals representing the difference between

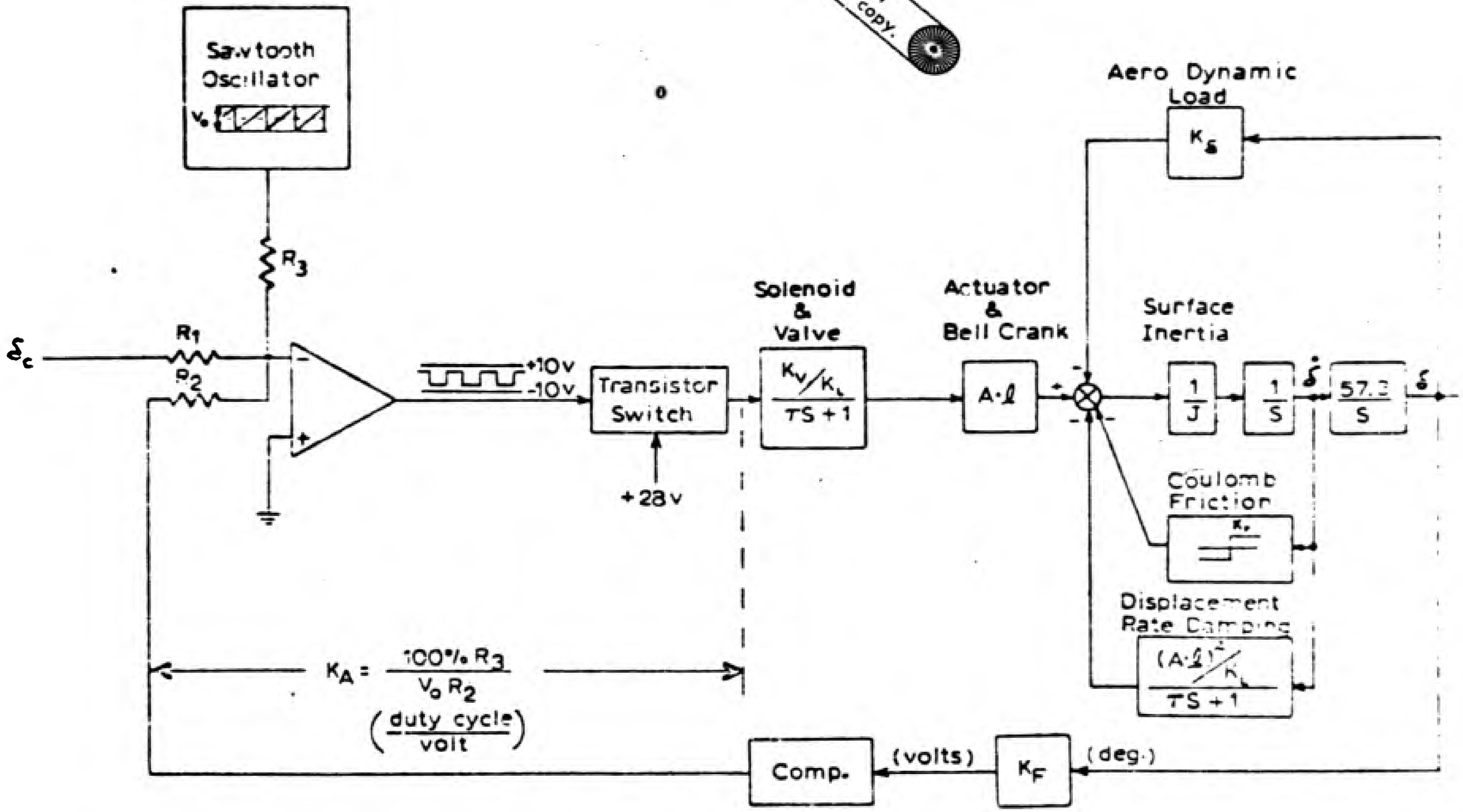


Figure 3.2.9 - Actuator Position Loop BLOCK DIAGRAM

the autopilot commanded in fin position δ_{ci}, and the actual fin positions, δ_i. The output of the gain, deadband, and limiting functions is denoted by a_2 (see Figure 3. 2. 10),which successively takes on values corresponding to each of the four fins. The output, a_2, is successively used as the forcing function for the time derivatives of the fin positions, $\dot{\delta}_i$.

The equations and logic for the gain, limiting and deadband functions are (where the input is $\delta_{ci} - \delta_i$, and the output is a_2):

$$\text{(error)} \quad a_1 = \delta_{ci} - \delta_i$$

$$\text{(limiting)} \; a_{1s} = \begin{Bmatrix} a_1 - b_h, \text{ if } a_1 > 0 \\ a_1 + b_h, \text{ if } a_1 < 0 \end{Bmatrix}$$

$$\text{(deadband)} \;\; \text{if } (\left| a_{1s} \right| \leq b_h), \text{ set } a_{1s} = 0$$

$$\text{(gain)} \qquad a_2 = \frac{g_1}{C_r} \; a_{1s}$$

$$\text{(limiting} \qquad \begin{Bmatrix} \text{If } (a_2 < (-h_2) \;) \text{ set } a_2 = -h_2 \\ \text{If } (a_2 > h_1) \text{ set } a_2 = h_1 \end{Bmatrix}$$

A flow chart of these functions is shown in Figure 3. 2. 10.

If the low frequency actuator model (OPTACT $\leq$ 0) is used, then the time derivatives of fin position, $\dot{\delta}_i$, i = 1, 2, 3, 4 are defined by the following equations:

$$\dot{\delta}_i = \frac{a_2 - g_2 \, f_m \, \delta_i}{\left(1 + \frac{g_2}{\omega_1} \, f_m\right)}$$

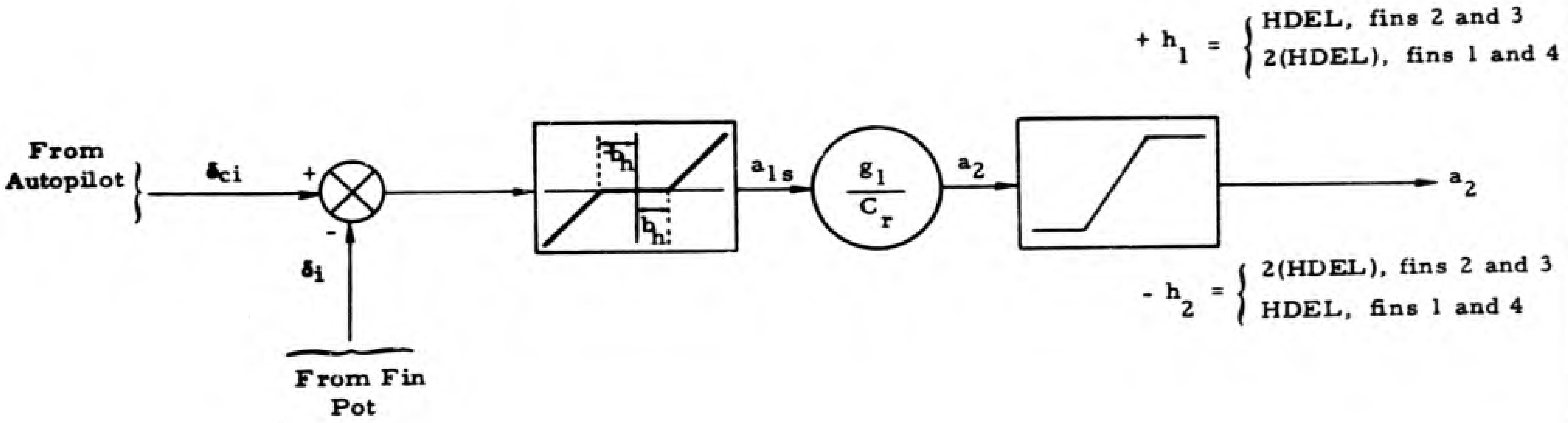


Figure 3.2.10 - Actuator Gain, Deadband, and Limiting

where g_2 and ω_1 are input constants, and f_m is the fin aerodynamic moment computed by:

$$f_m = C_{HD}\, Q\, S_A\, S_L/12,$$

where Q is dynamic pressure, S_A the aero reference area, S_L the aero reference length, and C_{HD} is the fin aero moment coefficient. C_{HD} is a function of Mach number and is obtained by a linear interpolation of the tabled values of Mach number and fin aerodynamic moment coefficients given in Table 3.2.1, below.

Mach Number	Fin Aerodynamic Hinge Moment Coefficient
0.00	0.0013
0.40	0.0014
0.80	0.0018
0.95	0.0022
1.05	0.0032
1.40	0.0023

Table 3.2.1. Fin Aerodynamic Hinge Moment Coefficient

Corresponding to limits of fin movement on the missile, the program limits the maximum fin positions to ± 19°. This is implemented by setting the fin position derivatives, $\dot{\delta}_i$, to zero whenever the absolute value of any of the current fin position, δ_i, exceeds 19°, and the sign of $\dot{\delta}_i$ is such that it would cause a greater fin deflection.

A block diagram of the low frequency actuator is depicted in Figure 3.2.11. (Note that because the differential equation contains the non-constant coefficient, f_m, a transfer function representation cannot properly be made.)

The high frequency actuator model is described by the following equations:

$$\dot{B}_i = \omega_1 (a_2 - B_i)$$

$$E_i = B_i - g_2 f_m \delta_i$$

$$\ddot{\omega}_i = \omega_n(\omega_n(E_i - \omega_i) - 2 z_n \dot{\omega}_i)$$

$$\delta_i = \frac{\dot{\omega}_i}{\omega_1} + \omega_i$$

where B_i, ω_i, and their derivatives are intermediate variables in the calculation of the fin position derivatives $\dot{\delta}_i$, and ω_1, g_2, ω_n, and z_n are input constants.

Fin position limiting to ± 19° is accomplished as described for the case OPTACT ≤ 0.

The equations above are linear with constant coefficients, and, thus, can be represented by transfer functions. The block diagram is depicted in Figure 3.2.12. Those elements which are identifiable with corresponding elements in the position loop block diagram (Figure 3.2.9) are labeled according to the component they represent. The two block diagrams,

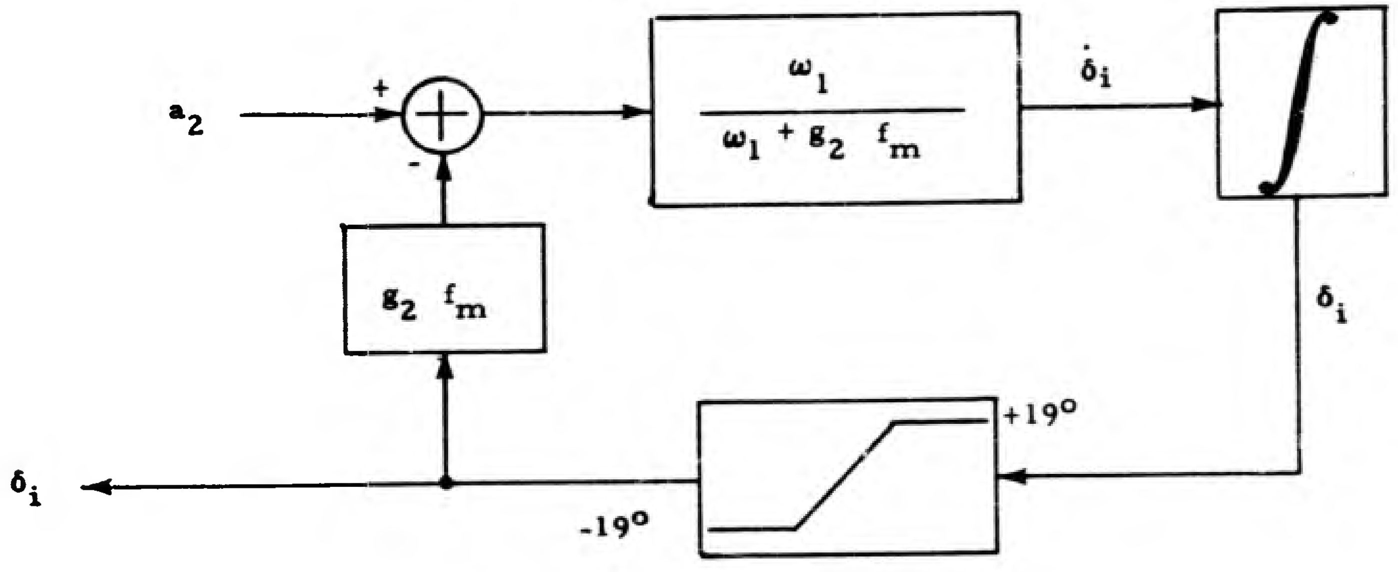


Figure 3.2.11 - Low Frequency Actuator (OPTACT≤0)

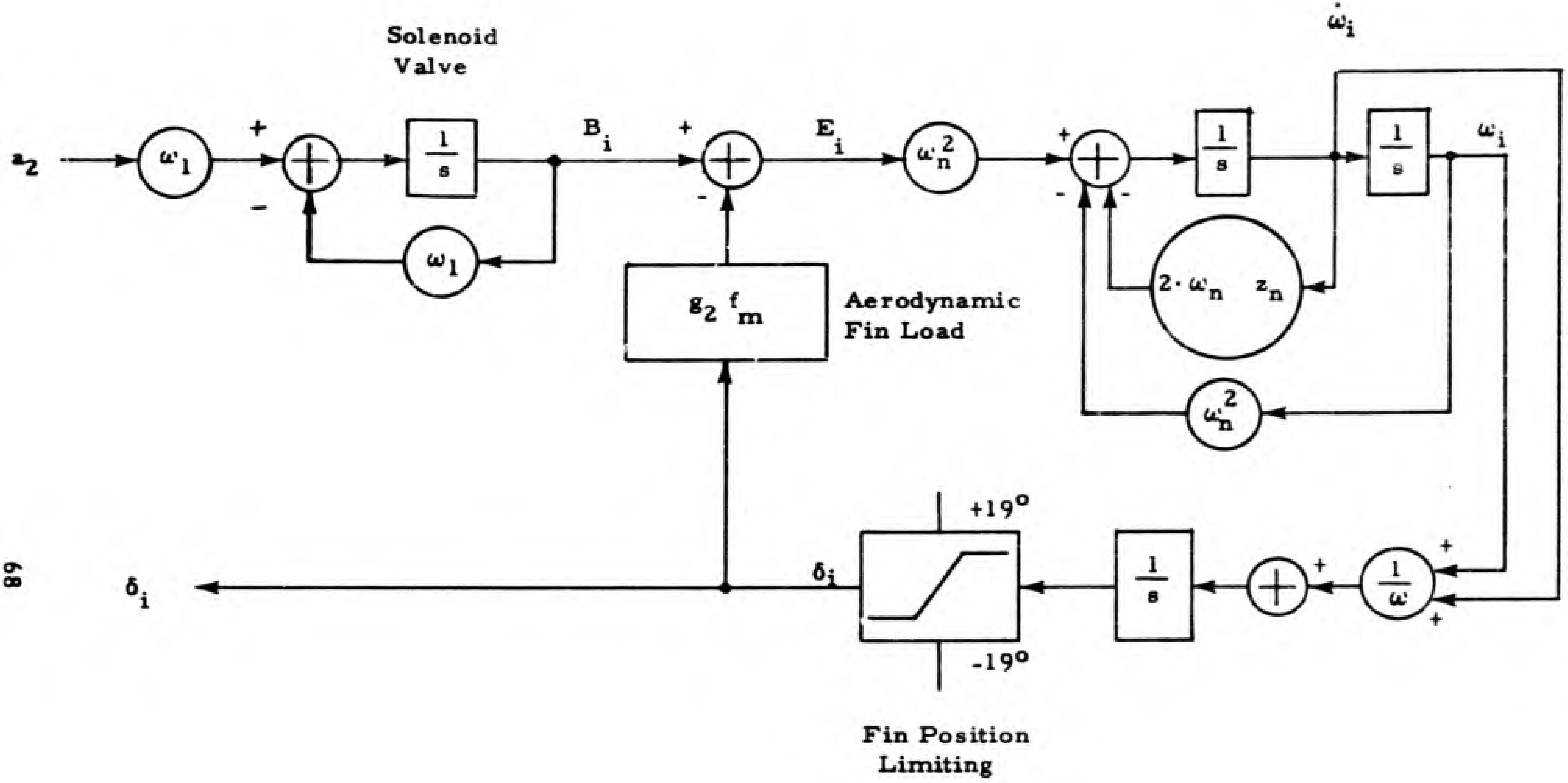


Figure 3.2.12 - High Frequency Actuator (OPTACT > 0)

however, do not represent identical systems. In particular, the portion of the diagrams past the point where the aerodynamic fin load feedback is summed is different.

Since the option under discussion (OPTACT > 0) has not been utilized, no attempt has been made to determine the reasons for the differences.

3.2.6.3 Initialization Subroutine

The Actuator Initialization Subroutine, C4I, moves the indicies of the "C" array which contain derivatives utilized in the Actuator Module into the array IPL(I). The array IPL(I) is used to point the numerical integration logic to the elements of the "C" array which are to be numerically integrated.

C4I also contains, as an entry point, the initialization routine for A1, the Aero Table Look-up Subroutine. This function is discussed in the Initialization Subroutine Section of Subroutine A1.

3.2.6.4 Input/Output Variables and Cross Reference of C-Array

1. Input from data cards:

FORTRAN SYMBOL	SYMBOL USED IN TEXT	C INDEX	DEFINITION
OPTACT	OPTACT	1140	Switch designating which of two computer actuator models to use
CR	C_r	1145	Actuator modulation gain
HDEL	HDEL	1147	Fin torque limit gain
W1	ω_1	1148	Solenoid valve bandwidth
ZN	z_n	1149	Damping of fin position loop
G1	g_1	1151	Actuator forward loop gain
BH	b_h	1152	Deadband
WN	ω_n	1153	Natural frequency of fin position loop
G2	g_2	1154	Fin aero load weighting factor
H1	h_1	1155	Positive fin torque limit
H2	h_2	1156	Negative fin torque limit
RFAREA	S_A	1306	Aerodynamic Coefficient Reference Area (FT^2)
RFLGTH	S_L	1307	Aerodynamic Coefficient Reference Length (FT)

2. Input from other modules

FORTRAN SYMBOL	SYMBOL USED IN TEXT	C INDEX	DEFINITION
PDYNMC	Q	203	Dynamic Pressure
VMACH	VMACH	204	Mach Number
BDELTC(1)	δ c1	856	Commanded Fin Position from autopilot, Fin #1
BDELTC(2)	δ c2	857	Commanded Fin Position from autopilot, Fin #2
BDELTC(3)	δ c3	858	Commanded Fin Position from autopilot, Fin #3
BDELTC(4)	δ c4	859	Commanded Fin Position from autopilot, Fin #4

3. Outputs

FORTRAN SYMBOL	SYMBOL USED IN TEXT	C INDEX	DEFINITION
BDELT(1)	$\delta 1$	1103	Actual Fin Position, Fin #1
BDELT(2)	$\delta 2$	1107	Actual Fin Position, Fin #2
BDELT(3)	$\delta 3$	1111	Actual Fin Position, Fin #3
BDELT(4)	$\delta 4$	1115	Actual Fin Position, Fin #4

3.2.7 D1 - Translational Dynamics Module

3.2.7.1 Function Description

The Translational Dynamics Module calculates translational accelerations of the missile in both the Earth and Body coordinate systems, and moves the missile acceleration components to the velocity integrator locations in the "C" array.

The missile velocity components obtained on the previous integration step are moved to the position integrator locations in the "C" array. Missile "g"-acceleration in Body Axes are obtained for information purposes. If the target motion option is selected on input (OPTARG > 0) target motion derivatives are also calculated based upon input values of linear acceleration and a turning term. The target acceleration and velocity components are moved to the target velocity and position locations in the "C" array.

The module logic is straightforward, except for the target motion logic. Several input parameters are available for specifying target motion during the simulation. One, ATHRST, represents simply the magnitude of the target acceleration vector. A second parameter, ATURNT, represents one variable in the calculation of the rate of change of the Earth plane azimuth angle ($\dot{\psi}$) of the velocity vector, where the second variable in the calculation is the instantaneous magnitude of the velocity vector itself. The initial azimuth angle (ψ_0) can be input, as can an elevation angle (γ_0)

where the sign conventions of these angles are shown in Figure 3.2.13.

ATHRST is input in g's to specify the magnitude of target acceleration along the instantaneous velocity vector. ATURNT is used in the equation for $\dot{\psi}$ as follows:

$$\dot{\psi} = \frac{\text{ATURNT}}{\text{VTARG}} \text{ (AGRAV) (CRAD).}$$

Thus, $\dot{\psi}$ is directly proportional to ATURNT. Since $\dot{\psi}$ is inversely proportional to VTARG (which will be increasing with time if ATHRST > 0), however, $\dot{\psi}$ will be constantly decreasing with time. Thus, $\dot{\psi}$ can be controlled on input for any desired value of VTARG, but the value of for all other times will be a function of VTARG. As an example of specifying an input value for ATURNT, suppose it is desired to have $\dot{\psi}$ = 1 deg/sec when VTARG = 1 ft/sec. Then, solving for ATURNT:

$$\text{ATURNT} = \frac{(\dot{\psi})\ \text{(VTARG)}}{\text{(AGRAV)}\ \ \text{(CRAD)}}$$

$$= \frac{\dfrac{1\ \text{deg}}{\text{sec}} \quad \dfrac{1\ \text{ft}}{\text{sec}}}{\dfrac{32\ \text{ft}}{\text{sec}^2\ \text{g}} \quad 57.3\ \text{deg}} \cong (5.47)\ (10^{-4})\ \text{g's}$$

The value of $\dot{\psi}$, however, will be larger for VTARG < 1 and smaller for VTARG > 1. A summary of the kinds of target motion that can be achieved with the input target motion parameters is shown in Table 3.2.2.

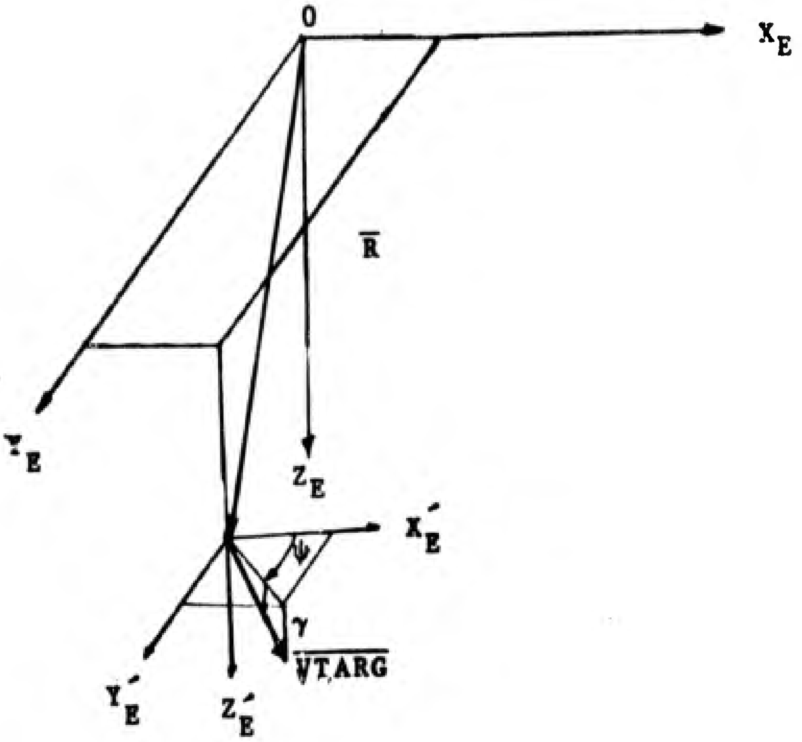


Figure 3.2.13 - Target Motion Velocity Vector*

---

*NOTE: A system (X_E', Y_E', Z_E'), parallel to the Earth Axis System, has been defined at the terminal point of the radius vector to emphasize the fact that γ and ψ specify the instantaneous direction of the target velocity vector, rather than the radius vector.

ATHRST	ATURNT	VTARG	γ	ψ	Target Motion*
0	0	0	-	-	None
0	0	>0	γ_o	ψ_o	Constant velocity of target away from origin in the direction specified by (γ_o, ψ_o).
> 0	0	V_o	γ_o	ψ_o	Constant acceleration of target away from origin in the direction (γ_o, ψ_o), with initial velocity V_o.
0	> 0	$V_o > 0$	γ_o	ψ_o	Constant magnitude of velocity, initially in the direction of (γ_o, ψ_o), with direction of velocity vector changing at a constant rate determined by ATURNT and V_o.
>0	> 0	V_o	γ_o	ψ_o	Constantly accelerating target away from origin in initial direction of (γ_o, ψ_o), but with ψ increasing (at a decreasing rate) with time.

---

* Note all of the above mentioned motions will automatically be initiated from the origin of the Earth Axes whenever OPTN2>0, because subroutine D1I sets the target position coordinates to zero. Also, some combinations are not shown because they would not result in target motion due to the manner in which the equations are written. Thus, for example, the combination of ATHRST=0, ATURNT>0, VTARG=0, (γ and ψ arbitrary) would result in no target motion.

3.2.7.2 Equations

Total Acceleration in Body Axes:

$$\begin{bmatrix} AXBA \\ AYBA \\ AZBA \end{bmatrix}_B = \frac{1}{DMASS} \begin{bmatrix} FXBA \\ FYBA \\ FZBA \end{bmatrix}_B$$

where:

FXBA, FYBA, and FZBA are calculated in the Forces and Moments Module (A2) and represent the sum of the aerodynamic and thrust forces, and DMASS is the instantaneous total vehicle mass calculated in the Engine Module (A3).

Total Acceleration in Earth Axes:

$$\begin{bmatrix} AXE \\ AYE \\ AZE \end{bmatrix}_E = M^T \begin{bmatrix} AXBA \\ AYBA \\ AZBA \end{bmatrix}_B$$

where M^T is the transpose of the Earth-to-Body transformation matrix, calculated in the Rotational Dynamics Module (D2).

Earth Axes Velocity Integrator Values

$$\begin{bmatrix} VXED \\ VYED \\ VZED \end{bmatrix} = \begin{bmatrix} AXE \\ AYE \\ AZE + AGRAV \end{bmatrix},$$

where AGRAV is the acceleration due to gravity. The "Velocity Integrator" locations in the "C" array (equivalenced to VXED, VYED, and VZED in the Translation Dynamics Module (D1)) are the actual "C" locations which are subsequently moved to the array DER(I) for use in the numerical integration routine AMRK.

Body Axes Velocity Integrator Values:

$$\begin{bmatrix} VDXB \\ VDYB \\ VDZB \end{bmatrix}_B = M \begin{bmatrix} VXED \\ VYED \\ VZED \end{bmatrix}_E$$

g-Accelerations in Body Axes:

$$\begin{bmatrix} ANGX \\ ANGY \\ ANGZ \end{bmatrix}_B = \frac{1}{g} \begin{bmatrix} VDXB \\ VDYB \\ VDZB \end{bmatrix}_B$$

The g-accelerations are calculated for information, and can be optionally printed.

Earth Axes Position Integrator Values:

$$\begin{bmatrix} RXED \\ RYED \\ RZED \end{bmatrix}_B = \begin{bmatrix} VXE \\ VYE \\ VZE \end{bmatrix}_B$$

Target Motion (Calculated if OPTARG > 0):

$$\text{VTARGD} \quad (\text{ATHRST})\ (\text{AGRAV}),$$

where VTARG is the target velocity integrator position in the "C" array, and ATHRST is the input value of target acceleration in g's.

$$\text{BPSITD} = (\text{ATURNT})\ (\text{AGRAV})\ (\text{CRAD})/\text{VTARG},$$

where BPSITD is the target velocity vector azimuth angle (see Figure 3. 2. 13), ATURNT is an input value (with units of g's) which partially determine the rate of change of the azimuth angle.

3. 2. 7. 3 Initialization Subroutine

The Translational Dynamics Initialization Subroutine (DII) calculates initial missile position and velocity components according to one of seven combinations of input option switches OPTN2 and OPTN4. Table 3. 2. 3 depicts the input variables required for each option, where the variables are defined as:

$$\left.\begin{array}{l}\text{RXE}\\ \text{RYE}\\ \text{RZE}\\ \text{VXE}\\ \text{VYE}\\ \text{VZE}\end{array}\right\} \quad \begin{array}{l}\text{Initial missile position and velocity}\\ \text{coordinates in Earth Axes System}\end{array}$$

BPSIO	Euler angles orienting Body Axes with respect to the Earth Axes (see Figure A-3.1 in Appendix A-3).
BTHTO	
BPHIO	
BTHTG	Initial gimble angles in pitch and yaw, respectively.
BPSIG	
RSLANT	Initial slant range from missile to target
RSLOV	Angle between Line-of-Sight and Earth plane. (Sign convention: BSLOV >0 when LOS is above the Earth plane.)
VWXE	Wind velocity components in Earth Axes.
VWYE	
VWZE	
VMACH	Initial missile mach number

Table 3.2.3. Input Data Options for Initial Values

	Unrestricted Inputs	Restricted Inputs					
		No Rail Dynamics		Rail Dynamics Included			
		Direct Fire		Direct Fire		Indirect Fire	
OPTN4	0	0	0	1	1	2	2
OPTN2	0	1	2	1	2	1	2
RXE	X	I	X	I	X	I	X
RYE	X	0	0	0	0	0	0
RZE	X	I	X	I	X	I	X
VXE	X	I	I	I	I	I	I
VYE	X	I	I	I	I	I	I
VZE	X	I	I	I	I	I	I
RTXE	X	0	0	0	0	0	0
RTYE	X	0	0	0	0	0	0
RTZE	X	0	0	0	0	0	0
BPSIO	X	I	I	X	X	X	X
BTHTO	X	I	I	X	X	X	X
BPHIO	X	0	0	0	0	0	0
BTHTG	X	X	X	I	I	X	X
BPSIG	X	X	X	I	I	X	X
RSLANT		X	I	X	I	X	I
BSLOV		X		X		X	
VWXE		X	X	X	X	X	X
VWYE		X	X	X	X	X	X
VWZE		X	X	X	X	X	X
(VMACH)* (VMWTE)		X	X	X	X	X	X

X - denotes variables that must be specified by input data
I - denotes variables that are computed internally
0 - denotes variables that are either set to zero initially or will be computed as zero.

*** Input VMACH if OPTN6=0; input VMWTE if OPTN6=1.**

3.2.7.4 Input/Output and Cross Reference of C-Array

1. Input from data cards

FORTRAN SYMBOL	SYMBOL USED IN TEXT	C INDEX	DEFINITION
AGRAV	AGRAV	1627	Acceleration due to gravity (ft/sec^2)
ATHRST	ATHRST	1629	Input constant target acceleration (g's)
ATURNT	ATURNT	1630	Target turning term
BGAMT	γ	1631	
OPTARG	OPTARG	1639	Target motion option switch (OPTARG >0 for target motion)
CRAD	CRAD	1751	Radians to degrees conversion

2. Input from other modules

FORTRAN SYMBOL	SYMBOL USED IN TEXT	C INDEX	DEFINITION
FXBA	FXBA	1300	Components of total forces acting on missile in Body Coordinate System (LBS)
FYBA	FYBA	1301	
FZBA	FZBA	1302	
CFA11	CFA11	1703	Elements of the Earth-to-Body transformation matrix, M.
CFA12	CFA12	1707	
CFA13	CFA13	1711	
CFA21	CFA21	1715	
CFA22	CFA22	1719	
CFA23	CFA23	1723	
CFA31	CFA31	1727	
CFA32	CFA32	1731	
CFA33	CFA33	1735	

3. Output

FORTRAN SYMBOL	SYMBOL USED IN TEXT	C INDEX	DEFINITION
VXE	VXE	1603	Components of missile velocity in Earth fixed coordinate system (FT-SEC)
VYE	VYE	1607	
VZE	VZE	1611	
RXE	RXE	1615	Components of missile position in Earth Fixed coordinate system (FT).
RYE	RYE	1619	
RZE	RZE	1620	

3.2.8 D2 - Rotational Dynamics Module

3.2.8.1 Function Description

The Rotational Dynamics Module calculates the numerical values for the derivatives of the vehicle rotation rates (ω_P, ω_Q, and ω_R), and the elements (C_{ij}, i, j = 1, 2, 3) of the Earth-to-Body axis transformation Matrix, M.

3.2.8.2 Equations

The vehicle rotation derivatives are calculated from values of the respective moments (calculated in the Forces and Moments Module (A2)) and input values of the moments of inertia. The equations are:

$$\dot{\omega}_P = \frac{M_x}{I_{xx}} \qquad \text{(Roll)}$$

$$\dot{\omega}_Q = \frac{M_y + (I_z - I_x)\ (\omega_P)\ (\omega_R)}{I_{yy}} \qquad \text{(Pitch)}$$

$$\dot{\omega}_R = \frac{M_z + (I_x - I_y)\ (\omega_P)\ (\omega_Q)}{I_{zz}} \qquad \text{(Yaw)}$$

where:

M_α, α = x, y, z, denotes the moment of force about x, y, and z Body axes, respectively.

$I_{\alpha\alpha}$, $\alpha\alpha$ = x, y, z, denotes the moment of inertia about the x, y, and z Body axes, respectively,

and ω_P, ω_Q, and ω_R are in radians/second.*

If OPTN3 >0 (input roll option switch, C(3503)), the calculation of $\dot{\omega}_P$ is not accomplished, and missile roll is not simulated (a constant roll rate is maintained).

The derivatives ($\dot{C}_{ij}$, i, j = 1, 2, 3) of the elements of the Earth to Body transformation Matrix, M, are calculated using the following equations:

$$\dot{C}_{11} = (C_{21})\ (\omega_R) - (C_{31})\ (\omega_Q)$$
$$\dot{C}_{12} = (C_{22})\ (\omega_R) - (C_{32})\ (\omega_Q)$$
$$\dot{C}_{13} = (C_{23})\ (\omega_R) - (C_{33})\ (\omega_Q)$$
$$\dot{C}_{21} = (C_{31})\ (\omega_P) - (C_{11})\ (\omega_R)$$
$$\dot{C}_{22} = (C_{32})\ (\omega_P) - (C_{12})\ (\omega_R)$$
$$\dot{C}_{23} = (C_{33})\ (\omega_P) - (C_{13})\ (\omega_R)$$
$$\dot{C}_{31} = (C_{11})\ (\omega_Q) - (C_{21})\ (\omega_P)$$
$$\dot{C}_{32} = (C_{12})\ (\omega_Q) - (C_{22})\ (\omega_P)$$
$$\dot{C}_{33} = (C_{13})\ (\omega_Q) - (C_{23})\ (\omega_P)$$

---

* Note that in the 6DOF simulation program, ω_P, ω_Q, and ω_R are in degrees/sec., so the factor CRAD = 57.295778 appears where appropriate.

3.2.8.3 Assumptions

The derivation of $\dot{\omega}_P$, $\dot{\omega}_Q$, and $\dot{\omega}_R$ requires the assumptions that: (1) the products of inertia (I_{xy}, I_{xz}, I_{yz}, etc.) are zero and (2) the time derivatives of the principal moments of inertia ($\dot{I}_{xx}$, $\dot{I}_{yy}$, $\dot{I}_{zz}$) are zero. The point at which these assumptions are required in deriving these equations is discussed in Appendix A-1.

A derivation of the matrix element derivations, ($\dot{C}ij$), is accomplished in Appendix A-2. No limiting assumptions, other than that the Earth fixed coordinate system is inertial, are required in the derivation.

3.2.8.4 Initialization Subroutines

A Rotational Dynamics Initialization subroutine (D21), calculates the initial Earth to Body axis transformation Matrix, M, from the input initial Euler angles ψ, θ, and Φ, which are defined as follows:

1. A positive rotation (toward Y_E from X_E) of ψ (degrees) about the Z_E axis, giving a system e' (X', Y', Z').

2. A negative rotation (the X' axis up from the plane of the Earth axis system) of θ about the Y' axis, giving a system e^*(X^*, Y^*, Z^*),

3. A positive rotation (toward Z^* from Y^*) of Φ degrees about the X^* axis, given the Body Axis system E_B(X_B, Y_B, Z_B).

The angles are depicted in the diagram below, where (X_E, Y_E, Z_E) and X_B, Y_B, Z_B) represent the Earth and Body axes, respectively.

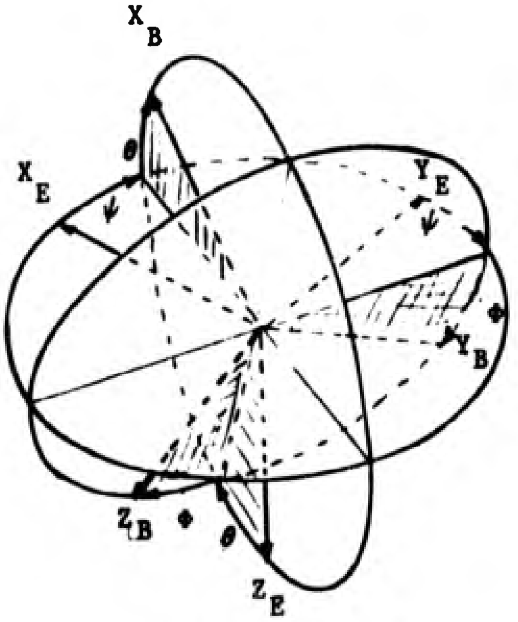


Figure 3. 2. 14

The resulting transformation, M, from the Earth axis system to the Body axis system is given by:

$$M = \begin{bmatrix} \text{COS } \psi \text{ COS } \theta & \text{SIN } \psi \text{ COS } \theta & -\text{SIN } \theta \\ \text{COS } \psi \text{ SIN } \theta \text{ SIN } \Phi & \text{COS } \psi \text{ COS } \Phi & \text{COS } \theta \text{ SIN } \Phi \\ -\text{SIN } \psi \text{ COS } \Phi & +\text{SIN } \psi \text{ SIN } \theta \text{ SIN } \Phi & \\ \text{COS } \psi \text{ SIN } \theta \text{ COS } \Phi & \text{SIN } \psi \text{ SIN } \theta \text{ COS } \Phi & \text{COS } \theta \text{ COS } \Phi \\ +\text{SIN } \psi \text{ SIN } \Phi & -\text{COS } \psi \text{ SIN } \Phi & \end{bmatrix}$$

Thus, a vector $\overline{V}_E$ expressed in the Earth axis system is given in the Body axis system as $\overline{V}_B = M\overline{V}_E$.

The transformation is derived for reference in Appendix A-3.

3.2.8.5 Input/Output and Cross Reference of C-Array

1. Input from data cards

FORTRAN SYMBOL	SYMBOL USED IN TEXT	C INDEX	DEFINITION
CRAD	-----	1751	Radians to degrees conversion factor

2. Inputs from other modules

FORTRAN SYMBOL	SYMBOL USED IN TEXT	C INDEX	DEFINITION
FMIX	FMIX	1748	Instantaneous values of moments of inertia (SLUG-FT^2)
FMIY	FMIY	1749	
FMIZ	FMIZ	1750	
FMXBA	FMXBA	1303	Components of the total moments of force about the body coordinate system (FT-LBS)
FMYBA	FMYBA	1304	
FMZBA	FMZBA	1305	

3. Output

FORTRAN SYMBOL	SYMBOL USED IN TEXT	C INDEX	DEFINITION
CFA11	C_{11}	1703	Elements of the Earth-to-Body transformation matrix, M
CFA12	C_{12}	1707	
CFA13	C_{13}	1711	
CFA21	C_{21}	1715	
CFA22	C_{22}	1719	
CFA23	C_{23}	1723	
CFA31	C_{31}	1727	
CFA32	C_{32}	1731	
CFA33	C_{33}	1735	
WP	ω_P	1739	Roll rate about X_B - axis (DEG/SEC)
WQ	ω_Q	1743	Pitch rate about Y_B - axis (DEG/SEC)
WR	ω_R	1747	Yaw rate about Z_B - axis (DEG/SEC)

3.2.9 G2-WINDS

3.2.9.1 Function Description

The velocity components of a wind of constant speed and direction are computed. The velocity components are referred to the earth fixed coordinate system. The winds are parallel to a flat earth with the direction and magnitude specified by input data. The wind components are set to zero until the vehicle leaves the launch rail and are assumed to be zero above a certain altitude specified by input data.

3.2.9.2 Equations

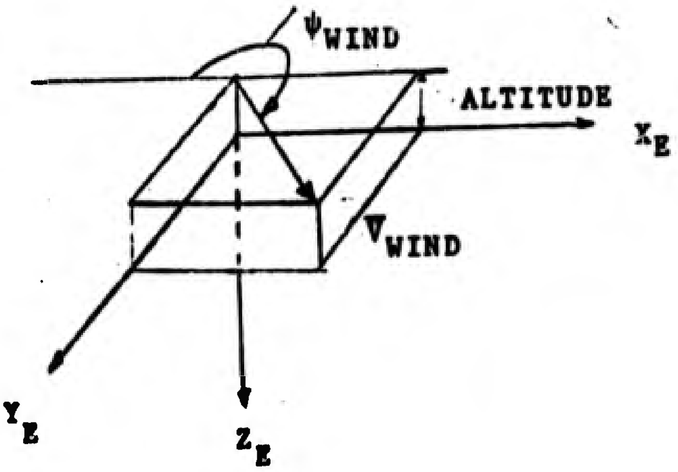


Figure 3.2.15 - Wind Components

$$V_{XWIND} = -V_{WIND} \cos(\psi_{WIND})$$

$$V_{YWIND} = -V_{WIND} \sin(\psi_{WIND})$$

$$V_{ZWIND} = 0$$

3.2.9.3 Initialization Subroutine

No initialization module exists for G2. However, due to the manner in which the initial value of the missile velocity relative to the ground (VXE, VYE, VZE) is computed in module D1I, it is necessary for the program to have the wind velocity component initial values before executing D1I. Since no wind initialization module currently exists, and since module G2 is called after module D1I, it is mandatory that the wind component initial values (V_{XWIND}, V_{YWIND}, V_{ZWIND}) be input on Type 3 data cards. (To be consistent with the ground rule that the wind is a horizontal wind, the z-component must be input as zero or left alone.)

3.2.9.4 Input/Output and Cross Reference of C-Array

1. Input from data cards

FORTRAN SYMBOL	SYMBOL USED IN TEXT	C INDEX	DEFINITION
OPTNW		50	Wind option selector 0. - no winds 1. - include steady winds
BPSIW	ψ_{WIND}	51	Wind direction: angle between wind velocity vector and the negative X-axis (DEG)
VWTE	V_{WIND}	52	Wind speed with respect to the ground (FT/SEC)
RHW		53	Altitude above which all wind speeds are zero (FT)

2. Output

FORTRAN SYMBOL	SYMBOL USED IN TEXT	C INDEX	DEFINITION
VWXE	V_{XWIND}	100	Components of the wind velocity relative to the Earth fixed coordinate system (FT/SEC)
VWYE	V_{YWIND}	101	
VWZE	V_{ZWIND}	102	

3.2.10 G3-AIR DATA

3.2.10.1 Function Description

Subroutine G3 computes the missile velocity with respect to the wind, and computes two atmospheric properties, density and speed of sound. The magnitude of the missile velocity, the atmospheric density and speed of sound are then used to compute mach number and dynamic pressure.

3.2.10.2 Equations

Missile Velocity with respect to Wind

$$\bar{V}_{MW} = \bar{V}_{MISSILE} - \bar{V}_{WIND}$$

Missile Altitude

$$h = -R_{ZE} + R_{ZO}$$

where

R_{ZE} - Z-component of missile position in Earth coordinates

R_{ZO} - Altitude of Earth coordinate system above ground

Atmospheric Density

$$\rho = \frac{\rho_O}{1 + kh + k_1 h^3}$$

where

$$\rho_0 = .076475$$

$$k = .3325 \times 10^{-4}$$

$$k_1 = .02315 \times 10^{-12} \quad .$$

Atmospheric Speed of Sound

$$a = k_2 h + a_o,$$

where

$$a_o = 1117.3$$

$$k_2 = -.00392 \quad .$$

Dynamic Pressure

$$Q = 1/2 \rho \left| \bar{V}_{MW} \right|^2 / g_o \quad .$$

Mach Number

$$M = \frac{\left| \bar{V}_{MW} \right|}{a} \quad .$$

3.2.10.3 Input/Output and Cross Reference of C-Array

1. Input from data cards

FORTRAN SYMBOL	SYMBOL USED IN TEXT	C INDEX	DEFINITION
RHZRO	R_{ZO}	208	Altitude of Earth fixed coordinate system above ground (FT)

2. Input from other modules

FORTRAN SYMBOL	SYMBOL USED IN TEXT	C INDEX	DEFINITION
VWXE	$\overline{V}_{WIND}$	100	Wind velocity components in Earth fixed coordinate system (FT/SEC)
VWYE		101	
VWZE		102	
VXE	$\overline{V}_{MISSILE}$	1603	Velocity components of missile in Earth fixed coordinate system (FT/SEC)
VYE		1607	
VZE		1611	
RZE	R_{ZE}	1623	Z-component of missile position in Earth fixed coordinate system (FT)

3. Output

FORTRAN SYMBOL	SYMBOL USED IN TEXT	C INDEX	DEFINITION
VMWXE VMWYE VMWZE	$\bar{V}_{MW}$	200 201 202	Components of missile velocity relative to the wind in the Earth fixed coordinate system (FT/SEC)
PDYNMC	Q	203	Dynamic pressure (LBS/FT^2)
VMACH	M	204	Mach Number
DRHO	ρ	205	Atmospheric density (LBS/FT^3)
VSOUND	a	206	Atmospheric speed of sound (FT/SEC)
VAIRSP	$\|\bar{V}_{MW}\|$	207	Magnitude of missile velocity relative to wind (airspeed) (FT/SEC)
RH	h	209	Altitude of missile above ground (FT)

3.2.11 G4-TERMINAL GEOMETRY

3.2.11.1 Function Description

This subroutine monitors the LOS vector $\overline{\Delta R}(t)$ at the end of each integration step to determine if the point of closest approach (target plane intersection) to the target has been passed. If the closest approach point has been passed, integration is halted and the time of closest approach (t_o), the position of the target with respect to the missile ($\overline{\Delta R_o}$), and the miss distance (RMISS) are computed. After computation of the miss distance, LCONV is set to 2 to flag subroutine STGE3 that the trajectory computation is to be terminated.

The time and position at closest approach are computed by linear interpolating the time (t_1) and position ($\overline{\Delta R_1}$) values just prior to closest approach and the time (t_2) and position ($\Delta\overline{R}_2$) values following closest approach. The time and position values following closest approach exist as current values of time (t) and position ($\overline{\Delta R}$) on the trajectory, while the time (t_1) and position ($\overline{\Delta R_1}$) just prior to closest approach were saved on the previous pass-through this subroutine.

G4 also monitors the altitude (RDELZ) of the missile relative to the target. If the altitude becomes negative, it is assumed that ground impact has occurred. If ground impact does occur, the run termination switch, LCONV, is set to 2 to flag STGE3 to terminate trajectory computation.

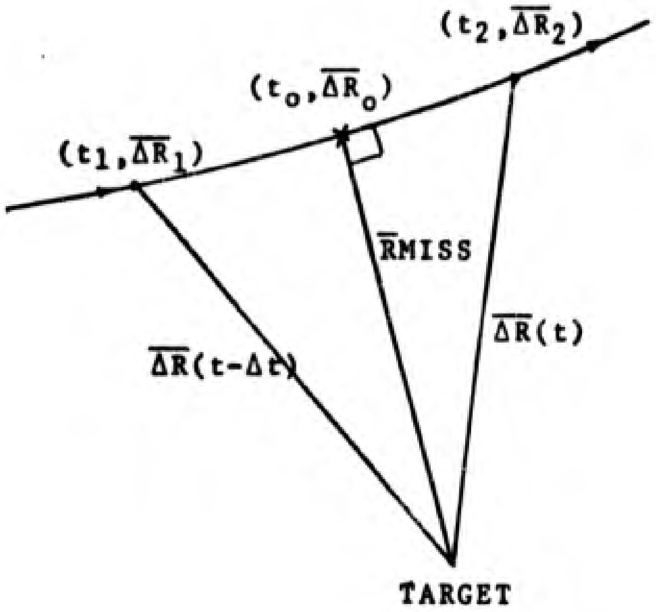


Figure 3.2.16

3.2.11.2 Equations

To monitor the LOS vector, the position of the target is first transformed into a flight path coordinate system aligned with missile velocity by

$$\overline{RFP} = [M] \ \overline{\Delta R}$$

where

$$\overline{\Delta R} = \begin{bmatrix} \Delta X \\ \Delta Y \\ \Delta Z \end{bmatrix}$$

position of target with respect to missile in a system parallel to the Earth fixed coordinate system.

$$\overline{RFP} = \begin{bmatrix} XFP \\ YFP \\ ZFP \end{bmatrix}$$

position of target with respect to missile in flight plane coordinate system.

and

$$[M] = \begin{bmatrix} COS\gamma_H COS\gamma_V & SIN\gamma_H COS\gamma_V & -SIN\gamma_V \\ -SIN\gamma_H & COS\gamma_H & 0 \\ COS\gamma_H SIN\gamma_H & SIN\gamma_H SIN\gamma_H & COS\gamma_H \end{bmatrix} .$$

The above transformation involves a rotation from the Earth fixed axes through the heading angle γ_H and the flight path angle γ_y to the flight plane coordinate system as shown in Figure 3.2.17.

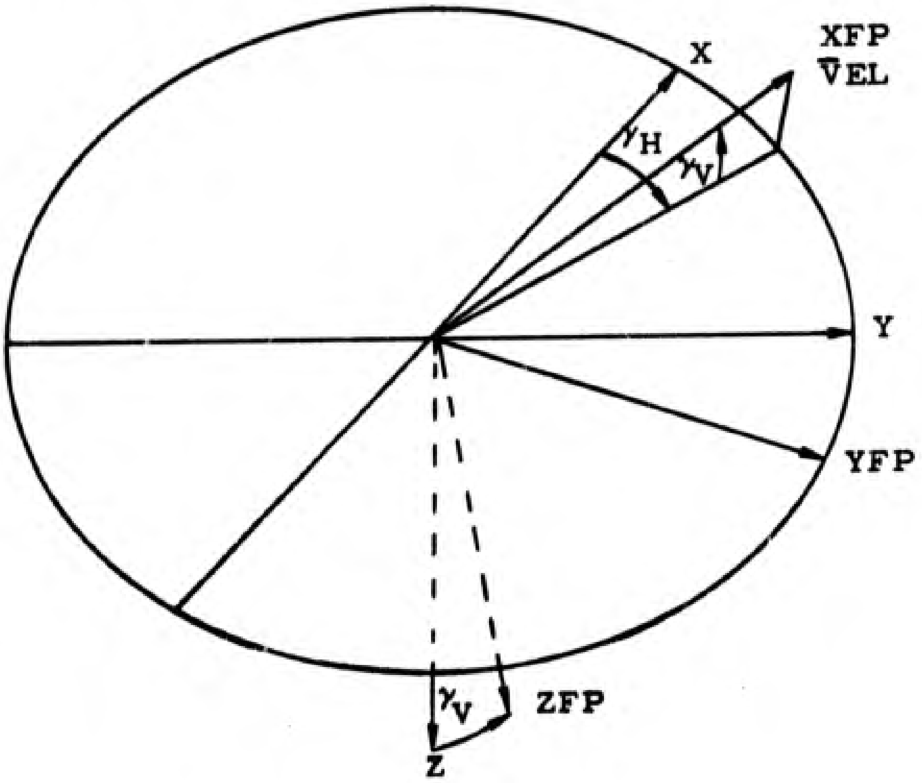


Figure 3.2.17 - Flight Plane Coordinate System

Transforming the LOS vector into the flight plane coordinate system simplifies the determination of closest approach passage, because the X-component (XFP) of the LOS will be zero at closest approach. This condition exists because the X-axis (XFP) is coincident with the velocity vector and the LOS is normal to the velocity at the closest approach point.

To determine closest approach passage, the value of XFP is monitored. A positive XFP means the target lies ahead of the missile. A negative XFP means the target lies behind the missile. When the value of XFP changes sign from one integration step to the next, closest approach has been bracketed. At this time, integration is halted and the time (t_o) and position ($\overline{\Delta R}_o$) of closest approach is computed by interpolation. Interpolation is performed using XFP as the independent variable. Thus,

$$t_o = t_1 + P_{CT} (t_2 - t_1)$$

$$\overline{\Delta R}_o = \overline{\Delta R}_1 + P_{CT} (\overline{\Delta R}_2 - \overline{\Delta R}_1)$$

and

$$\overline{RFP}_o = \overline{RFP}_1 + P_{CT} (\overline{RFP}_2 - \overline{RFP}_1)$$

where

$$P_{CT} = \frac{XFP_o - XFP_1}{XFP_2 - XFP_1}$$

and since $XFP_o = 0$,

$$P_{CT} = \frac{-XFP_1}{XFP_2 - XFP_1}$$

3.2.11.3 Assumptions and Limitations

1. The LOS is not monitored until a slant range becomes less than 500 feet.

2. For a slant range greater than 500 feet, the Z-component (RDELZ) of the target with respect to the missile is monitored to determine ground impact.

3. The flight plane coordinate system is aligned with the missile earth fixed velocity vector. If appreciable target motion exists, then the condition that the X-component (XFP) of $\overline{RFP}$ is equal to zero at the point of closest approach to the target is not precisely true.

3.2.11.4 Input/Output and Cross Reference of C-Array

1. Input from other modules

FORTRAN SYMBOL	SYMBOL USED IN TEXT	C INDEX	DEFINITION
BGAMH	γH	357	Horizontal flight path angle (heading)(DEG)
BGAMV	γV	358	Vertical flight path angle (DEG)
RANGE		371	Distance from target (FT)
RDELX	ΔX	1635	Position components of target (LOS)
RDELY	ΔY	1636	with respect to the missile in
RDELZ	ΔZ	1637	Earth fixed coordinate system (FT)
T	t	2000	Flight time (SEC)

2. Output

FORTRAN SYMBOL	SYMBOL USED IN TEXT	C INDEX	DEFINITION
LCONV	LCONV	2020	Integration termination switch
RMISS	RMISS	300	Miss distance (measured in target plane) (FT)
RXF	XFP	301	Position components of target with respect to missile in the flight plane coordinate system (FT)
RYF	YFP	302	
RZF	ZFP	303	

After bracketing the closest approach point and interpolation for the time and position, the following parameters are written out by this subroutine.

RMISS	Miss distance (FT)
t_o	Flight time at closest approach (SEC)
RDELX (ΔX) RDELY (ΔY) RDELZ (ΔZ)	Position components of the target with respect to the missile in the system centered at the missile and parallel to the Earth fixed frame (FT).
RYFP (YFP) RZFP (ZFP)	Position components of the target in the flight plane frame (the X-component is not output since it is assumed to be zero). (FT)

3.2.12 G5 - COORDINATE CONVERSION MODULE

3.2.12.1 Function Description

G5 computes all the angles pertaining to the position and velocity of the vehicle; the proportional navigation angles, and look vector angles; and makes three coordinate transformations. The components of the spin axes of the two degree of freedom gyros (inertial platform) are computed in the Earth fixed coordinate system along with the two Euler angles defining the orientation of each gyro relative to the body coordinate system. The angles computed and the transformations made in this subroutine are:

1. Euler angles of the body coordinate system relative to the Earth fixed coordinate system.
2. Inertial platform gyro spin axis position components and Euler angles.
3. Target position with respect to the missile; missile total velocity; and missile position with respect to the moving rail launcher.
4. Transformation of missile LOS from Earth fixed coordinate system into the body coordinate system.
5. Vertical and horizontal line-of-sight angles.

6. Vertical and horizontal proportional navigation angles.

7. Vertical and horizontal flight path angles of the missile with respect to the Earth fixed axes.

8. Transformation of the velocity relative to the air from the Earth fixed coordinate system into the body coordinate system.

9. Angle of attack with respect to the wind.

   a) Vertical (α) and horizontal (β) angles of attack.

   b) Alpha prime (α' - absolute angle of attack) and phi prime (Φ' - aerodynamic roll angle with respect to wind) angles (wind tunnel axes).

3.2.12.2 Equations

1. Euler angles of the Body Coordinate System

The three Euler angles (ψ, θ, Φ) are computed from elements of the Earth-to-Body transformation matrix [M]. The matrix is computed in D2, the rotational dynamics subroutine, and passed into this subroutine by the COMMON BLOCK C. The matrix is defined as

$$M = \begin{bmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{bmatrix}$$

The relationship between each element value and the three Euler angles is given in Appendix A-3. It can be seen from these relationships that

$$\Phi = \tan^{-1} \frac{C_{23}}{C_{33}} \qquad \text{(Euler roll)}$$

$$\theta = \tan^{-1} \frac{-C_{13}}{\sqrt{C_{11}^2 + C_{12}^2}} \qquad \text{(Euler pitch)}$$

$$\psi = \tan^{-1} \frac{C_{12}}{C_{11}} \qquad \text{(Euler yaw)}$$

2. Inertial Platform Position Components and Euler Angles

The inertial platform is composed of two degree of freedom gyros. Gyro number one is oriented with its spin axis aligned with the initial y-body axis of the vehicle. This gyro detects vehicle roll and yaw. Gyro number two is oriented with ist spin axis aligned with the initial z-body axis of the vehicle. This gyro detects pitch and roll. The gyros axes are oriented as shown in Figure 3.2.18.

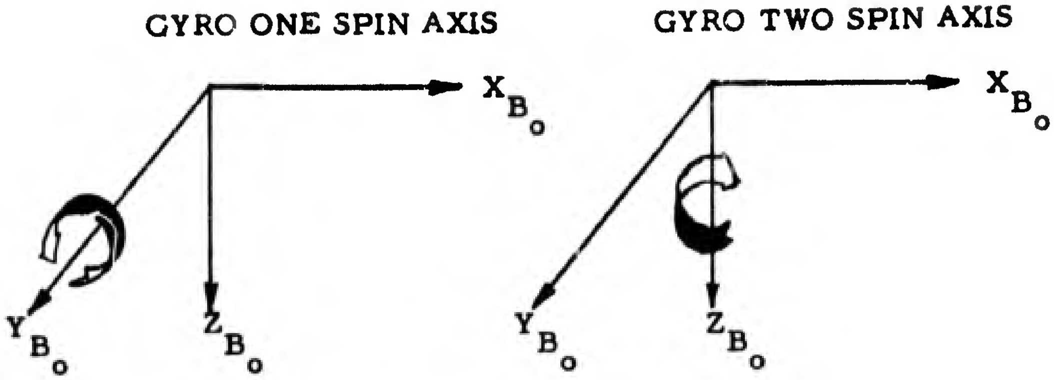


Figure 3.2.18 - Initial Orientation of The Two 2-DOF Gyros

Gyro initial orientation matrices are set in D2I, the rotational dynamics initialization subroutine. The gyro one matrix is then defined as:

$$Ao = \begin{bmatrix} Ao_{11} & Ao_{12} & Ao_{13} \\ Ao_{21} & Ao_{22} & Ao_{23} \\ Ao_{31} & Ao_{32} & Ao_{33} \end{bmatrix}$$

where

$$\begin{bmatrix} Ao_{11} & Ao_{12} & Ao_{13} \\ Ao_{21} & Ao_{22} & Ao_{23} \\ Ao_{31} & Ao_{32} & Ao_{33} \end{bmatrix} = \begin{bmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{bmatrix}_{@\ t = 0}$$

The elements are taken from the Earth-to-Body coordinate system transformation matrix [M] computed at the initial time point or, $[Ao] = [M]_{@\ t=0}$. This subroutine (G5) transforms the spin axis of each gyro from the initial body coordinate system through the Earth fixed system to the current body coordinate system and then computes the four angles detected by the two gryos. Thus,

$$\underset{\text{GYRO}}{\begin{bmatrix} X_B \\ Y_B \\ Z_B \end{bmatrix}} = [M]\ [Ao]^T \underset{\text{GYRO}}{\begin{bmatrix} X_{Bo} \\ Y_{Bo} \\ Z_{Bo} \end{bmatrix}}$$

Substituting the components of gyro one ($X_{Bo} = 0$, $Y_{Bo} = 1$, $Z_{Bo} = 0$) into the above equation gives

$$\begin{bmatrix} X_B \\ Y_B \\ Z_B \end{bmatrix}_1 = \begin{bmatrix} C_{11}Ao_{21} + C_{12}Ao_{22} + C_{13}Ao_{23} \\ C_{21}Ao_{21} + C_{22}Ao_{22} + C_{23}Ao_{23} \\ C_{31}Ao_{21} + C_{32}Ao_{22} + C_{33}Ao_{23} \end{bmatrix} ,$$

and substituting the components of gyro two ($X_{Bo} = 0$, $Y_{Bo} = 0$, $Z_{Bo} = 1$) into the same equation gives

$$\begin{bmatrix} X_B \\ Y_B \\ Z_B \end{bmatrix}_2 = \begin{bmatrix} C_{11}Ao_{31} + C_{12}Ao_{32} + C_{13}Ao_{33} \\ C_{21}Ao_{31} + C_{22}Ao_{32} + C_{23}Ao_{33} \\ C_{31}Ao_{31} + C_{32}Ao_{32} + C_{33}Ao_{33} \end{bmatrix} .$$

The two gyro-one Euler angles are shown in Figures 3.2.19 and 3.2.21 and are computed by

$$\Phi_1 = \tan^{-1}\left(\frac{Z_{B1}}{Y_{B1}}\right)$$

and

$$\Psi_1 = \tan^{-1}\left(\frac{-X_{B1}}{\left(\frac{Y_{B1}}{\cos\Phi_1}\right)}\right)$$

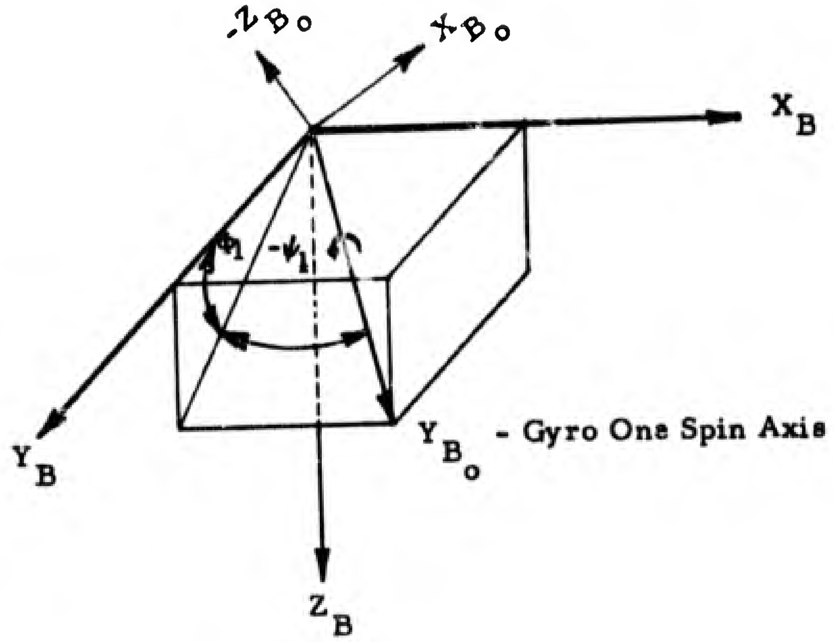


Figure 3.2.19 - Gyro-One Euler Angles

The two gyro-two Euler angels are shown in Figures 3.2.20 and 3.2.21 and are computed by

$$\theta_2 = \tan^{-1}\left(\frac{X_{B2}}{Z_{B2}}\right)$$

and

$$\Phi_2 = \tan^{-1}\left(\frac{-Y_{B2}}{\left(\frac{Z_{B2}}{\cos\theta_2}\right)}\right).$$

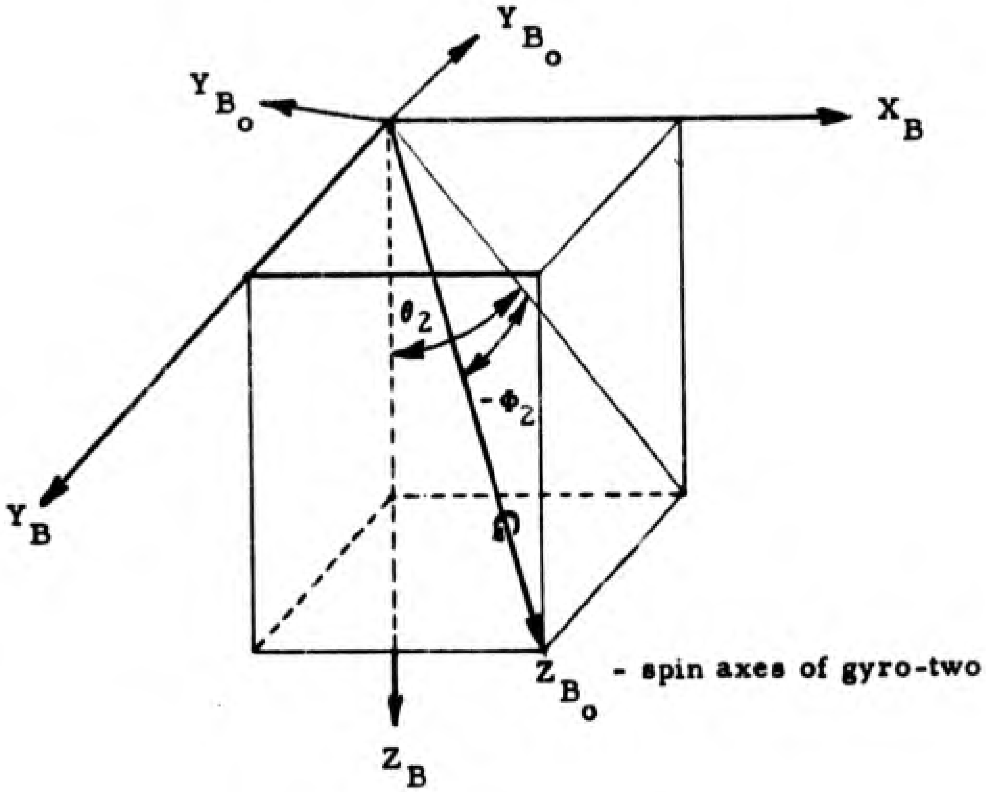


Figure 3.2.20 - Gyro-Two Euler Angles

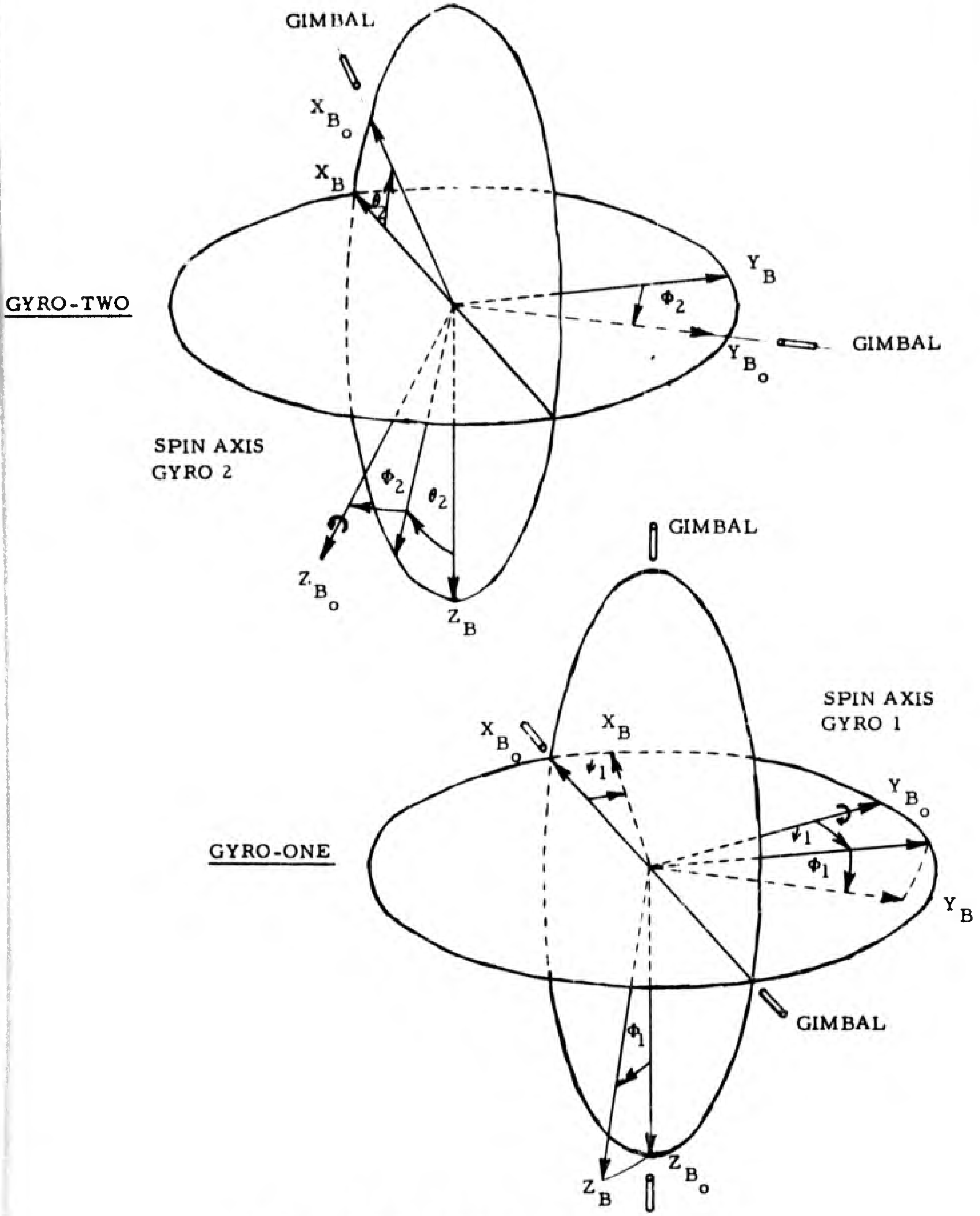


Figure 3.2.21 - 2DOF Gyro Angle Detection

3. Target Position (LOS vector) with Respect to the Missile; Missile total velocity; and Missile Position with Respect to the Moving Rail Launcher

Missile total velocity in the Earth fixed coordinate system is

$$V_{TOTE} = \sqrt{V_{XE}^2 + V_{YE}^2 + V_{ZE}^2}$$

Target position (LOS vector) with respect to the missile (Figure 3. 2. 22)

is

$$\overline{\Delta R}_{LOS} = \overline{R}_T - \overline{R}_E \qquad \text{(LOS - line-of-sight vector)}$$

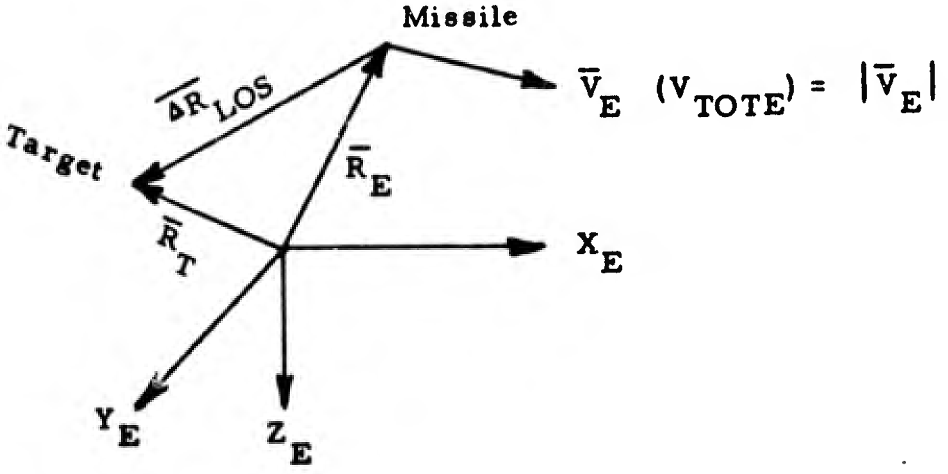


Figure 3. 2. 22 - Earth Fixed Coordinate System

Missile Position with respect to the moving rail launcher

To determine missile separation from the launch rail, the position and motion of the rail is required along with the position and motion of the missile. The motion of the rail is assumed to be composed of a constant linear velocity with respect to the Earth ($\overline{V}_o$, the missile initial velocity, see Figure 3.2.23) and an oscillatory and linear acceleration with respect to the vehicle initial body coordinate system ($\ddot{Z}_{RAIL}$). The initial position of the rail is assumed to be $\overline{R}_o$, the initial position of the missile. Thus, the position of the launcher as a function of time in the Earth fixed coordinate system is

$$\overline{R}_L = \overline{R}_o + \overline{V}_o t + \Delta\overline{R}_{os_L} ,$$

where the launcher position ($\Delta\overline{R}_{os_L}$) in the Earth frame due to the oscillatory motion is

$$\Delta\overline{R}_{os_L} = [M]_o \, \Delta\overline{R}_{os} \quad ,$$

and $[M]_o$ is the Earth-to-Body transformation matrix computed at the initial time point, or

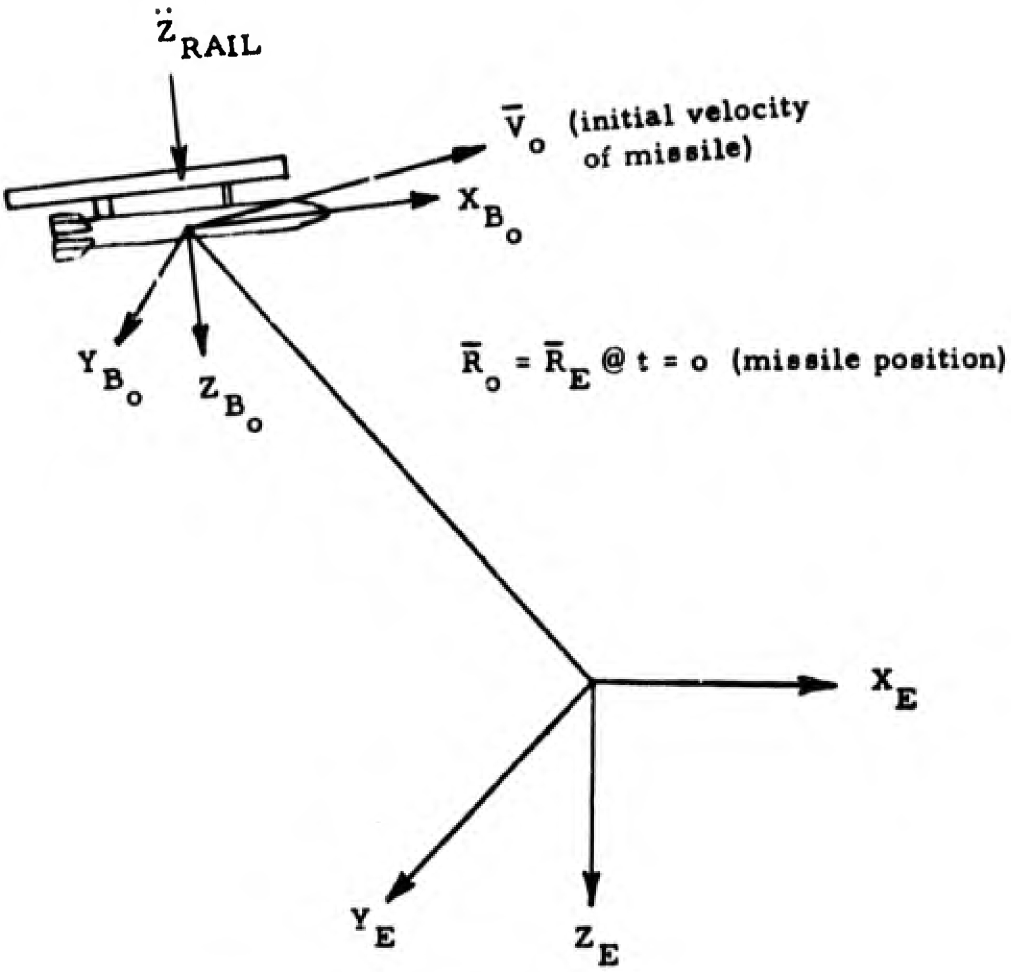


Figure 3.2.23 - Launcher Initial Position and Velocity

$$[M]_o = \begin{bmatrix} Ao_{11} & Ao_{12} & Ao_{13} \\ Ao_{21} & Ao_{22} & Ao_{23} \\ Ao_{31} & Ao_{32} & Ao_{33} \end{bmatrix} .$$

$\bar{R}_{os}$ is the position change of the launcher in the vehicle Body coordinate system due to oscillatory motion. The assumption is that all oscillatory motion occurs along the Z-component of the Body axis, thus the displacement of the launcher is along Z, or

$$\Delta \bar{R}_{os} = \Delta \bar{Z}_{RAIL} .$$

The oscillatory acceleration of the rail launcher is assumed to be along the Z-Body axis only, and is described by the waveshape given in Figure 3.2.24. The equation fitting the waveshape of Figure 3.2.24 is

$$\ddot{z}_{RAIL} = A \sin (\omega t + X) + A_2 .$$

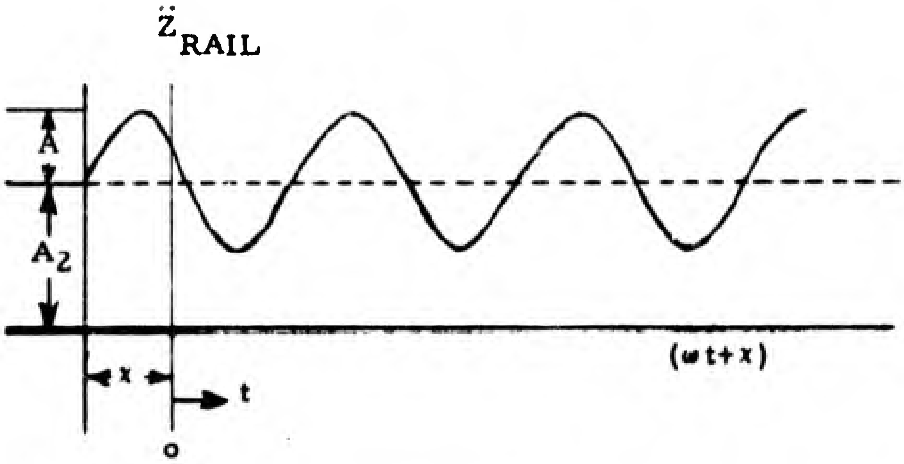


Figure 3.2.24 - Launcher Oscillatory Acceleration Profile

To determine rail displacement due to oscillation, the above equation is integrated twice to give

$$\Delta Z_{RAIL} = - \frac{A}{\omega^2} \sin(\omega t + \chi) + \frac{1}{2} A_2 t^2 + A_3 t + A_4 \ .$$

The constants of integration are determined based on the following assumptions:

1. $\Delta Z_{RAIL} = 0 \ @ \ t = 0$

   $\therefore A_4 = \frac{A}{\omega^2} \sin(\chi)$

2. $A_3 = 0$

3. $$\left.\begin{array}{lll} A & - & \text{amplitude of oscillation} \\ \omega & - & \text{frequency of oscillation} \\ \chi & - & \text{phase of oscillation at } t = 0 \\ A_2 & - & \text{linear acceleration component} \\ & & \text{of oscillation} \end{array}\right\} \text{Input Quantities}$$

Substituting the above assumptions into the ΔZ_{RAIL} equation gives

$$\Delta Z_{RAIL} = \frac{A}{\omega^2} [\sin\chi - \sin(\omega t + \chi)] + \frac{1}{2} A_2 t^2 \ .$$

Thus, launcher position in the Earth fixed frame then becomes

$$\bar{R}_L = \bar{R}_o + \bar{V}_o t + [M]_o \begin{bmatrix} 0 \\ 0 \\ \Delta Z_{RAIL} \end{bmatrix} .$$

Missile position with respect to the launcher then becomes

$$\Delta \bar{\bar{R}}_{ML} = \bar{R}_E - \bar{R}_L ,$$

with the magnitude of the separation being

$$R_{ANG0} = \sqrt{\Delta X^2_{ML} + \Delta Y^2_{ML} + \Delta Z^2_{ML}} .$$

4. Transformation of the Missile LOS from the Earth Fixed Coordinate System into the Body Coordinate System

$$\Delta \bar{R}_{B_{LOS}} = [M] \Delta \bar{R}_{LOS} . \quad \text{(LOS in Body System)}$$

5. Vertical and horizontal line-of-sight (LOS) angles (Figure 3.2.25)

$$\lambda_H = \tan^{-1} \frac{-\Delta Y_{LOS}}{\Delta X_{LOS}}$$

$$\lambda_V = \tan^{-1} \frac{-\Delta Z_{LOS}}{\sqrt{\Delta X^2_{LOS} + \Delta Y^2_{LOS}}}$$

Slant range from missile to the target is given as

$$\text{RANGE} = \sqrt{\Delta X^2_{LOS} + \Delta Y^2_{LOS} + \Delta Z^2_{LOS}} \ .$$

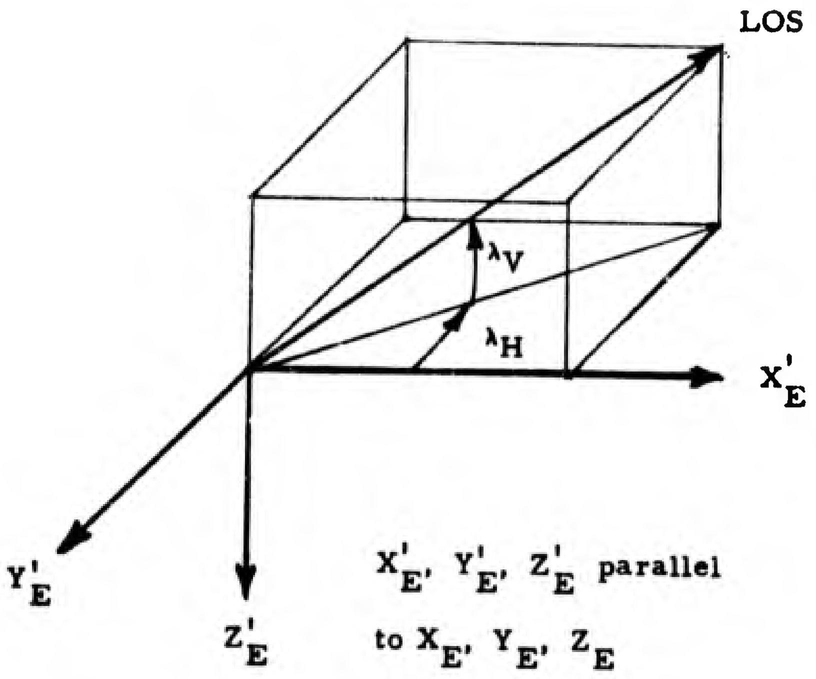


Figure 3.2.25 - Vertical and Horizontal LOS Angles

6. Vertical and Horizontal Proportional Navigation

Vertical and horizontal proportional navigation angles are defined in the coordinate system shown in Figure 3.2.26. The vertical and horizontal proportional navigation angles gives the orientation of the missile velocity vector relative to the line-of-sight vector (LOS). In deriving these angles, the assumption is made that the velocity of the target is negligible with respect to the missile velocity.

This coordinate system is established with the LOS vector as the LV_1 axis; LV_2 is normal to the LOS vector and lying in a horizontal plane; and LV_3 completes a right-handed system. The proportional navigation angles are the angles defining the orientation of the missile velocity with respect to the LOS. These angles are shown in Figure 3.2.26.

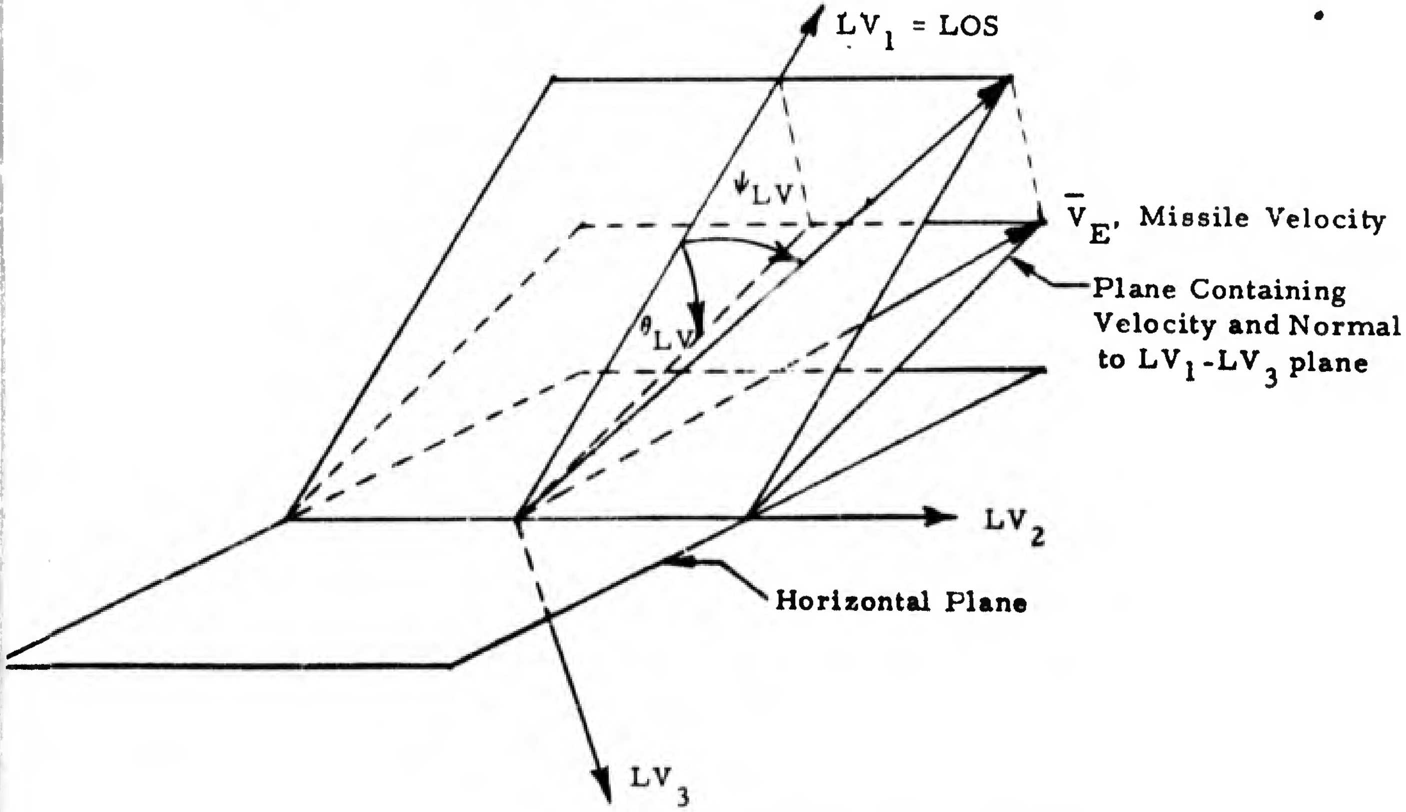


Figure 3.2.26 - Proportional Navigation Angles

The LOS coordinate system is established by

$$\hat{LV}_1 = \Delta\hat{R}_{LOS},$$

$$\hat{LV}_2 = (\hat{k}_E \times \hat{LV}_1) / |\hat{k}_E \times \hat{LV}_1| \text{ , *}$$

and

$$\hat{LV}_3 = \hat{LV}_1 \times \hat{LV}_2.$$

The magnitude of $\hat{k}_E \times \hat{LV}_1$ may be found by considering that

$$\hat{k}_E \times \hat{LV}_1 = \hat{k}_E \times \Delta\hat{R}_{LOS}$$

and that

$$\hat{k}_E \times \Delta\hat{R}_{LOS} = (\Delta R \sin\beta)\hat{\gamma}.$$

Now, $\Delta R \sin\beta$ is simply the projection of ΔR into the X_E-Y_E plane.

Thus,

$$\Delta R \sin\beta = \sqrt{\Delta X^2_{LOS} + \Delta Y^2_{LOS}} .$$

Therefore,

$$\hat{LV}_2 = (\hat{k}_E \times \hat{LV}_1) / \sqrt{\Delta X^2_{LOS} + \Delta Y^2_{LOS}} .$$

---

* $\hat{k}_E$ is a unit vector along the Z_E - axis, and x represents a cross product.

To determine the navigation angles, the missile velocity is projected onto each axis of the LV coordinate system. This produces the components of the velocity in this system. Thus, the angles are computed by

$$\theta_{LV} = \tan^{-1} \frac{V_E \cdot \hat{LV}_3}{V_E \cdot \hat{LV}_1} *$$

and

$$\psi_{LV} = \tan^{-1} \frac{V_E \cdot \hat{LV}_2}{V_E \cdot \hat{LV}_1} .$$

7. Vertical and Horizontal Flight Path Angles in the Earth Fixed Coordinate System (see Figure 3.2.27)

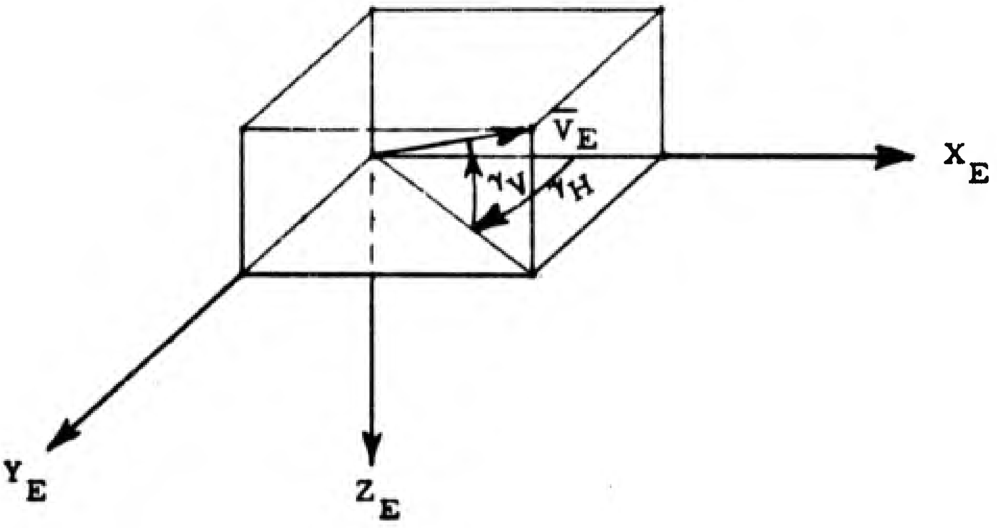


Figure 3.2.27. Vertical (γ_V) and Horizontal (γ_H) Flight Path Angles

---

* The dot " • " represents a dot product.

$$\gamma_V = \tan^{-1}\left(\frac{-V_{ZE}}{\sqrt{V_{XE}^2 + V_{YE}^2}}\right)$$

and

$$\gamma_H = \tan^{-1}\left(\frac{V_{YE}}{V_{XE}}\right)$$

8. Transformation of the Velocity Relative to the Wind From the Earth Fixed Coordinate System Into the Body Coordinate System

The velocity of the missile with respect to the wind, calculated in Subroutine G3, is transformed into the Body axes by

$$\bar{V}_{MWB} = [M] \quad \bar{V}_{MWE} \quad ,$$

where

$$\bar{V}_{MWB} = \begin{bmatrix} V_{MWU} \\ V_{MWV} \\ V_{MWW} \end{bmatrix} \cdot$$

9. Angle of Attack with Respect to Wind Axes.

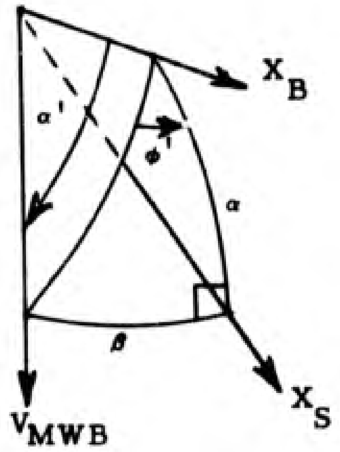
The geometry relating to the angles of attack of the missile with respect to the wind is illustrated in Figure 3. 2. 1 of Section 3. 2. 1, and Figure 3. 2. 28 below. The calculation of these angles is bypassed if the engine is thrusting (Flag QBURN ≤ 0) and the missile is still on the launch rail (RANGO $\leq$ RAIL).

a. Vertical angle of attack (α) and horizontal angle of attack (β) (Figure 3. 2. 28).

$$\alpha = \tan^{-1}\left(\frac{V_{MWW}}{V_{MWU}}\right)$$

and

$$\beta = \tan^{-1}\left(\frac{V_{MWV}}{V_{MWU}}\right) .$$



The X_B - X_S plane contains the Z_B axis

Figure 3. 2. 28. Angles of Attack

b. From spherical trigonometry and assuming a small angle of sideslip β, (the small angle β allows a right triangle to be formed by the chords of the angles α, β, and α') the absolute angle of attack, α', and the roll angle of the missile with respect to the wind axes, Φ', may be computed based on small angle approximations by

$$\alpha'^{2} = \alpha^{2} + \beta^{2}$$

and

$$\Phi' = \tan^{-1}\left(\frac{\alpha}{\beta}\right).$$

Conventionally, small angle approximations limits the angle magnitudes to about 10°. However, in the derivation of the above equations, half angles ($\alpha/2$, $\beta/2$) appear in the formulas that are simplified through the small angle approximation. Thus, since

$$\frac{\alpha}{2} \leq 10^{\circ}$$

and

$$\frac{\beta}{2} \leq 10^{\circ}$$

then

$$\alpha \leq 20^{\circ}$$

and

$$\beta \leq 20^{\circ}$$

is allowed.

3.2.12.3 Input/Output and Cross Reference of C Array

1. Input from Data Cards

FORTRAN SYMBOL	SYMBOL USED IN TEXT	C INDEX	DEFINITION
AGV	A	1330	Amplitude of the oscillatory acceleration acting on the missile due to the launcher vibrating (g's).
CFREQ	ω	1331	Frequency of the oscillatory acceleration acting on the missile due to the launcher vibrating (CPS).
CPHASE	χ	1332	Phase relationship at time=0 of the oscillatory acceleration acting on the missile due to the launcher vibrating (DEG).
AGV2	A_2	1332	Magnitude of the linear component of acceleration acting on the missile due to the launcher (g's).

2. Inputs from Other Modules

FORTRAN SYMBOL	SYMBOL USED IN TEXT	C INDEX	DEFINITION
VMWXE	$\bar{V}_{MWE}$	200	Components of missile velocity relative to the wind in Earth Fixed system (FT/SEC)
VMWYE		201	
VMWZE		202	
VAIRSP		207	Velocity magnitude of missile relative to wind (FT/SEC)
QBURN		1405	Engine burnout switch (1=burnout)
VXE	V_{XE}	1603	Components of missile velocity in Earth Fixed Coordinate (FT/SEC)
VYE	V_{YE}	1607	
VZE	V_{ZE}	1611	
RXE	$\bar{R}_E$	1615	Position components of missile in Earth fixed coordinate system (FT)
RYE		1619	
RZE		1623	
RTXE	$\bar{R}_T$	1651	Position components of target in Earth fixed coordinate system (FT)
RTYE		1655	
RTZE		1659	
RXO	$\bar{R}_O$	1668	Initial position components of missile and rail in fixed system (FT)
RYO		1669	
RZO		1670	
VXO	$\bar{V}_O$	1671	Initial velocity - components of missile and rail in Earth fixed system (FT)
VYO		1672	
VZO		1673	
CFA11	C_{11}	1703	Elements of the Earth-to-Body transformation matrix, M
CFA12	C_{12}	1707	
CFA13	C_{13}	1711	
CFA21	C_{21}	1715	
CFA22	C_{22}	1719	
CFA23	C_{23}	1723	
CFA31	C_{31}	1727	
CFA32	C_{32}	1731	
CFA33	C_{33}	1735	
A021	Ao_{21}	1755	Position components of gyro 1 (inertial platform) spin axis unit vector in Earth Fixed coordinate system
A022	Ao_{22}	1756	
A023	Ao_{23}	1757	
A031	Ao_{31}	1758	Position components of gyro 2 (inertial platform) spin axis unit vector in Earth fixed coordinate system
A032	Ao_{32}	1759	
A033	Ao_{33}	1760	
T	t	2000	Flight time (SEC)

3. Output

FORTRAN SYMBOL	SYMBOL USED IN TEXT	C INDEX	DEFINITION
BTHT	θ	350	Euler pitch angle of body system (DEG).
BPSI	ψ	351	Euler yaw angle of body system (DEG).
BPHI	Φ	352	Euler roll angle of body system (DEG).
BPH1	Φ_1	353	Euler roll angle of inertial platform (gyro 1) with respect to the body system (DEG)
BTH2	θ_2	354	Euler pitch angle of inertial platform (gyro 2) with respect to the body system (DEG).
BPS1	ψ_1	355	Euler yaw angle of the inertial platform (gyro 1) with respect to the body system (DEG).
BPH2	Φ_2	393	Euler roll angle of inertial platform (gyro 2) with respect to the body system (DEG).
VTOTE	V_{TOTE}	356	Missile total velocity magnitude in the Earth fixed frame (FT/SEC)
BGAMH	γ_H	357	Horizontal flight path angle (heading) measured clockwise from X_E - axis (DEG)
BGAMV	γ_V	358	Vertical flight path angle measured positive up from the local horizontal (DEG).
BTHLV	θ_{LV}	363	Vertical proportional navigation angle, measured positive downward from the LOS vector to the projection of the missile velocity into the vertical plane that contains the LOS vector (DEG).
BPSLV	ψ_{LV}	364	Horizontal proportional navigation angle measured positive clockwise from the LOS vector to the projection of the missile velocity into the plane containing the LOS vector and normal to the vertical plane (DEG).
BLAMV	λ_V	365	Vertical line-of-sight angle, measured from the local horizontal to the LOS vector (DEG).
BLAMH	λ_H	366	Horizontal line-of-sight angle, measured from X_E axis to the projection of the LOS vector into the local horizontal plane (DEG).

FORTRAN SYMBOL	SYMBOL USED IN TEXT	C INDEX	DEFINITION
BALPHA	α	367	Vertical angle of attack, measured from the projection of the missile velocity with respect to the wind into the X_B-Z_B plane and the X_B axis (DEG).
BALPHY	β	368	Horizontal angle of attack (sideslip) measured from the projection of the missile velocity with respect to the wind into the X_B - Z_B plane and the velocity relative to the wind (DEG).
BALPHP	α'	369	Total angle of attack between vehicle X_B body-axis and the velocity relative to the wind (DEG).
BPHIP	ϕ'	370	Aerodynamic roll angle between the plane containing the Z_B-X_B axes and the plane containing X_B-velocity relative to the wind (DEG).
RANGE	RANGE	371	Slant range from missile to the target (FT)
RXBA	ΔX_{BLOS}	372	X-component of missile LOS ($\overline{\Delta R}_{BLOS}$) in body coordinates (FT).
RYBA	ΔY_{BLOS}	373	Y-component of missile LOS ($\overline{\Delta R}_{BLOS}$) in Body coordinates (FT).
RZBA	ΔZ_{BLOS}	374	Z-component of missile LOS ($\overline{\Delta R}_{BLOS}$) in Body coordinates (FT).
RANGO	R_{ANGO}	380	Magnitude of the separation distance between the rail launcher and the missile. (Measured from rear end of launcher to the rear lug of missile (FT).
RXL	ΔX_{ML}	390	X-component of the missile position with respect to the launcher (FT).
RYL	ΔY_{ML}	391	Y-component of the missile relative to the launcher in the Earth fixed coordinate frame (FT).
RZL	ΔZ_{ML}	392	Z-component of the missile relative to the launcher in the Earth fixed coordinate frame (FT).
RDELX	ΔX_{LOS}	1635	X-component of missile LOS ($\overline{\Delta R}_{LOS}$) in earth fixed coordinate system (FT).
RDELY	ΔY_{LOS}	1636	Y-component of missile LOS ($\overline{\Delta R}_{LOS}$) in earth fixed coordinate system(FT).
RDELZ	ΔZ_{LOS}	1637	Z - component of missile LOS ($\overline{\Delta R}_{LOS}$) in Earth fixed coordinate system (FT)

3.2.13 S1 - Seeker Module

3.2.13.1 Function Description

The Seeker Module, S1, currently used in the 6DOF simulation program models either a vidicon tracker or a quadrant (laser) tracker, depending upon whether the input option switch, OPTNSK, is greater than zero, or less than or equal to zero, respectively.

3.2.13.2 Equations

1. Line-of-Sight Angular Offset Calculation

The initial portion of the module logic which determines target position (modeled as a point source) in the gimbal axes is common to both trackers. This operation consists of transforming the X, Y, and Z components of target position from the body axes to the gimbal axes, then calculating the angular offsets (from the gimbal boresight axis) in pitch and yaw.

As is depicted in Figure 3.2.29, the gimbal axes, (X_G, Y_G, Z_G), orientation with respect to the body axes, (X_B, Y_B, Z_B), is specified by two rotations; the first through an angle of ψ_G about the Z_B axis, and the second through an angle of θ_G about Y_G (the sign of the angles following the right hand rule convention).

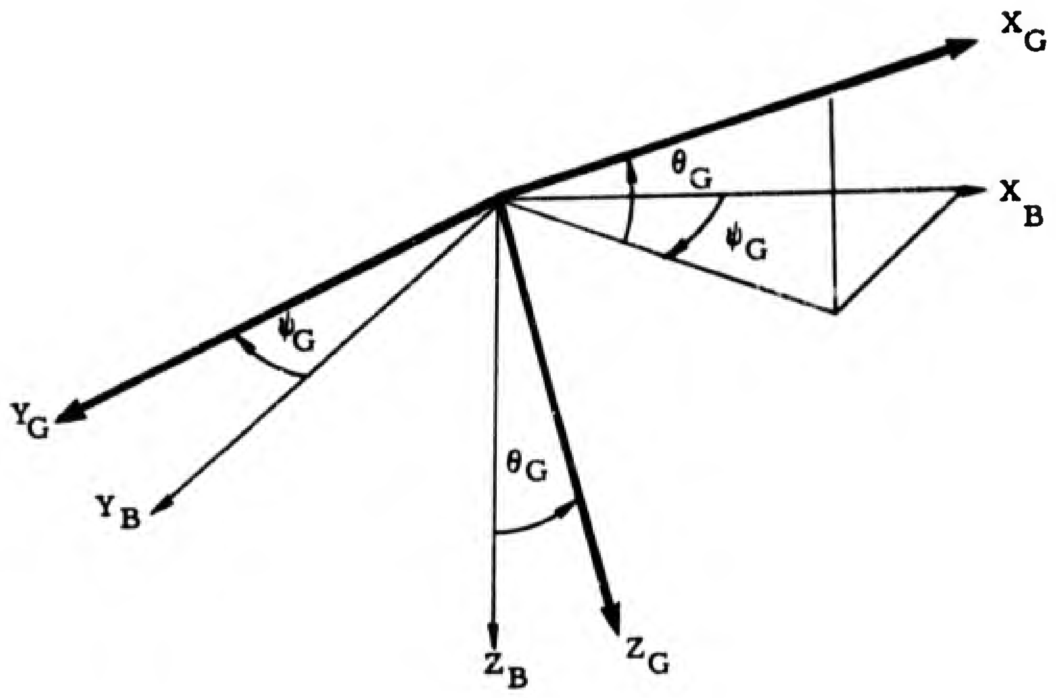


Figure 3.2.29

The transformation matrix, M, from the body to the gimbal axes is given by:

$$M_G = \begin{bmatrix} \mathrm{COS}\,\theta_G \mathrm{COS}\,\psi_G & \mathrm{SIN}\,\psi_G \mathrm{COS}\,\theta_G & -\mathrm{SIN}\,\theta_G \\ -\mathrm{SIN}\,\psi_G & \mathrm{COS}\,\psi_G & 0 \\ \mathrm{SIN}\,\theta_G \mathrm{COS}\,\psi_G & \mathrm{SIN}\,\theta_G \mathrm{SIN}\,\psi_G & \mathrm{COS}\,\theta_G \end{bmatrix}$$

Hence, the target position coordinates in the gimbal axes are:

$$\begin{bmatrix} X_G \\ Y_G \\ Z_G \end{bmatrix}_G = M_G \begin{bmatrix} X_B \\ Y_B \\ Z_B \end{bmatrix}_B$$

where X_B, Y_B, and Z_B are the target position coordinates (LOS vector) in the body axes (computed in the Coordinate Transformation Module, G5).

The offset angles from the gimbal boresight axis (X_G), β_Z and β_Y, follow directly from the geometry depicted in Figure 3. 2. 30.

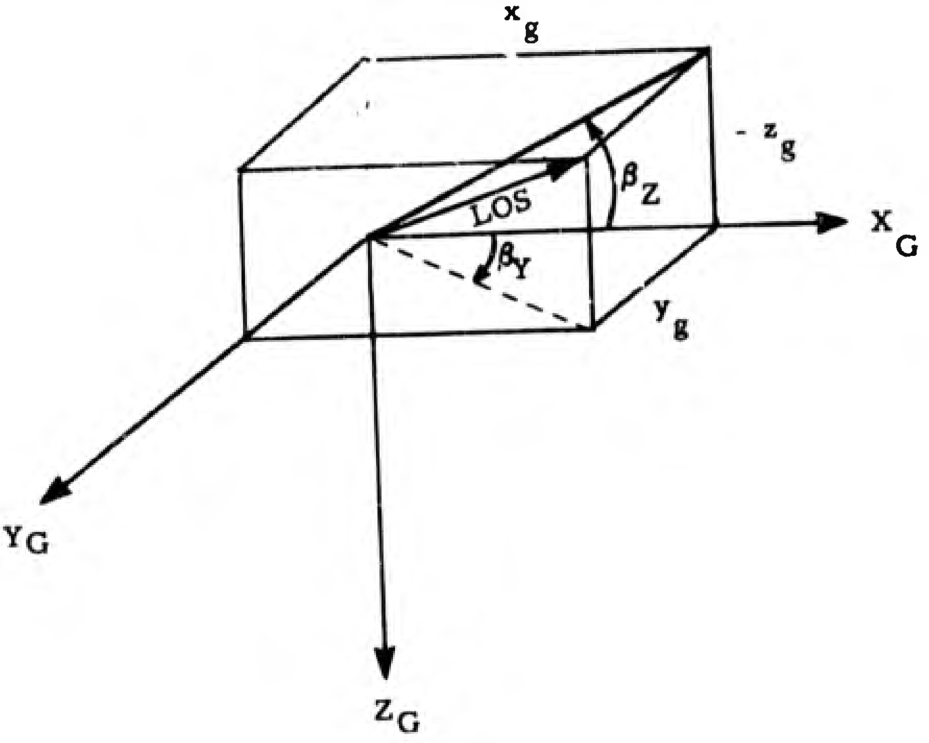


Figure 3. 2. 30. Angular Offset of LOS from Seeker Boresight Axis (X_G)

$$\beta_z = \tan^{-1} \frac{-z_g}{x_g}$$

$$\beta_y = \tan^{-1} \frac{y_g}{x_g}$$

β_z and β_y are used in both the vidicon and the quadrant trackers to generate an autopilot error signal.

2. Vidicon Tracker

In the vidicon tracker mode (OPTNSK >0), a check is made to determine if slant range (RANGE) is less than optical break-lock range (RBK), where break-lock range is the range at which seeker saturation occurs. At this point, no new error signals are calculated in the seeker module, the seeker module returns control to the calling subroutine (AUXSUB), and the last error signals calculated continues to be fed to the autopilot.

With slant range greater than optical break-lock range, the pitch and yaw error signals to the autopilot (ω_z and ω_y) are calculated as the product of an input gain (GEO) and the LOS offset angles, β_z and β_y, i.e.,

$$\omega_z = \text{GEO} \ \beta_z$$

$$\omega_y = \text{GEO} \ \beta_y$$

Program control is then transferred to the common vidicon and quadrant tracker output calculations (see Paragraph 4).

3. Quadrant Tracker

In the quadrant tracker mode, the logic is somewhat more complex than the simple linear error signals produced by the vidicon model. Included in the quadrant tracker logic is the capability to model the following features:

- Laser pulse rate controllable by input data.
- Pulse hold logic retains detector signal between pulses.
- Maximum target acquisition range
- Target in or out of field of view
- Range dependent detector dead band
- Direct (pre-launch lock-on to target) or indirect (launch without lock-on to target) fire modes
  - Gyro caged along missile longitudinal axis until lock-on (indirect fire only)
  - Higher tracker signal gains (independently in pitch and yaw)switched in during target acquisition* (indirect fire only).

---

* Target acquisition in this case refers to the phase of target acquisition occurring between the time that the target comes into the field-of-view and the times that it crosses the midpoint of the field-of-view in pitch and yaw. This provides the higher gain needed to prevent the target from passing completely through the field-of-view before the missile can respond to the error signals.

- Smoothing (integration) of seeker error signals to autopilot optional
    - Response time of seeker error signal to detector signal can be varied by input constant.

The equations and logic for the quadrant tracker are:

(a) Laser pulse rate and tracker pulse hold logic:

If $(t < s)$, hold last detector pulse and go to seeker error signal computation

$$s = s + \Delta t,$$

where t is the time of flight, Δt is the laser pulse period, and s is a step function of time.

The relation between t, Δt, and s is shown in Figure 3.2.31.

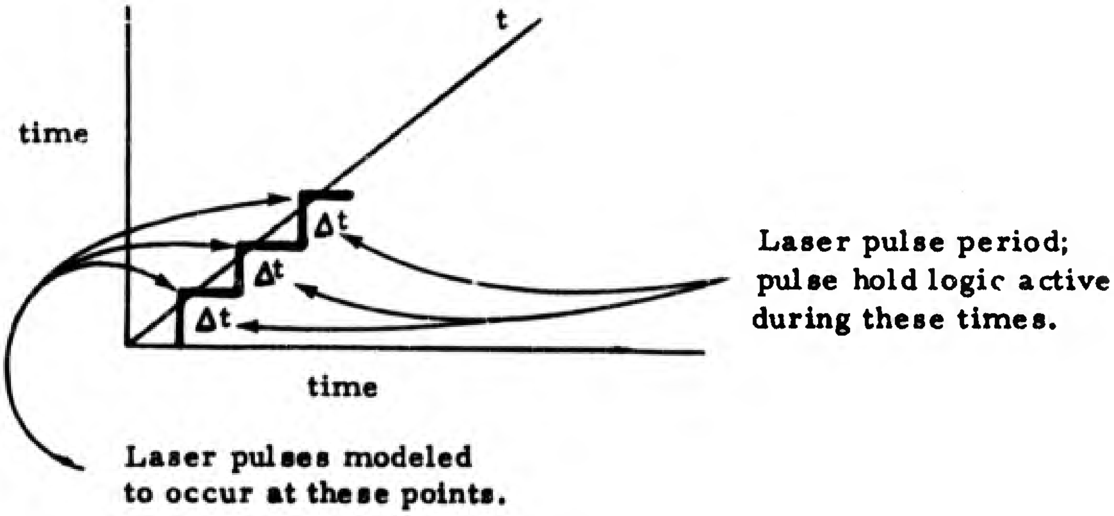


Figure 3.2.31. Pulse Hold Logic

(b) Maximum acquisition range sensing:

If (RANGE > RLOCK), effectively set pitch and yaw error signals to zero and exit tracker module, because slant range (RANGE) is beyond maximum acquisition range (RLOCK).

(c) Target in or out of field-of-view determination:

If $$\frac{\beta_y^2}{\Phi_y^2} + \frac{\beta_z^2}{\Phi_z^2} > 1,$$

effectively set pitch and yaw error signals to zero and exit tracker module, because target point source is outside the elliptical field-of-view determined by Φ_y and Φ_z.

The geometry of this determination is shown in Figure 3. 2. 32.

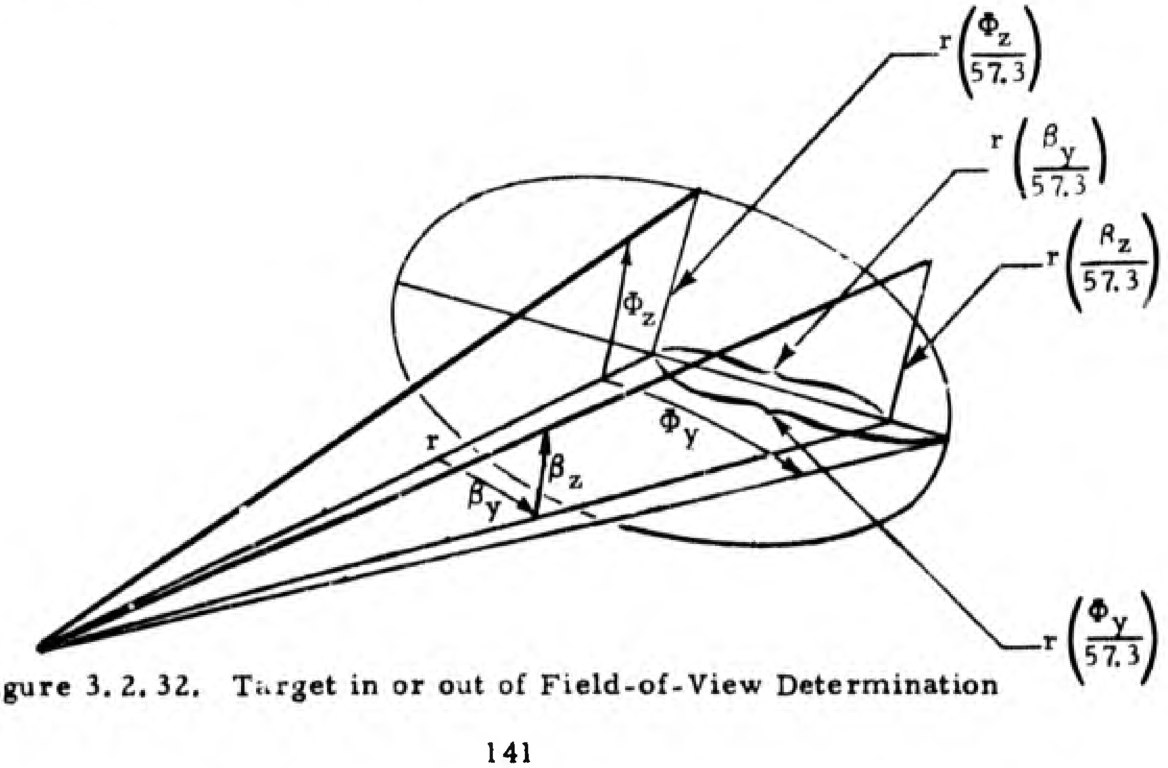


Figure 3. 2. 32. Target in or out of Field-of-View Determination

From Figure 3.2.32, using small angle approximations (specifically, the Y and Z coordinates of FOV and target position are taken to be the arc lengths, e.g. $Y_{FOV} \approx r \frac{\Phi y}{57.3}$, with angles in degrees), the equation for the FOV ellipse boundary is:

$$\frac{Y^2}{r^2\left(\frac{\Phi_Y}{57.3}\right)^2} + \frac{Z^2}{r^2\left(\frac{\Phi_Z}{57.3}\right)^2} = 1$$

If the point (Y, Z) is such that the left hand side is greater than 1, then (X, Z) is outside the FOV ellipse. Replacing Y and Z with the target components, $r\left(\frac{\beta_Y}{57.3}\right)$ and $r\left(\frac{\beta_Z}{57.3}\right)$, and cancelling the r's and the 57.3, have that

$$\frac{\beta_Y^2}{\Phi_Y^2} + \frac{\beta_Z^2}{\Phi_Z^2} > 1$$

is the condition for the target being outside the FOV.

(d) Detector output signal:

$$BZ = \begin{cases} +1, \text{ if } \beta_Z > 0 \\ -1, \text{ if } \beta_Z < 0 \end{cases}$$

$$BY = \begin{cases} +1, \text{ if } \beta_Y > 0 \\ -1, \text{ if } \beta_Y < 0 \end{cases}$$

(e) Range dependent deadband determination:

$$TKDB = \frac{BDB}{2} \left(\frac{RANGE}{32810} \right)^2 ,$$

where TKDB is one-half the total range dependent deadband, BDB is the input total deadband normalized to ten kilometers (32810 ft.), and range is the missile to target slant range. A rationale for computing deadband in this manner is that reflected laser energy will vary as $\frac{1}{(RANGE)^2}$ so the photoelectric current differential between the four quadrants will also vary as a function of $\frac{1}{(RANGE)^2}$. Since the photoelectric current differential also varies by the amount of image center offset from the detector center, there will be a minimum detectable image offset due to detector and amplifier noise.

(f) Target in or out of deadband determination:

$$\text{If } (|\beta_z| < TKDB) \ BZ = 0$$
$$\text{If } (|\beta_y| < TKDB) \ BY = 0,$$

i.e., set pitch or yaw detector outputs to zero if target angular offset in pitch or yaw are within the deadband.

(g) If target is in FOV, uncage gyro and save pitch and yaw detector error signals for gain switching during acquisition phase:

$$u_z = \beta_z$$

$$u_y = \beta_y$$

Cage = 1

(h) Generate autopilot error signals and seeker gimbal torque error signals:

There are two seeker model options for calculating autopilot and seeker gimbal torque error signals. One option directly outputs the detector error signal times a gain factor. The second integrates the detector gain and permits a seeker signal shaping term and a response time factor to be input. In both of these options, the same signals are used for both the autopilot and the seeker gimbal torque.

A third option (which can be exercised only if the second of the previously discussed two options is used) allows the response of the seeker gimbal torque signal to be controlled by input independently of the autopilot signal.

(1) Detector error signal times a gain output directly:

$$\omega_z = g_s \beta_z$$

$$\omega_y = g_s \beta_y ,$$

where ω_z and ω_y are the error signals to the autopilot and the seeker gimbal torque, and g_s is an input gain constant. (Control is then transferred to common vidicon/quadrant tracker output logic.)

(2) Detector error signal integration with input signal response time control.

$$\omega_z = \frac{g_s \beta_z + \Phi}{\omega_\ell} + \int_o^t \left(g_s \beta_z + \Phi\right) dt,$$

$$\omega_y = \frac{g_s \beta_y + \Phi}{\omega_\ell} + \int_o^t \left(g_s \beta_y + \Phi\right) dt,$$

where Φ is an input term which can be used to change the operating level of the detector error signal*, and ω_ℓ is an input seeker signal response time factor.

The block diagram for ω_z (input = $g_s \beta_z$) is shown in Figure 3. 2. 33. An analoguous block represents β_y.

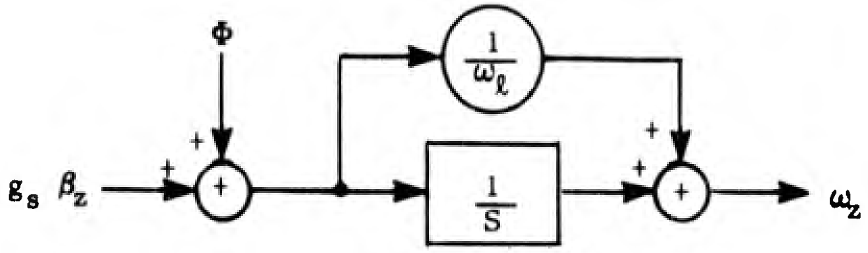


Figure 3. 2. 33 - Error Signal Processing When $\omega_\ell > 0$

---

*A positive input value of Φ, for example, would cause the detector output signal to ride on the value of $\Phi > 0$, which would positive bias the pitch and yaw error signals to the autopilot. Φ can be used to simulate gyro drift.

The pulse hold logic causes β_z (and β_y) to behave as a positive or negative unit step function, depending upon whether the point source target is in the upper or lower two quadrants of the detector. Consequently, the term

$$\frac{g_s \ \beta_z + \Phi}{\omega_\ell}$$

is a step function with amplitude $\pm g_s$, when $\Phi = 0$, $\omega_\ell = 1$. The integrated term in ω_z, therefore, is a ramp function with slope $\pm g_s$. The quantity contributed by the term $\frac{g_s \ \beta_z + \Phi}{\omega_\ell}$, superimposed on the ramp function, provides the output signal, ω_z, with a faster response to a detector error signal change than would be obtained with only the integrated term.

(3) Independent* seeker gimbal torque response (utilized only when $\omega_n > 0$).

The equations for the seeker gimbal torque error signals are:

$$\dot{\gamma}_z = \omega_n \left(\omega_z - \gamma_z\right)$$

$$\dot{\gamma}_y = \omega_n \left(\omega_y - \gamma_y\right)$$

---

* The seeker gimbal torques generated by the additional differential equations are independent of the autopilot error signals in the sense that the autopilot error signals are unaffected by the additional equations. The forcing function, however, for the gimbal torque is the same signal sent to the autopilot.

$$\rho_z = \frac{\dot{\gamma}_z}{\omega_c} + \gamma_z$$

$$\rho_y = \frac{\dot{\gamma}_y}{\omega_c} + \gamma_y$$

The block diagram for the seeker gimbal torque signals is shown in Figure 3. 2. 34 for the z-channel (an analogous block can be drawn for the y-channel).

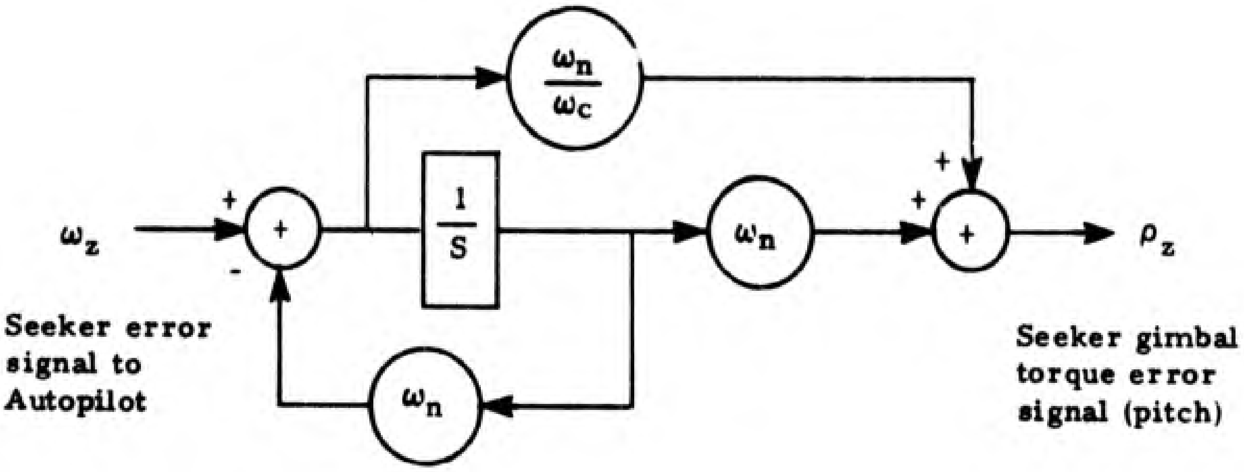


Figure 3. 2. 34 - Addition Seeker Gimbal Torque Error Signal Processing When $\omega_n > 0$

Combining the block diagram showing the seeker signal error processing (Figure 3. 2. 33) with the block showing the seeker gimbal torque processing (Figure 3. 2. 34) and including the logic switching accomplished by the program for zero values of ω_λ and ω_n, results in the signal processing block shown in Figure 3. 2. 35 for the z-channel (a similar block represents the y-channel).

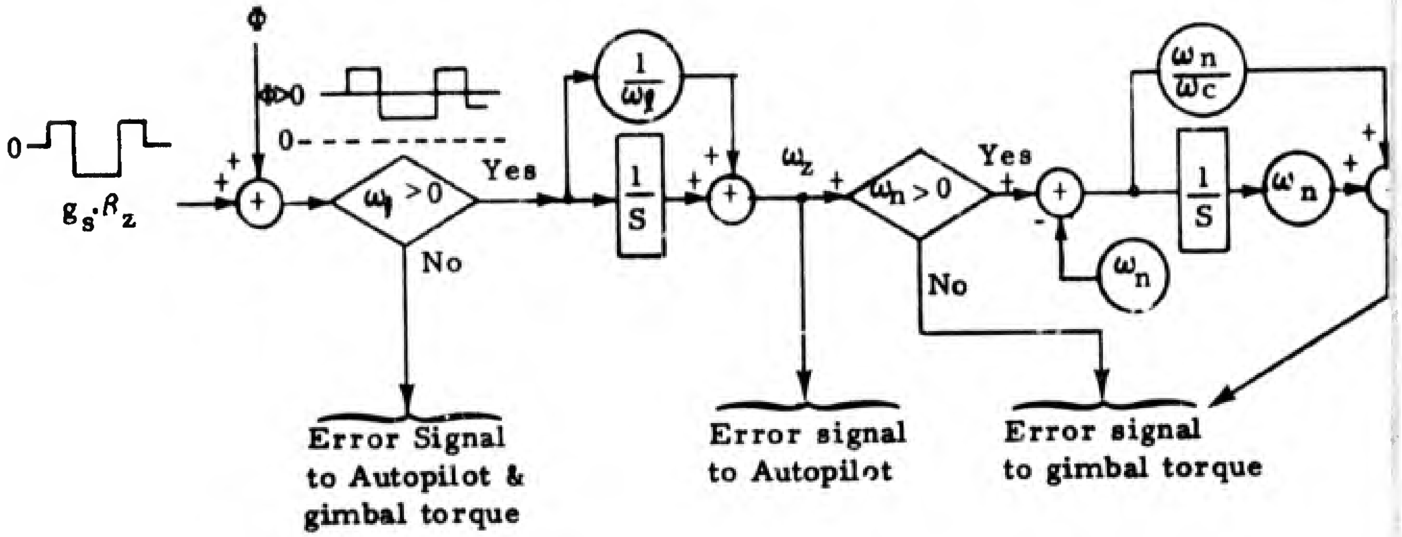


Figure 3. 2. 35 - Signal Processing Block

(i) Acquisition Mode Gain Switching

Before the target image enters the FOV in the Indirect Fire option, the seeker gyro is caged (CAGE=0) and no error signals are output from the seeker to either the autopilot or for the seeker gimbal processing torque. Once the target enters the FOV, the seeker gyro is uncaged, the sign (+ or -) of the detector error signal is retained by variables u_z and u_y, respectively, and higher acquisition mode gain, g_x, is used for the pitch and yaw error signals. When the error signals

---

* Acquisition Mode here refers to the system state in Indirect Fire between the two that the target enters the FOV and the time the target crosses from the upper (lower) two quadrants to the lower (upper) two quadrants (in pitch), with analogous conditions for yaw.

change sign, the error signal gain is switched back to the terminal homing value, g_s.

The gain switching occurs independently in pitch and yaw. Print statements contained in the acquisition mode gain switching logic automatically print the time at which acquisition gain is switched out and terminal homing gain is switched in. Also printed at this time are the target angular offset angles from the seeker boresight axis. The printouts occur independently in pitch and yaw.

4. Output Calculations*

After calculation of the seeker error signals for the autopilot and seeker gyro gimbal torques, the Seeker Module transforms missile body rotational rates to the seeker gyro gimbal axes, calculates seeker gimbal angle derivatives. In the pre-acquisition case of a caged seeker gyro (CAGE = 0), all error signal outputs are forced to zero. Control is then returned to the calling subroutine, AUXSUB. The common (to both vidicon and quadrant trackers) output calculations are performed by the following equations:

---

*Common to both Vidicon and Quadrant tracker logic.

Transform Missile Body Rates to Seeker Gimbal Axes:

$$\underset{\substack{\text{Gimbal}\\ \text{Axes}}}{\begin{bmatrix}\dot{\Phi}_{X_G}\\ \dot{\Phi}_{Y_G}\\ \dot{\Phi}_{Z_G}\end{bmatrix}} = M_G \underset{\substack{\text{Body}\\ \text{Axes}}}{\begin{bmatrix}\dot{\Phi}_{X_B}\\ \dot{\Phi}_{Y_B}\\ \dot{\Phi}_{Z_B}\end{bmatrix}} ,$$

where M_G is the body axes to seeker gimbal axes transformation matrix, the right hand vector has body rates in the body axes as components, and the left hand vector has the same body rates in the gimbal axes as components. Because the seeker is not gimballed about the X_B axis, the $\dot{\Phi}_{XB}$ component is not calculated.

Gimbal Coupling terms

$$C_y = S_\omega \, (- \theta_G + \psi_G)$$

$$C_z = S_\omega \, (- \theta_G - \psi_G)$$

where S_ω is an input spring restraining torque constant θ_G, ψ_G, $\dot{\theta}_G$, and $\dot{\psi}_G$ are the seeker gimbal yaw and pitch angles and their derivatives, obtained on the last integration step.

Calculate Gimbal Angle Derivatives

$$\dot{\theta}_G = \rho_y + C_z - \dot{\Phi}_{y_G}$$

$$\dot{\psi}_G = \rho_z + C_y - \frac{\dot{\Phi}_{z_G}}{\alpha_{33}}$$

where C_z and C_y are the gimbal coupling terms, $\dot{\phi}_{y_G}$ and $\dot{\phi}_{z_G}$ are the missile body rate components about the Y_G and Z_G seeker gimbal axes, α_{33} is the (3, 3) element from the body to seeker gimbal axes transformation matrix, M_G and ρ_y and ρ_y represent the seeker gimbal processing torque signals. (If $\omega_n > 0$, or are equal to ω_y and ω_z, if $\omega_n \leq 0$).

3.2.13.3 Assumptions and Limitations

The following assumptions are made in the 6DOF seeker model:

1. The target is modeled as a point source.
2. The seeker two degree of freedom gyro is modeled without dynamics.
3. Reflected pulses are always present at the input PRF, and are always detected when in the FOV.
4. Detector image resolution deadband is a function of range only.
5. The seeker pitch to yaw and yaw to pitch coupling terms assume an inertially stabilized platform.
6. Seeker gimbal angle limiting is not modeled.

The first assumption results in all of the seeker optics, detector characteristics, such as image resolution, saturation, signal to noise, etc., being ignored.

The implications of the second assumption are essentially self-explanatory; the simulated gyro behaves as a perfect gyro (with the exception of the coupling terms and drift).

The third and fourth assumptions ignore the possibility of lost pulses, attenuation of reflected signals due to atmospheric conditions designator malfunction, countermeasures, etc.

The fifth assumption is not strictly true in a gyro which is precessed to follow the LOS. However, the assumption would not be severely violated with a relatively stabilized LOS-seeker boresight angle. If the 6DOF model is to be used to evaluate candidate seekers components (e. g., seekers) however, more precise models may be required.

3.2.13.4 Initialization Subroutine

The Seeker Initialization Subroutine, SII, enters the indices of the elements of the "C" array which contain Seeker Module generated derivatives into successive locations in the IPL array. The IPL array points the numerical integration logic to the elements of the "C" array which are to be numerically integrated. SII initializes the IPL array with the indices of all potentially utilized seeker related derivatives, despite the fact that the number of derivatives actually used by the Seeker Module depends upon whether the input values of the variables ω_λ and

ω_n are zero or positive. The only potentially detrimental effect of this is to cause unnecessary numerical integrations to occur.

S11 also initializes the cage-uncage switches used in S1 as a function of the setting of the input option switch, OPTN4. Specifically, if OPTN4 $\leq$ 1, (signifying that simulation is for direct fire (lock-on before launch), the seeker gyro caging switches are set to the uncaged value (CAGE, TKRY, TKRZ, and TRKZY all equal to 1), reflecting the fact that the gyro is uncaged before direct fire launch. If OPTN4 > 1 (an indirect fire mode simulation), the seeker gyro caging switches are set to the cage value (zero).

3.2.13.5 Input/Output and Cross Reference of C-Array

1. Input From Data Cards

FORTRAN SYMBOL	SYMBOL USED IN TEXT	C INDEX	DEFINITION
RLOCK	RLOCK	445	Maximum quadrant tracker acquisition range
DT	Δt	446	Laser designator pulse repetition time
BDB	BDB	447	Tracker deadband (degrees) at 10 kilometers
CFOVZ	θ_z	448	Total seeker field-of-view in pitch (degrees)
CFOVY	θ_y	449	Total seeker field-of-view in yaw (degrees)
GSX	GSX	450	Seeker quadrant tracker error signal acquisition gain
SEPS	Φ	451	Integration shift rate
SWP	S_ω	452	Spring restraining torque constant
RBK	RBK	453	Vidicon Tracker breaklock range (image saturation range)
GEO	GEO	454	Vidicon tracker error signal gain
OPTNSK	OPTNSK	455	Vidicon/Quadrant tracker option switch
GS	g_s	456	Quadrant tracker error signal gain
WSL	ω_λ	457	Quadrant tracker error signal gain term (affects error signal amplitude and response time)
WSN	ω_n	458	Quadrant tracker gimbal torque signal gain term.
WL2	ω_c	459	Quadrant tracker gimbal torque signal gain term

2. Input from Other Modules

FORTRAN SYMBOL	SYMBOL USED IN TEXT	C INDEX	DEFINITION
RANGE	RANGE	371	Missile to Target Slant Range (LOS vector magnitude)
RXBA	X_B	372	X-Component of Slant Range (In Body Axes)
RYBA	Y_B	373	Y-Component of Slant Range (In Body Axes)
RZBA	Z_B	374	Z-Component of Slant Range (In Body Axes)
WP	$\dot{\Phi}\ X_B$	1739	Missile Roll Rate (In Body Axes)
WQ	$\dot{\Phi}\ Y_B$	1743	Missile Pitch Rate (In Body Axes)
WR	$\dot{\Phi}\ Z_B$	1747	Missile Yaw Rate (In Body Axes)

3. Outputs

FORTRAN SYMBOL	SYMBOL USED IN TEXT	C INDEX	DEFINITION
WLAMQ	ω_z	403	Seeker Pitch Error Signal to Autopilot
WLAMR	ω_y	407	Seeker Yaw Error Signal to Autopilot
CAGE	CAGE	461	Seeker Gyro Cage-Uncage Switch to Autopilot
TKRZ	-	462	Acquisition Gain Switch for Pitch to Autopilot
TKRY	-	463	Acquisition Gain Switch for Yaw to Autopilot

3.3 EXECUTIVE SUBROUTINES

3.3.1 Module Initialization Executive Subroutine (AUXI)

The module initialization subroutines are called by subroutine AUXI according to the input module data control cards (Type "2" input data cards - see Section 4 for input card formats).* AUXI is called by the MAIN program after the input subroutine, OINPT1, is called, and prior to entering the integration loop. AUXI contains calls to a number of initialization subroutines which exist only as non-functional entry points in the subroutine DUMMY. Thus, for example, AUXI calls A2I, which by the naming convention would correspond to the Initialization Subroutine for the Forces and Moments Module, A2. However, no subroutine A2I exists, which results in the compiler accessing the entry point A2I in DUMMY, which simply returns control back to AUXI.

Table 3.3.1 shows the functional module initialization subroutines, along with the module number which initiates the subroutine call in AUXI.

---

*AUXI will automatically call an initialization subroutine for each module specified on a Type "2" input card, i.e., no input cards are required for initialization subroutines.

NAME	FUNCTION	SUBROUTINE NUMBER
A1I*	AERO Coefficient Module Initialization	2
A3I	Engine Module Initialization	4
A5I	Upper and Lower derivative bounds for integration step size switching	6
C1I	Autopilot (High Frequency) Module Initialization	7
C2I	Autopilot (Low Frequency) Module Initialization	8
C4I	Actuator Initialization	10
D1I	Translational Dynamics Module Initialization	17
D2I	Rotational Dynamics Module Initialization	18
S1I	Seeker Module Initialization	28

Table 3. 3. 1 - Module Initialization Subroutine Numbers

---

* A1I is an entry point in C4I.

3.3.2 Module Executive Subroutine (AUXSUB)

Subroutine AUXSUB is an executive routine which calls the modules input on Type "2" data cards (see Section 4 for input data card format) in the order of input. AUXSUB is called by the MAIN program. It is in the integration loop, and consequently, is called by MAIN at every integration step. AUXSUB contains calls to modules that exist only as non-functional entry points in subroutine DUMMY. Calls to these non-existent modules, however, will never be made when correct input Type "2" data cards are used.

Table 3.3.2 depicts the existing modules and their associated module numbers which, when input on Type "2" data cards, cause AUXSUB to execute the modules in input order.

AUXSUB performs the auxiliary function of moving derivatives into the "C" array (calculated by the modules) to sequential locations in the array DER(I). The derivative values contained in this array are subsequently numerically integrated by subroutine AMRK.

MODULE NAME	FUNCTION	MODULE NUMBER
A1	AERO Coefficient Table Look-Up	2
A2	Forces and Moments Module	3
A3	Engine Module	4
C1	Autopilot (High Frequency) Module	7
C2	Autopilot (Low Frequency) Module	8
C4	Actuator Module	10
D1	Translational Dynamics Module	17
D2	Rotational Dynamics Module	18
G2	Winds Module	23
G3	Air Data Module	24
G4	Terminal Geometry Module	25
G5	Coordinate Transformation Module	26
S1	Seeker Module	28

Table 3. 3. 2 - Modules and Associated Module Numbers

3.3.3 SUBL1 - Executive Subroutine

This executive subroutine is functionally inactive. It is called by MAIN, and in turn calls INPT1, OUPT1, STGE1, CNTR1, RNDM1, AUXA1, AUXB1, and AUXC1, depending upon the subroutine numbers stored in the array, SUBNO (I). However, all of the subroutine names called (INPT1, OUPT1, etc.) exist only as entry points in subroutine DUMMY. Consequently, SUBL1 performs no function whatsoever.

3.3.4 SUBL2 - Executive Subroutine

3.3.4.1 Function Description

Executive subroutine, SUBL2, is called by MAIN immediately after the module initialization subroutine, AUXI, is called. SUBL2, in turn, calls INPT2, OUPT2, STGE2, CNTR2, RNDM2, AUXA2, AUXB2, and AUXC2, depending upon the subroutine numbers stored in the array, SUBNO(I)*. The array, SUBNO (I), is initialized in OINPT1 by the subroutine numbers entered on Type "1" input data cards, with the current configuration of the simulation program, two Type "1" cards are input to enter subroutine numbers "3" and "4" into the SUBNO(I) array. This will cause subroutines OUPT2 and STGE2 (described in Sections 3.5.2 and 3.4.11, respectively) to be called by SUBL2.

---

*All of the named subroutines except OUPT2 and STGE2 exist only as entry points in the subroutine, DUMMY.

3.3.4.2 Input/Output and Cross Reference of C-Array

FORTRAN SYMBOL	C INDEX	DEFINITION
NOSUB	2461	Number of subroutines specified on Type "1" input data cards.
SUBNO(I)	2462	Subroutine numbers entered on Type "1" input data cards.

3.3.5 SUBL3 - Executive Subroutine

Executive subroutine, SUBL3, is called by MAIN at every integration step. SUBL3, in turn, calls INPT3, OUPT3, STGE3, CNTR3, RNDM3, AUXA3, AUXB3, and AUXC3, depending upon the subroutine numbers stored in the array, SUBNO(I).* The array, SUBNO(I), is initialized in OINPT1 by the subroutine numbers entered on Type "1" input data cards. With the current configuration of the simulation program, two Type "1" cards are input to enter subroutine numbers "3" and "4" into the SUBNO(I) array. This will cause SUBL3 to call subroutines OUPT3 and STGE3 (described in Sections3.5.2 and 3.4.12).

3.3.5.1 Input/Output and Cross Reference of C-Array

FORTRAN SYMBOL	C INDEX	DEFINITION
NOSUB	2461	Number of subroutines specified on Type "1" input data cards.
SUBNO(I)	2462	Subroutine numbers entered on Type "1" input data cards.

---

*All of the named subroutines except INPT3 and OUPT3 exist only as entry points in the subroutine, DUMMY.

3.3.6 Table Look-Up Subroutines: TABLE, TABL2 and TABL3

These three subroutines are used for the express purpose of passing tabular aerodynamic data to the special function subroutines FINTP1, FINTP2, and FINTP3. These special function subroutines perform a linear interpolation between two points in three types of tabular data tables. These data tables are:

1. TABLE ⟶ FINTP1 - one dependent, one independent variable
2. TABL2 ⟶ FINTP2 - one dependent, two independent variables
3. TABL3 ⟶ FINTP3 - one dependent, three independnet variables.

The independent variable tables must be monotonically increasing data arrays.

The argument list variables of the three subroutines are:

1. TABLE (X, XI, YI, NX, XK, XLABEL, Y)

   X - Instantaneous value of independent variable

   XI - Independent variable array (table)

   YI - Dependent variable array (table)

   NX - Number of points in each table

   XK - Flag used in FINTP1

XLABEL - Reserved for holorith character name of data table

Y - Linear interpolated value of the dependent variable.

2. TABL2 (X, Y, XYI, ZI, NXY, XINTER, XLABEL, Z)

X - Instantaneous value of the first independent variable XI

Y - Instantaneous value of the second independent variable YI

XYI - This single array contains both the first (XI) and second (YI) independent variable tabular values. The first independent variable (XI) values occupies the leading element locations and the second independent variable (YI) values occupies the remaining element locations.

ZI - Dependent variable array. This array contains all dependent variable tabular values. The dependent variable values as a function of the first independent variable XI are sub-groups of data in the array. The sub-groups are then a function of the second independent variable YI. (The sub-groups must be ordered such that they are a function of YI with YI monotonically increasing.)

NXY(I) - Number of points in the data tables

I = 1, number of first independent variable (XI) data points

I = 2, number of second independent variable (YI) data points

XINTER - Flag used in FINTP2

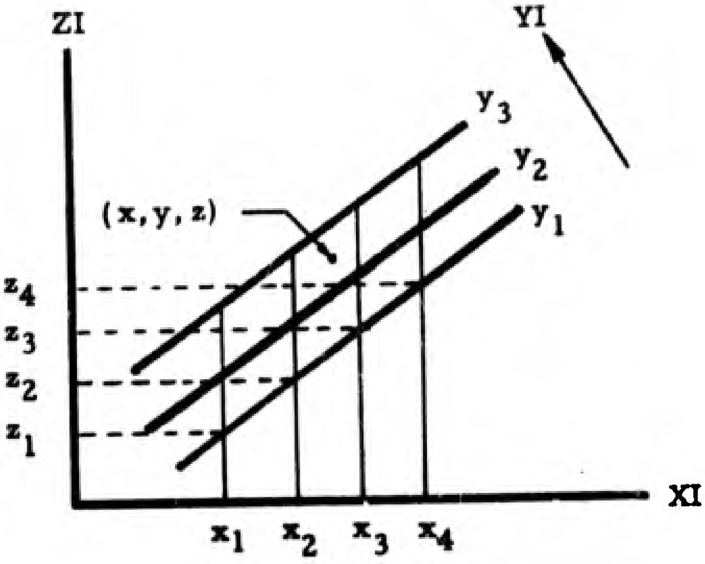
XLABEL - Reserved for holorith character name of data table

Z - Linear interpolated value of the dependent variable.

Example of TABL2 use.

Assume the following set of data.

ZI	XI	YI
z_1	x_1	y_1
z_2	x_2	
z_3	x_3	
z_4	x_4	
z_5	x_1	y_2
z_6	x_2	
z_7	x_3	
z_8	x_4	
z_9	x_1	y_3
z_{10}	x_2	
z_{11}	x_3	
z_{12}	x_4	



The tabular values above would be assigned to the argument list arrays as follows:

$$XYI = x_1, x_2, x_3, x_4, y_1, y_2, y_3$$

$$ZI = \left| \overbrace{z_1, z_2, z_3, z_4}^{y_1} \right| \overbrace{z_5, z_6, z_7, z_8}^{y_2} \left| \overbrace{z_9, z_{10}, z_{11}, z_{12}}^{y_3} \right|$$

The number-of-point variables would be set as:

NXY(1) = 4

NXY(2) = 3

The discrete point (x, y) would be passed into the subroutine, a linear interpolation would be performed between the data points that bracketed (x, y) and a singular value of z would be returned. Thus,

$$\left.\begin{array}{l} X = x \\ Y = y \end{array}\right\} \quad \text{input}$$

Z = z - returned value

3. TABL3 (X, Y, Z, XYZI, WI, NXYZ, XINTER, XLABEL, W)

X - Instantaneous value of the first independent variable XI

Y - Instantaneous value of the second independent variable YI

Z - Instantaneous value of the third independent variable ZI

XYZI - This single array contains all three independent variable tabular values. The first independent variable (XI) values appears first, the second independent variable (YI) values appear second, and the third independent variable (ZI) values appear last.

WI - Dependent variable array. This array contains all dependent variable tabular values. The dependent variable values as a function of the first independent variable (XI) appears as a sub-group of the second independent variable YI with this grouping a sub-group of the third independent variable ZI.

NXYZ(I) - Number of points in the data tables

I = 1, number of first independent variable (XI) data points

I = 2, number of second independent variable (YI) data points

I = 3, number of third independnet variable (ZI) data points

XINTER - Flag used in FINTP2

XLABEL - Reserved for holorith character name of data table

W - Linear interpolated value of the dependent variable.

3.4 AUXILIARY AND FUNCTION SUBROUTINES

3.4.1 Aerodynamic Coefficient Table Error Indication Subroutines: AERROR and TERROR

Subroutine AERROR is used to indicate that the numberical range of an aerodynamic coefficient data table has been exceeded by the trajectory being flown. The message "OUT OF AERO TABLE ARGUMENT ARRAY" and the table name is printed out, along with a dump of the C-array element string that contains all the aerodynamic table values. The integration termination switch, LCONV, is then set to 2 to flag Subroutine STGE3 that the run is to be terminated.

Subroutine TERROR is used to indicate that an entire aerodynamic coefficient table is missing. The message "NO AERO POINTS SPECIFIED FOR ARG" and the missing table name is written out. The run is then terminated by a CALL EXIT statement.

(NOTE: These two subroutines are not being utilized by the program. No provision has been made to detect these errors in the aero tables not to call these subroutines.)

3.4.2 AMRK-Numerical Integration Routine

3.4.2.1 Function Description

Numerical integration of the state vector derivatives is accomplished using a standard Adams-Moulton technique with a Runge-Kutta start.

Numerical solutions to second order differential equations are obtained sequentially by quadrature of two first order differential equations. The Adams-Moulton portion of the routine includes a variable step size which halves or doubles the step size as a function of the difference between the predicted and corrected values of the integrals. The step size is bounded by input upper and lower limits.

3.4.2.2 Equations

The four-point Adams-Moulton routine requires the initial four derivatives for each state vector derivative. These initial values are obtained from three calls to the four pass Runge-Kutta routine, plus the derivatives representing initial conditions. The equations are standard. Specifically, if $\bar{y}_{i+1}^{[1]} = f(t_i, \bar{y}_i)$, then $\bar{y}_{i+1}$ is obtained from:

$$\bar{y}_{i+1} = \bar{y}_i + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4),$$

where

$$k_1 = f(t_i, \bar{y}_i)\, \delta t$$

$$k_2 = f(t_i + \frac{\delta t}{2}, \bar{y}_i + \frac{k1}{2})\, \delta t$$

$$k_3 = f(t_i + \frac{\delta t}{2}, \bar{y}_i + \frac{k2}{2})\, \delta t$$

$$k_4 = f(t_i + \delta t, \bar{y}_i + k_3)\, \delta t .$$

$\bar{y}'_{i+1}$ is obtained by calling (in AMRK) the subroutine AUXSUB, which calls the functional modules to calculate new derivatives based on $\bar{y}_{i+1}$. The derivatives $\bar{y}'_{i+1}$, obtained in each complete iteration through the Runga-Kutta routine and AUXSUB, are stored in an array ("T") for use in starting the Adams-Moulton routine. Thus, after three complete iterations, the derivatives, $\bar{y}'_o$, at the initial point, t_o, and the derivatives, $\bar{y}'_1$, $\bar{y}'_2$, and $\bar{y}'_3$, at points $t_o + \delta t$, $t_o + 2\delta t$, and $t_o + 3\delta t$, are stored in the "T" array.

At subsequent integration steps, the Adams-Moulton routine performs the following two calculations:

$$\text{Predicted: } (\bar{y}_{i+1})_p = \bar{y}_i + \delta t (p_1 \bar{y}'_i - p_2 \bar{y}_{i-1} + p_3 \bar{y}'_{i-2} - p_4 \bar{y}'_{i-3})$$

$$\text{Corrected: } (\bar{y}_{i+1})_c = \bar{y}_i + \delta t (c_1 \bar{y}'_{i+1} + c_2 \bar{y}'_i - c_3 \bar{y}'_{i-1} + c_4 \bar{y}'_{i-2})$$

where $\bar{y}'_{i+1}$, appearing in the equation for $(\bar{y}_{i+1})_c$

is obtained by calling AUXSUB after $(\overline{y}_{i+1})_p$ is determined. The coefficients are[2]:

$$p_1 = \frac{55}{24}, \; p_2 = \frac{59}{24}, \; p_3 = \frac{37}{24}, \; p_4 = \frac{9}{24}$$

$$c_1 = \frac{9}{24}, \; c_2 = \frac{19}{24}, \; c_3 = \frac{5}{24}, \; c_4 = \frac{1}{24} .$$

The absolute value of $(\overline{y}_{i+1})_c - (\overline{y}_{i+1})_p$ for each element of the vectors is then compared with preset (by subroutine A5I) upper and lower bounds stored in the arrays "EU" and "EL". If the absolute value of the difference for any element of the vectors exceeds EU_i, an attempt is made to half the step size. If halving the step size decreases it below the integration step size, HMIN (an input value), the step size is not changed, and the value of $(\overline{y}_{i+1})_p$ is accepted for $\overline{y}_{i+1}$. An input option switch (RITE >0) triggers a printout of the simulation time, the absolute value of the difference between the predictor and the corrector, and the state vector element number, whenever the upper bound, EU_i, is exceeded. The step size can be doubled in a manner analogous to halving it if the absolute values of the difference between predicted and corrected values are less than preset lower bounds, EL_i. However, in this case, <u>all</u> of the values must be less than EL_i before an attempt will be made to double the step size. Again, the step size will not be changed if doubling the step size would increase it above the input step size upper bound, HMAX.

---

[2] W. E. Milne, Numerical Solutions of Differential Equations, Dover Publications, Inc., (1970).

When the step size is halved, all four derivatives ($\overline{y}_i$, $\overline{y}_{i-1}$, $\overline{y}_{i-2}$, and $\overline{y}_{i-3}$) must be defined for the same step size.* Since two of the derivatives thus required ($\overline{y}_{i-1}$ and $\overline{y}_{i-3}$) have not been previously defined, they can be obtained either by backing the time variable to the appropriate points and recomputing derivatives, or by linear interpolation from the existing derivatives. The latter alternative is used in AMRK, and is illustrated in Figure 3.4.2 where the initial situation is depicted in Figure 3.4.1. Once these derivatives are obtained, the procedure of calculating predicted and corrected values continues as previously described.

The occurrence of two step size halvings in sequence would require that some derivatives be obtained by interpolating between values previously obtained by interpolation. Rather than permit this to occur, AMRK returns control to the Runge-Kutta routine to recalculate the derivatives, starting at the third preceding time point. Operation of AMRK then proceeds as previously described.

When the step size is doubled, the derivatives must, as before, be defined at equally spaced intervals. However, in this case, the needed derivatives (essentially every other previous one) have already been defined and are stored in the "T" array. Hence, all that is required is to redefine the "T" array such that the derivatives used in the predictor and

---

*Because the coefficients of the predictor and corrector equations are derived based on the assumption of equally spaced intervals (see Reference 2 on page 172.

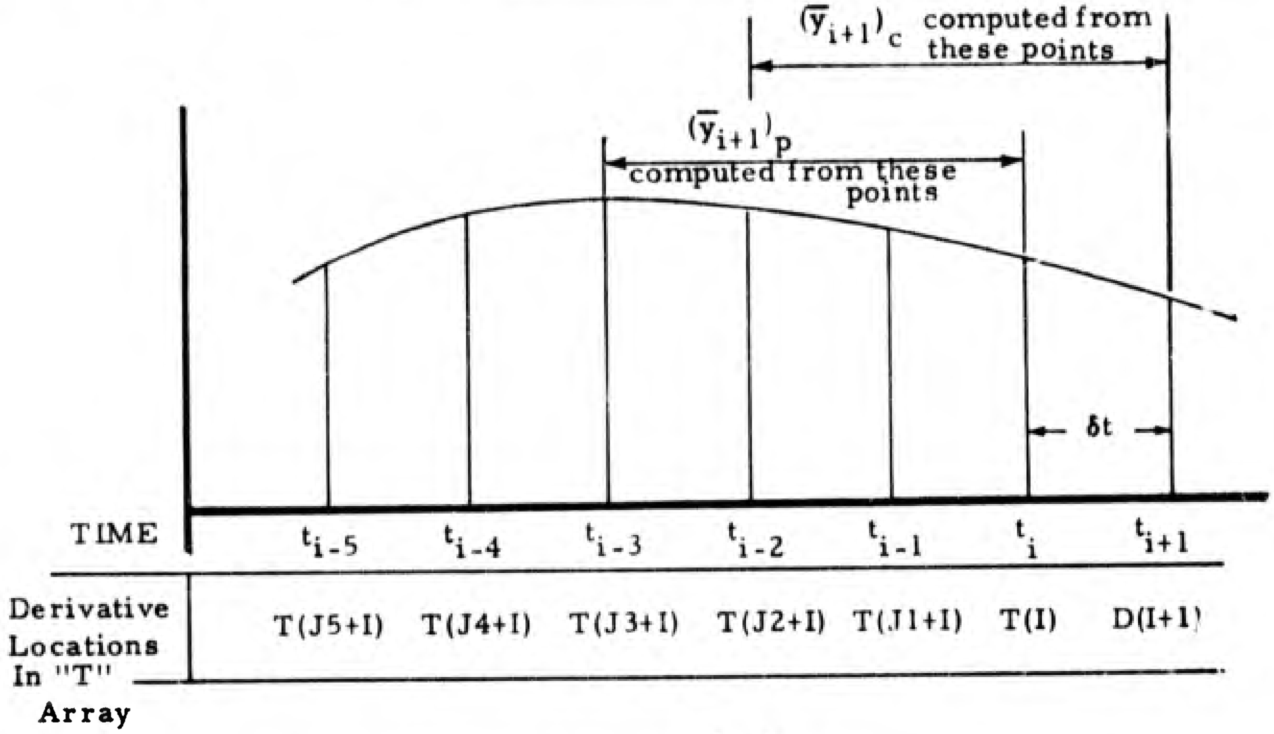


Figure 3.4.1 - Predictor, $(\bar{y}_{i+1})_p$, and Corrector $(\bar{y}_{i+1})_c$, with Current Step Size

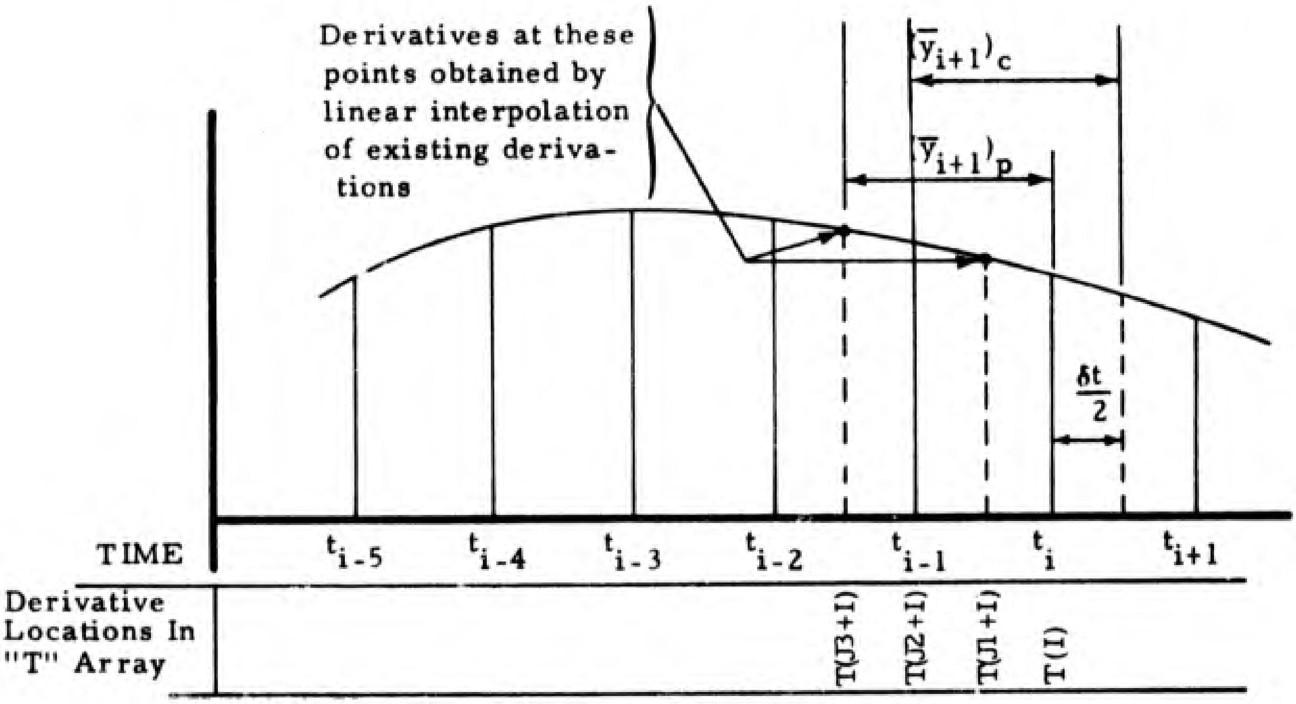


Figure 3.4.2 - New Predictor, $(\bar{y}_{i+1})_p$, and Corrector $(\bar{y}_{i+1})_c$, after Halving Step Size

corrector equations will represent the proper time points. Because of the limitation of the site of the "T" array, however, doubling is prohibited from occurring until two integrations have occurred after a previous doubling (it is clear that successive doublings would require an increasing number of past derivatives to be stored). Doubling of the step size is also inhibited until two integration steps after the step size has been halved. In this case, the reason is that the derivatives in the "T" array represent both the full and the halved step sizes, and hence, are not defined at equally spaced intervals. Step size doubling is illustrated in Figures 3.4.3, 3.4.4, and 3.4.5, where the initial state is shown in Figure 3.4.1.

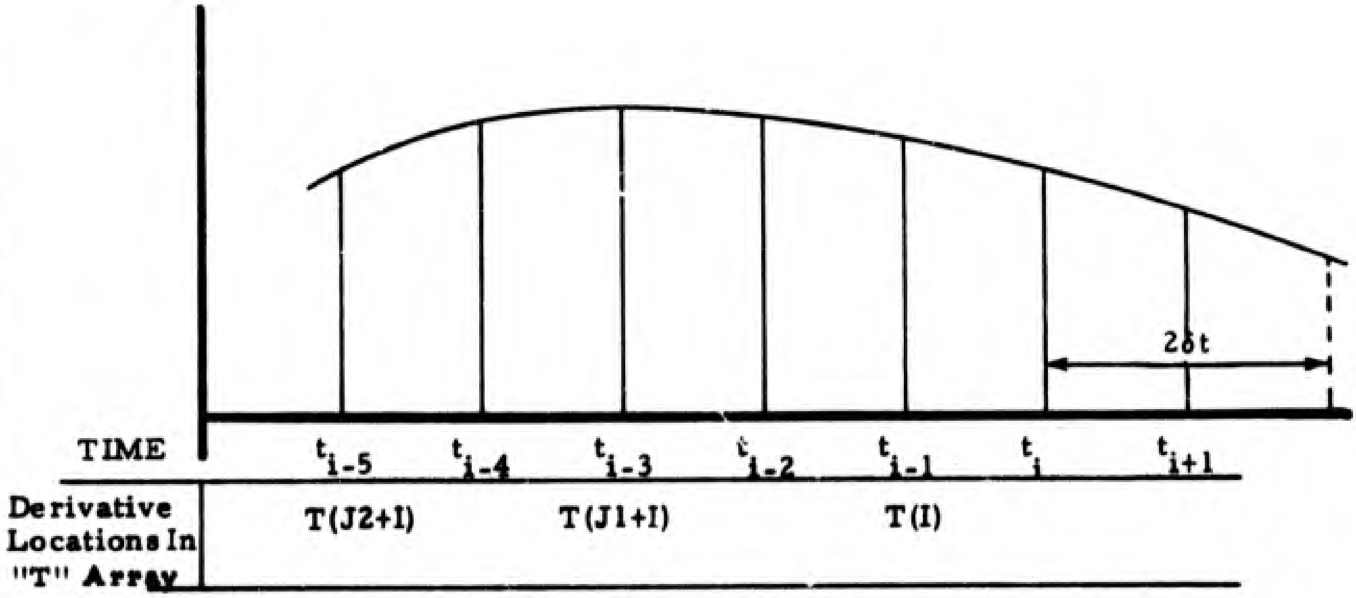


Figure 3.4.3 - Derivative Locations in "T" Array Immediately after Doubling Step Size

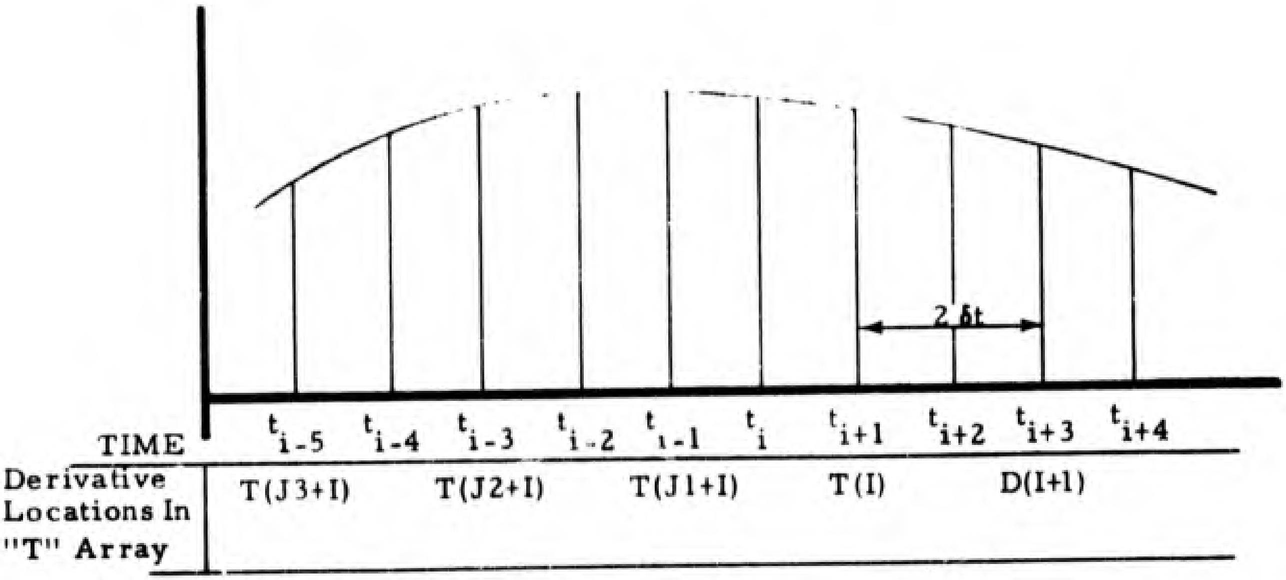


Figure 3.4.4 - Derivative Locations In "T" Array after Doubling Step Size and Calling AUXSUB (functional modules)

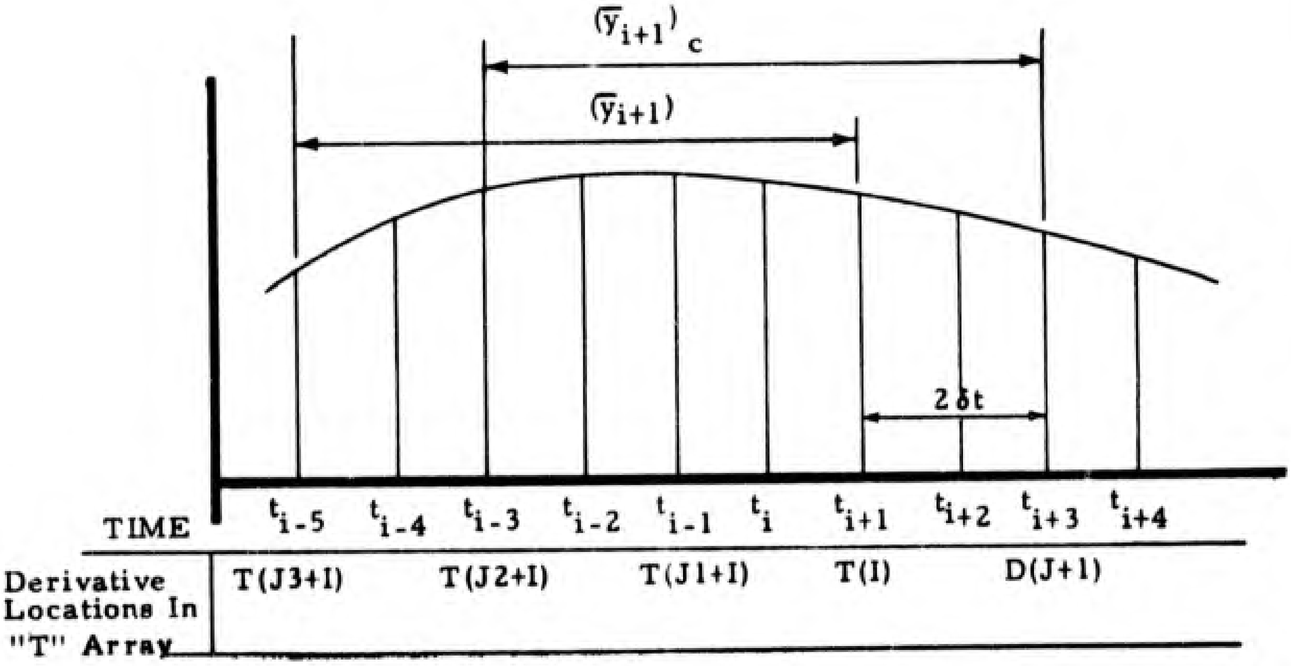


Figure 3.4.5 - Predictor, $(\bar{y}_{i+1})_p$, and Corrector, $(\bar{y}_{i+1})_c$, after Doubling Step Size

3.4.2.3 Input/Output and Cross Reference of C-Array

1. Input from data cards

FORTRAN SYMBOL	SYMBOL USED IN TEXT	C INDEX	DEFINITION
HMIN	HMIN	2662	Integration Step Size Lower bound (SEC)
HMAX	HMAX	2663	Integration Step Size Upper bound (SEC)
D(1)	δt	2664	Initial integration step size (SEC)
RITE	RITE	1971	Option Switch for step size halving printout (>0 = Print)

2. Input from other modules

FORTRAN SYMBOL	SYMBOL USED IN TEXT	C INDEX	DEFINITION
D(I)	y'_i	2664	Value of Ith state element of state vector
EL(I)	EL_i	2765	Lower bound to absolute value between predicted and corrected integrated value of Ith derivative.
EU(I)	EU_i	2865	Upper bound to absolute value between predicted and corrected integrated value of Ith derivative.
V(I)	y_i	2965	Value of Ith state variable.

3. Outputs

FORTRAN SYMBOL	SYMBOL USED IN TEXT	C INDEX	DEFINITION
V(I)	y_i	2965	Integrated value of Ith element of state vector

3.4.3 Trig Special Function Routines: COSD, SIND and ATAND

The special function trig routines COSD, SIND and ATAND allow the user to work in units of degrees when dealing with angles. Use of the Fortran supplied functions COS, SIN and ATAN2 is made through these intermediate special function routines.

Input to and use of the special functions COSD and SIND are:

```
X = COSD(Z)
Y = SIND(Z)
```

where Z is an angle in degrees.

Output of the special function ATAND is

```
Z = ATAND(Y, X)
```

where Z is an angle in degrees with assignment to the proper quadrant and a range of ± 180 degrees.

3.4.4 BLOCK DATA - Aerodynamic Coefficient Tables

3.4.4.1 Function Description

The aerodynamic coefficient data tables are stored by BLOCK DATA and the data tables are assigned to labeled COMMON blocks in BLOCK DATA. The aerodynamic data tables are passed to the table look-up subroutine, Module A1, through the labeled COMMON blocks for linear interpolation within the tables.

3.4.4.2 The Data Tables Stored by BLOCK DATA

Axial force coefficient at zero angle of attack

$$C_A = f_1 \text{ (MACH)}$$

C_A						
MACH	0	.8	.9	1.1	1.4	2.0

Pitching moment coefficient at zero angle of attack

$$C_{mo} = f_2 \text{ (MACH)}$$

C_{mo}						
MACH	.5	.85	.95	1.05	1.25	2.0

Normal force coefficient

$$C_{N'}(\alpha') = f_1 \text{ (MACH, } \alpha')$$

$C_{N'}$								MACH
								.5
								.85
								.95
								1.05
								1.25
α'	0	2	4	8	12	16	20	

Pitching moment coefficient

$$C_{m'}(\alpha') = f_2(\text{MACH}, \alpha')$$

α'	0	2	4	8	12	16	20	MACH
$C_{m'}$								.5
								.85
								.95
								1.05
								1.25

Normal force coefficient due to fin pitch position

$$\left(\frac{C_N'}{\delta q}\right)_{\phi'=0} = f_6(\text{MACH}, \alpha')$$

α'	0	2	4	8	12	16	20	MACH
$\frac{C_N'}{\delta_q}$								.5
								.85
								.95
								1.05
								1.25

Pitch control moment coefficient due to pitch control

$$\left(\frac{C_m'}{\delta q}\right)_{\phi'=0} = f_4(\text{MACH}, \alpha)$$

α'	0	2	4	8	12	16	20	MACH
$\frac{C_{m'}}{\delta_q}$								.5
								.85
								.95
								1.05
								1.25

Incremental pitching moment coefficient due to aerodynamic roll angle (ϕ')

$$\Delta C_{m'} = F_3 \text{ (MACH, } \alpha' \text{)}$$

								MACH
$\Delta C_{m'}$								.5
								.85
								.95
								1.05
								1.25
α'	0	2	4	8	12	16	20	

Incremental yawing moment due to aerodynamic roll angle (ϕ')

$$\Delta C_{n'} = f_8 \text{ (MACH, } \alpha' \text{)}$$

							MACH
$\Delta C_{n'}$							0
							.7
							.9
							1.1
							1.4
							2.0
α'	0	4	8	12	16	20	

Pitch damping moment coefficient

$$C_{mq} = f_{12} \text{ (MACH, } \alpha \text{)}$$

							MACH
C_{mq}							0
							.7
							.9
							1.1
							1.4
	-1.75						2.0
α	0	4	8	12	16	20	

Incremental side force coefficient due to aerodynamic roll angle (ϕ')

$$\Delta C_{Y'} = f_7 (\text{MACH}, \alpha')$$

$\Delta C_{Y'}$							MACH
							0
							.7
							.9
							1.1
							1.4
							2.0
α'	0	4	8	12	16	20	

Roll damping moment coefficient

$$C_{\ell p} = f_{13} (\text{MACH}, \alpha)$$

$C_{\ell p}$							MACH
							0
							.7
							.9
							1.1
							1.4
							2.0
α'	0	4	8	12	16	20	

Incremental normal force coefficient due to aerodynamic roll angle (ϕ')

$$\Delta C_{N'} = f_5 (\text{MACH}, \alpha')$$

$\Delta C_{N'}$								MACH
								.5
								.85
								.95
								1.05
								1.25
α'	0	2	4	8	12	16	20	

Rolling moment coefficient due to fin roll control

$$\left(\frac{C_{\ell'}}{\delta_p}\right)_{\Phi'=0} = f_{11}\,(\text{MACH}, \alpha')$$

α'	0	2	4	8	12	16	20	MACH
$\frac{C_{\ell'}}{\delta_p}$								.5
								.95
								1.25

Incremental rolling moment due to aerodynamic roll angle (Φ')

$$\Delta C_{\ell'} = f_{10}\,(\text{MACH}, \alpha')$$

α'	0	2	4	8	12	16	20	MACH
$\Delta C_{\ell'}$								.5
								.95
								1.25

Incremental rolling moment due to aerodynamic roll angle (Φ') and launch lugs

$$\Delta C_{\ell'_{\text{LUGS}}} = f_9\,(\text{MACH}, \alpha')$$

α'	0	2	4	8	12	16	20	MACH
C LUGS								.5
								.95
								1.25

3.4.4.3 Output

BLOCK DATA is not an executable subroutine, therefore, no output is allowed. However, BLOCK DATA does assign the data tables (dependent and independent) to array names and to labeled COMMON blocks for storage in core.

The following table gives the array name of the dependent and the independent variable(s), and their COMMON block assignments. In the table below, the dependent variable name is listed first, followed by its independent variable name(s). Where two independent variables are involved, they are listed in the order in which they appear in the COMMON block (both independent variable tables are assigned to one COMMON Block when two independent variables are involved.

FORTRAN SYMBOL		SYMBOL USED IN TEXT		COMMON BLOCK
Dependent Variable	Independent Variable(s)	Dependent Variable	Independent Variable(s)	
CXO	VM1	C_A	MACH	CXOF CA1
CMO	VM4	C_{mo}	MACH	CMOF CA4
CZP	BA3 VM3	$C_{N'(\alpha')}$	α' MACH	CZPF CA3 CA3
CMP	BA3 VM3	$C_{m'(\alpha')}$	α' MACH	CMPF CA3 CA3
CZD1	BA3 VM3	$\frac{C_{N'}}{\delta_q}$	α' MACH	CZDF CA3 CA3
CMD1	BA3 VM3	$\frac{C_{m'}}{\delta_q}$	α' MACH	CMDF CA3 CA3
CM2	BA3 VM3	$\Delta C_{m'}$	α' MACH	CM2F CA3 CA3
CN4	BA2 VM2	$\Delta C_{n'}$	α' MACH	CN4F CA2 CA2
CMQ	BA2 VM2	C_{mq}	α MACH	CMQF CA2 CA2

FORTRAN SYMBOL		SYMBOL USED IN TEXT		COMMON BLOCK
Dependent Variable	Independent Variable(s)	Dependent Variable	Independent Variable(s)	
CY4		$\Delta C_Y'$		CY4F
	BA2		α'	CA2
	VM2		MACH	CA2
CLP		$C_{\ell p}$		CLPF
	BA2		α'	CA2
	VM2		MACH	CA2
CZ2		$\Delta C_N'$		CZ2F
			α'	CA3
			MACH	CA3
CLD1		$\frac{C_\ell'}{\delta_p}$		CLDF
	BA5		α'	CA5
	VM5		MACH	CA5
CL4		$\Delta C_\ell'$		CL4F
	BA5		α'	CA5
	VM5		MACH	CA5
CL2		${\Delta C_\ell'}_{Lugs}$		CL2F
	BA5		α'	CA5
	VM5		MACH	CA5

3.4.5 DUMMY - Auxiliary Subroutine

Subroutine DUMMY contains, as ENTRY points, all of the subroutine names called in the simulation program. This feature allows the program to be modularly altered (e.g., removing a module or subroutine or substituting a module or subroutine) without the necessity of locating and removing all calls to non-existent subroutines. A disadvantage to this approach is that the inadvertent omission of a needed subroutine will be detectable only by erroneous output results, because the subroutine will exist as an entry point in DUMMY, and a call to the subroutine will not cause a premature program termination.

3.4.6 DUMPO - Auxiliary Subroutine

Subroutine DUMPO prints the entire "C" array when called. With the current program configuration, DUMPO can be called a maximum of seven times; six times by the input variable DOC and one time by the input variable OPTN10.

DOC allows C-array dumping for six integration intervals, starting with t=0, and continuing for the next five integration steps. The input variable, DOC, is set to the number of "C" array dumps fewer than six desired. Thus, if the six dumps are desired, DOC would be set to zero on input. If no dumps were desired, DOC would be set to six, etc. DUMPO is called by subroutine OUPT2 to provide a "C" array dump at t=0 if DOC is input less than six. Subsequent calls (if any) of DUMPO are made from subroutine OUPT3, within the integration loop.

If the input variable, OPTN10, is set greater than zero, the MAIN program will call DUMPO at the end of a case and dump the C-array.

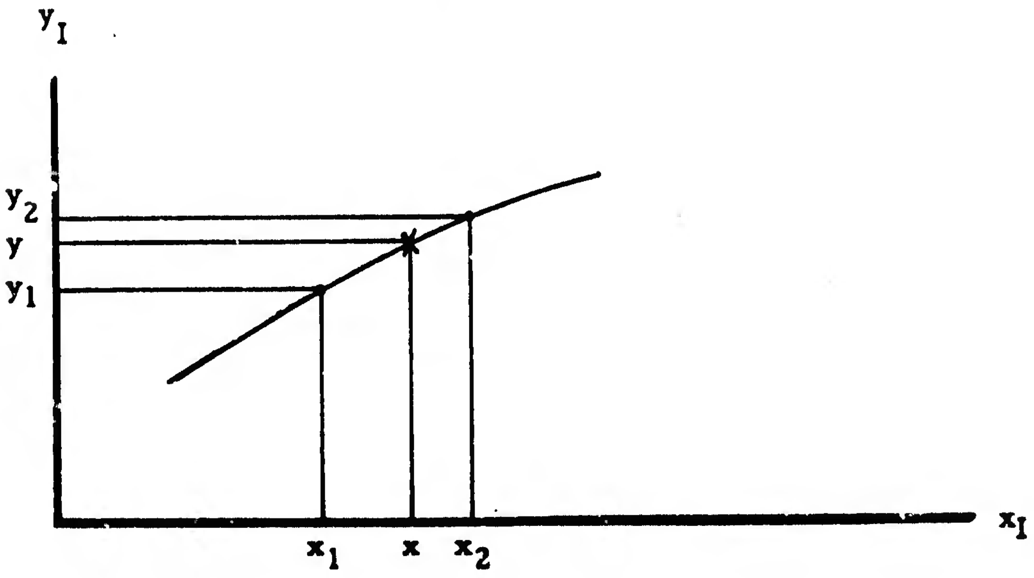
3.4.7 FINTP1 - Aero Table Linear Interpolation of a Function with one Dependent and One Independent Variable

Function FINTP1 utilizes the following equation to linearly interpolate between the two points for the value of the dependent variable when the dependent variable is a function of only one independent variable.

$$y = Pct\ (y_2 - y_1) + y_1$$

where:

$$P_{ct} = \left(\frac{x - x_1}{x_2 - x_1}\right) .$$



The argument list variables, which are internal inputs to the subroutine, are defined as:

X - Instantaneous value of the independent variable XI
XI - Independent variable array
YI - Dependent variable array
N - Number of points in the arrays
F - Flag that allows bypassing the computation of PCT.
XL - (Not Used)

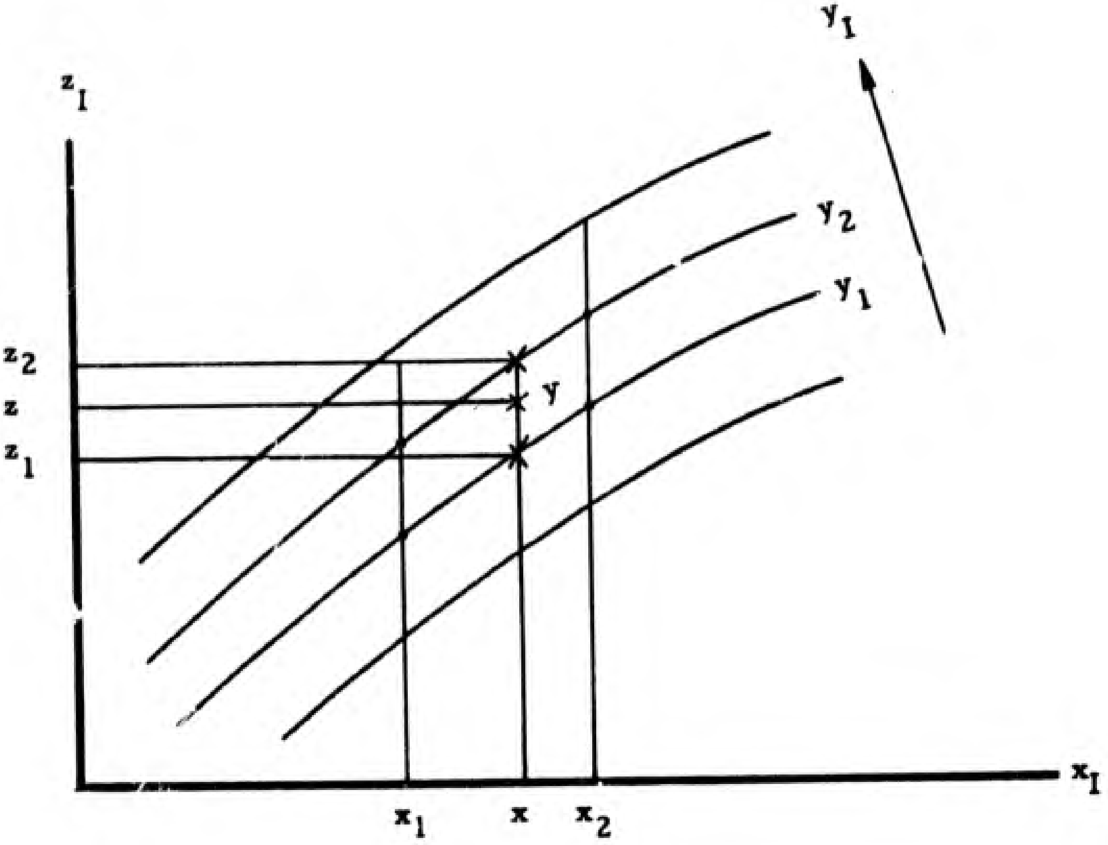
3.4.8 FINTP2 - Aero Table Linear Interpolation of a Function With One Dependent and Two Independent Variables

Function FINTP2 utilizes the following equations in conjunction with FUNCTION FINTP2 to linearly interpolate between two points for the value of the dependent variable when the dependent variable is a function of two independent variables.

$$z = Pct\ (z_2 - z_1) + z_1$$

where

$$P_{ct} = \left(\frac{y - y_1}{y_2 - y_1}\right)$$



The z_1 and z_2 values are computed in FUNCTION FINTP1 at the value x of the independent variable XI.

The argument list variables, which are internal inputs to the subroutine, are defined as:

X	- Instantaneous value of the first independent variable XI
Y	- Instantaneous value of the second independent variable YI
XI	- First independent variable array
YI	- Second independent variable array
ZI	- Dependent variable array

NXO - Number of points in the first independent variable array

NY - Number of points in the second independent variable array

NX - Number of points in the first independent array

F - Flag that allows bypassing the computation of PCT

XL - (Not Used)

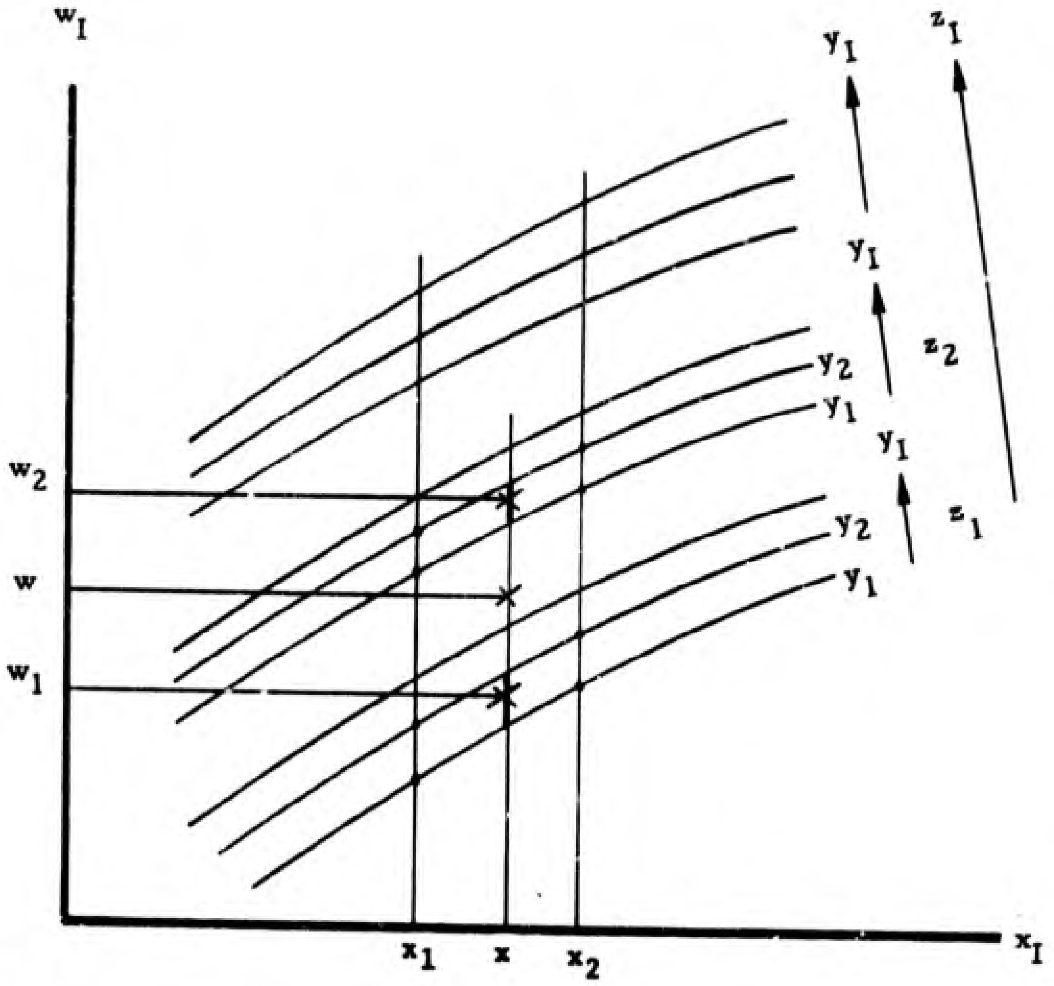
3.4.0 FINTP3 - Aero Table Linear Interpolation of a Function With One Dependent and Three Independent Variables

Function FINTP3 utilizes the following equations in conjunction with FUNCTION FINTP2 to linearly interpolate between two points for the value of the dependent variable when the dependent variable is a function of three independent variables.

$$w = P_{ct} (w_2-w_1)+w_1$$

where

$$P_{ct} = \left(\frac{z - z_1}{z_2-z_1}\right)$$



The w_2 and w_1 values are computed in FUNCTION FINTP2 at the value y of the independent variable YI. The z_2 and z_1 values are computed in FUNCTION FINTP1 at the value x of the independent variable XI.

The argument list variables, which are internal inputs to the subroutine, are defined as:

X	-	Instantaneous value of the first independent variable XI.
Y	-	Instantaneous value of the second independent variable YI.

XI - First independent variable array

YI - Second independent variable array

ZI - Third independent variable array

WI - Dependent variable array

NZ - Number of points in the ZI array

NY - Number of points in the YI array

NX - Number of points in the XI array

F - Flag to bypass computation of PCT in FUNCTION FINTP2

XL - (Not used)

3.4.10 RESET - Multi-Case Variable Reinitialization

Subroutine RESET is called by MAIN prior to the second and subsequent trajectory simulation cases. Variables for which the reset flag was set on the Type 3 input data cards will be reinitialized to the initial values specified on the corresponding Type 3 cards. Thus, if a Type 3 card for C(1615) (missile X-coordinate) specified an initial value of −1000 ft., and the reset flag (columns 46-60) was 1., then subroutine RESET would reinitialize C(1615). The operation of RESET consists of effectively setting C(LISTNO(I)) equal to VALUE (I), for I ranging from one to NOLIST. NOLIST, and the arrays LISTNO(I) and VALUE(I), are defined by the input subroutine OINPT1. NOLIST specifies the number of elements in the C-array which are to be reset, LISTNO(I) specifies the index of the C-array of the Ith element to reset, and VALUE(I) specifies the initial value to which C(LISTNO(I)) will be reset.

3.4.11 Subroutine STGE2

STGE2 initializes the STGE3 switches KCONV, LCONV, and KSTEP. These switches are initialized as:

KCONV = 0

LCONV = 0

KSTEP = 1

3.4.12 Subroutine STGE3

3.4.12.1 Function Description

STGE3 controls normal trajectory computation termination and calls OUPT3 for output following trajectory termination. There are four modes of normal trajectory computation termination handled by STG3. The events defining these four modes are:

1. Flight time (T) reaches the input maximum flight time boundary (TF).
2. Missile reaches the point of closest approach to the target (trajectory intersects the target plane). (Flagged in G4)
3. Missile flies into the ground. (Flagged in G4).

4. Missile actual aerodynamic flight regime is out of range of the aerodynamic tabular data flight regime. (Flagged in AERROR).

Trajectory computation will be terminated by the event that occurs first. When termination is to occur as a result of one of the above events, STGE3 sets the switch KSTEP equal to 2 to flag MAIN that the case is to be terminated.

Mode 1 is handled directly by STGE3. The actual flight time T is compared to the input maximum flight time boundary TF. If T is within .01 second of TF, STGE3 sets the switch KSTEP for case termination. However, if T exceeds TF, STGE3 initiates an integration step size halving scheme to iterate for a final integration step size to force T to fall within the .01 second tolerance of TF. If, however, T does not fall within this tolerance of TF after ten interval halving passes, the case will be automatically terminated.

Modes 2, 3 and 4 are handled indirectly by STGE3. These events are determined in other modules with the module making the determination setting the switch LCONV equal to 2 to flag STGE3 that an event has occurred and that trajectory computation is to terminate. STGE3 monitors LCONV and when a value of 2 appears, STGE3 sets the switch KSTEP equal to 2 to flag MAIN that the case is to be terminated.

3.4.12.2 Initialization Subroutine

Subroutine STGE2 initially sets the flags LCONV=0, and KSTEP=1. STGE2 is called by Subroutine SUBL2 at the start of trajectory initialization.

3.4.12.3 Definition of Variables Used in STGE3

FORTRAN SYMBOL	C INDEX	SOURCE MODULE(S) OR INPUT DATA	DEFINITION
T	2000	AMRK	Number of seconds into flight simulation (Flight time)
TF	2001	INPUT	Upper bound to simulation flight time (flight time cutoff), SEC)
PCNT	2003	OUPT2 OUPT3 STGE3	Time boundary in seconds below which no output/ plotting from OUPT3 is performed.
STEP	2010	INPUT	Switch that specifies the nature of the next case that follows termination of this case. (STEP=11 indicates that no additional case follows.)
KSTEP	2011	STGE2 STGE3	Switch used to flag MAIN that run termination has occurred. (Set to 1 in STGE2; set to 2 in STGE3) 1-continue integration 2-stop integration.
LCONV	2020	STGE2 G4 AERROR	Switch used to flag STGE2 that run termination is to occur. 0-continue integration 1- ? 2-terminate computation
DER(1)	2664	INPUT	Initial integration step size (SEC)

FORTRAN SYMBOL	C INDEX	SOURCE MODULE(S) OR INPUT DATA	DEFINITION
KCONV	2021	STGE2 OUPT2 STGE3	Pass counter for step size halving in the iteration loop to force T within the hard wired tolerance of .01 sec of TF. (Set to zero in STGE2 and OUPT2)
N	2561	Program Variables	Counter for loading IPL Array
NPT	1975		Integration mode control switch
NJ	1974		number of derivations

3.4.13 ZERO-Auxiliary Subroutine

Subroutine ZERO zeros certain elements of the C array. The variable names and the corresponding C array indices are:

VARIABLE NAME	C ARRAY INDEX
NPLOT	1984
OPOINT	2023
NOMOD	2361
NOSUB	2461
NOLIST	3066
NOOUT	3167
LOSTAT	3338
NOSTAT	3339
NORNDM	3440

With the current configuration of the simulation program, all elements of the C array are initially set to zero, thus making the function of this subroutine redundant.

3.5 INPUT/OUTPUT SUBROUTINES

3.5.1 OINPT1 - Input Subroutine

Subroutine OINPT1 reads the input data cards. The input information is stored in arrays determined by the data card type number (see Section 4 for input data card formats). Table 3.5.1 lists the input data card types and the array in which input information is stored. Other than the Type "6" card, which must come at the end of a data set, the cards can be input in any order. Within the module and staging subroutine control cards, however, the order must correspond to the desired execution order of the subroutines and modules.

The logic of OINPT1 reads variable values on Type "3" cards directly into the specified position of the C array, in the order in which the cards are read. Hence, a previously defined element of the C array will be overridden by the new value if the same C index is specified on another "3" card encountered later in the data card set. Thus, an input variable can be changed in the data card set by simply adding the desired card behind the original one(s) in the set. This cannot be done, however, with the data cards other than Type "3" cards. Adding an additional Type "2" (module selection) card, for example, without removing the one to be replaced will cause both modules to be executed in the integration loop.

DATA CARD TYPE	FUNCTION	DATA FIELD			STORED IN ARRAY		
		1	2	3	1	2	3
1	Selects output and staging subroutines	Subroutine Number			Subroutine Number		
2	Selects Modules	Module Number			Module Number		
3	Sets Input Variable to Initial Values	"C"Array Index	Index Variable Value	Re-set Flag		C(at index specified by Data Field #1). Also stored in VALUE if re-set flag is set.	LISTNO
4	Selects Variables to print at prescribed time intervals	Alpha Name Desired for print	"C"Array Index		ONAME1 ONAME2	OUTNO	
7	Selects Variables to plot at prescribed time intervals	Alpha Name Desired for plot	"C"Array Index		VLABLE	OUTPLT	
6	Flags end of data set	-	-	-	-	-	-

Table 3.5.1 - Input Data Card Information Storage Arrays

3.5.2 OUPT3 - Output Subroutine

3.5.2.1 Function Description

OUPT3 controls print out of all output data generated during integration of the missile trajectory and stores data to be plotted in the plot array, GRAPH, if NOPLOT is equal to 1. Specifically, OUPT3 performs the following:

1. The C array dump option (DOC) that allows output of the C array at the first DOC-6 integration steps is exercised in this routine. OUPT3 checks the flag ITCNT (if ITCNT≤6, dump C array) to determine if a C array dump is desired. ITCNT is set initially to DOC + 1 in OUPT2 and bumped up by 1 each pass through OUPT3 until it exceeds 6. No C array dump is made if ITCNT is greater than 6.

2. The trajectory time (T) and integration step size (DER(1)) is printed out by OUPT3 following a change in integration step size made in the integration subroutine, AMRK.

3. Output data specified on Type "4" input data cards is output from OUPT3. The symbol names located in the second and third ALPHA fields of Type "4" data cards are printed out five symbols per row with double spacing between each row. This symbol block is output

before the first block of output data is printed and then every two pages of print thereafter. The value of the parameters specified by the C index on Type "4" cards is printed out at each time point interval specified by the input variable CPP. These data blocks are output until two pages of print are filled. At this time, the symbol block is reprinted, followed by a resumption of the data block output print.

4. OUPT3 fills the plot array, GRAPH (if NOPLOT= 4) with data to be plotted as specified by Type "7" input cards. The number of data points to be plotted is monitored as the array is filled and if the number exceeds 300, the warning message"***WARNING-PLOTTING ARRAY FILLED-ONLY FIRST 300 POINTS PLOTTED***," is printed out. (OUPT3 does not control the plot routines. This is done from the MAIN program.)

3.5.2.2 Initialization Subroutine

OUPT2 sets KCONV, the run termination switch to zero, initializes certain output and plot flags and calls DUMPO to execute the first C array dump if DOC<6. OUPT2 transfers the initial values of all plot variables (specified by Type "7" input cards) into the plot variable storage array, GRAPH.

3.5.3 PLOT4 - PLOT Routine

3.5.3.1 Function Description

Data for up to fifteen variables for a given trajectory simulation will be automatically saved for plotting if any combination of three input plot options are exercised. The available options and the corresponding option switches are:

1. PLOTN4 = 4 provides data output to tape for missile and target X-Z plots.

2. PLOTN2 = K provides data output to tape for plotting K-1 dependent variables versus one specified independent variable (thus, K is set equal to the sum of the number of dependent variables and the one dependent variable).

3. PLOTNO = L provides data output to tape for plotting variables against time, where L is set such that the value of (PLOTNO-PLOTN2-PLOTN4) equals the number of variables to be plotted using this option (equivalently, PLOTNO must be set to the total number of variables to be plotted for all options, where each pair in 2., is counted as two variables.

To obtain actual plots, the plot tape is used as input to the Stomberg-Carson 4020 microfilm recorder.

The plot interval is set in seconds by the value of the input variable PPP. The number of time points is limited to a maximum of three-hundred. Whenever the quantity, (flight time)/PPP, is greater than three-hundred, a printout warns the user the maximum number of plot time points has been used, and all time points between the 299th and the last time point are ignored (the last time point is plotted).

The desired variable names to appear on the plot are input on Type "7" data cards. For the options previously listed in 1. and 2., both the abscissa and the ordinate are labeled by input values. In the option described in 3., the ordinate label is the input label, but the abscissa is automatically labeled "TIME".

The total number of points to be plotted will be reduced from the total number generated by the value of the input variable, PTLESS. Thus, if fifty points were generated during a simulation, and PTLESS were input with a value of thirty, then only the first twenty points would be plotted.

3.5.3.2 Input/Output and Cross Reference of C-Array

1. Input from data cards

FORTRAN SYMBOL	C INDEX	DEFINITION
PLOTN4	1982	Missile and target trajectory plot option switch. Set to 4 if option desired, 0 otherwise.
PLOTN2	1983	Number of variables to be plotted as pairs (e.g., X versus Y).
PTLESS	2007	Number of generated plot points to eliminate from end of trajectory
PLOTNO	2008	Total number of variables to plot.

If the PLOTN4 (missile and target trajectory plot) option is selected, the first four Type "7" input data cards (see Section 4 for card formats) must successively describe $X_{missile}$, $Z_{missile}$, X_{target}, and Z_{target}.

If the PLOTN2 (pairs of variables plotted, one versus the other) option is selected, the Type "7" cards must be input in pairs, with the abscissa variable first. These cards must directly follow the PLOTN4 cards (if any).

Any variables to be plotted versus time are input on Type "7" cards immediately following the PLOTN2 cards (if any).

2. Input from other modules

FORTRAN SYMBOL	C INDEX	DEFINITION
GRAPH (I, J)	3510	Storage array for the Ith time point of the Jth variable to be plotted. Defined in OUPT3.

4. INPUT/OUTPUT FORMAT

4.1 INPUT

4.1.1 Introduction

Program inputs fall into five major categories. The category of each data card input is identified to the input section of the program by the card type, punched in card column 1. The data categories and the corresponding card types are indicated in Table 4.1.1, below.

CARD TYPE (CARD COL. 1)	DATA CATEGORY
1	Output and staging subroutine selection
2	Module selection
3	Program parameter input values
4	Print variable selection
7	Plot variable selection
6	Data set termination indicator

Table 4.1.1 - Input Data Categories and Corresponding Card Type

The program is designed to sequentially simulate multiple trajectory cases in a single run. The input data sets, one for each case, are loaded back-to-back in the data deck (see Figure 4.1.1). The program input logic is structured to optionally allow the input data sets for successive cases to include only data which differs from or augments data in the immediately preceding case. Thus, for example, if case 2 differed from case 1 by only the value of the launch heading azimuth angle, then only the new value of launch heading azimuth angle would be required in data set 2.

Program variables which can be set by input are initialized to zero by the program prior to reading the input cards for the first case. Consequently, program constants (constant gains, deadband angles, etc.) which are to have values of zero do not require an input card. Program variables which are to be initialized to zero, but varies during simulation, however, will require an input card with the initial value reset flag set if multiple cases are being run, and the desired initial value of the variable is zero for each case.

The program input logic does not require that the cards be arranged according to card type sequence. Thus, Type 1, Type 2, etc., cards can appear anywhere in a data set; cards of a given type can even be separated by different card types. The order of appearance of the Type 2 (module selection cards), however, dictates the execution order of the

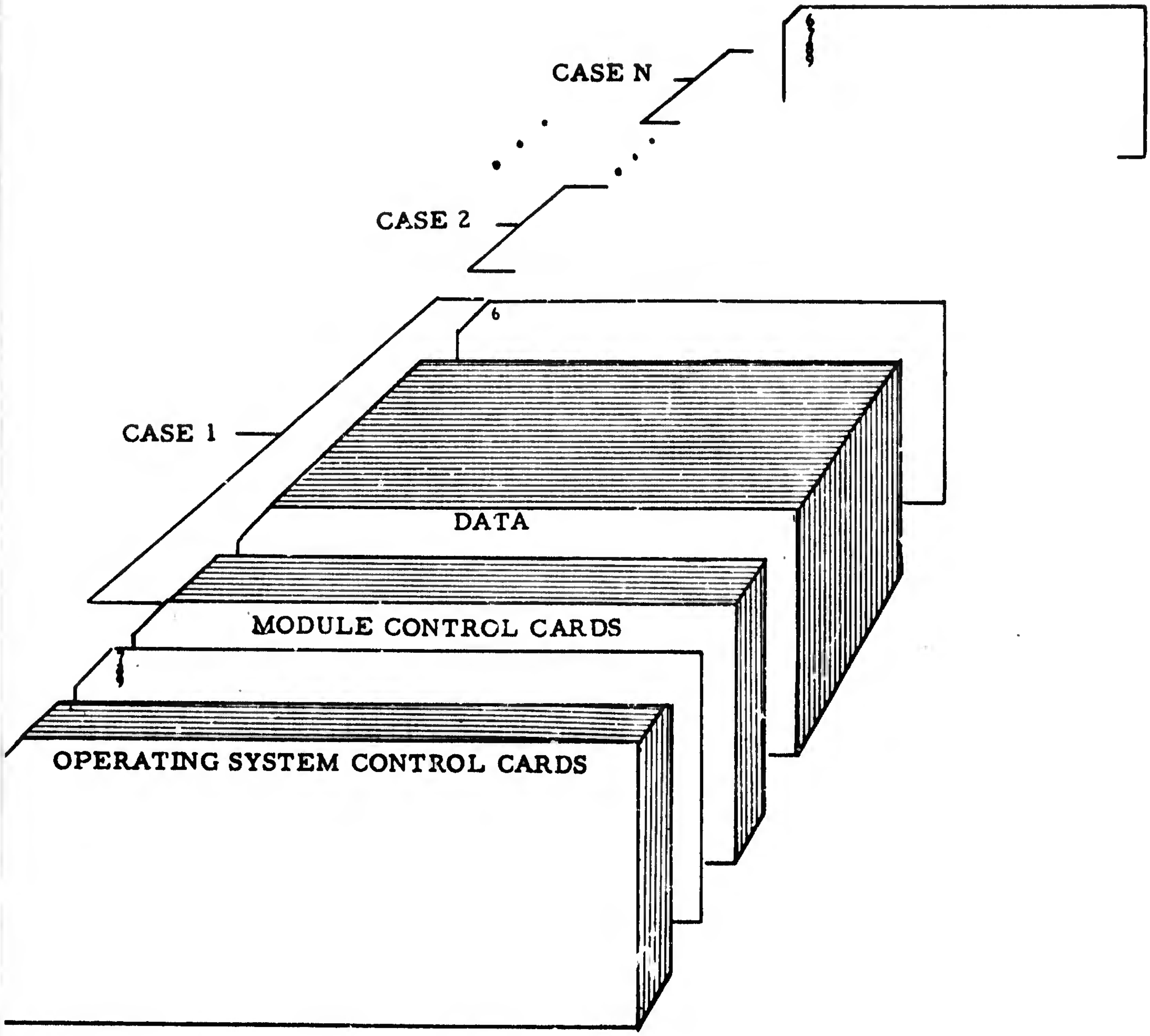
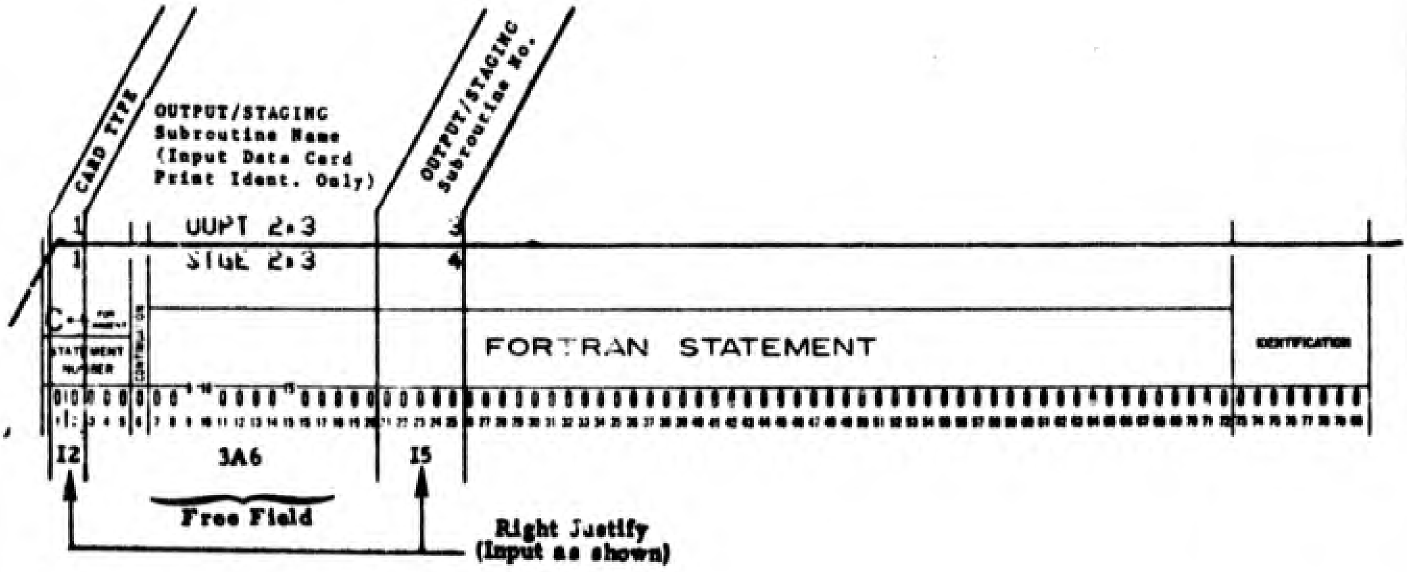


Figure 4.1.1 - Deck Set-Up

modules during a simulation. Because the module execution order is critical to a valid simulation, the order of appearance of the Type 2 cards within a data set must be as specified in Paragraph 3.

4.1.2 Card Type 1 - Output and Staging Subroutine Selection

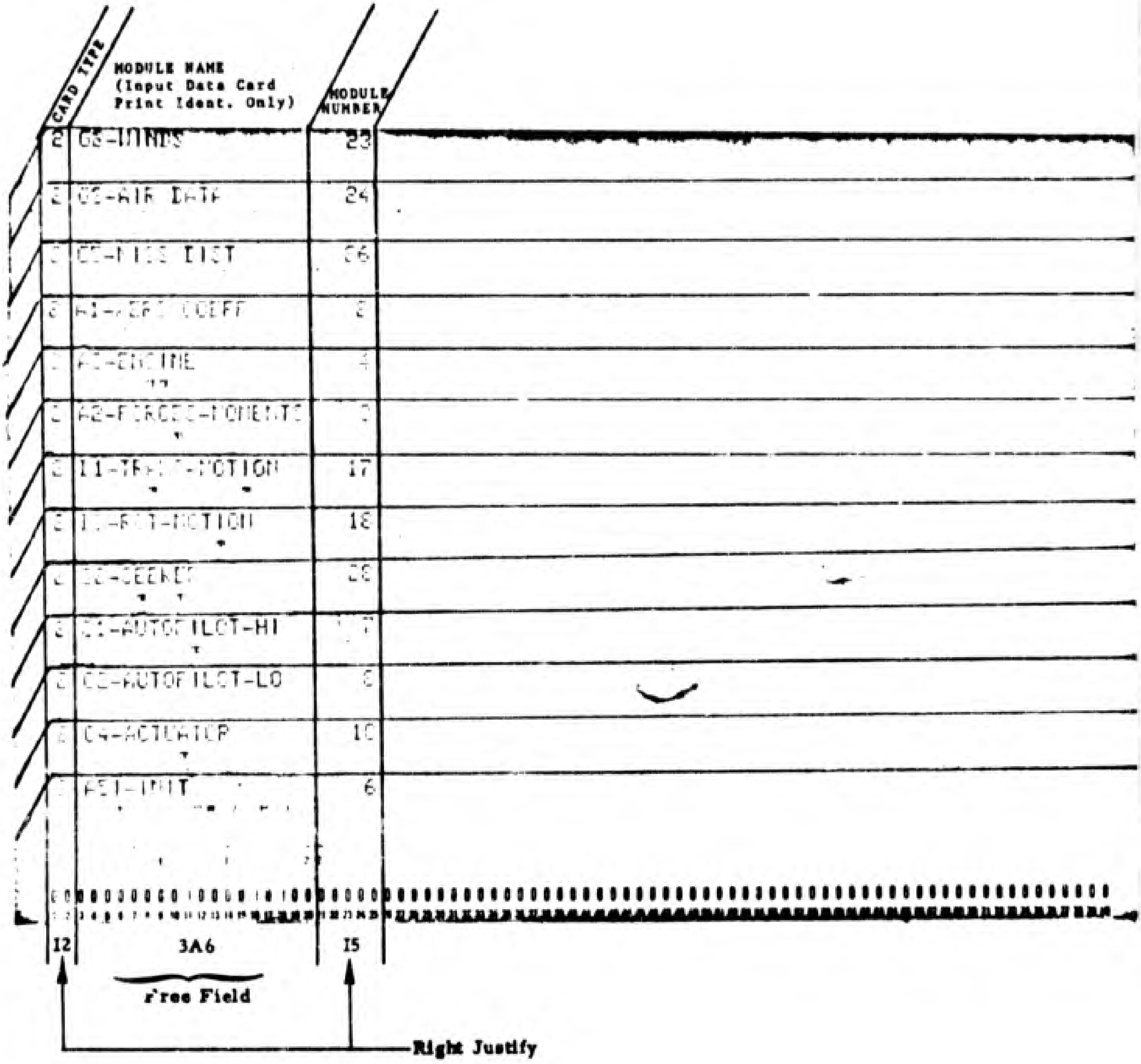
Data entered on Type 1 cards controls, in principle, which output and staging subroutines are used during a simulation. With the current configuration of the program, however, only one option is available. Therefore, the Type 1 cards must be input as shown below (the order of appearance in the deck is unimportant).



4.1.3 Card Type 2 - Module Section

Data entered on Type 2 cards determines which modules will be executed during a simulation. Moreover, as previously mentioned, the order of appearance in the deck of the Type 2 cards determines the order of module execution. Therefore, the cards must be input in the order as illustrated on the next page. The Winds module (G2) card can be omitted if a run is to be made with no winds. Two autopilot modules, one (C1) including high frequency components, and one (C2) modeling only low (less than 6 cps) frequency components, are available. C1 is shown on the following page, but can be replaced by C2 if the low frequency model is desired.

The module name field is used for data set print identification only. Any name, comment, etc., can be entered in this field. The module number must be the number (shown) recognized by the program.

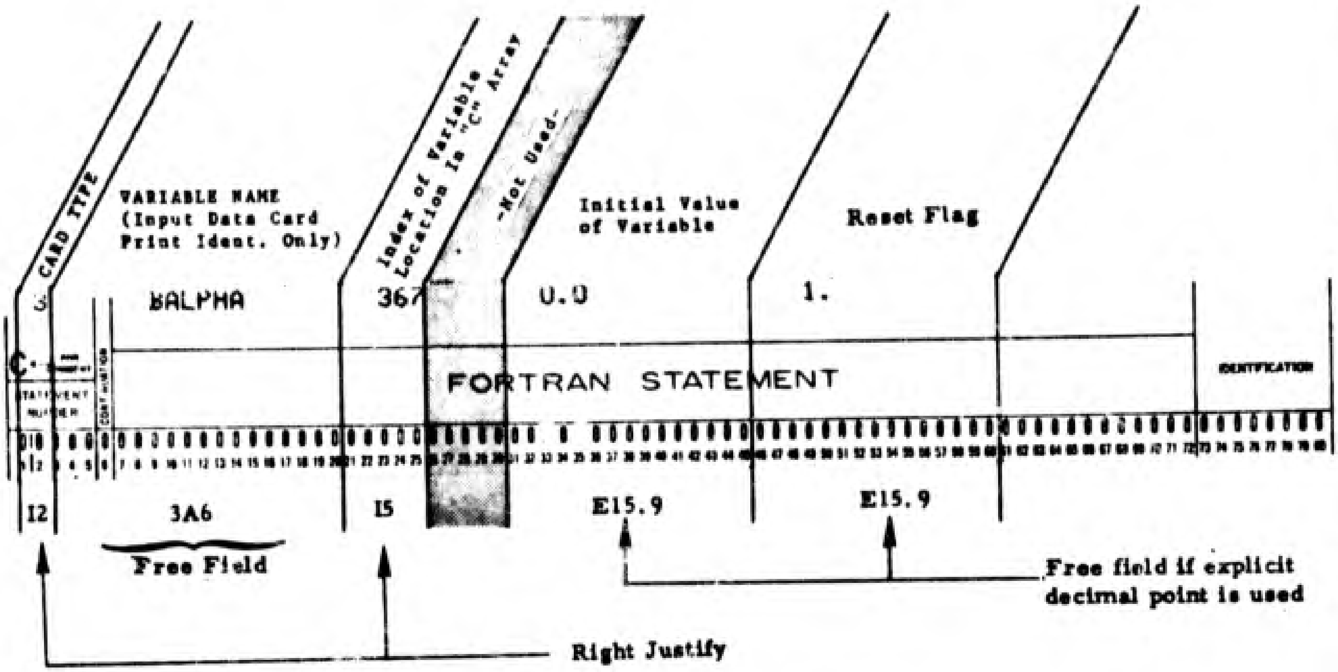


4.1.4 Card Type 3 - Program Parameter Input Values

The Type 3 cards specify initial parameter variable values. If the reset flag is set to 1, the program will reset the variable to the input value at the end of the case corresponding to the data set. Any subsequent cases, therefore, will use the same initial variable value, until a data set is encountered which changes the initial value. If multiple cases are being run, each variable that changes its value during the simulation should either have the reset flag set, or have an initial value defined for each case; otherwise, cases after the first will be initiated with the variable equal to its terminal value in the preceding case.

Initial values from Type 3 cards are read directly into a single array, using the number entered in columns 21 through 25 as the array subscript. Therefore, the index value must be right justified, and represent the correct array index. Table 4.1.2 shows the array index for each input variable. The name field is for print identification only. Because the initial value data on Type 3 cards is read directly into the location in the variable array specified on the card, two or more Type 3 cards with the same index entry will result in all but the last input value being obliterated. The last encountered value will be the one used for the parameter initial value.

A sample Type 3 card placing an initial value of zero into the program variable array at location 367 is shown below. The 1 in the Reset Flag field will cause the variable to be restored to an initial value of zero for subsequent cases.

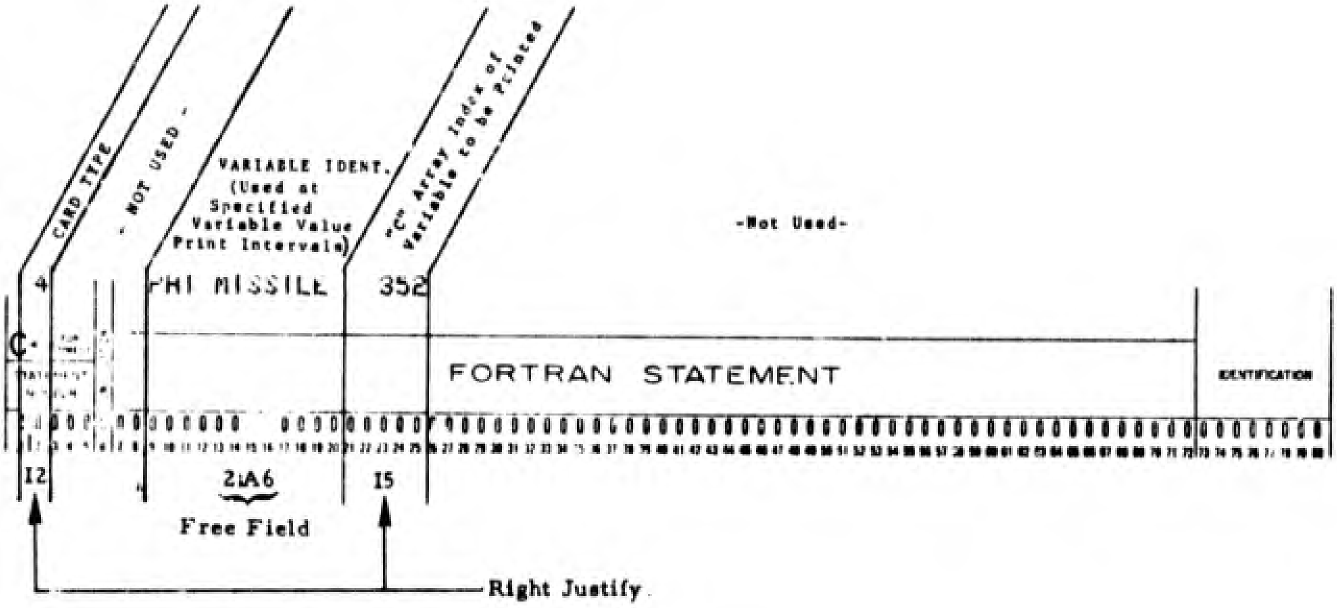


4.1.5 Card Type 4 - Print Variable Selection

The values of up to fifty variables can be printed at user specified simulation flight time intervals (CPP is the input print interval variable; see Table 4.1.2.). The array index of each variable to be printed is entered on a Type 4 card. In this case, the variable name is placed in columns 9 through 20 (rather than 3 through 20 as in Types 1, 2, and

3 cards), and this name appears in the printout to identify the variable. The print order will correspond to the order of the Type 4 cards in the data set.

An example of a Type 4 card is shown below, indicating that the value in the 352nd location of the variable array is to be printed, and is to be identified by the name "PHI MISSILE."



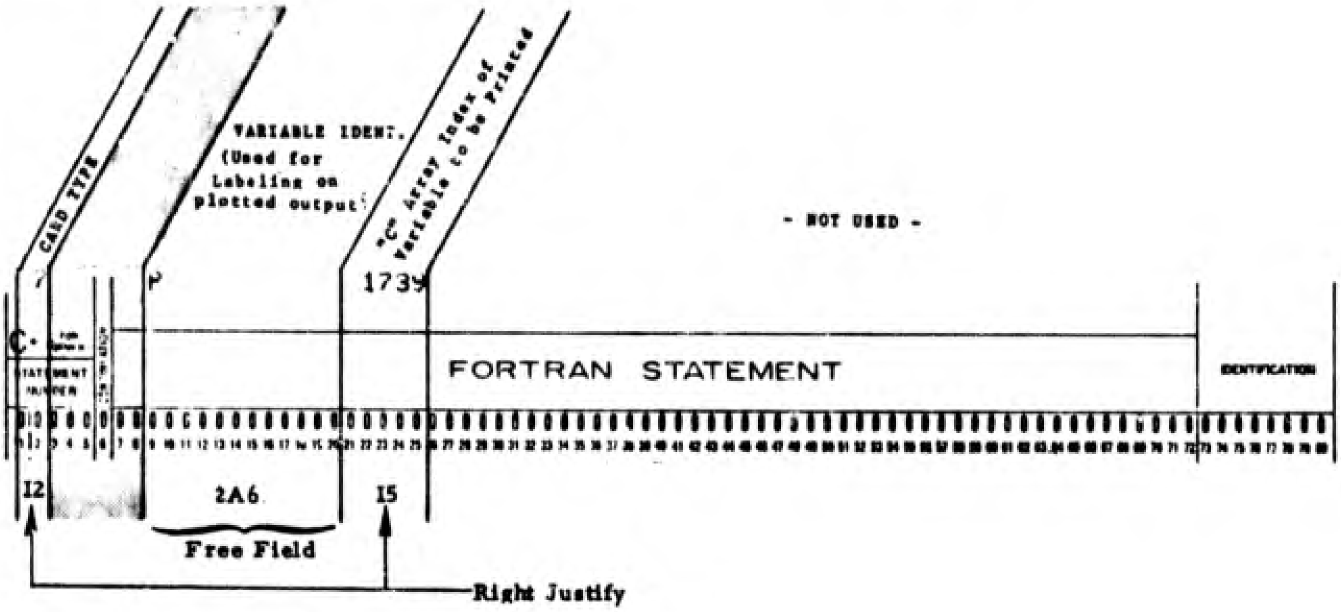
4.1.6 Card Type 7 - Plot Variable Selection

A plot tape with up to fifteen variables per case can be generated by the program. The variables to be plotted are entered on Type 7 cards in the same format as Type 4 cards. The plot sampling points will be taken every PPP seconds, until the end of the trajectory (up to a maximum of 300 sampling points), where PPP is an input variable (see Table 4. 1. 2).

The order of the Type 7 cards is dictated in part by the plot options selected. Any combination of the following three plot options can be selected: (1) missile and target trajectory (X-Z) plots; (2) paired (one versus the other) variable plots; (3) variables versus time plots (see Table 4. 1. 2 for option switch formats). The Type 7 cards must be input in the order corresponding to the three options, i. e. , if the trajectory plot option is selected, the Type 7 card for this option must be located in the deck ahead of Type 7 cards corresponding to the remaining two options. Type 7 cards corresponding to the paired plots must precede the variables versus time plots.

In addition, Type 7 cards for the trajectory plots must be input on four cards in the order $X_{missile}$, $Z_{missile}$, X_{target}, Z_{target}. Type 7 cards for the paired variable plots must have the abscissa variable first, followed by the ordinate variable. No internal order needs to be observed for the variable versus time plot cards.

An example of a Type 7 card is shown on the following page, where the variable to be plotted is at location 1739 of the variable array, and the plot variable label is "P". (The type of plot would be determined by the input option switches, and the location of the card with respect to the other Type 7 cards, as discussed in the previous paragraphs.)



4.1.7 Card Type 6 - Data Termination Indicator

This card signals the program that the end of a data set has been encountered. Include one at the end of each data set.

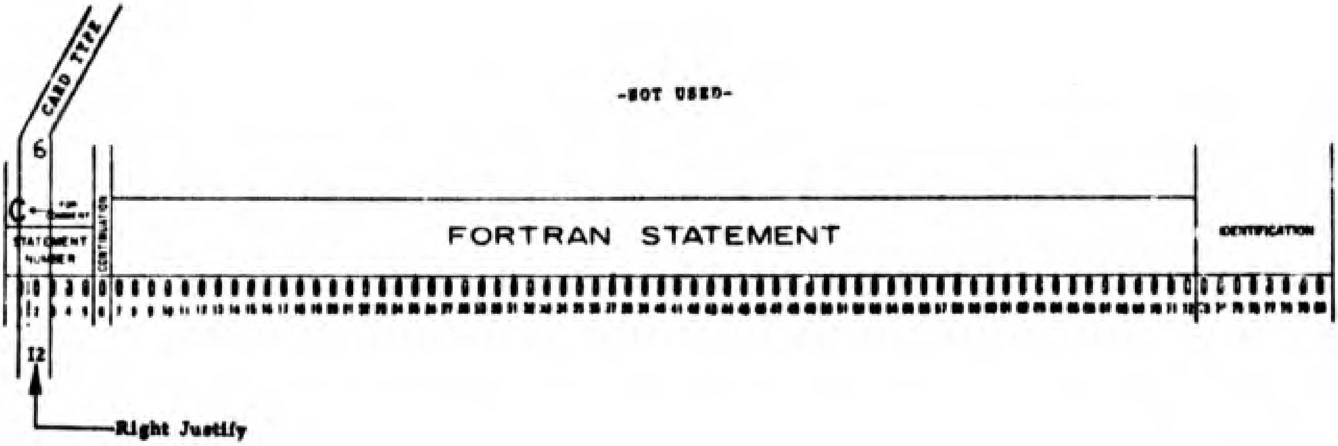


Table 4.1.2 - Input Data

FORTRAN SYMBOL	C INDEX	DEFINITION
		<u>INPUT TO EXECUTIVE ROUTINES</u>
		<u>Plot Control</u> (See Section 3.5.3)
PLOTN4	1982	Special plots of missile and target option switch 0 - no special plots 4 - provides data for missile to target plots
PLOTN2	1983	Number of variables in full page plot (multiple dependent variables to be plotted against one independent variable)
PLOTNO	2008	Total number of variables to be plotted (if PLOTNO = 0, all plot routines will be ignored) PLOTNO = L provides data output to tape plotting variables against time, where L is set such that the value of (PLOTNO-PLOTN2-PLOTN4) equals the number of variables to be plotted using this option (equivalently, PLOTNO must be set to the total number of variables to be plotted for all options, where each pair represented by PLOTN2 is counted as two variables).
PTLESS	2007	Number of last plotting points deleted
REPPLT	2006	Controls multi-case plot and print data input card read option 0 - Use new Type 4 and 7 data cards 1 - Use old plus new Type 4 and 7 cards -1 - Use new Type 7 cards
PPP	2005	Plotting interval (seconds) (number of points $\leq$ 300)

Table 4.1.2 Input Data (Cont.)

FORTRAN SYMBOL	C INDEX	DEFINITION
		Print Control
RITE	1971	Option switch for step size halving printout (>0=Print when integrated value exceeds upper limit set in A5I)
CPP	2015	Printing interval (SEC)
DOC	2013	Allows dumping of C-Array for the first (6-DOC) integration intervals (normally input 6.)
OPTN10	2022	C-Array dump option at end of run. 0 - no C-Array dump 1 - dump C-Array at end of run
		Integration Parameters
RKUTTA	1972	Runge-Kutta integration switch 0 - Adams Moulton with Runge Kutta start 1 - Stay in Runge Kutta
T	2000	Initial value of flight time (SEC)
TF	2001	Maximum allowable flight time (SEC)
DER(1)	2664	Initial value of integration step (SEC)*
HMIN	2662	Minimum value of integration step (SEC)*
HMAX	2663	Maximum value of integration step (SEC)*
		Multiple Run Control
STEP	2010	Flag that indicates the nature of the next case to be run. STEP may take on any one of eleven values. The input value of STEP determines the point in MAIN at which the next case will be start execution. Normal input values are: 2 - normal run follows or 11 - terminate case (The entry point in MAIN based on the value of STEP is indicated in Figure 4.1.2 for all values of STEP from 1 to 11.)

---

*If the low frequency model (Module C2) is called, no inputs will be accepted for HMIN (C(2662)), HMAX (C(2663)) or DER(1) (C(2664)), because a value of .005 seconds is hardwired into C2I for each of the variables.

Table 4. 1. 2 Input Data (Cont.)

FORTRAN SYMBOL	C INDEX	DEFINITION
		<u>Constants for Simulation</u>
AGRAV	1627	Gravity constant (Ft/Sec^2)
RHZRO	1665	Distance of Earth fixed coordinate above sea level (FT)
CRAD	1751	Radians to degrees conversion factor (57.295778 hard-wired in module D1I)
		<u>MISSILE WEIGHT, ENGINE AND AIRFRAME PARAMETERS</u>
CISP	1414	Specific impulse (Seconds)
DWT	1415	Missile total weight less propellant (LBS)
DWP	1416	Missile propellant weight (LBS)
RDCGO	1417	Launch value of C. G. shift (FT)
RDCGF	1418	Burnout value of C. G. shift (FT) (C. G. shift occurs only along X-axis)
FMIXF	1419	Burnout value of missile moment of inertia about roll axis (Slugs - FT^2)
FMIYF	1420	Burnout value of missile moment of inertia about pitch axis (Slugs - FT^2)
FMIZF		(The program automatically sets FMIZF=FMIYF)
RLCGO	1421	Distance between launch C. G. and rear lug (FT)
THRUST		Thrust table built into MODULE A3
RFAREA	1306	Aerodynamic cross sectional reference area (FT^2)
RFLGTH	1307	Aerodynamic chord reference length (FT)
RLUG	1316	Distance between lugs (FT)
BDP	1231	Initial value of combined fin setting for roll control (DEG)
BDQ	1232	Initial value of combined fin setting for pitch control (DEG)
BDR	1233	Initial value of combined fin setting for yaw control (DEG)
		<u>MISSILE AND TARGET INITIAL POSITION AND VELOCITY VALUES AND MISSILE INITIAL ATTITUDE SETTING</u>
		(Input of the following parameters is governed by the two input options, OPTN2 and OPTN4; see Table 4. 1. 3)

Table 4. 1. 2 - Input Data (Cont.)

FORTRAN SYMBOL	C INDEX	DEFINITION
RXE	1615	Initial components of missile in Earth fixed coordinate system (FT)
RYE	1619	
RZE	1623	
VXE	1603	Initial velocity components of missile in Earth fixed coordinate system (FT/SEC)
VYE	1607	
VZE	1611	
BPSIO	1754	Initial body yaw Euler angle (DEG)
BTHTO	1753	Initial body pitch Euler angle (DEG)
BPHIO	1752	Initial body roll Euler angle (DEG)
BTHTG	427	Initial seeker pitch gimbal angle (DEG)
BPSIG	431	Initial seeker yaw gimbal angle (DEG)
RTXE	1651	Target position in Earth fixed coordinate system (normally zero) (FT)
RTYE	1655	
RTZE	1659	
RSLANT	1667	Initial magnitude of the LOS vector (Slant range)(FT)
BSLOV	1666	Initial value of the LOS vector angle (measured in a vertical plane from missile local horizontal to the LOS vector (DEG).
VWXE	100	Initial velocity components of a steady wind in the Earth Coordinate System (FT/SEC)
VWYE	101	
VWZE	102	
VMACH	204	Initial value of mach number
VMWTE	1674	Initial velocity of missile relative to the wind
		<u>LAUNCHER PARAMETERS (INPUT ONLY IF OPTN4>0)</u>
RAIL	1317	Rail length (between rear of front lug and end of rail)(FT)
AGV	1330	Amplitude of the oscillatory acceleration acting on the missile due to the launcher vibrating (g's).
CFREQ	1331	Frequency of the oscillatory acceleration acting on the missile due to the launcher vibration (CPS).
CPHASE	1332	Phase relationship at time=0 of the oscillatory acceleration action on the missile due to the launcher vibrating. (DEG)
AGV2	1333	Magnitude of the linear component of acceleration acting on the missile due to the launcher (g's).

Table 4.1.2 - Input Data (Cont)

FORTRAN SYMBOL	C INDEX	DEFINITION
		OPTIONS
OPNW	50	Wind option selector 0 = no winds (no input required) 1 = steady winds (see Table 4.1.2.1) for input data requirements
OPTNSK	455	Seeker option 0 = quadrant tracker 1 = vidicon tracker
OPTACT	1140	Actuator model option selector 0 = low frequency model 1 = high frequency model
QNALGN	1403	Thrust misalignment option selector 0 = no thrust misalignment (no inputs required) 1 = compute thrust misalignment forces (see Table 4.1.2.2 for input data requirements)
QBURN	1405	Engine burnout switch 0 = engine burn 1 = initialize trajectory after engine burnout (see Table 4.1.2.3 for additional inputs)
OPTM	1551	Guidance option 0 - normal guidance and flight mode 1 = fixes fin settings for duration of trajectory based on input values (see Table 4.1.2.4 for input data requirements)
OPTARG	1639	Target motion option 0 = no target motion (no additional input required) 1 = integrate target motion to update target position (see Table 4.1.2.5 for input data requirements)
OPTN2	3502	Initial position and velocity input option (see Table 4.1.3) 0 = Input position and velocity of missile and input position of target. Input Euler angles of missile and seeker gimbal angles. Input missile initial velocity relative to the wind (VMWTE or VMACH) and wind velocity components (if wind option OPTNW=1). 1 = Input RSLANT and BSLOV. Initial position components (RXE, RZE) are computed from RSLANT and BSLOV. RYE automatically set

Table 4.1.2 - Input Data (Cont.)

FORTRAN SYMBOL	C INDEX	DEFINITION
		to zero. Initial velocity components (VXE, VYE, VZE) computed from the wind velocity (VWXE, VWYE, VWZE which must be input) and missile velocity relative to the wind (VMACH or VMWTE, which also must be input). Target position automatically set to zero. 2 = Input initial position components RXE and RZE (RYE automatically set to zero). RSLANT computed from RXE and RZE. Initial velocity components (VXE, VYE, VZE) computed from wind velocity and missile velocity relative to the wind (VMACH or VMVTE, which also must be input.) Initial target position automatically set to zero.
OPTN3	3503	Roll rate option selector 0 = compute roll rate from moments about roll axis. 1 = maintain roll rate at zero or input constant
OPTN4	3504	Rail launch dynamics and fire selector option (see Table 4.1.3 for input data requirements) 0 = No rail dynamics computed; direct fire. Input initial seeker gimbal angles (BTHTG, BPSIG). The initial body Euler angles (BPSIO, BTHTO). are computed from the seeker gimbal angles under the constraint that the X-gimbal axis points along the LOS vector. (BPHIO is automatically set to zero). 1 = Rail dynamics computed; direct fire. Input initial body Euler angles (BPSIO, BTHTO). (BPHIO automatically set to zero). The initial seeker gimbal angles (BTHTG, BPSIG) are computed from the initial body Euler angles under constraint that the X-gimbal axis points along the LOS vector. (BPHIO is automatically set to zero). 2 = Rail dynamics computed; indirect fire. Input initial body Euler angles (BPSIO, BTHTO). (BPHIO automatically set to zero). The seeker is automatically caged at the input initial gimbal angle values. (BPSIG, BTHTG) until target acquisition.

Table 4.1.2 - Input Data (Cont.)

FORTRAN SYMBOL	C INDEX	DEFINITION
OPTN6	3506	Missile initial velocity (relative to wind) option 0 = input missile initial mach number (VMACH, C(204)). 1 = input missile initial velocity relative to the wind (VMWTE, C(1674))
		<u>SEEKER PARAMETERS</u>
BTHTG	427	Initial value of pitch gimbal angle (input if OPTN4≠1)
BPSIG	431	Initial value of yaw gimbal angle (input if OPTN4≠1)
DT	446	Laser designator pulse repetition time (SEC)
SWP	452	Spring restraining torque constant
OPTNSK	455	Vidicon/Quadrant tracker option switch
		0 -Quadrant Tracker (The following variables used only with Quadrant tracker)
RLOCK	445	Maximum quadrant tracker acquisition range (FT)
BDB	447	Tracker deadband at 10 kilometers (DEG)
CFOVZ	448	Total seeker field-of-view in pitch (DEG)
CFOVY	449	Total seeker field-of-view in yaw (DEG)
GSX	450	Seeker quadrant tracker error signal acquisition gain
SEPS	451	Integrator drift rate
GS	456	Quadrant tracker error signal gain
WSL	457	Zero location of integrator lead compensation
WSN	458	Pole location of additional lead-lag compensation
WL2	459	Zero location of additional lead-lag compensation
		1-Vidicon Tracker (The following variables used only with Vidicon tracker)
RBK	453	Vidicon Tracker breaklock range (image saturation range)(FT).
GEO	454	Vidicon tracker error signal gain
		<u>AUTOPILOT PARAMETERS</u>
TDY	860	Minimum time before trajectory pitch program will be switched to terminal homing guidance
GBIAS	861	Gravity bias program
GN	862	Seeker signal shaping gain

Table 4. 1. 2 - Input Data (Cont.)

FORTRAN SYMBOL	C INDEX	DEFINITION
WN2	863	Second order pole location for seeker signal shaping filter
WN1	864	First order pole location for shaping filter
WL	865	Zero location for seeker signal shaping filter
WLXX1	866	First zero in roll stabilization filter
WLXX2	867	Second zero in roll stabilization filter
WLJK1	868	First zero in pitch-yaw stabilization filter
WLJK2	869	Second zero in pitch-yaw stabilization filter
HJK	870	Pitch-yaw signal limiter
WXX	871	Natural frequency of roll stabilization filter
DXX	872	Damping ratio of roll stabilization filter
WJK	873	Natural frequency of pitch-yaw stabilization filter
DKJ	874	Damping ratio of pitch-yaw stabilization filter
GXX	875	Roll compensation gain
GJK	876	Pitch-yaw compensation gain
QBIAS	878	Pitch trajectory shaping bias
RBIAS	879	Yaw trajectory shaping bias
HXX	890	Roll signal limiter
		<u>ACTUATOR PARAMETERS</u>
OPTACT	1140	Actuator model option selector 0 = low frequency model 1 = high frequency model
CR	1145	Actuator modulation gain
HDEL	1147	Fin Torque limit gain
W1	1148	Solenoid valve bandwidth
ZN	1149	Damping of fin position loop
G1	1151	Actuator forward loop gain
BH	1152	Deadband
WN	1153	Natural frequency of fin position loop
G2	1154	Fin aero load weighting factor
H1	1155	Positive fin torque limit
H2	1156	Negative fin torque limit

Table 4.1.2.1. Wind Option - OPTNW = 1

FORTRAN SYMBOL	C INDEX	DEFINITION
BPSIW	51	Wind direction: angle between wind velocity vector and negative X_E-axis (DEG)
VWTE	52	Wind speed with respect to the Earth (steady horizontal winds only) (FT/SEC)
RHW	53	Altitude above which all the wind speeds are zero (FT)
		NOTE: If wind option OPTNW=1 is input, the initial value of the components of the wind (VWXE, VWYE, VWZE) must be input also.

Table 4.1.2.2. Thrust Misalginment Option - QNALGN = 1

FORTRAN SYMBOL	C INDEX	DEFINITION
BALPHT	1401	Thrust offset angle (α_T) measured from a line parallel to the X_B-axis and the thrust (DEG).
BPHIT	1402	Thrust offset angle (Φ_T) measured from the plane parallel to the X_B - Z_B plane to the plane containing the thrust and the line parallel to the X_B-axis (DEG).
RFXCG	1313	Thrust offset along X_B-axis (FT)
RFYCG	1314	Thrust offset along Y_B-axis (FT)
RFZCG	1315	Thrust offset along Z_B-axis (FT)

Table 4.1.2.3 Engine Burnout Switch - QBURN=1

FORTRAN SYMBOL	C INDEX	DEFINITION
WP	1739	Initial roll rate about X_B-axis (DEG/SEC)
WQ	1743	Initial pitch rate about Y_B-axis (DEG/SEC)
WR	1747	Initial yaw rate about Z_B-axis (DEG/SEC)
BALPHA	367	Initial vertical angle of attack (α , DEG)
BALPHY	368	Initial sideslip angle of attack (β , DEG)
BPHIP	370	Initial aerodynamic roll angle (Φ', DEG)

Table 4.1.2.4 Guidance Option Switch - OPTM=1

FORTRAN SYMBOL	C INDEX	DEFINITION
UDL1	1555	Angular setting of fin #1, measured positive as a counterclockwise rotation (DEG).
UDL2	1556	Angular setting of fin #2, measured positive as a counterclockwise rotation (DEG).
UDL3	1557	Angular setting of fin #3, measured positive as a clockwise rotation (DEG).
UDL4	1558	Angular setting of fin #4, measured positive as a clockwise rotation (SEG).

Table 4.1.2.5 Target Motion Option - OPTARG = 1

FORTRAN SYMBOL	C INDEX	DEFINITION
ATHRST	1629	Magnitude of the target acceleration vector (g's)
ATURNT	1630	Scaling factor for the angular rotation rate of the target radius vector (g's)
BGAMT	1631	Elevation angle of target, measured from the horizontal plane to the target position vector (DEG)

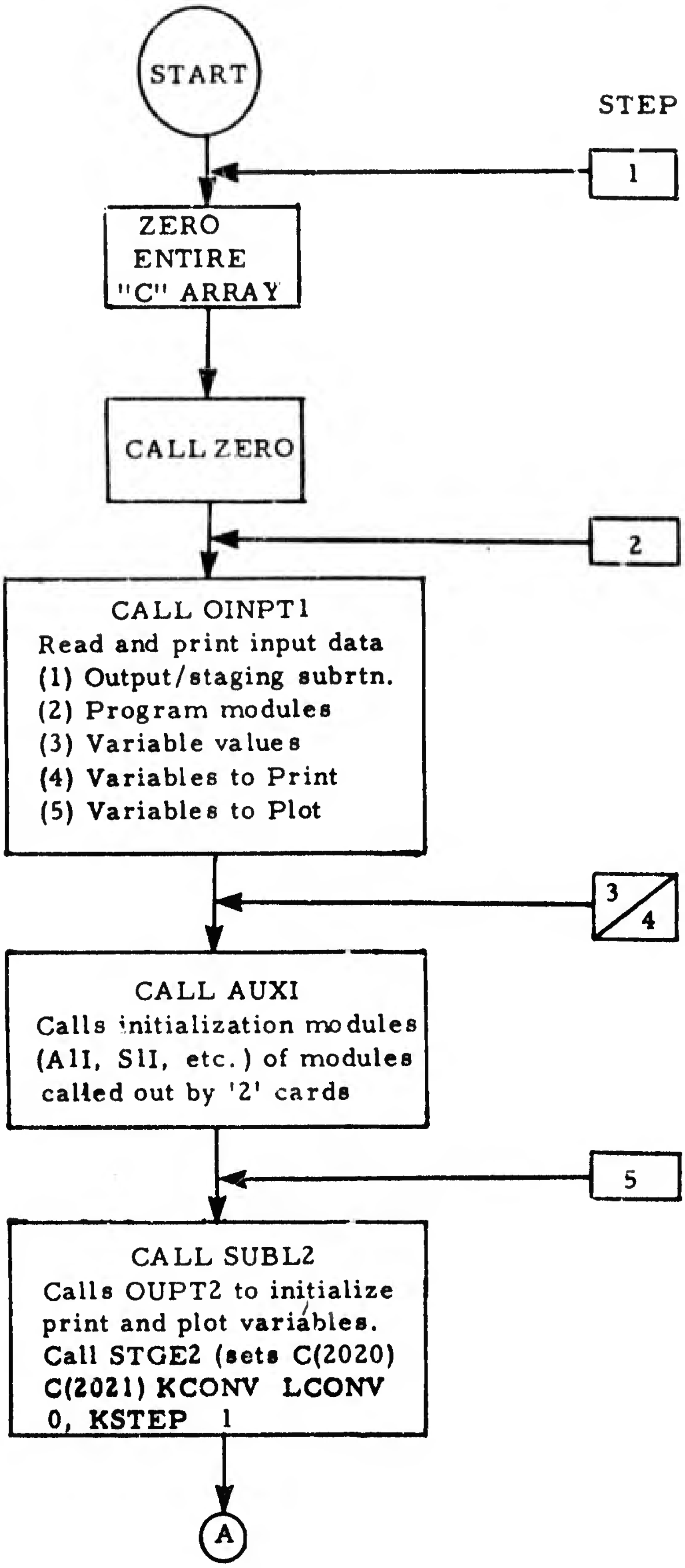


Figure 4.1.2 - Multiple Case Entry Point In MAIN -Step

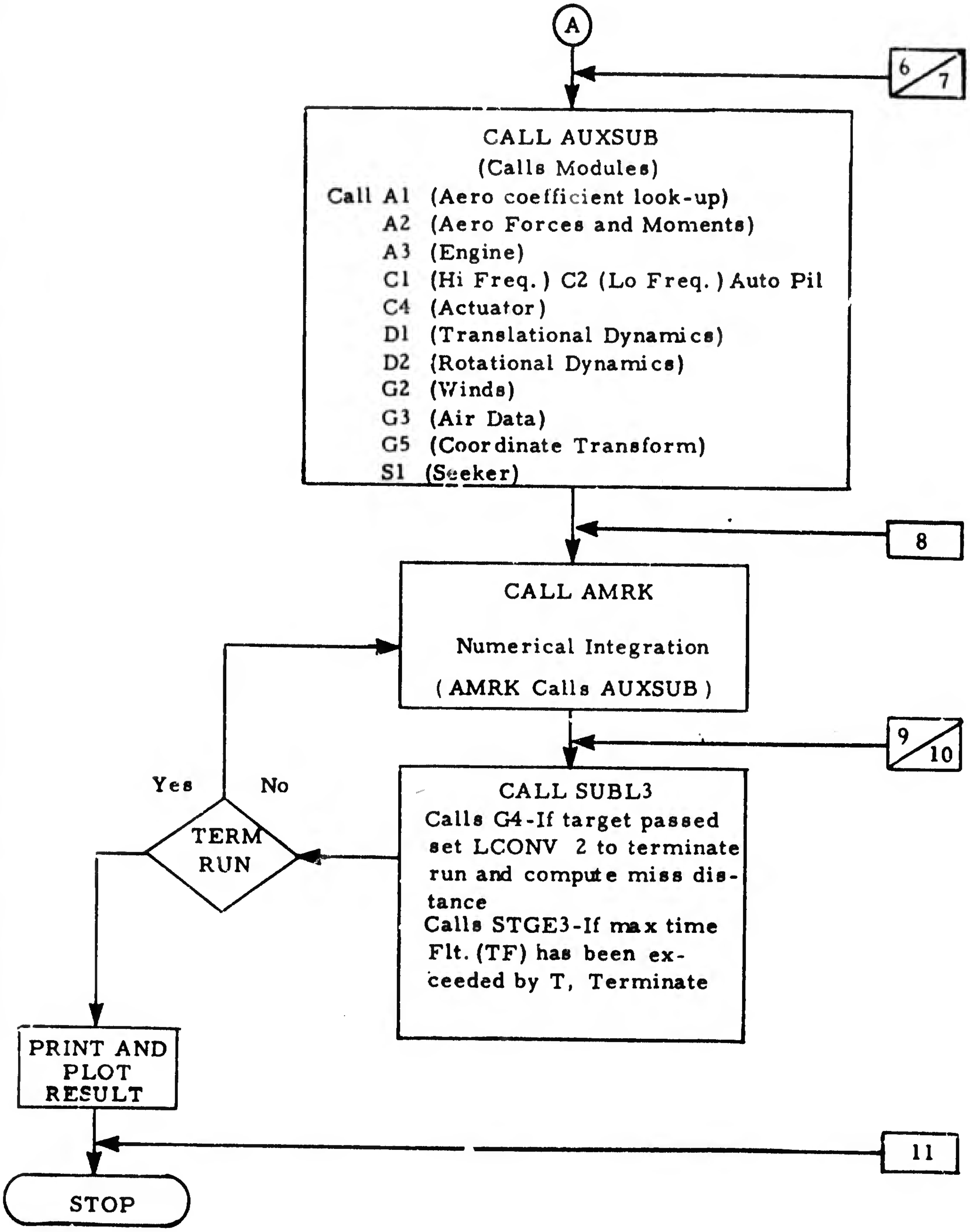


Figure 4.1.2 - Multiple Case Entry Point in MAIN-Step (Cont.)

Table 4.1.3 Input Data Options for Initial Values

	Unrestricted Inputs	Restricted Inputs					
		No Rail Dynamics		Rail Dynamics Included			
		Direct Fire		Direct Fire		Indirect Fire	
OPTN4	0	0	0	1	1	2	2
OPTN2	0	1	2	1	2	1	2
RXE	X	I	X	I	X	I	X
RYE	X	0	0	0	0	0	0
RZE	X	I	X	I	X	I	X
VXE	X	I	I	I	I	I	I
VYE	X	I	I	I	I	I	I
VZE	X	I	I	I	I	I	I
RTXE	X	0	0	0	0	0	0
RTYE	X	0	0	0	0	0	0
RTZE	X	0	0	0	0	0	0
BPSIO	X	I	I	X	X	X	X
BTHTO	X	I	I	X	X	X	X
BPHIO	X	0	0	0	0	0	0
BTHTG	X	X	X	I	I	X	X
BPSIG	X	X	X	I	I	X	X
RSLANT		X	I	X	I	X	I
BSLOV		X		X		X	
VWXE		X	X	X	X	X	X
VWYE		X	X	X	X	X	X
VWZE		X	X	X	X	X	X
(VMACH VMWTE)*		X	X	X	X	X	X

X - denotes variables that must be specified by input data

I - denotes variables that are computed internally

0 - denotes variables that are either set to zero initially or will be computed as zero.

---

* Input VMACH if OPTN6=0; input VMWTE if OPTN6=1.

4.2 OUTPUT

4.2.1 Introduction

The program provides several print and plot outputs. Some of the print outputs occur automatically. The remaining print outputs, and the plot outputs, are user options. A brief discussion and an example of each output is included below.

4.2.2 Automatic Printouts

- Input Conditions

The input data cards for each data set are printed out in card format at the end of the input phase for each case. An example, showing only one line for each type of input card, is included below:

```
    Input Data

1 OUPT 2, 3                 3   -0.                 -0.
2 G2-HE                    23   -0.                 -0.
3 CPP                    2015   -1.0000000E+00      1.0000000E+00
4      PHI MISSILE        352   -0.                 -0.
7      Y MSL             1619   -0.                 -0.
6                          -0   -0.                 -0.
```

• Front Lug Clears Rail

When the front lug clears the rail, a pitch down moment is present on the missile. Because of the potential significance of this transient, a printout occurs at this point, giving the time that the front lug clears the rail,' the velocity of the missile relative to the wind (airspeed), and the pitch moment.

```
FRONT LUG CLEARS RAIL     T =  8.79E-02     REL VEL =  1.82E+02     PITCH MOMENT = -5.63E+01
```

• Rear Lug Clears Rail

A printout, occurring at the time the rear lug clears the rail contains the time, the velocity of the missile relative to the wind (airspeed), and the rail lug force.

```
REAR LUG CLEARS RAIL    T =    .1125REL VEL =  203.079   RAIL FORCE =      0.00
```

• Engine Burnout Time

```
BURNOUT TIME=  3.0000 SEC.
```

• End Conditions

The three programmed simulation terminations are: (1) ground impacted with missile to target range greater than 500 feet; (2) input value of maximum flight time exceeded; and (3) target plane intercepted.

When simulation termination occurs due to conditions (1) and (2), no special end condition printouts are generated.

When simulation termination is due to target plane intercept, the program prints the following information:

- Miss distance (measured in the intercept plane)
- Flight time at intercept
- Components of LOS vector at intercept
- Y and Z components of miss distance (in the intercept plane)
- Values of variables requested on print control (Type 4) cards.

A sample printout is shown below:

```
MISS DISTANCE =   1.6816804E+00

FLIGHT  TIME  =   3.2203602E+00

       ROELX =  -8.5689802E-02       ROELY =  -6.2013629E-01       ROELZ =   1.5602482E+00

                                     RYFP  =  -6.2182298E-01       RZFP  =   1.5624933E+00


   3.2250000      2.1777081E-01    -4.3991287E+00    -1.8405179E-01     1.2313047E+03     5.1850654E+00
                 -1.5153321E+01     1.2347848E+00    -4.5042028E+00    -1.8437267E+01     5.4527014E-01
                 -1.7223131E+01     2.1573598E+00    -6.0323832E+00     5.0956807E+01    -1.3088714E+00
                  1.1035854E+00    -1.6423920E+03     3.4301747E+00     5.3754411E+00     3.2724875E+00
                  1.2330304E+03     5.1758421E+03     1.8369247E+03     0.               -4.9159452E+00
                  4.5043061E+00     6.0138828E+00    -8.5781373E-01    -2.8336426E+00    -6.5956854E-01
                  1.6305325E+02    -1.6267243E+02    -2.5947918E+00     1.2652449E+00    -2.0521672E+01
                  1.7283945E+02    -1.7289714E+02    -3.1802085E+00     3.1809931E+00    -2.2149839E+01
                  2.9643264E+03    -1.1483450E+03     3.1743738E+00    -7.8490428E-01     3.6099194E+00
                  0.                0.                0.                0.
```

4.2.3 Optional Printouts

Periodic printouts of variables requested by input (on Type 4 cards) occur every CPP seconds, where CPP is an input parameter (see Table 4.1.2). The variable names input on Type 4 cards appear only once, followed by the instantaneous values of the variables. Subsequent interval printouts include only the variable values. A sample is shown in Figure 4.2.1.

"The values of every location in the C array will be printed for every integration step, for a maximum of (6-DOC) integration steps, where DOC is an input variable. The printouts are unlabeled, containing only the C array values. The identification of the variables associated with each element of the C array is given in Table 4.2.1. Because the printout includes only numerical values, no sample is shown.

4.2.4 Plots

A plot tape for any combination of three types of plots will be generated with appropriate settings of the input plot option switches (discussed in Section 3.5.3). The plots are generated from the plot tape on a Stromberg Carlson 4020 Recorder. The three types of plots are: 1) missile and target trajectory (X-Z) plots (Figure 4.2.2); 2) paired (one variable vs. another) plots (Figure 4.2.3); and 3) time plots (Figure 4.2.4).

```
TIME        ROLL FLAP        P                PHI MISSILE      VX EARTH         X MSL

            Q                PITCH FLAP       THETA MISSLE     VY EARTH         Y MSL

            R                YAW FLAP         PSI MISSILE      VZ EARTH         Z MSL

            MACH             ALPHA DEG        ALPHA PRIME      RANGE            ACCEL Z

            AIR SPEED        BETA DEG         POYNMC           FTHRST           ACCEL Y

            BTH              BPS              GAMMAH           GAMMAV           BXX

            THETA LOS-V      EPS Z            WLAMQ            THETA GIMBAL     Q COM

            PSI LOS-V        EPS Y            WLAMR            PSI GIMBAL       R COM

            RANGO            FIN1             FIN2             FIN3             FIN4

            VTXE             RTXE             RTYE             RTZE


                                                               1.0507241E+02    -3.2794873E+03
 .0050000   0.               0.               0.               0.                0.
            0.              -4.8705361E-02    0.               5.8738201E-15    -1.0000000E+02
            0.              -1.3909445E-03    0.               3.2810116E+03     4.2402003E-14
            9.4074361E-02    0.               0.               2.2780000E+03     0.
            1.0507241E+02    0.               1.3078175E+01   -3.2029827E-15    -0.
            0.               0.               0.              -1.7606440E+00     9.2219160E-01
           -1.7465545E+00    1.4089522E-02   -1.0250000E+00    1.4362413E-02     1.0553086E-01
            0.              -1.4355742E-02    1.0250000E+00   -4.7314416E-02    -5.0096305E-02
            1.2699720E-02   -4.7314416E-02   -5.0096305E-02    0.
            0.               0.               0.
TIME=     .0775000  STEP SIZE=     2.5000000E-03
TIME=     .0825000  STEP SIZE=     5.0000000E-03

   FRONT LUG CLEARS RAIL      T =  8.75E-02      REL VEL =  1.82E+02      PITCH MOMENT = -5.63E+01
TIME=     .0950000  STEP SIZE=     2.5000000E-03
TIME=     .1025000  STEP SIZE=     5.0000000E-03

      REAR LUG CLEARS RAIL     T =     .1125REL VEL =  203.079   RAIL FORCE =     0.00
TIME=     .1200000  STEP SIZE=     2.5000000E-03
TIME=     .1275000  STEP SIZE=     5.0000000E-03
TIME=     .1800000  STEP SIZE=     2.5000000E-03
TIME=     .1875000  STEP SIZE=     5.0000000E-03
TIME=     .2300000  STEP SIZE=     2.5000000E-03
TIME=     .2375000  STEP SIZE=     5.0000000E-03
TIME=     .2800000  STEP SIZE=     2.5000000E-03
TIME=     .2850000  STEP SIZE=     5.0000000E-03
TIME=     .3775000  STEP SIZE=     2.5000000E-03
TIME=     .3825000  STEP SIZE=     5.0000000E-03
TIME=     .4800000  STEP SIZE=     2.5000000E-03
TIME=     .4850000  STEP SIZE=     5.0000000E-03
TIME=     .5275000  STEP SIZE=     2.5000000E-03
TIME=     .5325000  STEP SIZE=     5.0000000E-03
TIME=     .5800000  STEP SIZE=     2.5000000E-03
TIME=     .5850000  STEP SIZE=     5.0000000E-03
TIME=     .5775000  STEP SIZE=     2.5000000E-03
TIME=     .5325000  STEP SIZE=     5.0000000E-03
TIME=     .5500000  STEP SIZE=     2.5000000E-03
TIME=     .5850000  STEP SIZE=     5.0000000E-03
TIME=     .7775000  STEP SIZE=     2.5000000E-03
TIME=     .7825000  STEP SIZE=     5.0000000E-03
TIME=     .8850000  STEP SIZE=     2.5000000E-03
TIME=     .8875000  STEP SIZE=     5.0000000E-03
```

Figure 4.2.1 - Sample Printout

```
TIME=       .9325000  STEP SIZE=       2.5000000E-03
TIME=       .9375000  STEP SIZE=       5.0000000E-03
TIME=       .9850000  STEP SIZE=       2.5000000E-03
TIME=       .9900000  STEP SIZE=       5.0000000E-03


    1.0000000        2.1906934E-01     -6.0164208E+00     -5.2660236E-01      8.7603204E+02     -2.7826206E+03
                     4.3042423E-01      1.8166337E+00      1.4266825E+00     -2.1948404E+00     -4.0572681E-01
                     3.7035458E+00      6.8351735E-01      5.6289970E-01      1.1278474E+01     -8.8651103E+01
                     7.8437301E-01      2.1577707E+00      2.2768733E+00      2.7840324E+03      1.1296696E-01
                     8.7610739E+02     -7.2676455E-01      9.0939453E+02      1.5333333E+03      4.0356639E-01
                    -1.4267428E+00     -5.6272520E-01     -1.4355049E-01     -7.3761145E-01     -5.4061775E-01
                    -1.0871466E+00     -8.0529636E-02     -9.6354166E-01     -3.1656074E+00      2.2166971E+00
                    -1.5191940E-01      3.0312655E-02      1.0135417E+00     -5.1540926E-01     -3.2669614E-01
                     3.9754169E+02      9.1404701E-01      2.2810817E+00      1.3521857E+00      2.7192204E+00
                     0.                 0.                 0.                 0.


    2.0000000        2.7043258E-02      8.0364217E-02     -8.8581618E-02      1.3524916E+03     -1.5968591E+03
                    -1.6461121E+00     -1.2936814E-01     -2.5709236E+00     -2.9403798E+00      7.6958225E-01
                    -2.1781136E+00     -1.5060671E-01     -4.2649906E-01      4.2855938E+01     -6.7545178E+01
                     1.2113972E+00     -7.5552310E-01      8.1559184E-01      1.5902872E+03      1.8472711E+00
                     1.3531734E+03      3.0720497E-01      2.1714210E+03      3.0000000E+02     -5.2992743E-01
                     2.5709267E+00      4.2606977E-01     -1.2033558E-01     -1.8149005E+00     -1.0771207E-01
                    -6.0713413E-01      7.9349589E-02      4.8953244E-02      6.8795349E-02     -5.9932232E+00
                    -9.2677946E-02     -1.3749344E-02     -9.9895834E-01      4.1250657E-01      1.5783985E+00
                     1.4834962E+03     -5.8076930E-03     -3.0702111E-01      4.0284822E-02     -2.5292860E-01
                     0.                 0.                 0.                 0.

BURNOUT TIME=  3.0000 SEC.

    3.0000000       -1.4220793E-01      1.3124161E-01      5.3944386E-01      1.2665175E+03     -2.7586837E+02
                    -1.3333453E+01     -6.1372097E-01     -1.0326083E+00      5.6962497E+00      1.6542302E+00
                    -1.9902097E+01      3.4530486E-01     -1.9865680E+00      3.1118628E+01     -1.3434481E+01
                     1.1345397E+00      1.2592611E+00      2.5627213E+00      2.7620025E+02     -4.6855183E-01
                     1.2673615E+03      2.2549304E+00      1.9087795E+03      0.                -2.1129642E+00
                     1.0325520E+00      1.9862453E+00      2.5768998E-01     -2.3112695E+00      5.0363027E-01
                    -4.7659102E-01     -6.8803184E-02     -1.4697917E+00     -1.6714719E+00     -1.0494157E+01
                     6.0078864E-01     -9.7345852E-02     -2.0052084E+00      1.7536741E+00     -1.9542275E+01
                     7.7053174E+03     -8.1681789E-01     -1.2620818E-01     -1.1012338E+00     -4.1062404E-01
                     0.                 0.                 0.                 0.

  MISS DISTANCE =    1.6816804E+00

  FLIGHT  TIME  =    3.2203602E+00

         RDELX =  -9.5689802E-02         RDELY =  -6.2013629E-01        RDELZ =    1.5602482E+00

                                         RYFP  =  -6.2182298E-01        RZFP  =    1.5624933E+00


    3.2250000        2.1777081E-01     -4.3991287E+00     -1.8405179E-01      1.2313847E+03      5.1850654E+00
                    -1.5153321E+01      1.2347848E+00     -4.5042820E+00     -1.8437267E+01      5.4527014E-01
                    -1.7223131E+01      2.1573598E+00     -6.032[illegible]E+00      5.0955807E+01     -1.3008714E+00
                     1.1033894E+00     -1.6423920E+01      5.6301743E+00      5.3794411E+00      3.2724875E+00
                     1.2330304E+03      5.1784621E+03      1.8369247E+03      0.                -4.9159452E+00
                     4.9043061E+00      6.0138828E+00     -8.5781373E-01     -2.8336626E+00     -5.5956854E-01
                     1.6393325E+02     -1.6287243E+02     -2.9947918E+00      1.2652549E+00     -2.0521672E+01
                     1.7203945E+02     -1.7285714E+02     -3.1302085E+00      5.1593961E+00     -2.2149339E+01
                     2.3643204E+03     -1.1483459E+03      3.1743738E+00     -7.0480420E-01      3.6699154E+00
                     0.                 0.                 0.                 0.
```

Figure 4.2.1 - Sample Printout (Cont.)

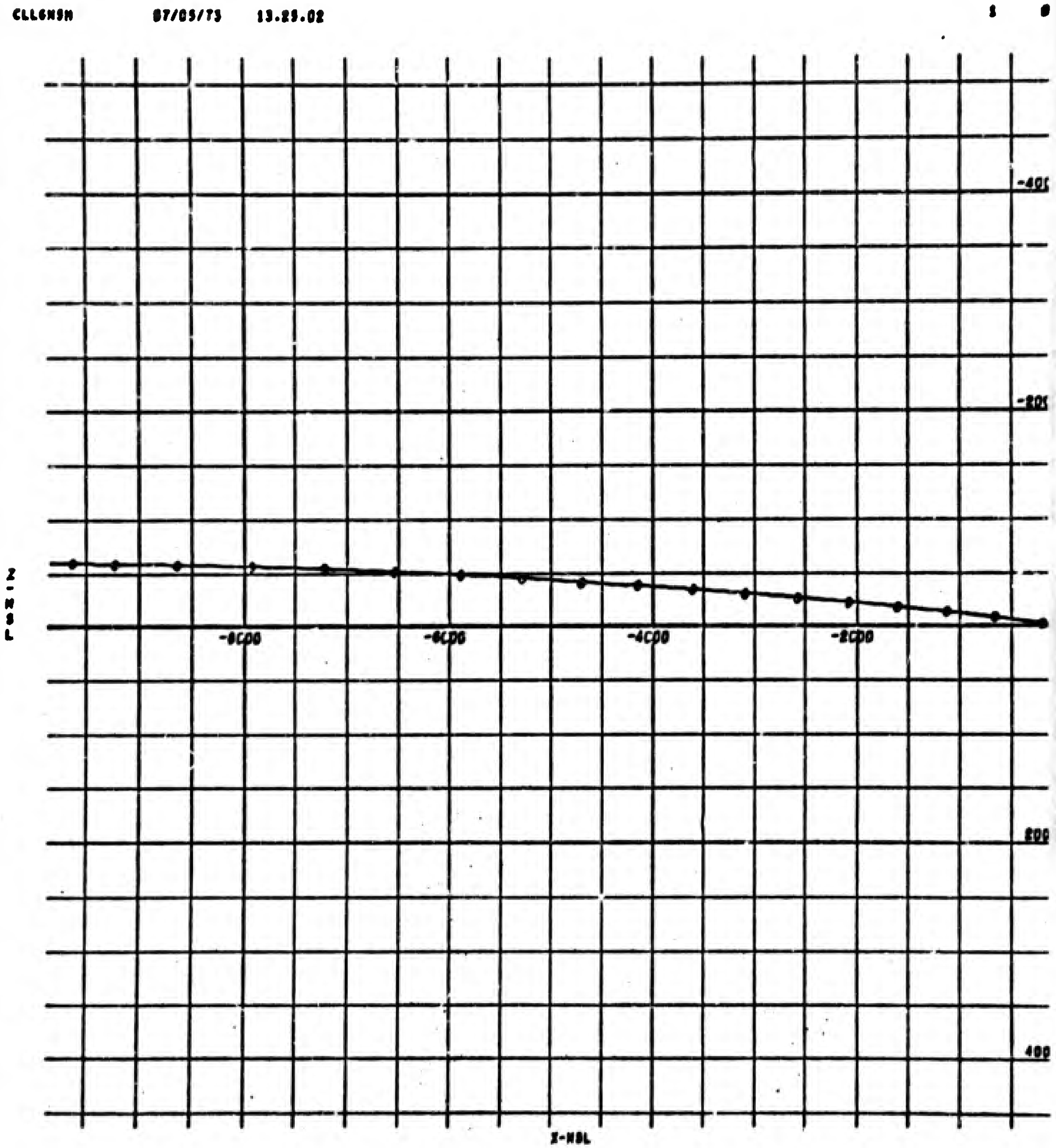


Figure 4. 2. 2 - Missile and Target Trajectory

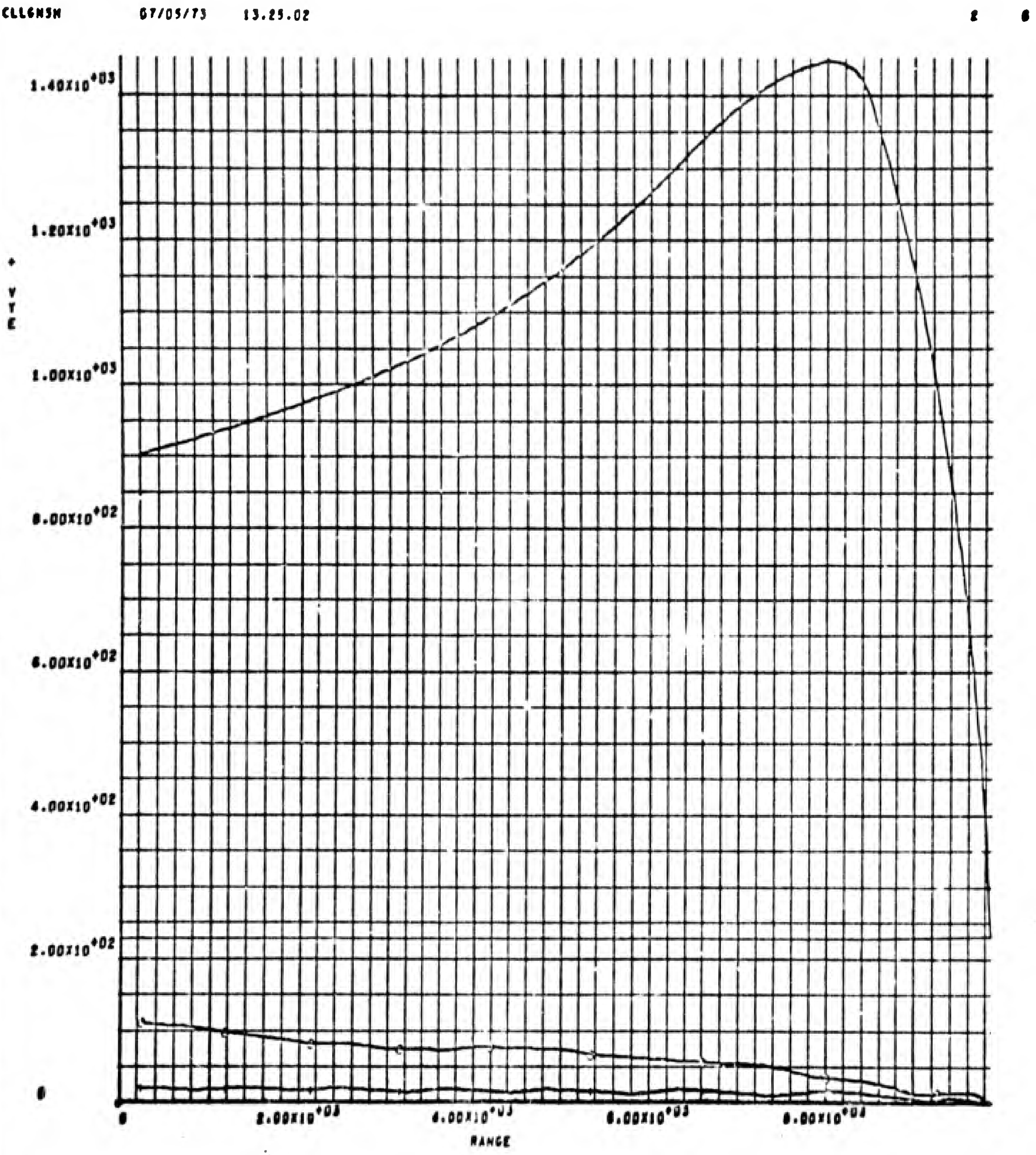


Figure 4.2.3 - Multiple Dependent Variables
vs. A Specified Dependent
Variable

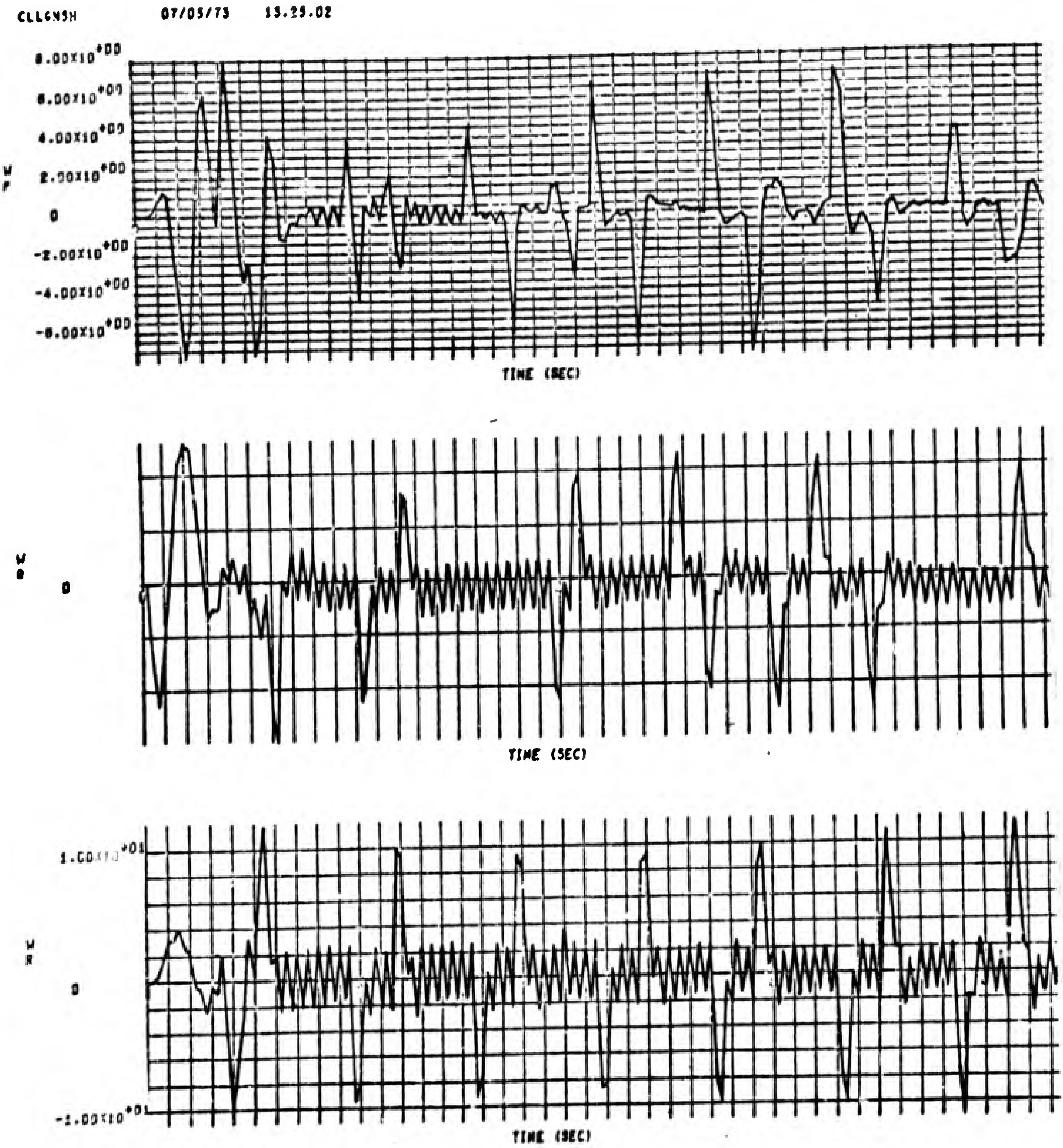


Figure 4.2.4 - Time Plots

Table 4.2.1

The following Table gives a list of all program variables with associated C Index numbers for the computer program.

Table 4.2.1

Module	Data Source	C Index	Fortran Name	Initial Value	Default Value	Definition
G2	Input	50	OPTNW	Input	0	Wind option switch 0-no winds, 1-steady horizontal winds
G2	Input	51	BPSIW	Input	0	Wind direction: angle between wind velocity and negative X_E-axis, (DEG)
G2	Input	52	VWTE	Input	0	Wind speed (FT/SEC)
G2	Input	53	RHW	Input	0	Altitude above which all wind velocity is zero (FT)
G2		54				Not Used
G2		↓				
G2		99				
G2	G2	100	VWXE	Input	0	Wind velocity components in the Earth fixed coordinate system.(FT/SEC)
G2	G2	101	VWYE	Input	0	
G2	G2	102	VWZE	Input	0	
G2		103				Not Used
G2		↓				
G2		199				
G3	G3	200	VMWXE			Components of missile velocity with respect to the wind in Earth fixed coordinate system (FT/SEC)
G3	G3	201	VMWYE			
G3	G3	202	VMWZE			
G3	G3	203	PDYNMC			Dynamic pressure (LB/FT^2)
G3	G3	204	VMACH	see def		Mach Number (Input if OPTN6=0)
G3	G3	205	DRHO			Air density (LB/FT^3)
G3	G3	206	VSOUND	D11		Speed of sound (FT/SEC) (Initial value computed in D11)
G3	G3	207	VAIRSP	D11		Magnitude of missile velocity with respect to the wind, (FT/SEC)
G3	Input	208	RHZRO	Input	0	Reference radius of Earth at sea level (FT)
G3	G3	209	RH			Missile Altitude (FT)
G3		210				Not Used
G3		↓				
G3		299				
G4	G4	300	RMISS			Miss distance (computed at point of closest approach) (FT)
G4	see def	301	RXF		0	Position components of the target in the flight plane coordinate system defined in Section 3.2.11 (FT) (RXF is not calculated (Hardwired value of 0)
G4	G4	302	RYF			
G4	G4	303	RZF			
G4		304				Not Used
G4		↓				
G4		349				
G5	G5	350	BTHT	C(1753)	0	Euler pitch angle in body system (DEG)
G5	G5	351	BPSI	C(1754)	0	Euler yaw angle in body system (DEG)
G5	G5	352	BPHI	0	0	Euler roll angle in body system (DEG)

Table 4. 2. 1 (Continued)

Module	Data Source	C Index	Fortran Name	Initial Value	Default Value	Definition
G5	G5	353	BPHI			Euler roll angle of inertial platform in body system (DEG)
G5	G5	354	BTH2			Euler pitch angle of inertial platform in body system (DEG)
G5	G5	355	BPSI			Euler yaw angle of inertial platform in body system (DEG)
G5	G5	356	VTOTE			Missile total velocity magnitude in Earth fixed system (FT/SEC)
G5	G5	357	BGAMH			Horizontal flight path angle (heading) measured positive clockwise from X_E-axis (DEG
G5	G5	358	BGAMV			Vertical flight path angle measured positive up from the local horizontal (DEG)
G5		359				Unused elements
G5		360				
G5		361				
G5		362				
G5	G5	363	BTHLV	0 (set in D1I)		Vertical proportional navigation angle, measured positive downward from the LOS vector to the projection of the missile velocity into the vertical plane that contains the LOS vector (DEG)
G5		364	BPSLV	0 (set in D1I)		Horizontal proportional navigation angle, measured positive clockwise from the LOS vector to the projection of the missile velocity into the plane containing the LOS vector and normal to the vertical plane (DEG)
G5		365	BLAMV			Vertical LOS angle, measured from the local horizontal to the LOS vector, positive up (DEG)
G5		366	BLAMH			Horizontal LOS angle, measured from the X_E'-axis to the projection of the LOS vector into the local horizontal plane (DEG)
G5		367	BALPHA	0 (set in A3I)		Vertical angle of attack, α, measured from the projection of the missile velocity with respect to the wind into the X_B-Z_B plane and the X_B axis (DEG)
G5		368	BALPHY	0 (set in A3I)		Horizontal angle of attack, β, measured from the projection of the missile velocity with respect to the wind into the X_B-Z_B plane and the velocity relative to the wind (DEG)
G5		369	BALPHP		0	Total angle of attack, α', (DEG)
G5		370	BPHIP	0 (set in A3I)		Aerodynamic roll angle, ϕ' (DEG)
G5		371	RANGE			Slant range (FT)
G5		372	RXBA			Components of the missile LOS in the body coordinate system (FT)
G5		373	RYBA			
G5	G5	374	RZBA			
G5		375				Unused elements
G5						
G5		379				

Table 4.2.1 (Continued)

Module	Data Source	C Index	Fortran Name	Initial Value	Default Value	Definition
G5	G5	380	RANGO			Magnitude of the separation distance from rear of launcher rail to the back of the rear lug (FT)
G5	↓	381				Not Used
G5		↓				
G5		389				
G5		390	RXL			Position components of the missile relative to the launcher in a system parallel to the Earth fixed coordinate system (FT)
G5		391	RYL			
G5		392	RZL			
G5		393	$\Phi 2$			Euler roll angle of the inertial platform (gyro 2) (DEG)
G5		394				
G5		395				
G5		396				Not Used
G5		↓				
G5	G5	399				
S1	S1	400				Not Used
S1	↓	401				
S1		402				
S1		403	WLAMQ			Seeker pitch error signal to autopilot
S1		404				Not Used
S1		405				
S1		406				
S1		407	WLAMR			Seeker yaw error signal to autopilot
S1		408	WLQD		0	Local variable derivative group in Seeker Module
S1		409				
S1		410				
S1		411	WLQ	see def		Local variable in Seeker Module (0 set initially in SII)
S1		412	WLRD		0	Local variable derivative group in Seeker Module
S1		413				
S1		414				
S1		415	WLR	see def		Local variable in Seeker Module (0 (Set in SII))
S1		416	WLQSD		0	Local variable derivative in Seeker Module
S1		417				
S1		418				
S1	S1	419	WLQS	see def		Local variable in Seeker Module (0 (Set in SII))

Table 4.2.1 (Continued)

Module	Data Source	C Index	Fortran Name	Initial Value	Default Value	Definition
S1	S1	420	WLRSD		0	Local variable derivative group in Seeker Module
S1		421				
S1		422				
S1		423	WLRS	see def		Local variable in Seeker Module (0 (Set in SII))
S1		424	BTHTGD			Time derivative group of pitch gimbal angle (DEG/SEC) (Input/Cal. in DII)
S1		425				
S1		426				
S1		427	BTHTG	see def	0/Cal.	
S1		428	BPSIGD			Time derivative group of yaw gimbal angle (DEG/SEC) (Input/Cal. in DII)
S1		429				
S1		430				
S1		431	BPSIG	see def	0/Cal.	
S1		432				Not Used
S1		433				
S1		434				
S1		435	BEPSZ		0	Local variable in Seeker Module
S1		436	BEPSY		0	Local variable in Seeker Module
S1		437	WZ		0	Local variable in Seeker Module
S1		438	WY		0	Local variable in Seeker Module
S1	S1	439	BGDEFL		0	Local variable in Seeker Module
S1		440				Not Used
S1						
S1		444				
S1	Input	445	RLOCK	Input	0	Maximum quadrant tracker acquisition range (FT)
S1	Input	446	DT		0	Laser Designator pulse repetition time (SEC)
S1	Input	447	BDB		0	Tracker deadband at 10 km (DEG)
S1	Input	448	CFOVZ		0	Total seeker field-of-view in pitch (DEG)
S1	Input	449	CFOVY		0	Total seeker field-of-view in yaw (DEG)
S1	Input	450	GSX		0	Seeker quadrant tracker error signal acquisition gain
S1	Input	451	SEPS		0	Integrator drift rate
S1	Input	452	SWP		0	Spring restraining torque constant
S1	Input	453	RBK		0	Vidicon tracker breaklock range (FT)
S1	Input	454	GEO		0	Vidicon tracker error signal gain
S1	Input	455	OPTNSK		0	Vidicon/Quadrant tracker option switch
S1	Input	456	GS		0	Quadrant tracker error signal gain
S1	Input	457	WSL		0	Quadrant tracker error signal gain term (affects error signal amplitude and response time)

Table 4. 2. 1 (Continued)

Module	Data Source	C Index	Fortran Name	Initial Value	Default Value	Definition
S1	Input	458	WSN	Input	0	Quadrant tracker gimbal torque signal gain term
S1	Input	459	SL2	Input	0	Quadrant tracker gimbal torque signal gain term
S1	Seeker	460	ST	See def		Local variable in Seeker Module (0 (set in S1I))
S1	Seeker	461	CAGE	See A		Seeker Gyro Cage (0) -Uncage (1) switch to autopilot
S1	Seeker	462	TKRZ	" "		Acquisition gain switch for pitch to autopilot
S1	Seeker	463	TKRY	" "		Acquisition gain switch for yaw to autopilot
S1	Seeker	464	TKRZY	" "		(Pitch and yaw gain switch coupling)
		465 ↓ 499				Not Used
		500 ↓ 799				Not Used
C1	Hi Freq Autopilot	800	WLQSDD	0	0	Local variable derivative group in High Frequency Autopilot
		801		-	-	
		802		-	-	
		803	WLQSP	0	0	
		804	WLQSD	0	0	
		805		-	-	Local variable derivative group in High Frequency Autopilot
		806		-	-	
		807	WLQS	0	0	
		808	WLQSSD	0	0	
		809		-	-	Local variable derivative group in High Frequency Autopilot
		810		-	-	
		811	WLQSS	C(861)+ C(878)	0	
		812	WLRSDD	0	0	Local variable derivative group in High Frequency Autopilot
		813		-	-	
		814		-	-	
		815	WLRSP	0	0	
		816	WLRSD	0	0	
		817		-	-	Local variable derivative group in High Frequency Autopilot
		818		-	-	
		819	WLRS	0	0	

A

If OPTN4 ≤ 1 (Set in S1I)	If OPTN4 > 1 (Set in S1I)
0	1
0	1
0	1
0	1

Table 4.2.1 (Continued)

Module	Data Source	C Index	Fortran Name	Initial Value	Default Value	Definition
C1	Hi Freq Autopilot	820	WLRSSD	0	0	Local variable derivative group in High Frequency Autopilot
		821		-	-	
		822		-	-	
		823	WLRSS	C(879)	0	
		824	BLQSSD	0	0	Local variable derivative group in High Frequency Autopilot
		825		-	-	
		826		-	-	
		827	BLQSS	0	0	
		828	BLRSSD	0	0	Local variable derivative group in High Frequency Autopilot
		829		-	-	
		830		-	-	
		831	BLRSS	-C(1233	0	
		832	BJJSDD	0	0	Local variable derivative group in High Frequency Autopilot
		833		-	-	
		834		-	-	
		835	BJJSP	0	0	
		836	BJJSD	0	0	Local variable derivative group in High Frequency Autopilot
		837		-	-	
		838		-	-	
		839	BJJS	C(831)	0	
		840	BKKSDD	0	0	Local variable derivative group in High Frequency Autopilot
		841		-	-	
		842		-	-	
		843	BKKSP	0	0	
		844	BKKSD	0	0	Local variable derivative group in High Frequency Autopilot
		845		-	-	
		846		-	-	
		847	BKKS	-C(831)	0	
		848	BXXSDD	0	0	Local variable derivative group in High Frequency Autopilot
		849		-	-	
		850		-	-	
		851	BXXSP	0	0	
		852	BXXSD	0	0	Local variable derivative group in High Frequency Autopilot
		853		-	-	
		854		-	-	
		855	BXXS	0	0	

Table 4. 2. 1 (Continued)

Module	Data Source	C Index	Fortran Name	Initial Value	Default Value	Definition
C1	Hi Freq Autopilot	856	BDELTC (1)	-	-	Autopilot generated fin commands to actuator for fins 1, 2, 3, and 4 (DEG)
		857	BDELTC (2)	-	-	
		858	BDELTC (3)	-	-	
		859	BDELTC (4)	-	-	
	Input	860	TDY	Input	0	Minimum time before trajectory pitch program will be switched to terminal homing guidance
		861	GBIAS		0	Gravity bias (DEG/SEC)
		862	GN		0	Seeker signal shaping gain
		363	WN2			Second order pole location for Seeker signal shaping filter
		864	WN1			First order pole location for shaping filter
		865	WL			Zero location for Seeker signal shaping filter
		866	WLXX1			First zero in roll stabilization filter
		867	WLXX2			Second zero in roll stabiliaztion filter
		868	WLJK1			First zero in ptich-yaw stabilization filter
		869	WLJK2			Second zero in pitch-yaw stabilization filter
		870	HJK			Pitch-yaw signal limiter
		871	WXX			Natural frequency of roll stabilization filter
		872	DXX			Damping ratio of roll stabilization filter
		873	WJK			Natural frequency of pitch-yaw stabilization filter
		874	DJK			Damping ratio of pitch-yaw stabilization filter
		875	GXX			Roll compensation gain
		876	GJK			Pitch-yaw compensation gain
		877	RES			Rate gyro term - Not Used
		878	QBIAS			Pitch trajectory shaping bias
		879	RBIAS			Yaw trajectory shaping bias
	C1	880	BXX	0		Local variables in High Frequency Autopilot
		881	BJJ			
		882	BKK			
		883	BXXSS			
		884	BJJSS			
		885	BKKSS			
		886	BTHTS			
		887	BPSIS			

Table 4. 2. 1 (Continued)

Module	Data Source	C Index	Fortran Name	Initial Value	Default Value	Definition
C1	C1	888	-	-	-	Not Used
		889	-	-	-	
		890	HXX	0	0	Roll signal limiter
		891				Not Used
		1099				
C4	C4	1100	BDELTD (1)	0	0	Fin No. 1 position derivative group
		1101		-	-	
		1102		-	-	
		1103	BDELT(1)	See def	-	(Computed in C4I)
		1104	BDELTD (2)	0	0	Fin No. 2 position derivative group
		1105	-	-	-	
		1106	-	-	-	
		1107	BDELT(2)	See def	-	
		1108	BDELTD (3)	0	0	Fin No. 3 position derivative group
		1109	-	-	-	
		1110	-	-	-	
		1111	BDELT(3)	See def	-	(Computed in C4I)
		1112	BDELTD (4)	0	0	Fin No. 4 position derivative group
		1113	-	-	-	
		1114	-	-	-	
		1115	BDELT(4)	See def	-	(Computed in C4I)
		1116	BDSD(1)	0	0	Local variable derivative group in the Actuator
		1117	-	-	-	
		1118	-	-	-	
		1119	BDS(1)	0	0	
		1120	BDSD(2)	0	0	Local variable derivative group in the Actuator
		1121	-	-	-	
		1122	-	-	-	
		1123	BDS(2)	0	0	

Table 4. 2. 1 (Continued)

Module	Data Source	C Index	Fortran Name	Initial Value	Default Value	Definition
C4	C4	1124	BDSD(3)	0	0	Local variable derivative group in the Actuator
↓	↓	1125	-	-	-	
↓	↓	1126	-	-	-	
↓	↓	1127	BDS(3)	0	0	
↓	↓	1128	BDSD(4)	0	0	Not Used
↓	↓	1129		-	-	
↓	↓	1130		-	-	
↓	↓	1131	BDS(4)	0	0	
↓	↓	1132				Not Used
↓	↓	↓				
↓	↓	1139				
↓	Input	1140	OPTACT	Input	0	Input switch to select high or low frequency actuator model (Low-OPTACT ≤ 0) (High-OPTACT > 0)
↓	C4	1141				Not Used
↓	↓	↓				
↓	↓	1144				
↓	Input	1145	CR	Input	0	Actuator modulation gain
↓	↓	1146	BDELH	-	-	Not Used
↓	↓	1147	HDEL	Input	0	Fin torque limit gain
↓	↓	1148	W1	↓	↓	Solenoid valve bandwidth
↓	↓	1149	ZN	↓	↓	Damping of fin position loop
↓	C4	1150	FMHD			
↓	Input	1151	G1	Input	0	Actuator forward loop gain
↓	↓	1152	BH	↓	↓	Deadband constant
↓	↓	1153	WN	↓	↓	
↓	↓	1154	G2	↓	↓	Fin aero load weighting factor
↓	↓	1155	H1	↓	↓	Positive fin torque limit
↓	↓	1156	H2	Input	0	Negative fin torque limit
↓		1157				Not Used
↓		1158				
↓		1159				

Table 4.2.1 (Continued)

Module	Data Source	C Index	Fortran Name	Initial Value	Default Value	Definition
C4 ↓	C4 ↓	1160	WDSDD (1)	0	0	Local variable 2nd order derivative group in the Actuator
		1161		-	-	
		1162		-	-	
		1163	WDSD(1)	0	0	
		1164				
		1165				
		1166	WDS(1)	0	0	
		1167	WDSDD (2)	0	0	Local variable 2nd order derivative group in the Actuator
		1168		-	-	
		1169		-	-	
		1170	WDSD(2)	0	0	
		1171		-	-	
		1172		-	-	
		1173	WDS(2)	0	0	
		1174	WDSDD (3)	0	0	Local variable 2nd order derivative group in the Actuator
		1175		-	-	
		1176		-	-	
		1177	WDSD(3)	0	0	
		1178		-	-	
		1179		-	-	
		1180	WDS (3)	0	0	
		1181	WDSDD (4)	0	0	Local variable 2nd order derivative group in the Actuator
		1182		-	-	
		1183		-	-	
		1184	WDSD(4)	0	0	
		1185		-	-	
		1186		-	-	
		1187	WDS(4)	0	0	
		1188 ↓ 1199				Not Used

Table 4.2.1 (Continued)

Module	Data Source	C Index	Fortran Name	Initial Value	Default Value	Definition
A1	Input	1200	OPTHNG		0	Undefined variable
		1201				Unused elements
		1202				
	A1	1203	CX			Total axial force coefficient value in body axis system
		1204	CY			Total side force coefficient value in body Axis system
		1205	CZ			Total normal force coefficient value in body axis system
		1206	CLP			Roll moment damping coefficient
		1207	CMQ			Pitch moment damping coefficient
		1208	CNR			Yaw moment damping coefficient
		1209	CL			Total rolling moment coefficient value in body axis system
		1210	CM			Total pitching moment coefficient value in body axis system
		1211	CN			Total yawing moment coefficient
	Table Look-up	1212	CXO			Axial force coefficient, wind axes
		1213	CZO			Normal force coefficient, wind axes
		1214	DCZ2			Incremental normal force coefficient due to aerodynamic roll angle (Φ'), wind axes
		1215	CZDQ			Normal force coefficient due to fin pitch position, wind axes
		1216	CZDF			Normal force coefficient due to fin yaw position, wind axes
		1217	DCY4			Incremental side force coefficient due to aerodynamic roll angle (Φ'), wind axes
		1218	CMO			Pitching moment coefficient, wind axes
		1219	DCMZ			Incremental pitching moment due to aerodynamic roll angle (Φ'), wind axes
		1220	CMDQ			Pitching control moment coefficient due to fin pitch position, wind axes
		1221	CMDR			Yaw control moment coefficient due to fin yaw position, wind axes
		1222	DCN4			Incremental yawing moment coefficient due to aerodynamic roll angle (Φ'), wind axes
		1223	DCL1			Unidentified
		1224	DCL4			Incremental rolling moment due to aerodynamic roll angle (Φ'), wind axes
		1225	CLDP			Roll damping moment coefficient, body axes
	G3	1226	VM	C(204)		Local mach number variable
	G5	1227	BAP	C(369)		Local variable for total angle of attack (α')
	A1	1228	BDL			Limited value of BDP (10° limit) (variable is not used)
		1229	BDM			Limited value of BDQ (10° limit) (variable is not used)
		1230	BDN			Limited value of BDR (10° limit) (variable is not used)
		1231	BDP	Input	0	Average value of the combined fin settings for roll control (DEG)
		1232	BDQ	Input	0	Average value of the combined fin settings for pitch control (DEG)
		1233	BDR	Input	0	Average value of the combined fin settings for yaw control (DEG)

Table 4.2.1 (Continued)

Module	Data Source	C Index	Fortran Name	Initial Value	Default Value	Definition
A1		1234				Unused elements
		1235				
		1236	CH1		0	Fin hinge moment coefficients (these coefficients are not used to calculate hinge moments. The hinge moments are calculated in C4 by coefficients stored by C4)
		1237	CH2		0	
		1238	CH3		0	
		1239	CH4		0	
		1240	CH11		0	Unidentified variables
		1241	CH21		0	
		1242	CH31		0	
		1243	CH41		0	
		1244				Unused elements
		1299				
A2	A2	1300	FXBA			X_B - Component of all external forces acting on the missile (axial forces) (LBS.)
		1301	FYBA			Y_B - Component of all external forces acting on the missile (side forces) (LBS.)
		1302	FZBA			Z_B - Component of all external forces acting on the missile (normal force), (LBS.)
		1303	FMXBA			X_B - Component of all external moments acting on the missile (rolling moment), (FT-LBS.)
		1304	FMYBA			Y_B - Component of all external moments acting on the missile (pitching moment), (FT-LBS.)
		1305	FMZBA			Z_B - Component of all external moments acting on the missile (yawing moment), (FT-LBS.)
	Input	1306	RFAREA	Input	0	Aerodynamic cross section reference area, (FT^2)
	Input	1307	RFLGTH	Input	0	Aerodynamic reference length used to compute aero moments, (FT)
	A3	1308	RDELCG	C(1417)	0	C. G. shift along X_B - axis, (FT)
	A2	1309	FMH1	*	*	Hinge moment about aerodynamics control surface, fin 1, (FT-LBS.)
	A2	1310	FMH2	*	*	Hinge moment about aerodynamics control surface, fin 2, (FT-LBS.)
	A2	1311	FMH3	*	*	Hinge moment about aerodynamics control surface, fin 3, (FT-LBS.)

---

* These four moments are computed as zero because the coefficients are set to zero. The hinge moments are recomputed in C4.

Table 4. 2. 1 (Continued)

Module	Data Source	C Index	Fortran Name	Initial Value	Default Value	Definition
A2		1313 1314 1315				Unused elements
A2	Input	1316	RLUG	Input	0	Distance between lugs, FT
A2	Input	1317	RAIL	Input	0	Rail length (distance between rear of front lug and end of rail), FT.
		1318 1319				Unused elements
	A3	1320	FMXTH	*	*	Thrust moment about X_B - axis, FT-LBS
	A3	1321	FMYTH	*	*	Thrust moment about Y_B-axis, FT-LBS
	A3	1322	FMZTH	*	*	Thrust moment about Z_B - axis, FT-LBS
	A2	1323	FMXLUG		0**	(1) X_B-component of moment counteracted by the front and rear lugs when both lugs are on the rail. (2) X_B-component of moment acting on the rear lug when the front lug has cleared the rail. (Rolling moment, FT-LBS.)
	A2	1324	FMYLUG		0**	(1) Y_B- component of moment counteracted by the front and rear lugs when both lugs are on the rail. (2) Y_B- component of moment acting on the rear lug when the front lug has cleared the rail. (Pitching moment, FT-LBS).
	A2	1325	FMZLUG		0**	(1) Z_B - component of moment counteracted by the front and rear lug when the lugs are on the rail. (2) Z_B- component of moment acting on the rear lug when the front lug has cleared the rail. (Yawing moment, FT-LBS).

* If QBURN is input as 1, these moments are set to zero in A3I.

** If OPTN4 = 0

Table 4.2.1 (Continued)

Module	Data Source	C Index	Fortran Name	Initial Value	Default Value	Definition
A2 A2 A2 A2		1326 1327 1328 1329				Not Used
A2	Input	1330	AGV	Input	0	Amplitude of the oscillatory acceleration acting on the missile due to the launcher vibrating, (g's.)
A2	Input	1331	CFREQ	Input	0	Frequency of the oscillatory acceleration acting on the missile due to the launcher vibrating, (CPS.)
A2	Input	1332	CPHAS	Input	0	Phase relationship at time =0 of the oscillatory acceleration acting on the missile due to the launcher vibrating, (DEG)
A2	Input	1333	AGV2	Input	0	Magnitude of the linear component of the acceleration acting on the missile due to the launcher, (g's.)
A2 A2 A2		1334 ↓ 1399				Not Used
A3		1400				Not Used
	Input	1401	BALPHT	Input	0	Thrust misalignment angles (used if QNALGN > 0)
	Input	1402	BPHIT	Input		
	Input	1403	QNALGN	Input	0	Thrust misalignment option switch (QNALGN >0 for thrust misalignment)
		1404	PCFTH			Not Used
	Input/ A3*	1405	QBURN	Input/ A3*	0	Engine burnout switch (*Set QBURN = 1 if simulation initial time is after engine burnout)
		1406 1407 1408				Not Used
	A3	1409	UDWP	0	0	Mass of burned propellant (SLUGS)
	A3	1410	FTHRST	Calculated in A3		Engine thrust (LBS)
	A3	1411	FTHX	"		(X, Y, and Z (body axes) components of thrust
	A3	1412	FTHY	"		FTHY = FTHZ = 0 and FTHX = FTHRST when QNALGN = 0)
	A3	1413	FTHZ	"		
	Input	1414	CISP	Input	0	Specific impulse (must be input as non-zero value)
	Input	1415	DWT	Input	0	Missile weight without propellant (LBS).
	Input	1416	DWP	Input	0	Propellant weight (LBS).
	Input	1417	RDCGD	Input	0	Initial x-coordinate (body axes) of C.G. (FT)
	Input	1418	RDCGF	Input	0	X-coordinate (body axes) of C.G. after engine burnout (FT)

Table 4. 2. 1 (Continued)

Module	Data Source	C Index	Fortran Name	Initial Value	Default Value	Definition
A3	Input	1419	FMIXF	Input	0	Burnout value of moment of inertia about x-axis (SLUG-FT^2)
	Input	1420	FMIYF	Input	0	Burnout value of moment of inertia about x-axis (SLUG-FT^2)
	Input	1421	RLCGO	Input	0	Initial value of x-coordinate (body axes) of C.G. (FT)
	A3	1422	RLCG	RLCGO	0	Instantaneous value of x-coordinate (body axes) of C.G. (FT)
		1423 → 1495				Not Used
	A3	1496	UMIPD	0	0	Local variable derivative group in Engine Model
	A3	1497				
	A3	1498				
	A3	1499	UMIP	0	0	
		1500 → 1550				Not Used
A1	Input	1551	OPTM	Input	0	Fixed fin position option switch (OPTM > 0 results in constant fin positions specified by input data)
A1	Input	1552	UPHI	Input	0	Fixed autopilot gyro Euler angles for fixed fin position option
A1	Input	1553	UPSI	Input	0	
A1	Input	1554	UTHT	Input	0	
A1	Input	1555	UDL1	Input	0	Input values of fixed fin positions for fins 1, 2, 3, and 4 (DEG)
A1	Input	1556	UDL2	Input	0	
A1	Input	1557	UDL3	Input	0	
A1	Input	1558	UDL4	Input	0	
		1559 → 1599				Not Used
D1	D1	1600	VXED	Computed in D1		X-component (earth axes) of velocity derivative group
		1601				
		1602				
		1603	VXE	Input/ D1I	0	VXE, VYE, and VZE can be input when OPTN2 = OPTN4 = 0, otherw'se they are internally defined.
		1604	VYED	Computed in D1		Y-component (earth axes) of velocity derivative group

Table 4.2.1 (Continued)

Module	Data Source	C Index	Fortran Name	Initial Value	Default Value	Definition	
D1	D1	1605					
		1606				Y-component (earth axes) of velocity derivative group	
		1607	VYE	Input/ D1I	0		VXE, VYE, and VZE can be input when OPTN2 = OPTN4 = 0, otherwise they are internally defined.
		1608	VZED	Computed in D1			
		1609				Z-component (earth axes) of velocity derivative group	
		1610					
		1611	VZE	Input/ D1I	0		
		1612	RXED	Computed in D1			
		1613				Position coordinate (earth axes) derivative group	
		1614					
		1615	RXE	Input/ D1I			See Table 4.1.3 for input instructions
		1616	RYED	Computed in D1			
		1617				Y position coordinate (earth axes) derivative group	
		1618					
		1619	RYE	Input/ D1I			
		1620	RZED	Computed in D1			
		1621				Z position coordinate (earth axes) derivative group	
		1622					
		1623	RZE	Input/ D1I			
		1624	AXBA	Calculated in D1		X, Y, and Z (body axes) components of translational accelerations (FT/SEC^2)	
		1625	AYBA	"			
		1626	AZBA	"			

Table 4. 2. 1 (Continued)

Module	Data Source	C Index	Fortran Name	Initial Value	Default Value	Definition
D1	D1	1627	AGRAV	Input		Acceleration due to gravity (FT/SEC^2)
D1	A3	1628	DMASS	(DWT+ DWP)/ 32.17		Instantaneous
	Input	1629	ATHRST	Input	0	Target acceleration (FT/SEC^2) (used only if OPTARG>0)
	Input	1630	ATURNT	Input	0	Target turning rate (DEG/SEC) (used only if OPTARG>0)
	Input	1631	BGAMT	Input	0	Target motion descent/ascent angle
	D1	1632	VDELX	Calcu- lated in D1		X, Y, and Z (earth axes) components of missile-target closing velocity
		1633	VDELY	"		
		1634	VDELZ	"		
		1635	RDELX			Not Used
		1636	RDELY			
		1637	RDELZ			
		1638	VCLSNG			
	Input	1639	OPTARG	Input	0	Target motion option switch (OPTARG O for target motion)
	D1	1640	VTARGD		0	Target velocity magnitude derivative group
	D1	1641				
	D1	1642			0	
	D1	1643	VTARG	Input	0	
		1644	BPSITD		0	Target azimuth angle derivative group
		1645				
		1646				
		1647	BPSIT	Input	0	
		1648	RTXED	VTXE		Target position derivative group for x-coordinate (earth axes) RTXE, RTYE, and RTZE can be input when OPTN2 = OPTN4 = 0, otherwise they are automatically set to an initial value of zero
		1649				
		1650				
		1651	RTXE	Input/	0	
		1652	RTYED	VTYE		Target position derivative group for y-coordinate (earth axes)
		1653				
		1654				
		1655	RTYE	Input/	0	

Table 4. 2. 1 (Continued)

Module	Data Source	C Index	Fortran Name	Initial Value	Default Value	Definition
D1	D1	1656	RTZED	VTZE		Target position derivative group for z-coordinate (earth axes)
		1657				
		1658				
		1659	RTZE	Input/ 0	0	
		1660	VTXE	Computed in D1	0 0	X, Y, and Z components (earth axes) of target velocity (FT/SEC)
		1661	VTYE	"	0	
		1662	VTZE	"	0	
		1663	VDXB	"		X, Y, and Z components (body axes) of missile acceleration (FT/SEC^2)
		1664	VDYB	"		
		1665	VDZB	"		
	Input	1666	BLOSV			Initial LOS elevation angle (DEG)
	Input	1667	RSLANT			Initial missile to target slant range (FT)
	D1	1668	RXO	RXE		Initial X, Y, and Z (earth axes) position coordinates (FT)
		1669	RYO	RYE		
		1670	RZO	RZO		
		1671	VXO	VXE		Initial X, Y, and Z velocity components (earth axes) (FT/SEC)
		1672	VYO	VYE		
		1673	VZO	VZE		
		1674	VMWTE	Input	0	Initial velocity of missile relative to wind (earth axes) (FT/SEC)
		1675				Not Used
		1676	ANGX	Calculated in D1		X, Y, and Z components (body axes) of acceleration (g's)
		1677	ANGY	"		
		1678	ANGZ	"		
		1679				Not Used
		1680				
		1681	ADIVE			
		1682				Not Used
		1699				

Table 4.2.1 (Continued)

Module	Data Source	C Index	Fortran Name	Initial Value	Default Value	Definition
D2	D2	1700	CFA11D	Calculated in D2		Derivative groups for Earth to Body Axes transformation matrix [CFA]
		1701				
		1702				
		1703	CFA11			
		1704	CFA12D			
		1705				
		1706				
		1707	CFA12			
		1708	CFA13D			
		1709				
		1710				
		1711	CFA13			
		1712	CFA21D			
		1713				
		1714				
		1715	CFA21			
		1716	CFA22D			
		1717				
		1718				
		1719	CFA22			
		1720	CFA23D			
		1721				
		1722				
		1723	CFA23			
		1724	CFA31D			
		1725				
		1726				
		1727	CFA31			
		1728	CFA32D			
		1729				
		1730				
		1731	CFA32			
		1732	CFA33D			
		1733				
		1734				
		1735	CFA33			

Table 4.2.1 (Continued)

Module	Data Source	C Index	Fortran Name	Initial Value	Default Value	Definition
D2	D2	1736	WPD	0	0	Derivative group for angular velocity (Roll) Angular velocities (WP, WQ, WR) can be input if simulation is initiated after engine burnout(QBURN = 1)
		1737				
		1738				
		1739	WP	0 /Input	0	
		1740	WQD		0	Derivative group for angular velocity (Pitch)
		1741				
		1742				
		1743	WQ	0/Input	0	
		1744	WRD	0	0	Derivative group for angular velocity (Yaw)
		1745				
		1746				
		1747	WR	0/Input	0	
	A3	1748	FMIX	Calcu-lated in A3		Instantaneous moments of inertia about X, Y, and Z body axes (SLUG/FT^2)
	A3	1749	FMIY	"		
	A3	1750	FMIZ	"		
	D1I	1751	CRAD	Set in D1I	0	Radians to degrees conversion (DEG/RADIAN)
	Input	1752	BPHIO			Initial value of body Euler roll angles (DEG)
	Input	1753	BTHTO			Initial value of body Euler pitch angle (DEG)
	Input	1754	BPSIO			Initial value of body Euler yaw angle (DEG)
		1755	A021	CFA21		Components of the unit vector along the autopilot platform spin axis, in the earth coordinate system
		1756	A022	CFA22		
		1757	A023	CFA23		
		1758	A031	CFA31		Components of the unit vector along the autopilot platform pitch-roll gyro spin axis, in the earth coordinate system
		1759	A032	CFA32		
		1760	A033	CFA33		
		1761 ↓ 1949				Not used
		1950 ↓ 1970				Not Used
AMRK	Input	1971	RITE	Input	0	Option switch for step size halving printout (0 - Print when integrated upper limit set in A5I is exceeded)

Table 4.2.1 (Continued)

Module	Data Source	C Index	Fortran Name	Initial Value	Default Value	Definition
MAIN	Input	1972	RKUTTA	Input	0	Runge Kutta integration switch (> 0 - stays in Runge Kutta; 0 - Adams Moulton with R. K. start)
AMRK	AMRK	1973	KASE	0 (Set in MAIN)		Initialization bypass switch
Program Variable		1974	NJ			Number of derivatives
Program Variable		1975	NPT			Integration mode control switch
		1976 ↓ 1979				Not Used
		1980	RN			Undefined Variable
		1981	RNT			Undefined Variable
PLOT4	Input	1982	PLOTN4	Input	0	Number of variables in trajectory plot (0 - no trajectory plots; 4 - trajectory plot)
PLOT4	Input	1983	PLOTN2	Input	0	Number of variables in full page plot (multiple dependent variables to be plotted against one independent variable)
OINPT1	OINPT1	1984	NPLOT	0 (Set in Main)		Total number of variables input on type 7 input data cards
OUPT3 ↓	OINPT1 ↓	1985 ↓ 1999	OUTPLT (1) ↓ OUTPLT (15)			Pointer to the index of the "C" array of variables to be plotted (Maximum of 15)
Program Variable		2000	T	Input	0	Trajectory flight time (SEC)
		2001	TF	Input	0	Maximum allowable flight (SEC)
		2002				Unused Element
OUPT3	OUPT3	2003	PCNT			Control variable for normal print output
OUPT3	OUPT3	2004	PPNT	C(2003)		Control variable for plot output
OUPT3	Input	2005	PPP	Input	0	Plot interval (SEC)
MAIN	Input	2006	REPPLT	Input	0	Controls multi-case plot and print data card read options 0 - Use new type 4 and 7 input cards 1 - Use old plus new type 4 and 7 input cards -1 - Use new type 7 input cards
MAIN	Input	2007	PTLESS	Input	0	Number of plot points to eliminate from end of trajectory before generating plot tape

Table 4.2.1 (Continued)

Module	Data Source	C Index	Fortran Name	Initial Value	Default Value	Definition
MAIN	Input	2008	PLOTNO	Input	0	Total number of variables to plot for all three options
MAIN	MAIN	2009	NOPLOT	C(2008)	0	Total number of variables to plot for all three options
MAIN	Input	2010	STEP	Input	0	Flag that indicates the nature of the next case to be run. STEP may take on any one of eleven values. The input value of STEP determines the point in MAIN at which the next case will be start execution. Normal input values are: 2 - normal run follows or 11 - terminate case (The entry point in MAIN based on the value of STEP is indicated in Figure 4.1.2 for all values of STEP from 1 to 11.)
MAIN	STGE3	2011	KSTEP	1 (Set in STGE2)		Case termination switch
MAIN	MAIN	2012	LSTEP	C(2010)	0	Integer value of STEP
OUPT2	Input	2013	DOC	Input	0	Allows dumping of "C" - array for the first 6-DOC integration intervals
OUPT3	OUPT2	2014	ITCNT	C(2013) +1	1	C-array dump switch
OUPT3	Input	2015	CPP	Input	0	Print interval (SEC)
OUPT3	OUPT3	2016	PGCNT	1 (Set in OUPT2)		Page counter for heading print
OUPT3	OUPT2	2017	DTCNT			Number of lines required for data print block plus four blank lines
		2018	TAPE			Undefined variable
		2019	TAPEND			Undefined variable
STGE3	G4	2020	LCONV	0(Set in STGE2)		Integration termination switch
STGE3	STGE3	2021	KCONV	0 (Set in STGE2)		Pass counter for step size halving in the iteration loop to force T within the hardwired tolerance of .01 sec of TF
MAIN	Input	2022	OPTN10	Input	0	C-array dump at end of run switch 0 - no dump 1 - dump C-array
MAIN	OUPT3	2023	OPOINT			Number of time points to plot
		2024				Not Used
OUPT3	PLOT4	2025	TIME(1)	C(2000)	0	Independent variable for plot Option 3. (trajectory plot sampling time points)
↓	↓	↓	↓			
OUPT3	PLOT4	2324	TIME (300			

Table 4. 2. 1 (Continued)

Module	Data Source	C Index	Fortran Name	Initial Value	Default Value	Definition
MAIN	OINTPL	2325	VLABLE (1)			Plot variable labels input by type 7 input cards
↓	↓	↓	↓			
MAIN	OINTPL	2354	VLABLE (30)			
OINTPL	Input	2355	IR (1)			Temporary location of card type number card
OINTPL	Input	2356	IR (2)			Temporary location of C index on input data cards
OINTPL	Input	2357	VR (1)			Temporary location of variable value on input data card
OINTPL	Input	2358	VR (2)			Temporary location of reset flag on input data card
		2359				Not used
		2360				Not used
AUXSUB	OINTPL	2361	NOMOD			Number of functional modules specified by type 2 input cards
AUXSUB	OINTPL	2362	MODNO (1)			Functional module numbers specified by type 2 input cards
↓	↓	↓				
AUXSUB	OINTPL	2460	MODNO (99)			
AUXSUB	OINTPL	2461	NOSUB		0	Number of functional modules
AUXSUB	OINTPL	2462	SUBNO(1)			Output and staging subroutine numbers specified by type 1 input cards
↓	↓	↓	↓			
AUXSUB	OINTPL	2560	SUBNO (99)			
PROGRAM VARIABLE	Sequentially in Initialization Subroutines	2561	N	0	0	Counter for loading IPL array
"	"	2562	IPL(1)			Pointer to the derivatives in the C-array
		↓	↓			
"	"	2661	IPL (100)			
AMRK	Input	2662	HMIN	Input	0	Integration step size limits (SEC)
AMRK	Input	2663	HMAX	Input	0	
AMRK	Input	2664	DER(1)	Input	0	Integration stop size (SEC)
AMRK	AMRK	2665	DER(2)			Derivative array used by AMRK
↓	↓	↓	↓			
AMRK	AMRK	2764	DER(101)			
AMRK	ASI	2765	EL(1)	Defined in ASI		Adams-Moulton lower integration step size switching limit
↓	↓	↓	↓			
AMRK	ASI	2764	EL(100)			

Table 4.2.1 (Continued)

Module	Data Source	C Index	Fortran Name	Initial Value	Default Value	Definition
AMRK	A5I	2865	EU(1)			Adams-Moulton upper integration step size switching limit
		↓	↓			
AMRK	A5I	2964	EU(100)			
AMRK	AMRK	2965	VAR(1)			Simulation flight time (SEC)
AMRK	AMRK	2966	VAR(2)			Integrated value array used in AMRK
↓	↓	↓	↓			
AMRK	AMRK	3065	VAR(101)			
RESET	OINPT1	3066	NOLIST			Number of variables to reset to initial values at end of case
RESET	OINPT1	3067	LISTNO(1)			C-array indices of variables to be reset to initial values at end of case
↓	↓	↓	↓			
RESET	OINPT1	3116	LISTNO (50)			
RESET	OINPT1	3117	VALUE(1)			Reset values of variables to be restores to initial values at end of case
		↓	↓			
RESET	OINPT1	3166	VALUE(50)			
OUPT3	OINPT1	3167	NOOUT			No. of variables to print of specified print intervals
OUPT3	OINPT1	3168	OUTNO(1)			C-index of variables to print
		↓	↓			
OUPT3	OINPT1	3217	OUTNO (50)			
OUPT3	OINPT1	3218	ONAME1 (1)			First six characters of print variable input NAMES
		↓	↓			
OUPT3	OINPT1	3267	ONAME1 (50)			
OUPT3	OINPT1	3268	ONAME2 (1)			Second six characters of print variable input NAMES
		↓	↓			
OUPT3	OINPT1	3317	ONAME2 (50)			
OUPT3	OINPT1	3318	ONAME3 (1)			Not Used
		↓	↓			
OUPT3	OINPT1	3327	ONAME3 (10)			
OUPT3	OINPT1	3328	ONAME4 (1)			Not Used
		↓	↓			
OUPT3	OINPT1	3337	ONAME4 (10)			

Table 4.2.1 (Continued)

Module	Data Source	C Index	Fortran Name	Initial Value	Default Value	Definition
OINPT1,	OINPT1	3338	LOSTAT	0 (Set in ZERO)		Not Used
OINPT1	OINPT1	3339	NOSTAT	0 (Set in ZERO)		Not Used
		3340 ↓ 3439	STATNO (1) ↓ STATNO (100)			Not Used
		3440	NORNDM			Not Used
		3441 ↓ 3490	RNDMO (1) ↓ RNDMO (50)			Not Used
		3491 ↓ 3501				Not Used
D1I	Input	3502	OPTN2	Input	0	Initial input option 0 - Input position and velocity of missile and seeker gimbal angles. 1 - Input RSLANT and BSLOV. Initial position components (RXE, RZE) are computed from RSLANT and BSLOV. 2 - Input initial position components RXE and RZE (RYE automatically set to zero). RSLANT computed from RXE and RZE.
D8	Input	3503	OPTN3	Input	0	Roll rate option selector 0 - compute roll rate from moments about roll axis 1 - maintain roll rate at zero
D1I	Input	3504	OPTN4	Input	0	Rail launcher dynamics and fire selector option 0 - no rail dynamics; direct fire 1 - compute rail dynamics; direct fire 2 - compute rail dynamics; indirect fire
		3505				Not Used
D1I	Input	3506	OPTN6	Input	0	Missile initial velocity (relative to wind) option (used only if OPTN4=0) 0 - Input missile initial mach number (MACH) 1 - Input missile initial velocity relative to the wind (VMWTE)

A1. ANGULAR ACCELERATION EQUATIONS

The angular acceleration equations implemented in the 6DOF program (subroutine D2) are:

$$\omega_{PD} = \frac{FMXBA}{FMIX} CRAD$$

$$\omega_{QD} = \frac{FMYBA \cdot CRAD + (FMIZ-FMIX)\ \frac{\omega_P \omega_Q}{CRAD}}{FMIY} \qquad (1)$$

$$\omega_{RD} = \frac{FMZBA \cdot CRAD + (FMIX-FMIY)\ \frac{\omega_P \omega_Q}{CRAD}}{FMIZ}$$

where

ω_{PD} denotes angular acceleration about X_b

ω_{QD} denotes angular acceleration about Y_b

ω_{RD} denotes angular acceleration about Z_b

FMXBA denotes applied moment about X_b

FMYBA denotes applied moment about Y_b

FMZBA denotes applied moment about Z_b

FMIX denotes moment of inertia about X_b

FMIY denotes moment of inertia about Y_b

FMIZ denotes moment of inertia about Z_b

Equation (1) is based on the assumption that: (1) FMIZ=FMIY (because of vehicle symmetry about X_b), and (2) the products of inertia are zero (not strictly true, even initially).

Equations (1) is derived as follows:

$\bar{\blacktriangle} = \bar{\bar{I}}\,\bar{\omega}$, where $\bar{\blacktriangle}$ is the angular momentum vector, $\bar{\bar{I}}$ is the inertial tensor, and $\bar{\omega}$ is the angular velocity vector, all in the Body coordinate system.

Now:

$\frac{d}{dt}\bar{\blacktriangle} = \frac{d}{dt}(\bar{\bar{I}}\bar{\omega}) = \bar{\blacktriangledown}$, i.e., the time derivative of the angular momentum vector equals the torque vector

Have:

$$\bar{\bar{I}} = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix}_{E_b} \tag{2}$$

$$\bar{\omega} = \frac{1}{CRAD} \begin{bmatrix} \omega P \\ \omega Q \\ \omega R \end{bmatrix}_{E_b}$$

So:

$$\bar{\bar{I}}\,\bar{\omega} = \left(I_{xx}(\omega_P)\bar{X}_b + I_{yy}(\omega_Q)\bar{Y}_b + I_{zz}(\omega_R)\bar{Z}_b\right)\frac{1}{CRAD} \quad ,$$

where $\bar{X}_b$, $\bar{Y}_b$, and $\bar{Z}_b$ are unit vectors in the body system.

Thus:

$$\begin{aligned} \frac{d}{dt}(\bar{\bar{I}}\bar{\omega}) = \frac{1}{CRAD} & \Big\{ I_{xx}(\dot{\omega}_P)\bar{X}_b + I_{xx}(\omega_P)\dot{\bar{X}}_b \\ & + I_{yy}(\dot{\omega}_Q)\bar{Y}_b + I_{yy}(\omega_Q)\dot{\bar{Y}}_b \\ & + I_{zz}(\dot{\omega}_R)\bar{Z}_b + I_{zz}(\omega_R)\dot{\bar{Z}}_b \Big\} \\ = {} & M_x\bar{X}_b + M_y\bar{Y}_b + M_z\bar{Z}_b \end{aligned} \tag{3}$$

where the assumption is made that

$$\dot{I}_{xx} = \dot{I}_{yy} = \dot{I}_{zz} = 0 \quad .$$

The time derivatives of the unit vectors $\bar{X}_b$, $\bar{Y}_b$, and $\bar{Z}_b$ are:

$$\dot{\bar{X}}_b = \frac{\omega_R\bar{Y}_b - \omega_Q\bar{Z}_b}{CRAD}$$

$$\dot{\bar{Y}}_b = \frac{\omega_P\bar{Z}_b - \omega_R\bar{X}_b}{CRAD}$$

$$\dot{\bar{Z}}_b = \frac{\omega_Q\bar{X}_b - \omega_P\bar{Y}_b}{CRAD}$$

Making these substitutions in Equation (3), and collecting terms, have:

$$\frac{1}{CRAD}\left\{\left(I_{xx}(\dot{\omega}P) - \frac{I_{yy}(\omega_Q)(\omega_R)}{CRAD} + \frac{I_{zz}(\omega_R)(\omega_Q)}{CRAD}\right)\bar{X}_b\right.$$

$$+\left(I_{yy}(\dot{\omega}_Q) + \frac{I_{xx}(\omega_P)(\omega_R)}{CRAD} - \frac{I_{zz}(\omega_R)(\omega_P)}{CRAD}\right)\bar{Y}_b$$

$$\left.+\left(I_{zz}(\dot{\omega}_R) - \frac{I_{xx}(\omega_P)(\omega_Q)}{CRAD} - \frac{I_{yy}(\omega_Q)(\omega_P)}{CRAD}\right)\bar{Z}_b\right\}$$

$$= M_x\bar{X}_b + M_y\bar{Y}_b + M_z\bar{Z}_b \ .$$

Simplifying:

$$\left(I_{xx}(\dot{\omega}_P) + \frac{(I_{zz} - I_{yy})(\omega_R)(\omega_Q)}{CRAD}\right)\bar{X}_b$$

$$+\left(I_{yy}(\dot{\omega}_Q) + \frac{(I_{xx} - I_{zz})(\omega_P)(\omega_R)}{CRAD}\right)\bar{X}_b$$

$$+\left(I_{zz}(\dot{\omega}_R) + \frac{(I_{yy} - I_{xx})(\omega_P)(\omega_Q)}{CRAD}\right)\bar{X}_b$$

$$= CRAD\,(M_x\bar{X}_b + M_y\bar{Y}_b - M_z\bar{Z}_b).$$

Equating like coefficients and solving for $(\dot{\omega}_P)$, $(\dot{\omega}_Q)$, $(\dot{\omega}_R)$, and noting that $I_{yy} = I_{zz}$, have that:

$$\dot{\omega}_P = \frac{M_x \cdot \text{CRAD}}{I_{xx}}$$

$$\dot{\omega}_Q = \left(M_y \cdot \text{CRAD} + \frac{(I_{zz} - I_{xx}) (\omega_P) (\omega_R)}{\text{CRAD}} \right) \frac{1}{I_{yy}} \qquad (4)$$

$$\dot{\omega}_R = \left(M_z \cdot \text{CRAD} + \frac{(I_{xx} - I_{yy}) (\omega_P) (\omega_Q)}{\text{CRAD}} \right) \frac{1}{I_{zz}}$$

Substituting the program names in Equation (4), yields Equation (1).

A2 - TIME DERIVATIVES OF THE EARTH TO BODY AXES TRANSFORMATION MATRIX, M

The time derivative of the earth to body axes transformation matrix, $\frac{d}{dt}$ (M) can be obtained as follows:

(1) let $\overline{X}_b$, $\overline{Y}_b$, $\overline{Z}_b$ and $\overline{X}_e$, $\overline{Y}_e$, $\overline{Z}_e$ denote unit vectors in the body and earth system, respectively.

(2) Assume the unit vectors in E_b have angular velocity components about $\overline{X}_b$, $\overline{Y}_b$, $\overline{Z}_b$ of ω_P, ω_Q, and ω_R (degrees/seconds), respectively.

Then the time derivatives of the unit vectors $M^T E_b$, written in terms of ω_P, ω_Q, and ω_R are (see Figure A-2. 1 for velocity components):

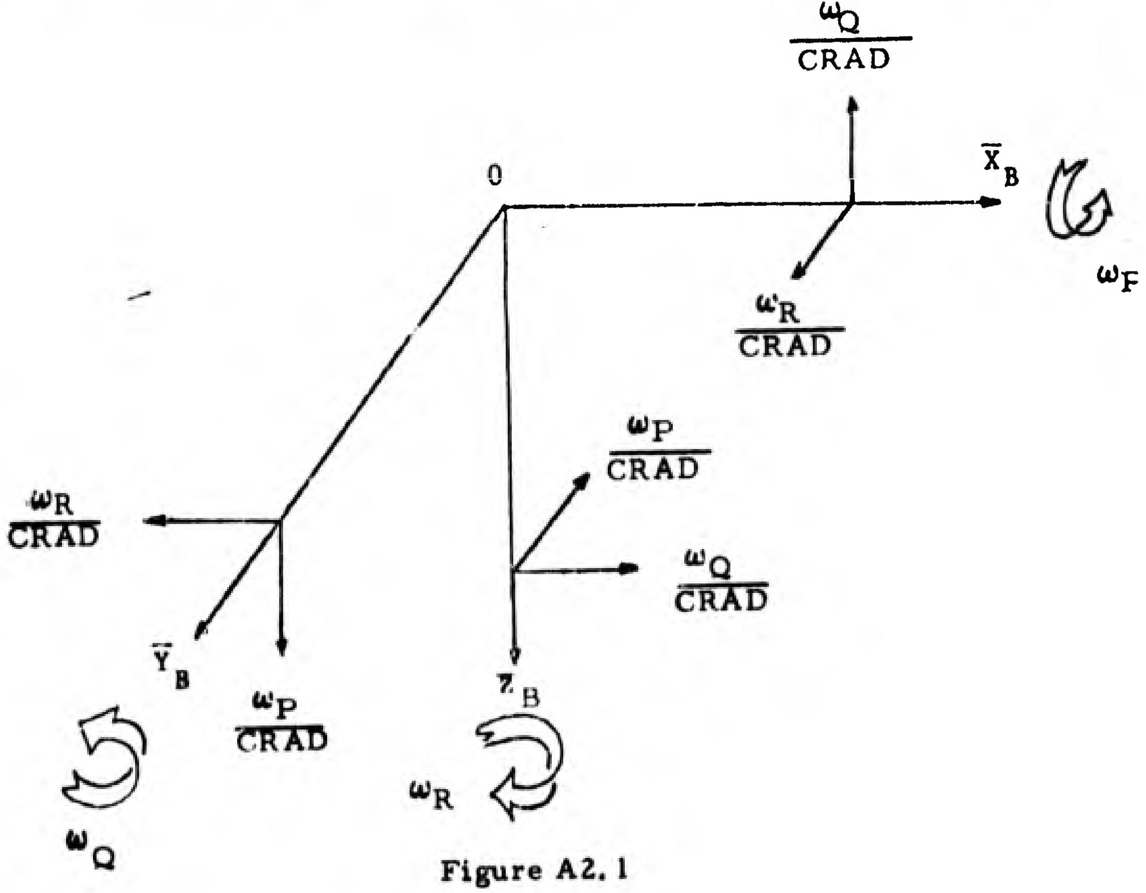


Figure A2. 1

$$\dot{\overline{X}}_b = \frac{0\overline{X}_b + \omega_R \overline{Y}_b - \omega_Q \overline{Z}_b}{CRAD} = \begin{bmatrix} 0 \\ \omega_R \\ -\omega_Q \end{bmatrix}_{E_b}$$

$$\dot{\overline{Y}}_b = \frac{-\omega_R \overline{X}_b + 0\overline{Y}_b + \omega_P \overline{Z}_b}{CRAD} = \begin{bmatrix} -\omega_R \\ 0 \\ \omega_P \end{bmatrix}_{E_b}$$

$$\dot{\overline{Z}}_b = \frac{\omega_Q \overline{X}_b - \omega_P \overline{Y}_b + 0\overline{Z}_b}{CRAD} = \begin{bmatrix} \omega_Q \\ -\omega_P \\ 0 \end{bmatrix}_{E_b}$$

where CRAD is the number of degrees per radian.

Transforming $\dot{\overline{X}}_b$, $\dot{\overline{Y}}_b$, $\dot{\overline{Z}}_b$ into the Earth coordinate systems E_e yields:

1.

$$\begin{bmatrix} \dot{X}_b \end{bmatrix}_{E_e} = M^T \dot{\overline{X}}_b = \begin{bmatrix} C_{11} & C_{21} & C_{31} \\ C_{12} & C_{22} & C_{32} \\ C_{13} & C_{23} & C_{33} \end{bmatrix} \dot{\overline{X}}_b = \begin{bmatrix} C_{21}\ \omega_R - C_{31}\ \omega_Q \\ C_{22}\ \omega_R - C_{32}\ \omega_Q \\ C_{23}\ \omega_R - C_{33}\ \omega_Q \end{bmatrix}_{E_e}$$

$$\begin{bmatrix} \dot{Y}_b \end{bmatrix}_{E_e} = M^T \dot{\overline{Y}}_b = \begin{bmatrix} -C_{11}\ \omega_R + C_{31}\ \omega_P \\ -C_{12}\ \omega_R + C_{32}\ \omega_P \\ -C_{13}\ \omega_R + C_{33}\ \omega_P \end{bmatrix}_{E_e}$$

$$\begin{bmatrix} \dot{Z}_b \end{bmatrix}_{E_e} = M^T \dot{\overline{Z}}_b = \begin{bmatrix} C_{11}\ \omega_Q - C_{21}\ \omega_P \\ C_{12}\ \omega_Q - C_{22}\ \omega_P \\ C_{13}\ \omega_Q - C_{23}\ \omega_P \end{bmatrix}_{E_e}$$

But the time derivatives of the unit vectors X_b, Y_b, Z_b, can also be expressed in the E_e system by first transforming into the E_e system and then taking the time derivatives. Thus:

$$\begin{bmatrix} X_b \end{bmatrix}_{E_e} = M^T \bar{X}_b = \begin{bmatrix} C_{11} & C_{21} & C_{31} \\ C_{12} & C_{22} & C_{32} \\ C_{13} & C_{23} & C_{33} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}_{E_b} = \begin{bmatrix} C_{11} \\ C_{12} \\ C_{13} \end{bmatrix}_{E_b}$$

$$\begin{bmatrix} Y_b \end{bmatrix}_{E_e} = M^T \bar{Y}_b = \begin{bmatrix} C_{21} \\ C_{22} \\ C_{23} \end{bmatrix}_{E_b}$$

$$\begin{bmatrix} Z_b \end{bmatrix}_{E_e} = \begin{bmatrix} C_{31} \\ C_{32} \\ C_{33} \end{bmatrix}_{E_b}$$

written in terms of the unit vectors $\bar{X}_e$, $\bar{Y}_e$, $\bar{Z}_e$, have:

$$\begin{bmatrix} X_b \end{bmatrix}_{E_e} = C_{11} \bar{X}_e + C_{12} \bar{Y}_e + C_{13} \bar{Z}_e$$

$$\begin{bmatrix} Y_b \end{bmatrix}_{E_e} = C_{21} \bar{X}_e + C_{22} \bar{Y}_e + C_{23} \bar{Z}_e$$

$$\begin{bmatrix} Z_b \end{bmatrix}_{E_e} = C_{31} \bar{X}_e + C_{32} \bar{Y}_e + C_{33} \bar{Z}_e$$

Since the E_e system is considered fixed, $\dot{\bar{X}}_e, \dot{\bar{Y}}_e, \dot{\bar{Z}}_e = 0$.   Thus:

2.

$$\frac{d}{dt}\begin{bmatrix}X_b\end{bmatrix}_{E_e} = \dot{C}_{11}\,\bar{X}_e + \dot{C}_{12}\,\bar{Y}_e + \dot{C}_{13}\,\bar{Z}_e = \begin{bmatrix}\dot{C}_{11}\\ \dot{C}_{12}\\ \dot{C}_{13}\end{bmatrix}_{E_e}$$

$$\frac{d}{dt}\begin{bmatrix}Y_b\end{bmatrix}_{E_e} = \begin{bmatrix}\dot{C}_{21}\\ \dot{C}_{22}\\ \dot{C}_{23}\end{bmatrix}_{E_e}$$

$$\frac{d}{dt}\begin{bmatrix}Z_b\end{bmatrix}_{E_e} = \begin{bmatrix}\dot{C}_{31}\\ \dot{C}_{32}\\ \dot{C}_{33}\end{bmatrix}_{E_e}$$

Since the corresponding expression obtained by the two methods represent the same vectors, have from 1 and 2 that:

$$\dot{C}_{11} = C_{21}\,\omega_R - C_{31}\,\omega_Q$$
$$\dot{C}_{12} = C_{22}\,\omega_R - C_{32}\,\omega_Q$$
$$\dot{C}_{13} = C_{23}\,\omega_R - C_{33}\,\omega_Q$$
$$\dot{C}_{21} = C_{31}\,\omega_P - C_{11}\,\omega_R$$
$$\dot{C}_{22} = C_{32}\,\omega_P - C_{12}\,\omega_R$$
$$\dot{C}_{23} = C_{33}\,\omega_P - C_{13}\,\omega_R$$
$$\dot{C}_{31} = C_{11}\,\omega_Q - C_{21}\,\omega_P$$
$$\dot{C}_{32} = C_{12}\,\omega_Q - C_{22}\,\omega_P$$
$$\dot{C}_{33} = C_{13}\,\omega_Q - C_{23}\,\omega_P$$

A3 - ORIENTATION OF BODY AND EARTH AXES

The Euler angles which orient the body axis system with respect to the Earth axis system are defined as follows:

(1) A positive rotation (toward Y_E from X_E) of ψ (degrees) about the Z_E axis, giving a system $E'(X^1, Y^1, Z^1)$;

(2) A negative rotation (The X^1 axis up from the plane of the Earth axis system) of θ about the Y^1 axis, giving a system $E''(X'', Y'', Z'')$;

(3) A positive rotation (toward Z'' from Y'') of [illegible] degrees about the X'' axis, giving the Body axis system $E_B(X_B, Y_B, Z_B)$

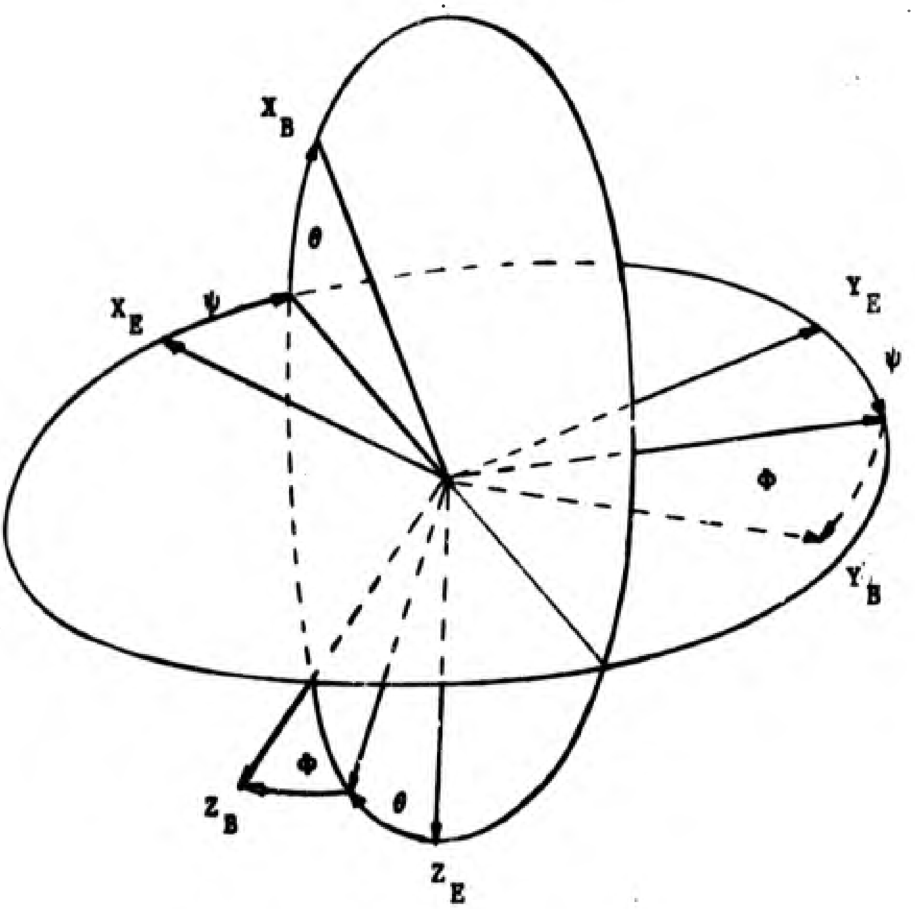


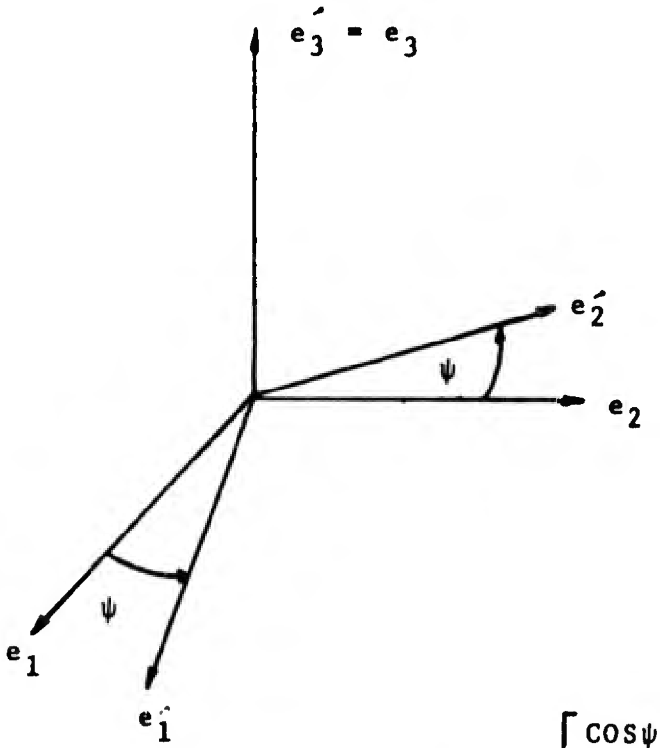
Figure A3.1

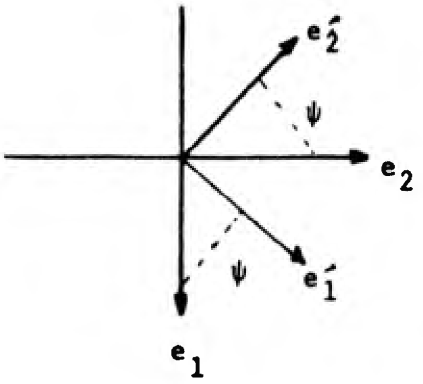
The transformation, M, from the Earth axis system to the Body axis system is given by:

$$M = \begin{bmatrix} \text{COS}\psi \ \ \text{COS}\ \theta & \text{SIN}\psi \ \ \text{COS}\ \theta & -\text{SIN}\ \theta \\ \text{COS}\psi \ \ \text{SIN}\ \theta \ \ \text{SIN}\ \Phi & \text{COS}\psi \ \ \text{COS}\ \Phi & \text{COS}\ \theta\ \text{SIN}\ \Phi \\ -\text{SIN}\psi \ \ \text{COS}\Phi & +\text{SIN}\psi \ \ \text{SIN}\theta\ \text{SIN}\ \Phi & \\ \text{COS}\psi \ \ \text{SIN}\ \theta \ \ \text{COS}\ \Phi & \text{SIN}\psi\ \text{SIN}\ \theta\ \text{COS}\ \Phi & \text{COS}\ \theta \ \ \text{COS}\ \Phi \\ +\ \text{SIN}\psi \ \ \text{SIN}\ \Phi & -\text{COS}\psi\ \text{SIN}\ \Phi & \end{bmatrix} \quad (1)$$

Thus, a vector $\overline{V}_E$ espressed in the Earth axis system is given in the Body axis system as $\overline{V}_B = M\overline{V}_E$.

First Rotation, ψ





$$e_1' = \text{COS}\psi\ e_1 + \text{SIN}\psi\ e_2$$

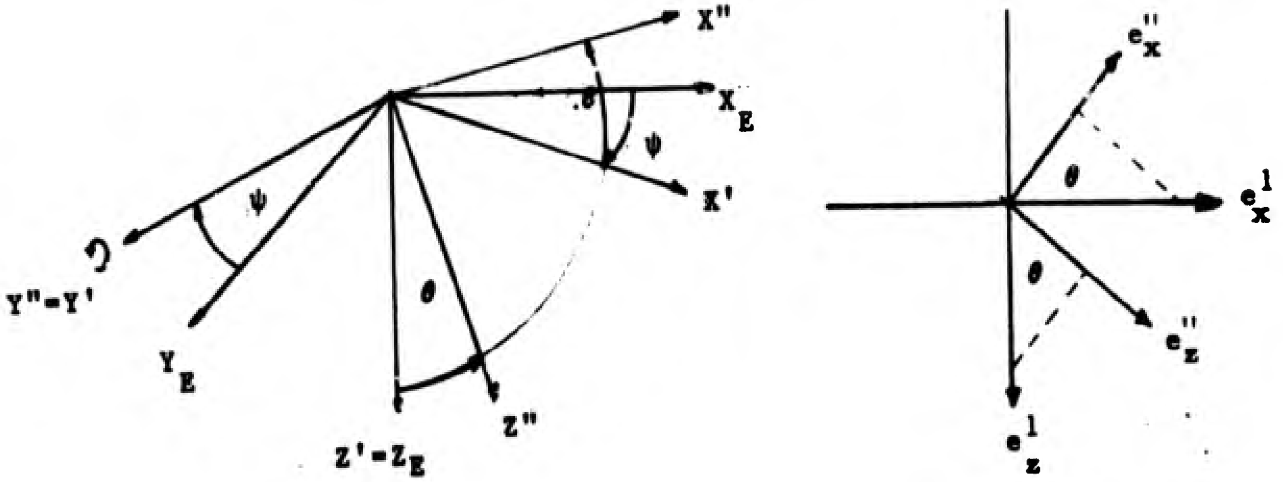
$$e_2' = -\text{SIN}\psi\ e_1 + \text{COS}\psi\ e_2$$

$$e_3' = e_3$$

$$M_1 = \begin{bmatrix} \text{COS}\psi & \text{SIN}\psi & 0 \\ -\text{SIN}\psi & \text{COS}\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Figure A3.2

Second Rotation



$$e_x'' = \text{COS}\,\theta\; e_x^1 - \text{SIN}\,\theta\; e_z^1$$

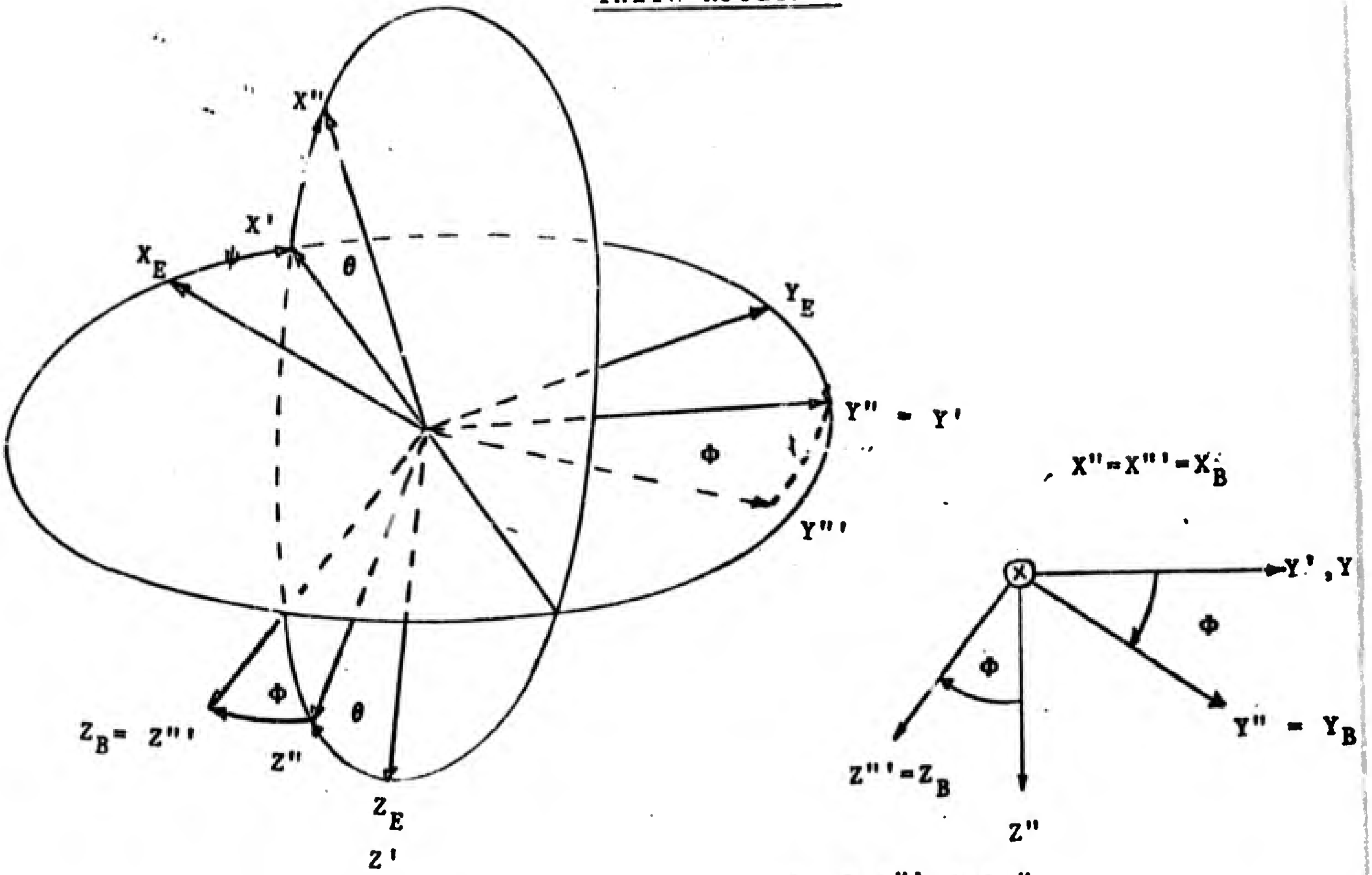
$$e_y'' = e_y^1$$

$$e_z'' = \text{SIN}\,\theta\; e_x^1 + \text{COS}\,\theta\; e_z^1$$

$$M_2 = \begin{bmatrix} \text{COS}\,\theta & 0 & -\text{SIN}\,\theta \\ 0 & 1 & 0 \\ \text{SIN}\,\theta & 0 & \text{COS}\,\theta \end{bmatrix}$$

Figure A3.3

Third Rotation



$$e_{X_B} = e_X''' = e_X''$$

$$e_{Y_B} = e_Y'' = \text{COS}\,\Phi\, e_Y'' + \text{SIN}\,\Phi\, e_Z''$$

$$e_{Z_B} = e_Z''' = -\text{SIN}\,\Phi\, e_Y'' + \text{COS}\,\Phi\, e_Z''$$

$$M_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \text{COS}\,\Phi & \text{SIN}\,\Phi \\ 0 & -\text{SIN}\,\Phi & \text{COS}\,\Phi \end{bmatrix}$$

Figure A3.4

Then, M, as defined in Equation (1), is obtained from the Matrix multiplication:

$$M = M_3\, M_2\, M_1\ .$$

A4 - ALPHABETICAL PROGRAM LISTING AND CROSS INDEX

SUBROUTINE	PAGE NUMBER
RESET	A4-129
SIND	A4-159
STGE2	A4-111
STGE3	A4-113
SUBL1	A4-105
SUBL2	A4-107
SUBL3	A4-109
S1	A4-57
S1I	A4-55
TABLE	A4-131
TABL2	A4-133
TABL3	A4-135
TERROR	A4-155
ZERO	A4-103

```
          PROGRAM MAIN(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,FILMFL)

          PROGRAM MAIN(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,FILMFL)
C
C*****DIMODS TO BE USED WITH FORTRAN A RK INTEGRATION ROUTINE
C
      COMMON    C(3510)              ,GRAPH              ,TEMPS(1000)         DIMC0030
      EQUIVALENCE (C(2662),HMIN   ), (C(2663),HMAX   ), (C(2664),DER    ),
C                 (C(2561),N      ), (C(2562),TOL    ), (C(2465),VAR    ),
C                 (C(2510),T      ), (C(2511),KSTEP  ), (C(2517),STEP   ),
C                 (C(2512),LSTEP  ), (C(2513),PLOTNO ), (C(2609),NCPLOT ),
C                 (C(2323),OPOINT ), (C(2025),TIME   ), (C(2325),VLAPLE ),
C                 (C(3167),NOOUT  ), (C(2522),OPTN10 ), (C(2526),REPPLT ),
C                 (C(2365),EU     ), (C(2765),EL     ), (C(2027),PTLESS )
      EQUIVALENCE                    (C(1971),RITE   ), (C(1972),RKUTTA )
      EQUIVALENCE (C(1973),KASE   ), (C(1974),NJ     ), (C(1975),NPT    )
      DIMENSION   GRAPH(300,15)      , TIME(300)
      DIMENSION   VLAPLE(2,15)       , IPL(100)         , DER(101)
      DIMENSION   VAR(101)           , EL(100)          , EU(100)
      EQUIVALENCE (C(1980),RN     )
      EQUIVALENCE (C(1981),RNT    )
      EQUIVALENCE (C(1982),PLOTN4)
      EQUIVALENCE (C(1983),PLOTN2)
      EQUIVALENCE (C(1984),NPLOT  )
      INTEGER OPOINT
      INTEGER OPT
      EXTERNAL AUXSUB
      CALL COUNTV
 1000 CALL ZERO
 1001 IF(PLOTNO.LE.0.)GOTO7
      IF(REPPLT.GT.0.)GOTO7
C
C                             REPPLT = 0.   USE NEW NO.4,7 (DISCARD OLD)
C                                      1.   USE OLD PLUS THOSE ADDED
C                                     -1.   USE NEW NO. 7 (DISCARD OLD)
      IF (REPPLT.GT.-1.0) NOOUT = 0
      NPLOT=0
    7 CALL CINPT1
      KASE=6                                                              DIM00110
      LSTEP = STEP.
      NPLOT4=PLOTN4
      NPLOT2=PLOTN2
      NOPLOT=PLOTNO                                                       PLOT
 1002 CALL  SUPL1
 1003 CALL  AUXI
 1004 CALL  SUPL2
 1005 DO 60 I = 2,N
      J = IPL(I-1)
      EL(I-1)=C(J+1)
      EU(I-1)=C(J+2)
      VAR(I) = C(J+3)
   60 DER(I) = C(J)
```

```
      PROGRAM MAIN(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,FILMPL)
      VAR(1) = T
      NPT=2
      IF(RKUTTA.GT.0.0) NPT=1
 1006 CALL   AUXSUB
 1007 NJ=N-1                                                            DIMC0185
      CALL AMPK(AUXSUB)                                                 DIMC0187
 1008 DO 50 I = 2, N
      J = IPL(I-1)
   50 C(J+3) = VAR(I)
      T = VAR(1)
 1009 CALL   SUBL3
      IF ( KSTEP .EQ. 1 ) GO TO 1007
      DO 155 JV=2,N
  155 VAR(JV)=0.
      CALL SA
      CALL PROCES
      IF (OPTN13 .GT. 0.) CALL CUMPO
      CALL RESET
      IF(LSTEP.EQ.5.OR.LSTEP.EQ.7.OR.NOPLOT.EQ.0)GOTO5
      CALL TIMEV(DELT)
      WRITE(6,96)DELT
   96 FORMAT(1H ,17HSTART PLOTTING ATF14.7)
      LESSPT=PTLESS
      OPOINT=OPOINT-LESSPT
      CALL PLOT4(GRAPH,OPOINT,VLABLE,TIME,NPLOT4,NPLOT2,NOPLOT)
      CALL PLOT2(GRAPH,OPOINT,VLABLE,TIME,NPLOT4,NPLOT2,NOPLOT)
      CALL PLOTN(GRAPH,OPOINT,VLABLE,TIME,NPLOT4,NPLOT2,NOPLOT)
      CALL TIMEV(DELT)
      WRITE(6,97)DELT
   97 FORMAT(1H ,18HPLOTTING ENDED AT F14.7)
    5 GO TO (1000,1001,1002,1003,1004,1005,1006,1007,1008,1009,1010),
     1 LSTEP
 1010 CONTINUE
      CALL EXIT
      STOP
      END
```

```
I N D E X
               PROGRAM MAIN(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,FILMPL)

  SYMBOL----------------------------------REFERENCES-------------------------------

  5        -   56       58*
  7        -   19       20       23*
  50       -   4400     46*
  60       -   3200     37*
  96       -   58WR     59*
  97       -   66WR     67*
  155      -   5400     51*
  1000     -   18*      68
  1001     -   19*      68
  1002     -   29*      68
  1003     -   30*      68
  1004     -   31*      68
  1005     -   32*      68
  1006     -   41*      68
  1007     -   42*      49       68
  1008     -   44*      68
  1009     -   48*      68
  1010     -   68       69*
  AMPK     -   43
  AUXI     -   36
  AUXSUB   -   16FX     41       43AG
  C        -    2CO      3EQ      4EQ      5EQ      9EQ      10EQ     11EQ     12EQ     13EQ     34
               35       36       37       46=
  COUNTV   -   17
  DFLT     -   57AG     58WR     65AG     66WR
  DEP      -    3EQ      70I     37=
  DUMPO    -   54
  EL       -    3EQ      80I     34=
  EU       -    3EQ      80I     35=
  EXIT     -   70
  GRAPH    -    2CO      60I     62AG     63AG     64AG
  HMAX     -    3EQ
  HMIN     -    3EQ
  I        -   3200     33       34       35       36       37       4400     45       46
  IPL      -    3EQ      70I     33       45
  J        -   33=      34       35       36       37       45=      46
  JV       -   5400     51
  KASE     -    5EQ     24=
  KSTEP    -    3EQ     49
  LESSPT   -   60=      61
  LSTEP    -    3EQ     25=      56       68
  MAIN     -   1
  N        -    3EQ     3200     42       4400     5100
  NJ       -    5EQ     42=
  NOOUT    -    3EQ     21=
  NOPLOT   -    3EQ     28=      56       62AG     63AG     64AG
  NPLOT    -   13EQ     22=
  NPLOT2   -   27=      62AG     63AG     64AG
```

```
I N D E X                                                                                        PAGE   4
            PROGRAM MAIN(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,FILMPL)

   NPLOT4   -     24=     62AG     63AG     64AG
   NPT      -      5EQ    39=      40=
   OINPT1   -     23
   OPOINT   -      3EQ    14IN     61=      62AG     63AG     64AG
   OPT      -     15IO
   OPTN10   -      3EQ    54
   PLOTN    -     64
   PLOTNO   -      3EQ    19       28
   PLOTN2   -     12EQ    27
   PLOTN4   -     11EQ    26
   PLOT2    -     63
   PLOT4    -     62
   PROGES   -     53
   PTLESS   -      3EQ    60
   REPPLT   -      3EQ    20       21
   RESET    -     55
   RITE     -      4EQ
   RKUTTA   -      4EQ    40
   RN       -      9EQ
   RNT      -     10EQ
   STEP     -      3EQ    25
   STOP     -     71
   SUBL1    -     29
   SUBL2    -     31
   SUBL3    -     44
   S8       -     52
   T        -      3EQ    38       47=
   TEMPS    -      2EQ
   TIME     -      3EQ    60I      62AG     63AG     64AG
   TIMEV    -     57      65
   VAR      -      3EQ    30I      36=      38=      46       47       51=
   VLABLE   -      3EQ    70I      62AG     63AG     64AG
   ZERO     -     18

   -+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
                                 BLOCK DATA

      1          BLOCK DATA
      2          COMMON
                *1/NC1/NC1(2)       /NC2/NC2(4)       /NC3/NC3(4)       /NC5/NC5(4)
                *1/CA1/VM1(6)       /CA2/BA2(6),VM2(5)
                *1/CA3/BA3(7),VM3(5)          /CA5/BA5(7),VM5(3)
      3          COMMON  /CMOF/CMO (6)   /CA4/VM4(6)
                *1/CZPF/CZP(35)     /CZ2F/CZ2(35)     /CMPF/CMP(35)     /CM2F/CM2(35)
                *1/CY4F/CY4(36)     /CN4F/CN4(36)     /CL4F/CL4(21)     /CL2F/CL2(21)
                *1/CZ0F/CZ01(35)                      /CM0F/CM01(35)
                *1/CHOF/CHO(36)     /CLPF/CLP(36)     /CLOF/CL01(21)
                *1/CXOF/CXO(6)
      4          DATA NC1/
      5          DATA NC2/
      6          DATA NC3/
      7          DATA NC5/
      8          DATA VM1/
      9          DATA VM2/
     10          DATA VM3/
     11          DATA VM4/
     12          DATA VM5/
     13          DATA BA2/
     14          DATA BA3/
     15          DATA BA5/
     16          DATA CXO/
     17          DATA CMO                                                      CM000010
     18          DATA CZP/
                                                                               CZ000010
                                                                               CZ000020
                                                                               CZ000030
                                                                               CZ000040
                                                                               CZ000050
     19          DATA CMP/
                                                                               CM000010
                                                                               CM000020
                                                                               CM000030
                                                                               CM000040
                                                                               CM000050
     20          DATA CZ01/
                                                                               CZ000010
                                                                               CZ000020
                                                                               CZ000030
                                                                               CZ000040
                                                                               CZ000050
     21          DATA CM01/
                                                                               CM000010
                                                                               CM000020
                                                                               CM000030
                                                                               CM000040
                                                                               CM000050
```

```
 I N D E X                                                               PAGE   6
                                  BLOCK DATA
        22       DATA CLM1/
                                                                         CLMP0010
                                                                         CLMP0030
                                                                         CLMF0050
        23       DATA CY4/
                                                                         DCY40010
                                                                         DCY40020
                                                                         DCY40030
                                                                         DCY40040
                                                                         DCY40050
                                                                         DCY40060
        24       DATA CN4/
                                                                         DCN40010
                                                                         DCN40020
                                                                         DCN40030
                                                                         DCN40040
                                                                         DCN40050
                                                                         DCN40060
        25       DATA CL4/
                                                                         DCL40010
                                                                         DCL40020
                                                                         DCL40030
        26       DATA CL2/
                                                                         DCL20010
                                                                         DCL20020
                                                                         DCL20030
        27       DATA CZ2/
                                                                         DCZ20010
                                                                         DCZ20020
                                                                         DCZ20030
                                                                         DCZ20040
                                                                         DCZ20050
        28       DATA CM2/
                                                                         DCM20010
                                                                         DCM20020
                                                                         DCM20030
                                                                         DCM20040
                                                                         DCM20050
        29       DATA CMQ/
                                                                         CMQ00010
                                                                         CMQ00020
                                                                         CMQ00030
                                                                         CMQ00040
                                                                         CMQ00050
                                                                         CMQ00060
        30       DATA CLP/
                                                                         CLP00010
                                                                         CLP00020
                                                                         CLP00030
                                                                         CLP00040
```

```
                                                          CLP00050
     31          END                                      CLP00060
```

```
I N D E X                                                                  PAGE   8
                                          BLOCK DATA

     SYMBOL                                 REFERENCES

     BA2       -     2CO      130A
     BA3       -     2CO      140A
     BA5       -     2CO      150A
     BLDATA    -     1
     CA1       -     2
     CA2       -     2
     CA3       -     2
     CA4       -     3
     CA5       -     2
     CLDF      -     3
     CLD1      -     3CO      220A
     CLP       -     3CO      300A
     CLPF      -     3
     CL2       -     3CO      260A
     CL2F      -     3
     CL4       -     3CO      250A
     CL4F      -     3
     CMDF      -     3
     CMD1      -     3CO      210A
     CMO       -     3CO      170A
     CMOF      -     3
     CMP       -     3CO      190A
     CMPF      -     3
     CMQ       -     3CO      290A
     CMQF      -     3
     CM2       -     3CO      280A
     CM2F      -     3
     CN4       -     3CO      240A
     CN4F      -     3
     CXO       -     3CO      160A
     CXOF      -     3
     CY4       -     3CO      230A
     CY4F      -     3
     CZDF      -     3
     CZD1      -     3CO      200A
     CZP       -     3CO      180A
     CZPF      -     3
     CZ2       -     3CO      270A
     CZ2F      -     3
     NC1       -     2         40A
     NC2       -     2         50A
     NC3       -     2         60A
     NC5       -     2         70A
     VM1       -     2CO       80A
     VM2       -     2CO       90A
     VM3       -     2CO      100A
     VM4       -     3CO      110A
     VM5       -     2CO      120A
```

```
-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
            SUBROUTINE G2                                                    A01
      C**WIND AND GUSTS MODULE
            COMMON C(3510)
      C**INPUT DATA
            EQUIVALENCE(C(2000),T)
            EQUIVALENCE (C(  50),OPTNW )
            EQUIVALENCE (C(  51),BPSIW )
            EQUIVALENCE (C(  52),VWTE  )
            EQUIVALENCE (C(  53),RHW   )
            EQUIVALENCE (C( 380),RANGO )
            EQUIVALENCE (C(1317),RAIL  )
      C**OUTPUT DATA
            EQUIVALENCE (C( 100),VWXE  )
            EQUIVALENCE (C( 101),VWYE  )
            EQUIVALENCE (C( 102),VWZE  )
      C**INPUTS FROM OTHER MODULES
            EQUIVALENCE (C( 209),RH    )
      C**CALCULATE WIND VELOCITY COMPONENTS
            IF ((OPTNW .LE. 0.) .OR. (RH .GT. RHW)) GO TO 10
            IF (RANGO .LT. RAIL) GO TO 10
            VWXE = -VWTE*COSD(BPSIW)
            VWYE = -VWTE*SIND(BPSIW)
            VWZE = 0.
            RETURN
         10 VWXE = 0.
            VWYE = 0.
            VWZE = 0.
            RETURN
            END
```

```
I N D E X                                                                          PAGE  10
                                        SUBROUTINE G2

SYMBOL      - - - - - - - - - - - - - - - - REFERENCES - - - - - - - - - - - - - - - -

10         -     14      15      20*
RPSIW      -     5EQ     16      17
C          -     7EQ     3EQ     4EQ     5EQ     6EQ     7EQ     8EQ     9EQ     10EQ    11EQ
                 12EQ    13EQ
COSD       -     16
G2         -     1
OPTNW      -     4EQ     14
RAIL       -     9EQ     15
RANGO      -     8EQ     15
RETURN     -     19      23
RW         -     13EQ    14
*RHW       -     7EQ     14
SIND       -     17
T          -     3EQ
VWTE       -     6EQ     16      17
VWXE       -     10EQ    16=     20=
VWYE       -     11EQ    17=     21=
VWZE       -     12EQ    18=     22=
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

```
      SUBROUTINE G3
C**AIR DATA MODULE G3
      COMMON C(3510)
C**INPUT DATA
      EQUIVALENCE (C(0208),RHZRO )
C**INPUTS FROM OTHER MODULES
      EQUIVALENCE (C(0100),VMXE  )
      EQUIVALENCE (C(0101),VMYE  )
      EQUIVALENCE (C(0102),VMZE  )
      EQUIVALENCE (C(1603),VXE   )
      EQUIVALENCE (C(1607),VYE   )
      EQUIVALENCE (C(1611),VZE   )
      EQUIVALENCE (C(1623),RZE   )
C**INPUTS FROM MAIN PROGRAM
C**STATE VARIABLE OUTPUTS
C**NONE
C**OTHER OUTPUTS
      EQUIVALENCE (C(0200),VMMXE )
      EQUIVALENCE (C(0201),VMMYE )
      EQUIVALENCE (C(0202),VMMZE )
      EQUIVALENCE (C(0203),PDYNMC)
      EQUIVALENCE (C(0204),VMACH )
      EQUIVALENCE (C(0205),DRHO  )
      EQUIVALENCE (C(0206),VSOUND)
      EQUIVALENCE (C(0207),VAIRSP)
      EQUIVALENCE (C(0209),RH    )
C**CALCULATE PRESENT ALTITUDE
      RH= -RZE+RHZRO
C**CALCULATE MISSILE VELOCITY WRT AIR MASS IN EARTH AXES
      VMMXE = VXE-VMXE
      VMMYE = VYE-VMYE
      VMMZE = VZE-VMZE
      VAIRSP = SQRT(VMMXE*VMMXE+VMMYE*VMMYE+VMMZE*VMMZE)
C**AIR DENSITY, SPEED OF SOUND, DYNAMIC PRESSURE, AND MACH
      DRHO=(.076475)/(1.+.3325E-04*RH+RH*RH*RH*.02315E-12)
      VSOUND = -.00392*RH+1117.3
      PDYNMC = (DRHO*VAIRSP*VAIRSP)/64.344
      VMACH = VAIRSP/VSOUND
      RETURN
      END
```

```
I N D E X                              SUBROUTINE G3

SYMBOL    - - - - - - - - - - - - - - - - REFERENCES - - - - - - - - - - - - - - - - - -

C          -    2EQ    3EQ    4EQ    5EQ    6EQ    7EQ    8EQ    9EQ    10EQ    11EQ
                12EQ   13EQ   14EQ   15EQ   16EQ   17EQ   18EQ   19EQ
DPHO       -    14EQ   25=    27
G3         -    1
PDYNMC     -    16EQ   27=
RETURN     -    29
RH         -    19EQ   23=    25     26
RHZRO      -    3EQ    20
RZE        -    10EQ   20
SQRT       -    24
VAIRSP     -    18EQ   24=    27     28
VMACH      -    15EQ   28=
VMWXE      -    11EQ   21=    24
VMWYE      -    12EQ   22=    24
VMWZE      -    13EQ   23=    24
VSOUND     -    17EQ   26=    28
VWXE       -    4EQ    21
VWYE       -    5EQ    22
VWZE       -    6EQ    23
VXE        -    7EQ    21
VYE        -    8EQ    22
VZE        -    9EQ    23

-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
    1            SUBROUTINE GS
           C**COORDINATE CONVERSION MODULE
    2            COMMON  C(3510)
           C
           C**INPUTS FROM OTHER MODULES
    3            EQUIVALENCE (C(0200),VMWXE )
    4            EQUIVALENCE (C(0201),VMWYE )
    5            EQUIVALENCE (C(0202),VMWZE )
    6            EQUIVALENCE (C(0277),VAIRSP)
    7            EQUIVALENCE (C(1317),RAIL  )
    8            EQUIVALENCE (C(1405),QPURN )
    9            EQUIVALENCE (C(1603),VXE   )
   10            EQUIVALENCE (C(1607),VYE   )
   11            EQUIVALENCE (C(1611),VZE   )
   12            EQUIVALENCE (C(1515),PXE   )
   13            EQUIVALENCE (C(1519),RYE   )
   14            EQUIVALENCE (C(1523),RZE   )
   15            EQUIVALENCE (C(1535),RDELX )
   16            EQUIVALENCE (C(1536),RDELY )
   17            EQUIVALENCE (C(1537),RDELZ )
   18            EQUIVALENCE (C(1651),RTXE  )
   19            EQUIVALENCE (C(1655),RTYE  )
   20            EQUIVALENCE (C(1659),RTZE  )
   21            EQUIVALENCE (C(1668),PXO   )
   22            EQUIVALENCE (C(1669),RYO   )
   23            EQUIVALENCE (C(1670),RZO   )
   24            EQUIVALENCE (C(1671),VXO   )
   25            EQUIVALENCE (C(1672),VYO   )
   26            EQUIVALENCE (C(1673),VZO   )
   27            EQUIVALENCE (C(1703),CFA11 )
   28            EQUIVALENCE (C(1707),CFA12 )
   29            EQUIVALENCE (C(1711),CFA13 )
   30            EQUIVALENCE (C(1715),CFA21 )
   31            EQUIVALENCE (C(1719),CFA22 )
   32            EQUIVALENCE (C(1723),CFA23 )
   33            EQUIVALENCE (C(1727),CFA31 )
   34            EQUIVALENCE (C(1731),CFA32 )
   35            EQUIVALENCE (C(1735),CFA33 )
   36            EQUIVALENCE (C(1751),CRAD  )
   37            EQUIVALENCE (C(1755),AC21  )
   38            EQUIVALENCE (C(1756),AC22  )
   39            EQUIVALENCE (C(1757),AC23  )
   40            EQUIVALENCE (C(1758),AC31  )
   41            EQUIVALENCE (C(1759),AC32  )
   42            EQUIVALENCE (C(1760),AC33  )
   43            EQUIVALENCE (C(2809),T     )
           C
           C**OTHER OUTPUTS
   44            EQUIVALENCE (C(0350),BTHT  )
   45            EQUIVALENCE (C(0351),BPSI  )
```

```
**** 7000 SCOPE 2.0 LVL  53T  04/28/73  12.35.47  05/29/73        73149

************ ADMCA RESEARCH CENTER ************

HH.MM.SS OEU SECOND ORIGIN
         17.05.1**CCP.LOC.   SCOPE 3.4            SOC ADTFC 03/2//73
17.05.09 00000.022 JOB.     -YY733,CM160000,CP70,T550.                          STPS 1          12
17.05.00 00000.022 JOB.     -FILE(TAPE5,RT=Z,BT=C)
17.05.00 00000.027 JOB.     -STAGE(TAPE5,HY,VSN=S34191,ST=CCP)
17.05.00 00000.524 LOD.     -FTN.
17.06.35 00002.840 USR.           2.707 CP SECONDS COMPILATION TIME
17.06.35 00002.841 LOD.     -LGO(PL=7777777)
17.06.08 00002.885 SYS.       LDF19 - FLS REQUIRED TO LOAD - 019152 OU.COG
17.06.08 00002.897 SYS.       LDC03 - EXECUTION INITIATED 08.EXP
17.06.08 00002.898 USR.      FORTRAN LIBRARY 53                03/05/73
17.06.38 00002.899 SYS.       JM260 - STAGE MT IN  LFN=TAPE5   VSN=S34191
       . 17.07.02 CCP.      MT22 BLOCKS READ -     00000067
         17.07.02 CCP.        JM517 - BLOCKS READ-             12
         17.07.02 CCP.        JM511 - WORDS  READ-          61425
17.07.29 00014.623 USR.         STOP
17.07.30 00014.626 SYS.       RM770 - MAXIMUM ACTIVE FILES                         5
17.07.30 00014.626 SYS.       RM771 - OPEN/CLOSE CALLS                           113
17.07.30 00014.626 SYS.       RM772 - DATA TRANSFER CALLS                     27,647
17.07.30 00014.627 SYS.       RM773 - CONTROL/POSITIONING CALLS                  192
17.07.30 00014.627 SYS.       RM774 - BM DATA TRANSFER CALLS                   1,524
17.07.30 00014.627 SYS.       RM775 - BM CONTROL/POSITIONING CALLS               102
17.07.30 00014.627 SYS.       RM776 - QUEUE MANAGER CALLS                        265
17.07.30 00014.627 SYS.       RM777 - RECALL CALLS                               204
17.07.30 00014.628 SYS.       SCM                   848.032  KWS
17.07.30 00014.628 SYS.       I/O                     0.162  MW
17.07.30 00014.628 SYS.       PMS                     6.151  MWS
17.07.30 00014.628 SYS.       USER                   11.759  SEC
17.07.30 00014.628 SYS.       JOB                    14.630  SEC
17.07.30 00014.628 SYS.       SC050 - 000372 SC/LC SWAPS
```

OMIT !

```
************          YYZ33EM  //// END OF LIST ////
************          YYZ33EM  //// END OF LIST ////
```

```
 46         EQUIVALENCE (C(0352),BPHI  )
 47         EQUIVALENCE (C( 353),BPH1  )
 48         EQUIVALENCE (C( 354),BTH2  )
 49         EQUIVALENCE (C( 355),BPS1  )
 50         EQUIVALENCE (C(0356),VTOTE )
 51         EQUIVALENCE (C(0357),BGAMH )
 52         EQUIVALENCE (C(0358),BGAMV )
 53         EQUIVALENCE (C(0363),BTHLV )
 54         EQUIVALENCE (C(0364),BPSLV )
 55         EQUIVALENCE (C(0365),BLAMV )
 56         EQUIVALENCE (C(0366),BLAMH )
 57         EQUIVALENCE (C(0367),BALPHA)
 58         EQUIVALENCE (C(0368),BALPHY)
 59         EQUIVALENCE (C(0369),BALPHP)
 60         EQUIVALENCE (C(0370),BPHIP )
 61         EQUIVALENCE (C(0371),RANGE )
 62         EQUIVALENCE (C(0372),RXBA  )
 63         EQUIVALENCE (C(0373),RYBA  )
 64         EQUIVALENCE (C(0374),RZBA  )
 65         EQUIVALENCE (C( 380),RANGO )
 66         EQUIVALENCE (C( 390),RXL   )
 67         EQUIVALENCE (C( 391),RYL   )
 68         EQUIVALENCE (C( 392),RZL   )
 69         EQUIVALENCE (C( 393),BPH2  )
      C
      C**CALCULATION OF HEADING, PITCH, ROLL EULER ANGLES IN DEGREES
 70         BPHI = ATAND(CFA23,CFA33)
 71         BTHT = ATAND(-CFA13,SQRT(CFA11*CFA11+CFA12*CFA12))
 72         BPSI = ATAND(CFA12,CFA11)
      C
      C**FREE GYRO MODELS (INITIAL GIMBAL ANGLES ARE ZERO)
 73         B21 = A021*CFA11 + A022*CFA12 + A023*CFA13
 74         B22 = A021*CFA21 + A022*CFA22 + A023*CFA23
 75         B23 = A021*CFA31 + A022*CFA32 + A023*CFA33
 76         B31 = A031*CFA11 + A032*CFA12 + A033*CFA13
 77         B32 = A031*CFA21 + A032*CFA22 + A033*CFA23
 78         B33 = A031*CFA31 + A032*CFA32 + A033*CFA33
 79         BPH1 = ATAND( B23,B22)
 80         BPS1 = ATAND(-B21,B22/COSD(BPH1))
 81         BTH2 = ATAND( B31,B33)
 82         BPH2 = ATAND(-B32,B33/COSD(BTH2))
      C
      C**CALCULATION OF TOTAL VELOCITY
 83         VTOTE = SQRT(VXE*VXE+VYE*VYE+VZE*VZE)
 84         RDELX = RTXE-RXE
 85         RDELY = RTYE-RYE
 86         RDELZ = RTZE-RZE
      C
 87         RXL = RXE - RXO - VXO*T
 88         RYL = RYE - RYO - VYO*T
```

```
         RZL = RZE - RZO - VZO*T
         RANGO = SQRT(RXL**2 + RYL**2 + RZL**2)
C
C**TRANSFORM MISSILE LOS FROM EARTH TO BODY AXES
         RXBA = RDELX*CFA11 + RDELY*CFA12 + RDELZ*CFA13
         RYBA = RDELX*CFA21 + RDELY*CFA22 + RDELZ*CFA23
         RZBA = RDELY*CFA31 + RDELY*CFA32 + RDELZ*CFA33
C
         UVP1 =  VXE*RDELX+VYE*RDELY
         UVP2 = RDELY*RDELX+RDELY*RDELY
         UVP3 = VZE*RDELZ
         UVP4 = SQRT(UVP2)
         RANGE = SQRT(UVP2+RDELZ**2)
C**VERTICAL AND HORIZONTAL LINE OF SIGHT ANGLES (EARTH AXES)
C
         BLAMH = ATAND(-RDELY,RDELX)
         BLAMV = ATAND(-RDELZ,UVP4)
C
C**VERTICAL AND HORIZONTAL PROPORTIONAL NAVIGATION ANGLES
         IF (VTOTE .EQ. 0.) GO TO 30
         VXP=(UVP1+UVP3)/RANGE
         VYP = (VYE*RDELX-VXE*RDELY)/UVP4
         VZP = (VZE*UVP2-RDELZ*UVP1)/(RANGE*UVP4)
         BTHLV  = ATAND(VZP,VXP)
         BPSLV  = ATAND(VYP,VXP)
C
         BGAMV = ATAND(-VZE,SQRT(VXE*VXE+VYE*VYE))
         BGAMH = ATAND(VYE,VXE)
C
C**VELOCITY WRT AIR IN BODY AXES
         VMWU = CFA11*VMWXE+CFA12*VMWYE+CFA13*VMWZE
         VMWV = CFA21*VMWXE+CFA22*VMWYE+CFA23*VMWZE
         VMWW = CFA31*VMWXE+CFA32*VMWYE+CFA33*VMWZE
C
C**VERTICAL AND HORIZONTAL ANGLES OF ATTACK
         IF (DBURN.LE.0. .AND. RANGO.LE.RAIL) GO TO 30
         BALPHA = ATAND(VMWW,VMWU)
         BALPHY = ATAND(VMWV,VMWU)
C
C**ALPHA PRIME AND PHI PRIME (WIND TUNNEL AXES)
         IF ((BALPHA-BALPHY).EQ.0.) GO TO 30
         BPHIP = ATAND(BALPHY,BALPHA)
      30 BALPHP=SQRT(BALPHA**2+BALPHY**2)
         RETURN
         END
```

```
I N D E X                                                                          PAGE  16
                                         SUBROUTINE G5

SYMBOL  - - - - - - - - - - - - - - - -  REFERENCES - - - - - - - - - - - - - - - -

TC       -    101     112     115     117*
ATANO    -    70      71      72      79      80      81      82      99     100     105
              106     107     108     113     114     116
AG21     -    37EQ    73      74      75
AG22     -    38EQ    73      74      75
AG23     -    39EQ    73      74      75
AG31     -    40EQ    76      77      78
AG32     -    41EQ    76      77      78
AG33     -    42EQ    76      77      78
BALPHA   -    57EQ    113=    115     116     117
BALPHP   -    59EQ    117=
BALPHY   -    58EQ    114=    115     116     117
BGAMH    -    51EQ    108=
BGAMV    -    52EQ    107=
BLAMH    -    56EQ    99=
BLAMV    -    55EQ    100=
BPHI     -    46EQ    70=
BPHIP    -    60EQ    116=
BPH1     -    47EQ    79=     80
BPH2     -    69EQ    82=
BPSI     -    45EQ    72=
BPSLV    -    54EQ    106=
BPS1     -    49EQ    80=
BTHLV    -    53EQ    105=
BTHT     -    44EQ    71=
BTH2     -    48EQ    81=     82
B21      -    73=     80
B22      -    74=     79      80
B23      -    75=     79
B31      -    76=     81
B32      -    77=     82
B33      -    78=     81      82
C        -    2CO     3EQ     4EQ     5EQ     6EQ     7EQ     8EQ     9EQ    10EQ    11EQ
              12EQ    13EQ    14EQ    15EQ    16EQ    17EQ    18EQ    19EQ    20EQ    21EQ
              22EQ    23EQ    24EQ    25EQ    26EQ    27EQ    28EQ    29EQ    30EQ    31EQ
              32EQ    33EQ    34EQ    35EQ    36EQ    37EQ    38EQ    39EQ    40EQ    41EQ
              42EQ    43EQ    44EQ    45EQ    46EQ    47EQ    48EQ    49EQ    50EQ    51EQ
              52EQ    53EQ    54EQ    55EQ    56EQ    57EQ    58EQ    59EQ    60EQ    61EQ
              62EQ    63EQ    64EQ    65EQ    66EQ    67EQ    68EQ    69EQ
CFA11    -    27EQ    71      72      73      76      91     109
CFA12    -    28EQ    71      72      73      76      91     109
CFA13    -    29EQ    71      73      76      91     109
CFA21    -    30EQ    74      77      92     110
CFA22    -    31EQ    74      77      92     110
CFA23    -    32EQ    70      74      77      92     110
CFA31    -    33EQ    75      78      93     111
CFA32    -    34EQ    75      78      93     111
CFA33    -    35EQ    70      75      78      93     111
```

```
I N D E X
                                      SUBROUTINE G5

COSD      -     80      82
CRAD      -     76EQ
G5        -      1
QBURN     -      8EQ   112
RAIL      -      7EQ   112
RANGE     -     61EQ    99=    102     104
RANGO     -     65EQ    99=    112
RDELX     -     15EQ    84=     91      92      93      94      95      99     103
RDELY     -     16EQ    85=     91      92      93      94      95      99     103
RDELZ     -     17EQ    86=     91      92      93      96      98     100     104
RETURN    -    118
RTXE      -     18EQ    84
RTYE      -     19EQ    85
RTZE      -     20EQ    86
RYPA      -     62EQ    91=
RXE       -     12EQ    84      87
RXL       -     66EQ    87=     90
RXO       -     21EQ    87
RYHA      -     63EQ    92=
RYE       -     13EQ    85      88
RYL       -     67EQ    88=     90
RYO       -     22EQ    88
RZPA      -     64EQ    93=
RZE       -     14EQ    86      89
RZL       -     68EQ    89=     90
RZO       -     23EQ    89
SQRT      -     71      83      90      97      98     107     117
T         -     43EQ    87      88      89
UVP1      -     94=    102     104
UVP2      -     95=     97      98     104
UVP3      -     96=    102
UVP4      -     97=    100     103     104
VAIRSP    -      6EQ
VMWU      -    109=    113     114
VMWV      -    110=    114
VMWW      -    111=    113
VMWXE     -      3EQ   109     110     111
VMWYE     -      4EQ   109     110     111
VMWZE     -      5EQ   109     110     111
VTOTE     -     56EQ    83=    101
VXE       -      9EQ    83      94     103     107     108
VXO       -     24EQ    87
VXP       -    102=    105     106
VYE       -     10EQ    83      94     103     107     108
VYO       -     25EQ    88
VYP       -    103=    106
VZE       -     11EQ    83      96     104     107
VZO       -     26EQ    89
VZP       -    104=    105

-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
       1         SUBROUTINE A1
       2         COMMON C(3512)
           C
           C**TABLE LOOKUP FOR AERO COEF
       3         COMMON
                *//NC1/NC1(2)      /NC2/NC2(4)      /NC3/NC3(4)      /NC5/NC5(4)
                */CA1/CA1(6)       /CA2/CA2(12)     /CA3/CA3(12)     /CA5/CA5(10)
                */CZPF/CZPF(35)    /CZ2F/CZ2F(35)   /CMPF/CMPF(35)   /CM2F/CM2F(35)
                */CY4F/CY4F(36)    /CN4F/CN4F(36)   /CL4F/CL4F(21)   /CL2F/CL2F(21)
                */CZOF/CZOF(35)                     /CMOF/CMOF(35)
                */CMQF/CMQF(36)    /CLPF/CLPF(36)   /CLOF/CLOF(21)
                */CXOF/CXOF(6)
       4         COMMON  /CMOF/CMOF(6)  /CA4/CA4(6)
           C
           C**INPUTS FROM OTHER MODULES
       5         EQUIVALENCE (C(0234),VMACH )
       6         EQUIVALENCE (C(0367),RALPHA)
       7         EQUIVALENCE (C(0368),RALPHY)
       8         EQUIVALENCE (C(0369),RALPHP)
       9         EQUIVALENCE (C(0373),RPHIP )
      10         EQUIVALENCE (C(1103),RDELT1)
      11         EQUIVALENCE (C(1107),RDELT2)
      12         EQUIVALENCE (C(1111),RDELT3)
      13         EQUIVALENCE (C(1115),BDELT4)
      14         EQUIVALENCE (C(1551),OPT4  )
      15         EQUIVALENCE (C(1555),UDL1  )
      16         EQUIVALENCE (C(1556),UDL2  )
      17         EQUIVALENCE (C(1557),UDL3  )
      18         EQUIVALENCE (C(1558),UDL4  )
           C
           C**INPUTS FROM MAIN PROGRAM
      19         EQUTVALENCE (C(2300),T     )
      20         EQUIVALENCE (C(2320),LCONV)
           C
           C**OUTPUT TO MODULES
      21         EQUIVALENCE (C(1202),OPTHNG)
      22         EQUIVALENCE (C(1203),CX    )
      23         EQUIVALENCE (C(1204),CY    )
      24         EQUIVALENCE (C(1205),CZ    )
      25         EQUIVALENCE (C(1206),CLP   )
      26         EQUIVALENCE (C(1207),CMQ   )
      27         EQUTVALENCE (C(1208),CNR   )
      28         EQUIVALENCE (C(1209),CL    )
      29         EQUIVALENCE (C(1210),CM    )
      30         EQUIVALENCE (C(1211),CN    )
           C
           C**OTHER OUTPUTS
      31         EQUIVALENCE (C(1212),CXO   )
      32         EQUIVALENCE (C(1213),CZO   )
      33         EQUIVALENCE (C(1214),DCZZ  )
```

```
      EQUIVALENCE (C(1215),C7DQ  )
      EQUIVALENCE (C(1216),C7DR  )
      EQUIVALENCE (C(1217),DCY4  )
      EQUIVALENCE (C(1218),CM0   )
      EQUIVALENCE (C(1219),DCM2  )
      EQUIVALENCE (C(1220),CMDQ  )
      EQUIVALENCE (C(1221),CMDR  )
      EQUIVALENCE (C(1222),DCN4  )
      EQUIVALENCE (C(1223),DCL1  )
      EQUIVALENCE (C(1224),DCL4  )
      EQUIVALENCE (C(1225),CLDP  )
      EQUIVALENCE (C(1226),VM    )
      EQUIVALENCE (C(1227),BAP   )
      EQUIVALENCE (C(1228),BDL   )
      EQUIVALENCE (C(1229),BDM   )
      EQUIVALENCE (C(1230),BDN   )
      EQUIVALENCE (C(1231),BDP   )
      EQUIVALENCE (C(1232),BDQ   )
      EQUIVALENCE (C(1233),BDR   )
      EQUIVALENCE (C(1236),CH1   )
      EQUIVALENCE (C(1237),CH2   )
      EQUIVALENCE (C(1238),CH3   )
      EQUIVALENCE (C(1239),CH4   )
      EQUIVALENCE (C(1240),CH11  )
      EQUIVALENCE (C(1241),CH21  )
      EQUIVALENCE (C(1242),CH31  )
      EQUIVALENCE (C(1243),CH41  )
C
      DATA IA,IB,IC,ID,IE,IF /7,6,4,5,8,3/
C MULTIPLE ANGLE FORMULAE AND ABSOLUTE VALUES OF ANGLE OF ATTACK
      USPHI = SIND(BPHIP)
      UCPHI = COSD(BPHIP)
      US2PHI = SIND(2.*BPHIP)
      US4PHI = SIND(4.*BPHIP)
      US2PH2 = US2PHI**2
C
C**LIMIT TABLE ARGUMENTS
      BDP = (-BDELT1 - BDELT2 + BDELT3 + BDELT4)/4.
      BDQ = ( BDELT1 + BDELT2 + BDELT3 + BDELT4)/4.
      BDR = (-BDELT1 + BDELT2 - BDELT3 + BDELT4)/4.
      IF (OPTN .LE. 0.) GO TO 15
      BDP = (-UDL1-UDL2+UDL3+UDL4)/4.
      BDQ = (+UDL1+UDL2+UDL3+UDL4)/4.
      BDR = (-UDL1+UDL2-UDL3+UDL4)/4.
   15 CONTINUE
      BDL = AMAX1(10.,ABS(BDP))
      BDM = AMAX1(10.,ABS(BDQ))
      BDN = AMAX1(10.,ABS(BDR))
      BAP = BALFHP
    * UAL = ABS(BALFHA)
```

```
      URT = ABS(RALPHY)
      VM = VMACH
      IF (RAP.GT.20.)RAP=20.
      IF (UAL.GT.20.)UAL=20.
      IF (URT.GT.20.)URT=20.
      IF (VM.GT.1.25) VM = 1.25
      IF (VM.LT..5) VM = .5
C
C**TABLE LOOKUP FOR AERO COEF
      IF (T.GT.0..AND.T.LE.UT) GO TO 1330
      UT = T
      CALL TABLE(     VM,CA1,CX0F,NC1,XF,4HCX0 ,CX0 )
      CALL TABLE(     VM,CA4,CM0F,NC4,XF,4HCM0 ,CM0 )
      XF=0.
      CALL TABL2(RAP,VM,CA3,CZ0F,NC3,XF,4HCZ0 ,CZ0 )
      CALL TABL2(RAP,VM,CA3,CMFF,NC3,XF,4HCMAG ,CM0 )
      CALL TABL2(RAP,VM,CA3,CM2F,NC3,XF,4HDCM2,DCM2)
      CALL TABL2(RAP,VM,CA3,CZDF,NC3,XF,4HCZDQ,CZDQ)
      CALL TABL2(RAP,VM,CA3,CZDF,NC3,XF,4HCZDR,CZDR)
      CALL TABL2(RAP,VM,CA3,CMDF,NC3,XF,4HCMDQ,CMDQ)
      CALL TABL2(RAP,VM,CA3,CMDF,NC3,XF,4HCMDR,CMDR)
      XF=0.
      CALL TABL2(RAP,VM,CA2,CY4F,NC2,XF,4HDCY4,DCY4)
      CALL TABL2(RAP,VM,CA2,CN4F,NC2,XF,4HDCN4,DCN4)
      XF=0.
      CALL TABL2(UAL,VM,CA2,CM0F,NC2,XF,4HCM0 ,CM0 )
      XF=0.
      CALL TABL2(URT,VM,CA2,CM0F,NC2,XF,4HCNP ,CNP )
      XF=0.
      CALL TABL2(RAP,VM,CA5,CL0F,NC5,XF,4HCLDP,CLDP)
      CALL TABL2(RAP,VM,CA5,CL4F,NC5,XF,4HDCL4,DCL4)
      CALL TABL2(RAP,VM,CA5,CL2F,NC5,XF,4HDCL2,DCL2)
      CM0 = CM0 - CMC
 1330 CONTINUE
C
C**AERO COEF WIND AXIS
      CZP = CZ0 + DCZ2*US2PH2 + CZDQ*UCPHI*BDQ - CZDR*USPHI*BDR
      CMP = CM0 + DCM2*US2PH2 + CMDQ*UCPHI*BDQ - CMDR*USPHI*BDR
      CNP =       DCN4*US4PHI + CMDQ*USPHI*BDQ + CMDR*UCPHI*BDR
      CYP =       DCY4*US4PHI + CZDQ*USPHI*BDQ + CZDR*UCPHI*BDR
C
C** TRANSFORMATION FROM WIND TO BODY AXIS
      CX = CX0
      CL = DCL2*US2PHI+ DCL4*US4PHI + CLDP*BDP
      CY =  CYP*UCPHI - CZP*USPHI
      CZ =  CYP*USPHI + CZP*UCPHI
      CN =  CNP*UCPHI - CMP*USPHI
      CM =  CNP*USPHI + CMP*UCPHI + CM0
      RETURN
      END
```

```
I N D E X                                                                         PAGE  21
                                          SUBROUTINE A1

SYMBOL    - - - - - - - - - - - - - - - - - - REFERENCES - - - - - - - - - - - - - - - - - - -

15        -    70      74*
1000      -    67      111*
ABS       -    75      76      77      79      80
AMAX1     -    75      76      77
A1        -     1
BALPHA    -     6EQ    79
BALPHP    -     8EQ    78
BALPHY    -     7EQ    80
BAP       -    46EQ    78=     82      92AG    93AG    94AG    95AG    96AG    97AG    98AG
              100AG   101AG   107AG   108AG   109AG
BDELT1    -    10EQ    67      68      69
BDELT2    -    11EQ    67      68      69
BDELT3    -    12EQ    67      68      69
BDELT4    -    13EQ    67      68      69
BDL       -    47EQ    75=
BDM       -    48EQ    76=
BDN       -    49EQ    77=
BDP       -    50EQ    67=     71=     75      117
BDQ       -    51EQ    68=     72=     76      112     113     114     115
BDR       -    52EQ    69=     73=     77      112     113     114     115
BPHIP     -     9EQ    62      63      64      65
C         -    2C0      5EQ     6EQ     7EQ     8EQ     9EQ    10EQ    11EQ    12EQ    13EQ
               14EQ    15EQ    16EQ    17EQ    18EQ    19EQ    20EQ    21EQ    22EQ    23EQ
               24EQ    25EQ    26EQ    27EQ    28EQ    29EQ    30EQ    31EQ    32EQ    33EQ
               34EQ    35EQ    36EQ    37EQ    38EQ    39EQ    40EQ    41EQ    42EQ    43EQ
               44EQ    45EQ    46EQ    47EQ    48EQ    49EQ    50EQ    51EQ    52EQ    53EQ
               54EQ    55EQ    56EQ    57EQ    58EQ    59EQ    60EQ
CA1       -     3      89AG
CA2       -     3     100AG   101AG   103AG   105AG
CA3       -     3      92AG    93AG    94AG    95AG    96AG    97AG    98AG
CA4       -     4      90AG
CA5       -     3     107AG   108AG   109AG
CH1       -    53EQ
CH11      -    57EQ
CH2       -    54EQ
CH21      -    58EQ
CH3       -    55EQ
CH31      -    59EQ
CH4       -    56EQ
CH41      -    60EQ
CL        -    24EQ    117=
CLDF      -     3      107AG
CLDP      -    44EQ    107AG   117
CLP       -    25EQ
CLPF      -     3
CL2F      -     3      109AG
CL4F      -     3      108AG
CM        -    29EQ    121=
```

```
I N D E X                                                                PAGE  22
                                     SUBROUTINE A1

      CMDF      -      3      97AG    98AG
      CMDQ      -    39EQ     97AG    113      114
      CMDP      -    40EQ     98AG    113      114
      CMO       -    37EQ     93AG    110=     113
      CMOF      -      4      93AG
      CMP       -   113=      120     121
      CMPF      -      3      93AG
      CMQ       -    26EQ    103AG
      CMQF      -      3     103AG   105AG
      CMS       -    96AG     112     121
      CM2F      -      3      94AG
      CN        -    30EQ     120=
      CNP       -   114=      120     121
      CNR       -    27EQ    105AG
      CN4F      -      3     101AG
      COSB      -     63
      CX        -    22EQ     116=
      CXP       -    31EQ     89AG    116
      CXPF      -      3      89AG
      CY        -    23EQ     118=
      CYP       -   115=      118     119
      CY4F      -      3     102AG
      CZ        -    24EQ     119=
      CZDF      -      3      95AG    96AG
      CZDQ      -    34EQ     95AG    112      115
      CZDR      -    35EQ     95AG    112      115
      CZO       -    32EQ     92AG    112
      CZP       -   112=      118     119
      CZPF      -      3      92AG
      CZZF      -      3
      DCL1      -    42EQ
      DCL2      -   109AG     117
      DCL4      -    43EQ    108AG    117
      DCM2      -    38EQ     94AG    113
      DCM4      -    41EQ    101AG    114
      DCY4      -    36EQ    100AG    115
      DCZ2      -    33EQ     112
      IA        -    610A
      IB        -    610A
      IC        -    610A
      ID        -    610A
      IE        -    610A
      IF        -    610A
      LCONV     -    20EQ
      NC1       -      3      89AG    90AG
      NC2       -      3     100AG   101AG    103AG    105AG
      NC3       -      3      92AG    93AG     94AG     95AG     96AG     97AG     98AG
      NC5       -      3     107AG   108AG    109AG
      OPTHNG    -    21EQ
      OPTM      -    14EQ      73
```

```
 I N D E X                                    SUBROUTINE A1                                  PAGE  23

 RETURN   -   122
 SINO     -   62      64      65
 T        -   19EQ    87      88
 TABLE    -   89      90
 TABL2    -   92      93      94      95      96      97      98      100     101     103
              105     107     108     109
 UAL      -   79=     83      103AG
 UAT      -   80=     84      105AG
 UCPHI    -   63=     112     113     114     115     118     119     120     121
 UDL1     -   15EQ    71      72      73
 UDL2     -   16EQ    71      72      73
 UDL3     -   17EQ    71      72      73
 UDL4     -   18EQ    71      72      73
 USPHI    -   62=     112     113     114     115     118     119     120     121
 US2PHI   -   64=     66      117
 US2PH2   -   66=     112     113
 US4PHI   -   65=     114     115     117
 UT       -   87      85=
 VM       -   45EQ    81=     85      86      89AG    90AG    92AG    93AG    94AG    95AG
              96AG    97AG    98AG    100AG   101AG   103AG   105AG   107AG   108AG   109AG
 VMACH    -   5EQ     81
 XF       -   89AG    90AG    91=     92AG    93AG    94AG    95AG    96AG    97AG    98AG
              99=     100AG   101AG   102=    103AG   104=    105AG   106=    107AG   108AG
              109AG

 -+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
      SUBROUTINE A3T
C**INITIALIZATION FOR ENGINE MODULE
      COMMON C(3910)
      DIMENSION IPL(101)
      EQUIVALENCE (C( 367),RALPHA)
      EQUIVALENCE (C( 368),RALPHY)
      EQUIVALENCE (C( 373),RPHIP )
      EQUIVALENCE (C(1318),RDELCG)
      EQUIVALENCE (C(1320),FMXTH )
      EQUIVALENCE (C(1321),FMYTH )
      EQUIVALENCE (C(1322),FMZTH )
      EQUIVALENCE (C(1405),QPUPN )
      EQUIVALENCE (C(1411),FTHX  )
      EQUIVALENCE (C(1412),FTHY  )
      EQUIVALENCE (C(1413),FTHZ  )
      EQUIVALENCE (C(1415),DWT   )
      EQUIVALENCE (C(1418),RDCGF )
      EQUIVALENCE (C(1419),FMIXF )
      EQUIVALENCE (C(1420),FMIYF )
      EQUIVALENCE (C(1528),DMASS )
      EQUIVALENCE (C(1739),WP    )
      EQUIVALENCE (C(1743),WQ    )
      EQUIVALENCE (C(1747),WR    )
      EQUIVALENCE (C(1748),FMIX  )
      EQUIVALENCE (C(1749),FMIY  )
      EQUIVALENCE (C(1750),FMIZ  )
      EQUIVALENCE (C(2561),N     )
      EQUIVALENCE (C(2562),IPL   )
      IPL(N  ) = 1496
      N = N+1
      C(1499) = 0.
C
      IF (QBURN .GT. 0.) GO TO 10
      WP = 0.
      WQ = 0.
      WR = 0.
      RALPHA = 0.
      RALPHY = 0.
      RPHIP = 0.
      RETURN
   10 CONTINUE
      FTHRST=0.
      FTHX=0.
      FTHY=0.
      FTHZ=0.
      FMXTH=0.
      FMYTH=0.
      FMZTH=0.
      DMASS = DWT/32.174
      RDELCG = RDCGF
```

```
      FMIX = FMIXF
      FMIY = FMIYF
      FMIZ = FMIYF
      RETURN
      END
```

```
I N D E X                                                                          PAGE  26
                                          SUBROUTINE A3I

     SYMBOL   - - - - - - - - - - - - - - -  REFERENCES  - - - - - - - - - - - - - - -

     13        -     3I        39*
     A3I       -     1
     RALPHA    -     4EQ       35=
     RALPHY    -     5EQ       36=
     RPHIP     -     6EQ       37=
     C         -     2CO       4EO       5EQ       6EQ       7EQ       8EQ       9EQ       10EQ      11EQ      12FQ
                     13EQ      14EQ      15EQ      16EQ      17EQ      18EQ      19EQ      20EQ      21EQ      22EQ
                     23EQ      24EQ      25EQ      26EQ      27EQ      34=
     DMASS     -     19EQ      47=
     DMT       -     15EQ      47
     FMIX      -     23EQ      49=
     FMIXF     -     17EQ      49
     FMIY      -     24EQ      50=
     FMIYF     -     19EQ      50        51
     FMIZ      -     25EQ      51=
     FMXTH     -      8EQ      44=
     FMYTH     -      9EQ      45=
     FMZTH     -     10EQ      46=
     FTHRST    -     40=
     FTHX      -     12EQ      41=
     FTHY      -     13EQ      42=
     FTHZ      -     14EQ      43=
     IFL       -     30I       27EQ      28=
     N         -     26EQ      28        29=
     QBUPN     -     11EQ      31
     RDCGF     -     16EQ      49
     RDELCG    -      7EQ      48=
     RETURN    -     38        52
     WP        -     20EQ      32=
     WO        -     21EQ      33=
     WR        -     22EQ      34=

     -+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

SUBROUTINE A3

```
    1          SUBROUTINE A3
         C**ENGINE MODULE
    2          COMMON C(3510)
         C
         C
         C** INPUT DATA
    3          EQUIVALENCE (C(1313),PFXCG )
    4          EQUIVALENCE (C(1314),PFYCG )
    5          EQUIVALENCE (C(1315),PFZCG )
    6          EQUIVALENCE (C(1401),RALPHT)
    7          EQUIVALENCE (C(1402),RPHIT )
    8          EQUIVALENCE (C(1403),QNALGK)
    9          EQUIVALENCE (C(1404),PCFTH )
   10          EQUIVALENCE (C(1405),PPURN )
   11          EQUIVALENCE (C(1414),CTSP  )
   12          EQUIVALENCE (C(1415),DWT   )
   13          EQUIVALENCE (C(1416),DWP   )
   14          EQUIVALENCE (C(1417),RDCGO )
   15          EQUIVALENCE (C(1418),RDCGF )
   16          EQUIVALENCE (C(1419),FMIXF )
   17          EQUIVALENCE (C(1420),FMIYF )
   18          EQUIVALENCE (C(1421),PLCGO )
         C
         C** INPUTS FROM OTHER MODULES
   19          EQUIVALENCE (C(2000),T     )
         C
         C** OUTPUTS
   20          EQUIVALENCE (C(1308),PDELCG)
   21          EQUIVALENCE (C(1320),FMXTH )
   22          EQUIVALENCE (C(1321),FMYTH )
   23          EQUIVALENCE (C(1322),FMZTH )
   24          EQUIVALENCE (C(1409),UDMP  )
   25          EQUIVALENCE (C(1410),FTHRST)
   26          EQUIVALENCE (C(1411),FTHX  )
   27          EQUIVALENCE (C(1412),FTHY  )
   28          EQUIVALENCE (C(1413),FTHZ  )
   29          EQUIVALENCE (C(1427),PLCG  )
   30          EQUIVALENCE (C(1428),DMASS )
   31          EQUIVALENCE (C(1748),FMIX  )
   32          EQUIVALENCE (C(1749),FMIY  )
   33          EQUIVALENCE (C(1750),FMIZ  )
         C
         C**STATE VARIABLES AND THEIR DERIVATIVES
   34          EQUIVALENCE (C(1496),UIMPD )
   35          EQUIVALENCE (C(1499),UIMP  )
         C**LOOK UP TABLE FOR THRUST
   36          DIMENSION NTH(2), THA(10), THF(10)
   37          DATA NTH/1,10/
   38          DATA THA/   0., .125, .250, .750,1.500,1.625,1.750, 2.00, 3.,100./
   39          DATA THF/2300.,1750.,1650.,1600.,1400.,1250., 600., 300., 0.,  0./
```

SUBROUTINE A3

```
      C
            IF (QPURN.GT.C.) RETURN
            CALL TABLE(T,THA,THF,NTH,XF,6HFTHRST,FTHRST)
      C
            IF (CNALGN) 20,20,10
         10 USINA=SIND(RALPHT)
            FTHX=FTHRST*COSD(RALPHT)
            FTHY=-FTHRST*USINA*SIND(RPHIT)
            FTHZ=FTHRST*USINA*COSD(RPHIT)
            FMXTH = -FTHY*RFZCG + FTHZ*RFYCG
            FMYTH =  FTHX*RFZCG + FTHZ*RFXCG
            FMZTH = -FTHX*RFYCG - FTHY*RFXCG
            GO TO 30
         20 FTHX=FTHRST
            FTHY=0.
            FTHZ=0.
            FMXTH=0.
            FMYTH=C.
            FMZTH=0.
         30 CONTINUE
      C
            UIMPD = FTHRST
            UDWP = UIMP/CISP
      C
            DMASS = (DWT+DWP-UDWP)/32.174
            RDELCG = RDCGO - (RDCGO - RDCGF)*UDWP/DWP
      C
            FMIX = FMIXF*(DWT+DWP-UDWP)/DWP
            FMIY = FMIYF*(DWT+DWP-UDWP)/DWP
            FMIZ = FMIY
            RLCG = RLCGO + RDELCG
            IF (FTHRST .GT. 0.) RETURN
      C
            WRITE (6,100) T
        100 FORMAT (//14H BURNOUT TIME=,F8.4,5H SEC.)
            QPURN=1.0
            RETURN
            END
```

```
I N D E X                                                                                        PAGE  29
                                          SUBROUTINE A3
      SYMBOL- - - - - - - - - - - - - - - - - - -REFERENCES- - - - - - - - - - - - - - - - - - - - -
      10        -     42      43*
      26        -     42      51*
      30        -     5.      57*
      100       -     67WR    68*
      A3        -     1
      RALPHT    -      6EQ    43      44
      BPHIT     -      7EQ    45      46
      G         -      2EQ     3EQ     4EQ     5EQ     6EQ     7EQ     8EQ     9EQ    10EQ    11EQ
                      12EQ    13EQ    14EQ    15EQ    16EQ    17EQ    18EQ    19EQ    20EQ    21EQ
                      22EQ    23EQ    24EQ    25EQ    26EQ    27EQ    28EQ    29EQ    30EQ    31EQ
                      32EQ    33EQ    34EQ    35EQ
      CISP      -     11EQ    59
      COSD      -     44      46
      DMASS     -     36EQ    50=
      DMP       -     13EQ    60      61      62      63
      DWT       -     12EQ    60      62      63
      FMIX      -     31EQ    62=
      FMIXF     -     16EQ    62
      FMIY      -     32EQ    63=     64
      FMIYF     -     17EQ    63
      FMIZ      -     33EQ    64=
      FMXTH     -     21EQ    47=     54=
      FMYTH     -     22EQ    48=     55=
      FMZTH     -     23EQ    49=     56=
      FTHRST    -     25EQ    41AG    44      45      46      51      58      66
      FTHX      -     26EQ    44=     48      49      51=
      FTHY      -     27EQ    45=     47      49      52=
      FTHZ      -     28EQ    46=     47      48      53=
      NTH       -     36DI    37DA    41AG
      PCFTH     -      9EQ
      QBURN     -     10EQ    40      69=
      QNALGN    -      8EQ    42
      RDCGF     -     15EQ    61
      RDCGO     -     14EQ    61
      RDELCG    -     25EQ    61=     65
      RETURN    -     40      66      70
      RFXCG     -      3EQ    48      49
      RFYCG     -      4EQ    47      49
      RFZCG     -      5EQ    47      48
      RLCG      -     29EQ    65=
      RLCGO     -     18EQ    65
      SIND      -     43      45
      T         -     19EQ    41AG    67WR
      TABLE     -     41
      THA       -     36DI    38DA    41AG
      THF       -     36DI    39DA    41AG
      UAMP      -     24EQ    59=     60      61      62      63
      UIMP      -     35EQ    59
```

```
I N D E X                                                        PAGE  30
                              SUBROUTINE A3

UIMPA    -     34EQ   5A=
USINA    -     43=  .  45       46
XF       -     41AG

-+-+-+-+-+-+-+-+-+-+-+-+-+->-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
  I N D E X
                                           SUBROUTINE A2

      1             SUBROUTINE A2
              C**AERO FORCE AND MOMENT MODULE     BODY AXES
      2             COMMON C(3517)
      3         101 FORMAT(1H0,4X,21HFRONT LUG CLEARS RAIL,5X,3HT =,1PE10.2,5X,
                *         9HWCL VEL =,1PE10.2,5X,14HPITCH MOMENT =,1PE10.2)
              C
              C**INPUT-DATA
      4             EQUIVALENCE (C(1336),PFAREA)
      5             EQUIVALENCE (C(1337),PFLGTH)
      6             EQUIVALENCE (C(1316),RLUG  )
      7             EQUIVALENCE (C(1317),RAIL  )
      8             EQUIVALENCE (C(1330),AGV   )
      9             EQUIVALENCE (C(1331),CFREQ )
     10             EQUIVALENCE (C(1332),CPHAS )
     11             EQUIVALENCE (C(1405),OBURN )
     12             EQUIVALENCE (C(1627),AGRAV )
              C
              C**INPUTS FROM OTHER MODULES
     13             EQUIVALENCE (C(0203),PDYNMC)
     14             EQUIVALENCE (C( 204),VMACH )
     15             EQUIVALENCE (C(0207),VAIRSP)
     16             EQUIVALENCE (C( 350),RTHT  )
     17             EQUIVALENCE (C( 380),RANGO )
     18             EQUIVALENCE (C(1203),CX    )
     19             EQUIVALENCE (C(1204),CY    )
     20             EQUIVALENCE (C(1205),CZ    )
     21             EQUIVALENCE (C(1206),CLP   )
     22             EQUIVALENCE (C(1207),CMQ   )
     23             EQUIVALENCE (C(1208),CNR   )
     24             EQUIVALENCE (C(1209),CL    )
     25             EQUIVALENCE (C(1210),CM    )
     26             EQUIVALENCE (C(1211),CN    )
     27             EQUIVALENCE (C(1236),CM1   )
     28             EQUIVALENCE (C(1237),CM2   )
     29             EQUIVALENCE (C(1238),CM3   )
     30             EQUIVALENCE (C(1239),CM4   )
     31             EQUIVALENCE (C(1320),FMXTH )
     32             EQUIVALENCE (C(1321),FMYTH )
     33             EQUIVALENCE (C(1322),FMZTH )
     34             EQUIVALENCE (C(1411),FTHX  )
     35             EQUIVALENCE (C(1412),FTHY  )
     36             EQUIVALENCE (C(1413),FTHZ  )
     37             EQUIVALENCE (C(1422),RLCG  )
     38             EQUIVALENCE (C(1723),CFA23 )
     39             EQUIVALENCE (C(1735),CFA33 )
     40             EQUIVALENCE (C(1739),WP    )
     41             EQUIVALENCE (C(1743),WQ    )
     42             EQUIVALENCE (C(1747),WR    )
     43             EQUIVALENCE (C(2869),T     )
              C
```

```
C**OUTPUTS                                                        A1C-2
      EQUIVALENCE (C(1300),FX3A  )                                A1C-2
      EQUIVALENCE (C(1301),FY3A  )                                A1C-2
      EQUIVALENCE (C(1302),FZ3A  )                                A1C-2
      EQUIVALENCE (C(1303),FMX3A )                                A1C-2
      EQUIVALENCE (C(1304),FMY3A )                                A1C-2
      EQUIVALENCE (C(1305),FMZ3A )                                A1C-2
      EQUIVALENCE (C(1708),RDELCG)  [illegible]
      EQUIVALENCE (C(1628),DMASS )
      EQUIVALENCE (C(1748),FMIX  )
      EQUIVALENCE (C(1749),FMIY  )
      EQUIVALENCE (C(1750),FMIZ  )
C
C**OTHER OUTPUTS
      EQUIVALENCE (C(1309),FMH1  )
      EQUIVALENCE (C(1310),FMH2  )
      EQUIVALENCE (C(1311),FMH3  )
      EQUIVALENCE (C(1312),FMH4  )
      EQUIVALENCE (C(1323),FMXLUG)
      EQUIVALENCE (C(1324),FMYLUG)
      EQUIVALENCE (C(1325),FMZLUG)
      EQUIVALENCE (C(3504),OPTN4)
C
C**FORCE VECTOR COMPONENTS
      UOS = PDYNMC*FFAREA
      UOSL = UOS*RFLGTH
C
      FX3A=UOS*(-CX)+FTHX
      FY3A=UOS*CY+FTHY
      FZ3A=UOS*CZ+FTHZ
C
C** AERO MOMENTS   (NOTE FACTOR OF 2.0 IN DAMPING COEFFICIENT)
      UL2V = 0.
      IF (VAIRSP .GT. 0.) UL2V = RFLGTH/(2.*VAIRSP)
      FMX3A = (CL + CLP*UL2V*HP) * UOSL                    + FMXTH
      FMY3A = (CM + CMQ*UL2V*HQ) * UOSL + FZ3A*RDELCG + FMYTH
      FMZ3A = (CN + CNR*UL2V*HR) * UOSL - FY3A*RDELCG + FMZTH
C
C** CALCULATE HINGE MOMENTS
      FMH1 = CH1*UOSL
      FMH2 = CH2*UOSL
      FMH3 = CH3*UOSL
      FMH4 = CH4*UOSL
C
C**MOMENTS AND FORCES DUE TO LUGS
      IF( (OPTN4 .GT. 0.).AND.(RANGO .LE. RAIL+RLUG) ) GO TO 70
      FYLUG = 0.
      FZLUG = 0.
      FMXLUG = 0.
      FMYLUG = 0.
```

SUBROUTINE A2

```
      FMZLUG = 0.
      IF (FLG2 .GT.0.) GO TO 74
      FLG2 = 1.
      WRITE(6,102) T,VAIRSP,UFZL2
  102 FORMAT (1H0,36H           REAR LUG CLEARS RAIL    T = ,F8.4,
     *  10HREL VEL = ,F8.3,16H   RAIL FORCE = ,F8.2)
      GO TO 74
   70 IF (RANGO .LE. RAIL) GO TO 72
      RZDD = AGV*SIND(360.*(GFREQ*T + GPHAS))
      FYLUG = -(FYBA + DMASS*AGRAV*CFA23 +  FMZBA*
     *            RLCG*DMASS/FMIZ)/(1. + DMASS*RLCG*RLCG/FMIZ)
      FZLUG = -(FZBA + DMASS*AGRAV*(CFA33-RZDD) +  FMYBA*
     *            RLCG*DMASS/FMIY)/(1. + DMASS*RLCG*RLCG/FMIY)
      FMXLUG = - FMXBA
      FMYLUG = FZLUG*RLCG
      FMZLUG =-FYLUG*RLCG
      IF (FLG1 .GT. 0.) GO TO 74
      FLG1 = 1.
      WRITE(6, 101) T, VAIRSP, FMYLUG
      GO TO 74
   72 CONTINUE
      RZDD = AGV*SIND(360.*(GFREQ*T + GPHAS))
      FYLUG = -(FYBA + DMASS*AGRAV*CFA23)
      FZLUG = -(FZBA + DMASS*AGRAV*(CFA33-RZDD))
      FMXLUG = - FMXBA
      FMYLUG = - FMYBA
      FMZLUG = - FMZBA
      FLG1 = 0.
   74 CONTINUE
C**TOTAL FORCE AND MOMENTS
      FYBA = FYBA + FYLUG
      FZBA = FZBA + FZLUG
      FMXBA = FMXBA + FMXLUG
      FMYBA = FMYBA + FMYLUG
      FMZBA = FMZBA + FMZLUG
      RETURN
      END
```

```
I N D E X                                                                                PAGE  34
                                         SUBROUTINE A2

SYMBOL  - - - - - - - - - - - - - - -     REFERENCES - - - - - - - - - - - - - - - - -

70         -     77      88*
72         -     80      99*
74         -     83      87      95      98      107*
101        -      3*     97WR
102        -     85WR    86*
AGPAV      -     12EQ    90      91      101     102
AGV        -      8EQ    89      100
A2         -      1
BTHT       -     16EQ
C          -      2EQ     4EQ     5EQ     6EQ     7EQ     8EQ     9EQ    10EQ    11EQ    12EQ
                 13EQ    14EQ    15EQ    16EQ    17EQ    18EQ    19EQ    20EQ    21EQ    22EQ
                 23EQ    24EQ    25EQ    26EQ    27EQ    28EQ    29EQ    30EQ    31EQ    32EQ
                 33EQ    34EQ    35EQ    36EQ    37EQ    38EQ    39EQ    40EQ    41EQ    42EQ
                 43EQ    44EQ    45EQ    46EQ    47EQ    48EQ    49EQ    50EQ    51EQ    52EQ
                 53EQ    54EQ    55EQ    56EQ    57EQ    58EQ    59EQ    60EQ    61EQ    62EQ
CFA23      -     38EQ    90      101
CFA33      -     39EQ    91      102
CFPEQ      -      9EQ    89      100
CH1        -     27EQ    73
CH2        -     28EQ    74
CH3        -     29EQ    75
CH4        -     30EQ    76
CL         -     24EQ    70
CLP        -     21EQ    70
CM         -     25EQ    71
CMQ        -     22EQ    71
CN         -     26EQ    72
CNR        -     23EQ    72
GPHAS      -     16EQ    89      100
CX         -     18EQ    65
CY         -     19EQ    66
CZ         -     20EQ    67
DMASS      -     51EQ    90      91      101     102
FLG1       -     95      96=     106=
FLG2       -     93      94=
FMH1       -     55EQ    73=
FMH2       -     56EQ    74=
FMH3       -     57EQ    75=
FMH4       -     58EQ    76=
FMIX       -     52EQ
FMIY       -     53EQ    91
FMIZ       -     54EQ    90
FMXRA      -     47EQ    70=     92      103     110=
FMXLUG     -     59EQ    85=     92=     103=    110
FMXTH      -     31EQ    70
FMYRA      -     48EQ    71=     91      104     111=
FMYLUG     -     60EQ    81=     93=     97WR    104=    111
FMYTH      -     32EQ    71
```

```
I N D E X
                                   SUBROUTINE A2

FMZRA    -    49EQ    72=     90      105     112=
FMZLUG   -    68EQ    82=     94=     105=    112
FMZTH    -    37EQ    72
FTHX     -    34EQ    65
FTHY     -    35EQ    66
FTHZ     -    36EQ    67
FXPA     -    44EQ    65=
FYPA     -    45EQ    66=     72      90      101     108=
FYLUG    -    78=     92=     94      101=    106
FZPA     -    46EQ    67=     71      91      102     109=
FZLUG    -    79=     91=     93      102=    109
OPTN4    -    62EQ    77
PDYNMS   -    13EQ    63
QPUPN    -    11EQ
RAIL     -     7EQ    77      88
RANGO    -    17EQ    77      88
RDLLCG   -    59EQ    71      72
RETURN   -   113
RFAREA   -     4EQ    63
RFLGTH   -     5EQ    64      69
RLCG     -    37EQ    90      91      93      94
RLUG     -     6EQ    77
PZOD     -    89=     91      100=    102
SIND     -    89      100
T        -    47EQ    85WR    89      97WR    103
UFZL2    -    85WR
UL2V     -    68=     69=     70      71      72
UOS      -    63=     64      65      66      67
UOSL     -    64=     70      71      72      73      74      75      76
VAIRSP   -    15EQ    69      85WR    97WR
VMACH    -    14EQ
WP       -    40EQ    70
WQ       -    41EQ    71
WR       -    42EQ    72

-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
         SUBROUTINE D1T
C**      TRANSLATIONAL DYNAMICS INITIALIZATION MODULE FOR D1
         COMMON C(3510)
         EQUIVALENCE (C(2561),M      )
         EQUIVALENCE (C(2562),IPL    )
         DIMENSION  IPL (100)
C
C** INPUT DATA
         EQUIVALENCE (C( 100),VHXE   )
         EQUIVALENCE (C( 101),VHYE   )
         EQUIVALENCE (C( 102),VHZE   )
         EQUIVALENCE (C( 274),VMACH  )
         EQUIVALENCE (C( 367),BALPHA)
         EQUIVALENCE (C( 368),BALPHY)
         EQUIVALENCE (C( 427),BTHTG )
         EQUIVALENCE (C( 431),BPSIG )
         EQUIVALENCE (C(1639),OPTARG)
         EQUIVALENCE (C(1666),BLOSV )
         EQUIVALENCE (C(1667),RSLANT)
         EQUIVALENCE (C(1674),VMATE )
         EQUIVALENCE (C(1751),CP40  )
         EQUIVALENCE (C(3502),OPTN2 )
         EQUIVALENCE (C(3504),OPTN4 )
         EQUIVALENCE (C(3506),OPTN6 )
C
C** OUTPUT TO MODULES
         EQUIVALENCE (C(1615),RXE    )
         EQUIVALENCE (C(1619),RYE    )
         EQUIVALENCE (C(1623),RZE    )
         EQUIVALENCE (C(1603),VXE    )
         EQUIVALENCE (C(1607),VYE    )
         EQUIVALENCE (C(1611),VZE    )
         EQUIVALENCE (C(1651),PTXE   )
         EQUIVALENCE (C(1655),PTYE   )
         EQUIVALENCE (C(1659),PTZE   )
         EQUIVALENCE (C(1668),RXO    )
         EQUIVALENCE (C(1669),RYO    )
         EQUIVALENCE (C(1670),RZO    )
         EQUIVALENCE (C(1671),VXO    )
         EQUIVALENCE (C(1672),VYO    )
         EQUIVALENCE (C(1673),VZO    )
         EQUIVALENCE (C(1752),BPHIO )
         EQUIVALENCE (C(1753),BTHTO )
         EQUIVALENCE (C(1754),BPSIO )
         EQUIVALENCE (C(1665),RHZRO  )
C
         IPL(N) = 1600
         IPL(N+1) = 1604
         IPL(N+2) = 1608
        *IPL(N+3) = 1612
```

```
            IPL(N+4) = 1616
            IPL(N+5) = 1620
            IPL(N+6) = 1640
            IPL(N+7) = 1644
            IPL(N+8) = 1648
            IPL(N+9) = 1652
            IPL(N+10) = 1656
            N = N+11
            C( 363) = 0.
            C( 364) = 0.
            CRAD = 57.295778
      C
            IF (OPTN2.LE.0.) RETURN
            IF (OPTARG .LE. 0.) N = N-5
      C
      C**CALCULATE MISSILE PARAMETER INITIAL CONDITIONS
            RYE=0.
            RTZE = 0.
            RTYE = 0.
            RTXE = 0.
            BPHIO = 0.
      C
            IF (OPTN2.GT.1.0) GO TO 10
            RXE=-RSLANT*COSD(RLOSV)
            RZE=RSLANT*SIND(RLOSV)
            GO TO 20
         10 RSLANT = SQRT(RZE**2 + RXE**2)
         20 RH = RHZRO - RZE
      C
            IF (OPTN4 .GT. 0.) GO TO 30
            BPSIO = CRAD*ASIN(SIND(BPSIG)*RSLANT/RXE)
            CPSIO  = COSD(BPSIO)
            TTHTG  = SIND(BTHTG)/COSD(BTHTG)
            BTHTO = ATAND((-RZE/RXE - TTHTG*CPSIO),(CPSIO - TTHTG*RZE/RXE))
            GO TO 40
         30 CONTINUE
            IF (OPTN4 .GT. 1.) GO TO 40
            UST = SIND(BTHTO)
            USP = SIND(BPSIO)
            UCP = COSD(BPSIO)
            UCT = COSD(BTHTO)
            RXBA = -RXE*UCP*UCT + RZE*UST
            RZBA = -RXE*UCP*UST - RZE*UCT
            RYBA =  RXE*USP
            BTHTG = ATAND(-RZBA,RXBA)
            BPSIG = -ATAND(-RYBA,(RXBA*COSD(BTHTG)-RZBA*SIND(BTHTG)))
         40 CONTINUE
      C
         24 VSOUND = 1117.3 - 0.392*RH
            IF (OPTN6 .LE. 0.) VMMTE = VMACH*VSOUND
```

SUBROUTINE D1T

```
    89          VMHXY = VMHTE*COSD(BALPHA - BTHTO)
          C
    90          VXE = VMXE + VMHXY*COSD(BALPHY + BPSIO)
    91          VYE = VMYE + VMHXY*SIND(BALPHY + BPSIO)
    92          VZE = VMZE + VMHTE*SIND(BALPHA - BTHTO)
          C
    93          RXO = RXE
    94          RYO = RYE
    95          RZO = RZE
    96          VXO = VXE
    97          VYO = VYE
    98          VZO = VZE
    99          RETURN
   100          END
```

```
I N D E X                                                                                     PAGE  39
                                       SUBROUTINE D1I

SYMBOL    - - - - - - - - - - - - - - - - REFERENCES - - - - - - - - - - - - - - - -

10        -   63      67*
20        -   64      68*
24        -   87*
30        -   69      75*
40        -   74      76      86*
ASIN      -   70
ATAND     -   73      84      85
BALPHA    -   18EQ    89      92
BALPHY    -   11EQ    90      91
BLOSV     -   15EQ    64      65
BPHIO     -   37EQ    62=
BPSIG     -   13EQ    70      85=
BPSIO     -   39EQ    70=     71      78      79      90      91
BTHTG     -   12EQ    72      84=     85
BTHTO     -   38EQ    73=     77      80      89      92
C         -    2EQ     3EQ     4EQ     6EQ     7EQ     8EQ     9EQ    10EQ    11EQ    12EQ
              13EQ    14EQ    15EQ    16EQ    17EQ    18EQ    19EQ    20EQ    21EQ    22EQ
              23EQ    24EQ    25EQ    26EQ    27EQ    28EQ    29EQ    30EQ    31EQ    32EQ
              33EQ    34EQ    35EQ    36EQ    37EQ    38EQ    39EQ    40EQ    53=     54=
COSD      -   64      71      72      79      80      85      89      90
CPSIO     -   71=     73
CROD      -   18EQ    55=     70
D1I       -    1
IPL       -    4EQ    50I     41=     42=     43=     44=     45=     46=     47=     48=
              49=     50=     51=
N         -    3EQ    41      42      43      44      45      46      47      48      49
              50      51      52=     57=
OPTARG    -   14EQ    57
OPTN2     -   19EQ    56      63
OPTN4     -   20EQ    69      76
OPTN6     -   21EQ    89
RETURN    -   56      99
RH        -   68=     87
RHZRO     -   40EQ    68
RSLANT    -   16EQ    64      65      67=     70
RTXE      -   28EQ    61=
RTYE      -   29EQ    60=
RTZE      -   30EQ    59=
RXRA      -   81=     84      85
RXE       -   22EQ    64=     67      70      73      81      82      83      93
RXO       -   31EQ    97=
RYRA      -   83=     85
RYE       -   23EQ    58=     94
RYO       -   32EQ    94=
RZRA      -   82=     84      85
RZE       -   24EQ    65=     67      68      73      81      82      93
RZO       -   33EQ    95=
SIND      -   65      70      72      77      78      85      91      92
```

```
SQRT     -    67
TTMTG    -    72=    73
UCP      -    79=    81     82
UCT      -    80=    81     82
USP      -    78=    83
UST      -    77=    81     82
VMACH    -     9EQ   84
VMWTE    -    17EQ   88=    89     92
VMWXY    -    89=    90     91
VSOUND   -    87=    88
VWXE     -     6EQ   90
VWYE     -     7EQ   91
VWZE     -     8EQ   92
VXE      -    25EQ   90=    96
VXO      -    34EQ   96=
VYE      -    26EQ   91=    97
VYO      -    35EQ   97=
VZE      -    27EQ   92=    98
VZO      -    36EQ   98=

-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-  -+-+
```

```
      SUBROUTINE 01
C**TRANSLATIONAL DYNAMICS MODULE
      COMMON C(3510)
C
C**INPUT DATA
      EQUIVALENCE (C(1627),AGRAV )
      EQUIVALENCE (C(1628),DMASS )
      EQUIVALENCE (C(1629),ATHRST)
      EQUIVALENCE (C(1630),ATURNT)
      EQUIVALENCE (C(1631),AGAMT )
      EQUIVALENCE (C(1639),CPTARG)
      EQUIVALENCE (C(1681),DOIVF )
      EQUIVALENCE (C(1751),CRAD  )
C
C**INPUTS FROM OTHER MODULES
      EQUIVALENCE (C(1300),FXBA  )
      EQUIVALENCE (C(1301),FYBA  )
      EQUIVALENCE (C(1302),FZBA  )
      EQUIVALENCE (C(1703),CFA11 )
      EQUIVALENCE (C(1707),CFA12 )
      EQUIVALENCE (C(1711),CFA13 )
      EQUIVALENCE (C(1715),CFA21 )
      EQUIVALENCE (C(1719),CFA22 )
      EQUIVALENCE (C(1723),CFA23 )
      EQUIVALENCE (C(1727),CFA31 )
      EQUIVALENCE (C(1731),CFA32 )
      EQUIVALENCE (C(1735),CFA33 )
      EQUIVALENCE (C(2990),T     )
C
C**STATE VARIABLE OUTPUTS
      EQUIVALENCE (C(1600),VXED  )
      EQUIVALENCE (C(1603),VXE   )
      EQUIVALENCE (C(1604),VYED  )
      EQUIVALENCE (C(1607),VYE   )
      EQUIVALENCE (C(1608),VZED  )
      EQUIVALENCE (C(1511),VZE   )
      EQUIVALENCE (C(1612),RXED  )
      EQUIVALENCE (C(1615),RXE   )
      EQUIVALENCE (C(1616),RYED  )
      EQUIVALENCE (C(1619),RYE   )
      EQUIVALENCE (C(1620),RZED  )
      EQUIVALENCE (C(1623),RZE   )
      EQUIVALENCE (C(1640),VTARGD)
      EQUIVALENCE (C(1643),VTARG )
      EQUIVALENCE (C(1644),RPSITD)
      EQUIVALENCE (C(1647),RPSIT )
      EQUIVALENCE (C(1648),RTXED )
      EQUIVALENCE (C(1651),RTXE  )
      EQUIVALENCE (C(1652),RTYED )
      EQUIVALENCE (C(1655),RTYE  )
```

SUBROUTINE D1

```
          EQUIVALENCE (C(1656),RTZED )
          EQUIVALENCE (C(1659),RTZE  )
C
C**OTHER OUTPUTS
          EQUIVALENCE (C(1624),AXBA  )
          EQUIVALENCE (C(1625),AYBA  )
          EQUIVALENCE (C(1626),AZBA  )
          EQUIVALENCE (C(1532),VDELX )
          EQUIVALENCE (C(1533),VDELY )
          EQUIVALENCE (C(1534),VDELZ )
          EQUIVALENCE (C(1535),PDELX )
          EQUIVALENCE (C(1536),PDELY )
          EQUIVALENCE (C(1537),PDELZ )
          EQUIVALENCE (C(1538),VCLSNG)
          EQUIVALENCE (C(1660),VTXE  )
          EQUIVALENCE (C(1561),VTYE  )
          EQUIVALENCE (C(1562),VTZE  )
          EQUIVALENCE (C(1563),VDXB  )
          EQUIVALENCE (C(1564),VDYB  )
          EQUIVALENCE (C(1665),VDZB  )
          EQUIVALENCE (C(1576),ANGX  )
          EQUIVALENCE (C(1577),ANGY  )
          EQUIVALENCE (C(1673),ANGZ  )
          EQUIVALENCE (C( 371),RANGE  )
C
C**ADD AERO AND THRUST FORCES TO GET TOTAL ACCELERATION IN BODY AXES
          AXBA = FXBA/DMASS
          AYBA = FYBA/DMASS
          AZBA = FZBA/DMASS
C
C**RESOLVE FROM BODY TO EARTH AXES
          AXE = CFA11*AXBA+CFA21*AYBA+CFA31*AZBA
          AYE = CFA12*AXBA+CFA22*AYBA+CFA32*AZBA
          AZE = CFA13*AXBA+CFA23*AYBA+CFA33*AZBA
C
C**INTEGRATE ACCELERATIONS
          VXED = AXE
          VYED = AYE
          VZED = AZE + AGRAV
C
C** CALCULATE TOTAL MISSILE ACCELERATION IN BODY AXES
          VDXB = CFA11*VXED + CFA12*VYED + CFA13*VZED
          VDYB = CFA21*VXED + CFA22*VYED + CFA23*VZED
          VDZB = CFA31*VXED + CFA32*VYED + CFA33*VZED
          ANGX = VDXB/32.174
          ANGY = VDYB/32.174
          ANGZ = VDZB/32.174
C
C**INTEGRATE VELOCITIES TO EARTH AXES  POSITION
     10   RXED = VXE
```

```
  82         RYED = VYE
  83         RZED = VZE
       C
       C**TARGET MOTION
  84         IF (OPTARG .LE. 0.) RETURN
  85         VTARGD = ATHRST*AGRAV
  86         BPSITD= 0.
  87         IF (VTARG.GT.0.) BPSITD= ATURNT*AGRAV*CRAD/VTARG
       C
  88         VTXE = VTARG*COSD(BGAMT)*COSD(BPSIT)
  89         VTYE = VTARG*COSD(BGAMT)*SIND(BPSIT)
  90         VTZE = VTARG*SIND(BGAMT)
       C
  91         RTXED = VTXE
  92         RTYED = VTYE
  93         RTZED = VTZE
       C
  94         VDELX = VTXE-VXE
  95         VDELY = VTYE-VYE
  96         VDELZ = VTZE-VZE
       C
  97         VELSNG = (RDELX*VDELX+RDELY*VDELY+RDELZ*VDELZ)/RANGE
  98         RETURN
  99         END
```

[illegible]

```
                              SUBROUTINE D1

    SYMBOL  - - - - - - - - - - - - - - - - REFERENCES - - - - - - - - - - - - - - - -

    1C       -    A1*
    ADIVE    -     9EQ
    AGRAV    -     3EQ    74     85     87
    ANGX     -    62EQ    78=
    ANGY     -    63EQ    79=
    ANGZ     -    64EQ    80=
    ATHRST   -     5EQ    85
    ATURNT   -     6EQ    87
    AXRA     -    46EQ    66=    69     70     71
    AXE      -    69=     72
    AYRA     -    47EQ    67=    69     70     71
    AYE      -    70=     73
    AZRA     -    48EQ    68=    69     70     71
    AZE      -    71=     74
    RGAMT    -     7EQ    88     89     90
    RPSIT    -    39EQ    88     89
    RPSITO   -    38EQ    86=    87=
    C        -     2EQ     3EQ    4EQ    5EQ    6EQ    7EQ    8EQ    9EQ   10EQ   11EQ
                  12EQ    13EQ   14EQ   15EQ   16EQ   17EQ   18EQ   19EQ   20EQ   21EQ
                  22EQ    23EQ   24EQ   25EQ   26EQ   27EQ   28EQ   29EQ   30EQ   31EQ
                  32EQ    33EQ   34EQ   35EQ   36EQ   37EQ   38EQ   39EQ   40EQ   41EQ
                  42EQ    43EQ   44EQ   45EQ   46EQ   47EQ   48EQ   49EQ   50EQ   51EQ
                  52EQ    53EQ   54EQ   55EQ   56EQ   57EQ   58EQ   59EQ   60EQ   61EQ
                  62EQ    63EQ   64EQ   65EQ
    CFA11    -    14EQ    69     75
    CFA12    -    15EQ    70     75
    CFA13    -    16EQ    71     75
    CFA21    -    17EQ    69     76
    CFA22    -    18EQ    70     76
    CFA23    -    19EQ    71     76
    CFA31    -    20EQ    69     77
    CFA32    -    21EQ    70     77
    CFA33    -    22EQ    71     77
    COSD     -    88      89
    GRAD     -    16EQ    87
    DMASS    -     4EQ    66     67     68
    D1       -     1
    FXRA     -    11EQ    66
    FYRA     -    12EQ    67
    FZRA     -    13EQ    68
    OPTARG   -     3EQ    84
    RANGE    -    65EQ    97
    RDELX    -    52EQ    37
    RDELY    -    53EQ    97
    RDELZ    -    54EQ    97
    RETURN   -    84      98
    RTXE     -    41EQ
    RTXEO    -    40EQ    91=
```

```
RTYE      -   43EQ
RTYED     -   42EQ    92=
RTZE      -   45EQ
RTZED     -   44EQ    93=
RXE       -   31EQ
RXED      -   30EQ    81=
RYE       -   33EQ
RYED      -   32EQ    82=
RZE       -   35EQ
RZED      -   34EQ    83=
SIND      -   89      90
T         -   23EQ
VCLSNG    -   55EQ    97=
VDELX     -   49EQ    94=     97
VDELY     -   50EQ    95=     97
VDELZ     -   51EQ    96=     97
VDXB      -   59EQ    75=     78
VDYB      -   60EQ    76=     79
VDZB      -   61EQ    77=     80
VTARG     -   37EQ    87      88      89      90
VTARGD    -   36EQ    85=
VTXE      -   56EQ    88=     91      94
VTYE      -   57EQ    89=     92      95
VTZE      -   58EQ    90=     93      96
VXE       -   25EQ    81      94
VXED      -   24EQ    72=     75      76      77
VYE       -   27EQ    82      95
VYED      -   26EQ    73=     75      76      77
VZE       -   29EQ    83      96
VZED      -   28EQ    74=     75      76      77
-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
      1          SUBROUTINE O2I
           C**ROTATIONAL DYNAMICS INITIALIZATION MODULE O2IFUL
      2          COMMON C(3510)
      3          DIMENSION  IPL (100)
           C**INPUT DATA
      4          EQUIVALENCE (C(1752),BPHIO )
      5          EQUIVALENCE (C(1753),BTHTO )
      6          EQUIVALENCE (C(1754),BPSIO )
           C**INPUTS FROM MAIN PROGRAM
      7          EQUIVALENCE (C(2561),N      )
      8          EQUIVALENCE (C(2562),IPL    )
           C**STATE VARIABLE OUTPUTS
      9          EQUIVALENCE (C(1703),CFA11 )
     10          EQUIVALENCE (C(1707),CFA12 )
     11          EQUIVALENCE (C(1711),CFA13 )
     12          EQUIVALENCE (C(1715),CFA21 )
     13          EQUIVALENCE (C(1719),CFA22 )
     14          EQUIVALENCE (C(1723),CFA23 )
     15          EQUIVALENCE (C(1727),CFA31 )
     16          EQUIVALENCE (C(1731),CFA32 )
     17          EQUIVALENCE (C(1735),CFA33 )
           C**OTHER OUTPUTS
     18          EQUIVALENCE (C(1755),AG21  )
     19          EQUIVALENCE (C(1756),AG22  )
     20          EQUIVALENCE (C(1757),AG23  )
     21          EQUIVALENCE (C(1758),AG31  )
     22          EQUIVALENCE (C(1759),AG32  )
     23          EQUIVALENCE (C(1760),AG33  )
           C**INITIAL CALCULATION OF EULER ANGLE MATRIX OF DIRECTION COSINES (CFA)
     24          USPHI = SIND(BPHIO)
     25          UCPHI = COSD(BPHIO)
     26          USTHT = SIND(BTHTO)
     27          UCTHT = COSD(BTHTO)
     28          USPSI = SIND(BPSIO)
     29          UCPSI = COSD(BPSIO)
     30          CFA11 = UCPSI*UCTHT
     31          CFA12 = USPSI*UCTHT
     32          CFA13 = -USTHT
     33          CFA21 = -USPSI*UCPHI+UCPSI*USTHT*USPHI
     34          CFA22 = UCPSI*UCPHI+USPSI*USTHT*USPHI
     35          CFA23 = UCTHT*USPHI
     36          CFA31 = UCPSI*USTHT*UCPHI+USPSI*USPHI
     37          CFA32 = USPSI*USTHT*UCPHI-UCPSI*USPHI
     38          CFA33 = UCTHT*UCPHI
           C
           C**INITIALIZE MATRIX COEF FOR FREE GYRO MODEL(S)
     39          AG31 = CFA31
     40          AG32 = CFA32
     41          AG33 = CFA33
     42          AG21 = CFA21
```

```
      A022 = CFA22
      A023 = CFA23
C
C**INTEGRATED PARAMETER LIST--(IPL)-FOR-HPO,HOD,HRD,AND CFAD
      IPL(N) = 1700
      IPL(N+1) = 1704
      IPL(N+2) = 1708
      IPL(N+3) = 1712
      IPL(N+4) = 1716
      IPL(N+5) = 1720
      IPL(N+6) = 1724
      IPL(N+7) = 1728
      IPL(N+8) = 1732
      IPL(N+9) = 1736
      IPL(N+10) = 1740
      IPL(N+11) = 1744
      N = N+12
      RETURN
      END
```

```
I N D E X                                                                        PAGE  48
                                       SUBROUTINE D2I
      SYMBOL---- - - - - - - - - - - - - - - - - -----REFERENCES--- - - - - - - - - - - - - -

      A021    -    18EQ    42=
      A022    -    19EQ    43=
      A023    -    20EQ    44=
      A031    -    21EQ    39=
      A032    -    22EQ    40=
      A033    -    23EQ    41=
      BPHIO   -     4EQ    24      25
      APSIO   -     6EQ    28      29
      ATHTO   -     5EQ    26      27
      C       -     2EQ     4EQ     5EQ     6EQ     7EQ     8EQ     9EQ    10EQ    11EQ    12EQ
                   13EQ    14EQ    15EQ    16EQ    17EQ    18EQ    19EQ    20EQ    21EQ    22EQ
                   23EQ
      CFA11   -     9EQ    30=
      CFA12   -    10EQ    31=
      CFA13   -    11EQ    32=
      CFA21   -    12EQ    33=     42
      CFA22   -    13EQ    34=     43
      CFA23   -    14EQ    35=     44
      CFA31   -    15EQ    36=     39
      CFA32   -    16EQ    37=     40
      CFA33   -    17EQ    38=     41
      COSD    -    25      27      29
      F2I     -     1
      IPL     -     3DI     8EQ    45=     46=     47=     48=     49=     50=     51=     52=
                   53=     54=     55=     56=
      N       -     7EQ    45      46      47      48      49      50      51      52      53
                   54      55      56      57=
      RETURN  -    58
      SIND    -    24      26      28
      UCPHI   -    25=     33      34      36      37      38
      UCPSI   -    29=     30      33      34      36      37
      UCTHI   -    27=     30      31      35      38
      USPHI   -    24=     33      34      35      36      37
      USPSI   -    28=     31      33      34      36      37
      USTHI   -    26=     32      33      34      36      37

      -+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
      SUBROUTINE D2
C**   ROTATIONAL DYNAMICS MODULE
      COMMON C(3510)
C
C**DATA INPUTS
      EQUIVALENCE (C(1748),FMIX  )
      EQUIVALENCE (C(1749),FMIY  )
      EQUIVALENCE (C(1750),FMIZ )
      EQUIVALENCE (C(1751),CRAD  )
      EQUIVALENCE (C(3503),OPTN3)
C
C**INPUTS FROM OTHER MODULES
      EQUIVALENCE (C(1303),FMXBA )
      EQUIVALENCE (C(1304),FMYBA )
      EQUIVALENCE (C(1305),FMZBA )
C
C**STATE VARIABLE OUTPUTS
      EQUIVALENCE (C(1700),CFA11D)
      EQUIVALENCE (C(1703),CFA11 )
      EQUIVALENCE (C(1704),CFA12D)
      EQUIVALENCE (C(1707),CFA12 )
      EQUIVALENCE (C(1708),CFA13D)
      EQUIVALENCE (C(1711),CFA13 )
      EQUIVALENCE (C(1712),CFA21D)
      EQUIVALENCE (C(1715),CFA21 )
      EQUIVALENCE (C(1716),CFA22D)
      EQUIVALENCE (C(1719),CFA22 )
      EQUIVALENCE (C(1720),CFA23D)
      EQUIVALENCE (C(1723),CFA23 )
      EQUIVALENCE (C(1724),CFA31D)
      EQUIVALENCE (C(1727),CFA31 )
      EQUIVALENCE (C(1728),CFA32D)
      EQUIVALENCE (C(1731),CFA32 )
      EQUIVALENCE (C(1732),CFA33D)
      EQUIVALENCE (C(1735),CFA33 )
      EQUIVALENCE (C(1736),WPD   )
      EQUIVALENCE (C(1739),WP    )
      EQUIVALENCE (C(1740),WQD   )
      EQUIVALENCE (C(1743),WQ    )
      EQUIVALENCE (C(1744),WRD   )
      EQUIVALENCE (C(1747),WR    )
C
C**INTEGRATE BODY ANGULAR RATES
      IF (OPTN3.GT.0.) GO TO 55
      WPD = CRAD*FMXBA/FMIX
   55 WQD = (CRAD*FMYBA+(FMIZ-FMIX)*WP*WR/CRAD)/FMIY
   65 WRD = (CRAD*FMZBA+(FMIX-FMIY)*WP*WQ/CRAD)/FMIZ
C
C**INTEGRATE ATTITUDE DIRECTION COSINES
   49 CFA11D=(CFA21*WR-CFA31*WQ)/CRAD
```

```
      CFA12D=(CFA22*WP-CFA32*WQ)/CRAD
      CFA13D=(CFA23*WR-CFA33*WQ)/CRAD
      CFA21D = (CFA31*WP-CFA11*WR)/CRAD
      CFA22D = (CFA32*WP-CFA12*WR)/CRAD
      CFA23D = (CFA33*WP-CFA13*WR)/CRAD
      CFA31D = (CFA11*WQ-CFA21*WP)/CRAD
      CFA32D = (CFA12*WQ-CFA22*WP)/CRAD
      CFA33D = (CFA13*WQ-CFA23*WP)/CRAD
      RETURN
      END
```

```
                                        SUBROUTINE 02

SYMBOL                                  REFERENCES

49        -    39*
55        -    35     37*
65        -    38*
C         -    2EQ    3EQ    4EQ    5EQ    6EQ    7EQ    8EQ    9EQ    10EQ   11EQ
               12EQ   13EQ   14EQ   15EQ   16EQ   17EQ   18EQ   19EQ   20EQ   21EQ
               22EQ   23EQ   24EQ   25EQ   26EQ   27EQ   28EQ   29EQ   30EQ   31EQ
               32EQ   33EQ   34EQ
CFA11     -    12EQ   42     45
CFA11D    -    11EQ   39=
CFA12     -    14EQ   43     46
CFA12D    -    13EQ   46=
CFA13     -    16EQ   44     47
CFA13D    -    15EQ   41=
CFA21     -    18EQ   39     45
CFA21D    -    17EQ   42=
CFA22     -    20EQ   40     46
CFA22D    -    19EQ   43=
CFA23     -    22EQ   41     47
CFA23D    -    21EQ   44=
CFA31     -    24EQ   39     42
CFA31D    -    23EQ   45=
CFA32     -    26EQ   40     43
CFA32D    -    25EQ   44=
CFA33     -    28EQ   41     44
CFA33D    -    27EQ   47=
CPAD      -    6EQ    36     37     38     39     40     41     42     43     44
               45     46     47
O2        -    1
FMIX      -    3EQ    36     37     38
FMIY      -    4EQ    37     38
FMIZ      -    5EQ    37     38
FMXBA     -    8EQ    36
FMYBA     -    9EQ    37
FMZBA     -    10EQ   38
OPTNS     -    7EQ    35
RETURN    -    48
WP        -    30EQ   37     38     42     43     44     45     46     47
WPD       -    29EQ   36=
WQ        -    32EQ   38     39     40     41     45     46     47
WQD       -    31EQ   37=
WR        -    34EQ   37     39     40     41     42     43     44
WRD       -    33EQ   39=
```

```
      SUBROUTINE S1I
C***SEEKER INIT. MODULE
      COMMON C(3510)
      DIMENSION IPL(100)
      EQUIVALENCE (C(2561),N      )
      EQUIVALENCE (C(2562),IPL    )
      EQUIVALENCE (C(3584),OPTN4  )
      IPL(N  ) = 424
      IPL(N+1) = 428
      IPL(N+2) = 404
      IPL(N+3) = 412
      IPL(N+4) = 416
      IPL(N+5) = 420
      N = N+6
      C( 411) = 0.
      C( 415) = 0.
      C( 419) = 0.
      C( 423) = 0.
      C( 460) = 0.
      IF (OPTN4 .GT. 1.) GO TO 30
      C( 461) = 1.
      C( 462) = 1.
      C( 463) = 1.
      C( 464) = 1.
      RETURN
   30 CONTINUE
      C( 461) = 0.
      C( 462) = 0.
      C( 464) = 0.
      C( 463) = 0.
      RETURN
      END
```

```
I N D E X                                                                         PAGE  53
                                      SUBROUTINE S1I

SYMBOL    - - - - - - - - - - - - - - - - - REFERENCES - - - - - - - - - - - - - - - - - -

30        -     19      25=
C         -     2EQ     4EQ     5EQ     6EQ     14=     15=     16=     17=     18=     20=
                21=     22=     23=     26=     27=     28=     29=
IPL       -     3DI     5EQ     7=      8=      9=      10=     11=     12=
N         -     4EQ     7       8       9       10      11      12      13=
OPTN4     -     6EQ     19
RETURN    -     24      30
S1I       -     1

*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
```

```
      SUBROUTINE S1
C**SEEKER MODULE
C
      COMMON C(3519)
      EQUIVALENCE (C(2000),T      )
  101 FORMAT (30H0      TARGET ACQUISITION   T = ,F8.4,
     *        10H   EPS Z = ,1PE11.3,10H   EPS Y = ,1PE11.3)
  102 FORMAT (30H0      PITCH PLANE TRACK    T = ,F8.4,
     *        10H   EPS Z = ,1PE11.3,10H   EPS Y = ,1PE11.3)
  103 FORMAT (30H0      YAW   PLANE TRACK    T = ,F8.4,
     *        10H   EPS Z = ,1PE11.3,10H   EPS Y = ,1PE11.3)
C
C**INPUT DATA
      EQUIVALENCE (C( 445),RLOCK )
      EQUIVALENCE (C( 446),DT    )
      EQUIVALENCE (C( 447),ROB   )
      EQUIVALENCE (C( 448),CFOVZ )
      EQUIVALENCE (C( 449),CFOVY )
      EQUIVALENCE (C( 450),GSX   )
      EQUIVALENCE (C( 451),SEPS  )
      EQUIVALENCE (C( 452),SWP   )
      EQUIVALENCE (C( 453),PPK   )
      EQUIVALENCE (C( 454),GFO   )
      EQUIVALENCE (C( 455),OPTNSK)
      EQUIVALENCE (C(0456),GS    )
      EQUIVALENCE (C(0457),WSL   )
      EQUIVALENCE (C(0458),WSM   )
      EQUIVALENCE (C(0459),WLZ   )
C
      EQUIVALENCE (C( 460),ST    )
      EQUIVALENCE (C( 461),CAGE  )
      EQUIVALENCE (C( 462),TKRZ  )
      EQUIVALENCE (C( 463),TKRY  )
      EQUIVALENCE (C( 464),TRKZY )
C
C**INPUTS FROM OTHER MODULES
      EQUIVALENCE (C(0371),RANGE )
      EQUIVALENCE (C(0372),RYBA  )
      EQUIVALENCE (C(0373),RYBA  )
      EQUIVALENCE (C(0374),RZBA  )
      EQUIVALENCE (C(1739),WP    )
      EQUIVALENCE (C(1743),WQ    )
      EQUIVALENCE (C(1747),WR    )
C
C**STATE VARIABLE OUTPUTS
      EQUIVALENCE (C(3409),WLQD  )
      EQUIVALENCE (C(3411),WLQ   )
      EQUIVALENCE (C(3412),WLRD  )
      EQUIVALENCE (C(3415),WLR   )
     *EQUIVALENCE (C(3416),WLQSD )
```

```
      EQUIVALENCE (C(0419),WLOS  )
      EQUIVALENCE (C(0421),WLRSQ )
      EQUIVALENCE (C(0423),WLPS  )
      EQUIVALENCE (C(0424),RTHTGD)
      EQUIVALENCE (C(0427),RTHTG )
      EQUIVALENCE (C(0428),RPSIGD)
      EQUIVALENCE (C(0431),RPSIG )
C
C**OTHER OUTPUTS
      EQUIVALENCE (C(0433),WLAMQ )
      EQUIVALENCE (C(0407),WLAMR )
      EQUIVALENCE (C(0435),DEPSZ )
      EQUIVALENCE (C(0436),DEPSY )
      EQUIVALENCE (C(0437),WZ    )
      EQUIVALENCE (C(0438),WY    )
      EQUIVALENCE (C(0439),RGDEFL)
C
C**DIRECTION COSINES FOR BODY TO PLATFORM TRANSFORMATION
      UCT = COSD(RTHTG)
      UST = SIND(RTHTG)
      UCP = COSD(RPSIG)
      USP = SIND(RPSIG)
      UB11 = UCT*UCP
      UB12 = UCT*USP
      UB13 =-UST
      UB21 =-USP
      UB22 = UCP
      UB23 = 0.
      UB31 = UST*UCP
      UB32 = UST*USP
      UB33 = UCT
C
C** CALCULATE TOTAL DEFLECTION OF GIMBALS
      RGDEFL=SQRT(RTHTG**2+RPSIG**2)
C
C**TRANSFORM LOS FROM BODY TO GIMBAL AXES
      RXG = UB11*RXBA+UB12*RYBA+UB13*RZBA
      RYG = UB21*RXBA+UB22*RYBA+UB23*RZBA
      RZG = UB31*RXBA+UB32*RYBA+UB33*RZBA
C
C**LOS ERRORS IN PLATFORM COORDINATES
      DEPSZ = ATAND(-RZG,RXG)
      DEPSY = ATAND( RYG,RXG)
C
C**SEEKER OUTPUT SIGNALS
      IF (OPTNSK.LE.0.) GO TO 89
C
C**VIDICON TRACKER
      IF (RANGE .LT. RBK) RETURN
      WLAMQ = GZO*DEPSZ
```

```
   75           WLAMR = GEO*BEPSY
   76           GO TO 70
          C
          C**QUADRANT TRACKER
   77        81 IF (T .LT. ST) GO TO 82
   78           ST = ST + DT
   79           IF (RANGE .GT. RLOCK) GO TO 81
   80           CZ = 2.*BEPSZ/CFOVZ
   81           CY = 2.*BEPSY/CFOVY
   82           IF (CZ**2 .GT. 1.-CY**2) GO TO 81
   83           BZ = SIGN(1.,BEPSZ)
   84           BY = SIGN(1.,BEPSY)
   85           TKDB = 6DB/2.*(RANGE/32816.)**2
   86           IF (ABS(BEPSZ).LT.TKDB) BZ = 0.
   87           IF (ABS(BEPSY).LT.TKDB) BY = 0.
   88           IF (CAGE .GT. 0.) GO TO 82
   89           UZ = BZ
   90           UY = BY
   91           CAGE = 1.
   92           WRITE(6,101) T, BEPSZ, BEPSY
   93           GO TO 82
   94        81 BZ = 0.
   95           BY = 0.
          C**SEEKER COMPENSATION
   96        82 WLAMQ = BZ*GS
   97           WLAMR = BY*GS
   98           WQP = WLAMQ
   99           WRP = WLAMR
  100           IF (WSL .LE. 0.) GO TO 83
  101           WLQD = WLAMQ
  102           WLRD = WLAMR
  103           WLQD = WLQC + SEPS
  104           WLRD = WLRD + SEPS
  105           WQP = WLQD/WSL + WLQ
  106           WRP = WLRD/WSL + WLR
  107           WLAMQ = WQP
  108           WLAMR = WRP
  109           IF (WSN .LE. 0.) GO TO 83
  110           WLQSD = WSN*(WQP - WLQS)
  111           WLRSD = WSN*(WRP - WLRS)
  112           WQP = WLQSD/WL2 + WLQS
  113           WRP = WLRSD/WL2 + WLRS
          C**SEEKER SWITCHING LOGIC
  114        83 IF (CAGE .LE. 0.) GO TO 30
          C  PITCH PLANE
  115        10 IF (TKRZ .GT. 0.) GO TO 20
  116           IF (BZ*UZ .GE. 0.) GO TO 12
  117           TKRZ = 1.
  118           WRITE(6,102) T, BEPSZ, BEPSY
  119           GO TO 20
```

```
   12 WLAMQ = BZ*GSX
      WQP = WLAMQ
      WLQD = 0.
      WLQSD = 0.
      UZ = BZ
C  YAW PLANE
   26 IF (TKRY .GT. 0.) GO TO 30
      IF (BY*UY .GE. 0.) GO TO 22
      TKRY = 1.
      WRITE(6,103) T, BEPSZ, BEPSY
      GO TO 30
   22 WLAMR = BY*GSX
      WRP = WLAMR
      WLRD = 0.
      WLRSD = 0.
      UY = BY
   30 CONTINUE
C
C**MISSILE BODY RATES IN GIMBAL AXES
      WZ = UB31*WP+UB32*WQ+UB33*WR
      WY = UB21*WP+UB22*WQ+UB23*WR
C
C**GIMBAL COUPLING
      UZK = SWP*(-BTHTG + BPSIG)
      UYK = SWP*(-BPSIG - BTHTG)
C
C**GIMBAL ANGLE DERIVATIVES
      BTHTGD = WQP + UZK - WY
      BPSIGD = WRP + UYK - WZ/UB33
C
      IF (GAGE .GT. 0.) RETURN
      WLAMQ = 0.
      WLAMR = 0.
      WLQD = 0.
      WLRD = 0.
      WLQSD = 0.
      WLRSD = 0.
      BTHTGD = 0.
      BPSIGD = 0.
      RETURN
      END
```

```
I N D E X                                                                    PAGE  58
                                        SUBROUTINE S1

SYMBOL   - - - - - - - - - - - - - - - - -    REFERENCES  - - - - - - - - - - - - - - - -

10       -   115*
12       -   116      120*
20       -   115      119      125*
22       -   126      130*
30       -    76      114      125      129      135*
80       -    72       77*
81       -    79       82       94*
82       -    77       88       93       96*
83       -   100      109      114*
101      -     4*      92WR
102      -     5*     118WR
103      -     6*     128WR
APS      -    86       87
ATAND    -    70       71
BOB      -     9EQ     85
BEPSY    -    49EQ     71=      75       81       84       87       92WR    118WR    128WR
BEPSZ    -    48EQ     72=      74       80       83       86       92WR    118WR    128WR
BGDEFL   -    52EQ     65=
BPSIG    -    45EQ     55       56       66      138      139
BPSIGD   -    44EQ    141=     150=
BTHTG    -    43EQ     53       54       66      138      139
BTHTGD   -    42EQ    140=     149=
BY       -    84=      87=      93       95=      97      126      133      134
BZ       -    83=      86=      89       94=      96      116      120      124
C        -     2EQ      3EQ      7EQ      8EQ      9EQ     10EQ     11EQ     12EQ     13EQ     14EQ
              15EQ     16EQ     17EQ     18EQ     19EQ     20EQ     21EQ     22EQ     23EQ     24EQ
              25EQ     25EQ     27EQ     28EQ     29EQ     30EQ     31EQ     32EQ     33EQ     34EQ
              35EQ     36EQ     37EQ     38EQ     39EQ     40EQ     41EQ     42EQ     43EQ     44EQ
              45EQ     46EQ     47EQ     48EQ     49EQ     50EQ     51EQ     52EQ
CAGE     -    23EQ     89       91=     114      142
CFOVY    -    11EQ     81
CFOVZ    -    16EQ     89
COSD     -    53       55
CY       -    81=      82
CZ       -    80=      82
DT       -     8EQ     78
GEO      -    16EQ     74       75
GS       -    19EQ     96       97
GSX      -    12EQ    129      130
OPTNSK   -    17EQ     72
RANGE    -    27EQ     73       79       85
RAK      -    15EQ     77
RETURN   -    73      142      151
RLOCK    -     7EQ     79
RXPA     -    28EQ     67       68       69
RXG      -    67=      70       71
RYPA     -    29EQ     67       68       69
RYG      -    68=      71
```

```
PZAA    -   30EQ    67     68     69
RZG     -   69=     70
SEPS    -   13EQ    107    104
SIGN    -   83      84
SIND    -   54      56
SQRT    -   66
ST      -   22EQ    77     78=
SWP     -   14EQ    138    139
S1      -    1
T       -   3EQ     77     92WR   118WR  120WR
TKDB    -   85=     86     87
TKRY    -   25EQ    125    127=
TKRZ    -   24EQ    115    117=
TRKZY   -   26EQ
UB11    -   57=     67
UB12    -   58=     67
UB13    -   59=     67
UB21    -   60=     68     137
UB22    -   61=     68     137
UB23    -   62=     68     137
UB31    -   63=     69     136
UB32    -   64=     69     136
UB33    -   65=     69     136    141
UCP     -   55=     57     61     63
UGT     -   53=     57     58     65
USP     -   56=     58     62     64
UST     -   54=     59     63     64
UY      -   96=     126    134=
UYK     -  139=     141
UZ      -   89=     116    124=
UZK     -  134=     146
WLAMO   -   46EQ    74=    96=    98     101    107=   129=   121    143=
WLAMR   -   47EQ    75=    97=    99     102    108=   130=   131    144=
WLO     -   35EQ    105
WLOD    -   34EQ    101=   103=   105    122=   145=
WLOS    -   39EQ    110    112
WLOSD   -   38EQ    110=   112    123=   147=
WLP     -   37EQ    106
WLPD    -   36EQ    102=   104=   106    132=   146=
WLPS    -   41EQ    111    113
WLPSD   -   40EQ    111=   113    133=   148=
WL2     -   21EQ    112    113
WP      -   31EQ    136    137
WQ      -   32EQ    136    137
WQP     -   98=     105=   107    110    112=   121=   140
WR      -   33EQ    136    137
WRP     -   99=     106=   108    111    113=   131=   141
WSL     -   19EQ    104    105    106
WSN     -   20EQ    109    110    111
WY      -   51EQ    137=   140
```

```
I N D E X                          SUBROUTINE S1

W2            SLEQ   136*    141
-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

FIRE AUTOPILOT INITIATION MODULE

```
C**HELFIRE AUTOPILOT INITIATION MODULE
C**HIGH FREQ. MODEL
      SUBROUTINE C1I
      COMMON C(3510)
      DIMENSION IPL(100)
      EQUIVALENCE (C(2561),N        )
      EQUIVALENCE (C(2562),IPL      )
C
      IPL(N   ) = 800
      IPL(N+1) = 804
      IPL(N+2) = 808
      IPL(N+3) = 812
      IPL(N+4) = 816
      IPL(N+5) = 820
      IPL(N+6) = 824
      IPL(N+7) = 828
      IPL(N+8) = 832
      IPL(N+9) = 836
      IPL(N+10) = 840
      IPL(N+11) = 844
      IPL(N+12) = 848
      IPL(N+13) = 852
      N = N+14
      C(803) = 0.
      C(807) = 0.
      C(811) = 0.
      C(815) = 0.
      C(819) = 0.
      C(823) = 0.
      C(827) = 0.
      C(831) = 0.
      C(835) = 0.
      C(839) = 0.
      C(843) = 0.
      C(847) = 0.
      C(851) = 0.
      C(855) = 0.
      C( 811) = C( 878) + C (861)
      C( 923) = C( 879)
      C( 931) =-C(1233)
      C( 947) =-C( 831)
      C( 939) = C( 831)
      IF (C(877) .LE. 0.) RETURN
      IPL(N   ) = 901
      IPL(N+1) = 905
      IPL(N+2) = 909
      C( 904) = 0.
      C( 908) = 0.
      C( 912) = 0.
      RETURN
```

FIRE AUTOPILOT INITIATION MODULE

```
     48          END
```

```
I N D E X
                         FIRE AUTOPILOT INITIATION MODULE

SYMBOL                        REFERENCES

C        -    2CO    4EQ    5EQ    21=    22=    23=    24=    25=    26=    27=
              28=    29=    30=    31=    32=    33=    34=    35=    36=    37=
              38=    39=    40     44=    45=    46=
CII      -    1
IPL      -    3DE    5EQ    6=     7=     8=     9=     10=    11=    12=    13=
              14=    15=    16=    17=    18=    19=    41=    42=    43=
N        -    4EQ    6      7      6      9      10     11     12     13     14
              15     16     17     10     19     20=    41     42     43
RETURN   -    40     47
```

FIRE AUTOPILOT MODULE

```
            C**HELLFIRE AUTOPILOT MODULE
            C**HIGH FREQ MODEL (USE DER = .0025)
      1           SUBROUTINE C1
      2           COMMON C(3510)
      3           DIMENSION HDELTC(4),VAR(101)
            C
            C**INPUT DATA
      4           EQUIVALENCE (C( 860),TOY    )
      5           EQUIVALENCE (C( 861),GPIAS  )
      6           EQUIVALENCE (C( 862),GM     )
      7           EQUIVALENCE (C( 863),WN2    )
      8           EQUIVALENCE (C( 864),WN1    )
      9           EQUIVALENCE (C( 865),WL     )
     10           EQUIVALENCE (C( 866),WLXX1  )
     11           EQUIVALENCE (C( 867),WLXX2  )
     12           EQUIVALENCE (C( 868),WLJK1  )
     13           EQUIVALENCE (C( 869),WLJK2  )
     14           EQUIVALENCE (C( 870),HJK    )
     15           EQUIVALENCE (C( 871),WXX    )
     16           EQUIVALENCE (C( 872),DXX    )
     17           EQUIVALENCE (C( 873),WJK    )
     18           EQUIVALENCE (C( 874),DJK    )
     19           EQUIVALENCE (C( 875),GXX    )
     20           EQUIVALENCE (C( 876),GJK    )
     21           EQUIVALENCE (C( 877),PES    )
     22           EQUIVALENCE (C( 878),DPIAS  )
     23           EQUIVALENCE (C( 879),RPIAS  )
     24           EQUIVALENCE (C( 880),HXX    )
            C
            C**INPUTS FROM OTHER MODULES
     25           EQUIVALENCE (C(  77),SPHI    )
     26           EQUIVALENCE (C(  87),STHT    )
     27           EQUIVALENCE (C(  97),SPSI    )
     28           EQUIVALENCE (C( 353),RPHI   )
     29           EQUIVALENCE (C( 354),RTHT   )
     30           EQUIVALENCE (C( 355),RPSI   )
     31           EQUIVALENCE (C( 403),WLAMO  )
     32           EQUIVALENCE (C( 407),WLAMR  )
     33           EQUIVALENCE (C( 461),CAGE   )
     34           EQUIVALENCE (C( 462),TKRZ   )
     35           EQUIVALENCE (C( 463),TKRY   )
     36           EQUIVALENCE (C(1551),CPTM   )
     37           EQUIVALENCE (C(1552),UPHI   )
     38           EQUIVALENCE (C(1553),UPSI   )
     39           EQUIVALENCE (C(1554),UTHT   )
            C
            C**INPUTS FROM MAIN PROGRAM
     40           EQUIVALENCE (C(2200),T      )
            C
            C** STATE VARIABLE OUTPUTS
```

```
      EQUIVALENCE (C( 800),WLOSDD)
      EQUIVALENCE (C( 803),WLOSF )
      EQUIVALENCE (C( 804),WLOSD )
      EQUIVALENCE (C( 807),WLOS   )
      EQUIVALENCE (C( 808),WLOSSD)
      EQUIVALENCE (C( 811),WLOSS )
      EQUIVALENCE (C( 812),WLRSDD)
      EQUIVALENCE (C( 815),WLRSP )
      EQUIVALENCE (C( 816),WLRSD )
      EQUIVALENCE (C( 819),WLRS   )
      EQUIVALENCE (C( 820),WLRSSD)
      EQUIVALENCE (C( 823),WLRSS )
      EQUIVALENCE (C( 824),BLOSSD)
      EQUIVALENCE (C( 827),BLOSS )
      EQUIVALENCE (C( 828),BLRSSD)
      EQUIVALENCE (C( 831),BLRSS )
      EQUIVALENCE (C( 832),BJJSDD)
      EQUIVALENCE (C( 835),BJJSF )
      EQUIVALENCE (C( 836),BJJSP )
      EQUIVALENCE (C( 839),BJJS   )
      EQUIVALENCE (C( 840),BKKSDD)
      EQUIVALENCE (C( 843),BKKSP )
      EQUIVALENCE (C( 844),BKKSD )
      EQUIVALENCE (C( 847),BKKS   )
      EQUIVALENCE (C( 848),BXXSDD)
      EQUIVALENCE (C( 851),BXXSP )
      EQUIVALENCE (C( 852),BXXSC )
      EQUIVALENCE (C( 855),BXXS   )
      EQUIVALENCE (C( 900),WNS    )
      EQUIVALENCE (C( 901),SNTH   )
      EQUIVALENCE (C( 904),STHP   )
      EQUIVALENCE (C( 905),SNPS   )
      EQUIVALENCE (C( 908),SPSP   )
      EQUIVALENCE (C( 909),SNPH   )
      EQUIVALENCE (C( 912),SPHP   )
C
C**OUTPUTS
      EQUIVALENCE (C( 856),BDELTC)
C
C**OTHER OUTPUTS
      EQUIVALENCE (C( 880),BXX    )
      EQUIVALENCE (C( 881),BJJ    )
      EQUIVALENCE (C( 882),BKK    )
      EQUIVALENCE (C( 883),BXXSS )
      EQUIVALENCE (C( 884),BJJSS )
      EQUIVALENCE (C( 885),BKKSS )
      EQUIVALENCE (C( 886),BTHTS )
      EQUIVALENCE (C( 887),BPSIS )
C
C**GUIDANCE SIGNAL SHAPING
```

```
C**GUIDANCE SWITCHING
      WLOSD = WLOSP
      WLRSD = WLRSP
      WLOSDD = WN2*(WN2*(WLAMQ - WLOS) - 2.*WLOSD)
      WLRSDD = WN2*(WN2*(WLAMR - WLRS) - 2.*WLRSD)
      WQC = GN*(WLOSD/WL+WLOS) + GBIAS
      WRC = GN*(WLRSD/WL+WLRS) + RBIAS
      IF (TKRZ.GT.0. .AND. T.GT.TOY) GO TO 4
      WLOSDD = 0.
      WQC = QBIAS + GBIAS
      IF (CAGE.GT.0. .AND. T.GT.TOY) WQC = WLAMQ + GBIAS
    4 IF (TKRY .GT. 0.) GO TO 5
      WLRSDD = 0.
      WRC = RBIAS
      IF (CAGE .GT. 0.) WRC = WLAMR
    5 CONTINUE
      WLOSSD = WN1*(WQC - WLOSS)
      WLRSSD = WN1*(WRC - WLRSS)
      BLOSSD = WLOSS
      BLRSSD = WLRSS
C
C**RATE GYRO DYNAMICS AND LIMITING
      BTHTS = -BTH2
      BPSIS = -BPS1
      BXX   = -BPH1
      IF (RES .LE. 0.) GO TO 10
      SNTH = WNS*(STHT - STHP)
      SNPS = WNS*(SPSI - SPSP)
      SNPH = WNS*(SPHI - SPHP)
      BTHTS = -RES*AINT(BTH2/RES) + SNTH
      BPSIS = -RES*AINT(BPS1/RES) + SNPS
      BXX   = -RES*AINT(BPH1/RES) + SNPH
   10 CONTINUE
      IF (OPTH .LE. 0.) GO TO 15
      BTHTS = UTHT
      BPSIS = UPST
      BXX   = UPHI
      BTH2 = -BTH2
      BPS1 = -BPS1
      BPH1 = -BPH1
   15 CONTINUE
C
C**SUMMATION OF RATE DAMPING AND GUIDANCE SIGNALS AND THEIR DERIVATIVES
      BJJ = (BLRSS-BPSIS) - (BLOSS-BTHTS)
      BKK =-(BLRSS-BPSIS) - (BLOSS-BTHTS)
C
C**GUIDANCE SIGNAL SHAPING AND LIMITING
      BXXSD = BXXSP
      BJJSD = BJJSP
      BKKSD = BKKSP
```

```
 128          BXXSDD = WXX*(WXX*(BXX - BXXS) - 2.*ZXX*BXXSD)
 129          BJJSDD = WJK*(WJK*(BJJ - BJJS) - 2.*ZJK*BJJSD)
 130          BKKSDD = WJK*(WJK*(BKK - BKKS) - 2.*ZJK*BKKSD)
 131          BXXSS = GXX*((BXXSDD + (WLXX1+WLXX2)*BXXSD)/(WLXX1*WLXX2) + BXXS)
 132          BJJSS = GJK*((BJJSDD + (WLJK1+WLJK2)*BJJSD)/(WLJK1*WLJK2) + BJJS)
 133          BKKSS = GJK*((BKKSDD + (WLJK1+WLJK2)*BKKSD)/(WLJK1*WLJK2) + BKKS)
 134          IF (ABS(BJJSS) .GT. HJK) BJJSS = SIGN(HJK,BJJSS)
 135          IF (ABS(BKKSS) .GT. HJK) BKKSS = SIGN(HJK,BKKSS)
 136          IF (ABS(BXXSS) .GT. HXX) BXXSS = SIGN(HXX,BXXSS)
        0
        C**COMMANDS TO ACTUATORS
 137          BDELTC(1) = BJJSS + BXXSS
 138          BDELTC(2) = BKKSS + BXXSS
 139          BDELTC(3) = BJJSS - BXXSS
 140          BDELTC(4) = BKKSS - BXXSS
 141          RETURN
 142          END
```

```
SYMBOL                                REFERENCES

4        -    91      95*
5        -    95      99*
13       -   107     114*
15       -   115     122*
ABS      -   134     135     136
ATNT     -   111     112     113
BDELTC   -    70I     75EQ   137=    138=    139=    140=
BJJ      -    78EQ   123=    129
BJJS     -    60EQ   129     132
BJJSD    -    59EQ   125=    129     132
BJJSDD   -    57EQ   129=    132
BJJSP    -    58EQ   126
BJJSS    -    81EQ   132=    134     137     139
BKK      -    79EQ   124=    130
BKKS     -    64EQ   130     133
BKKSD    -    63EQ   127=    130     133
BKKSDD   -    61EQ   130=    133
BKKSP    -    62EQ   127
BKKSS    -    82EQ   133=    135     138     140
BLOSS    -    54EQ   123     124
BLOSSD   -    53EQ   162=
BLPSS    -    56EQ   123     124
BLPSSD   -    55EQ   163=
BPH1     -    28EQ   106     113     121=
BPSIS    -    84EQ   165=    112=    117=    123     124
BPS1     -    30EQ   105     112     120=
BTHTS    -    83EQ   164=    111=    116=    123     124
BTH2     -    29EQ   104     111     119=
BXX      -    77EQ   166=    113=    118=    128
BXXS     -    68EQ   128     131
BXXSD    -    67EQ   125=    128     131
BXXSDD   -    65EQ   128=    131
BXXSP    -    66EQ   125
BXXSS    -    80EQ   131=    136     137     138     139     140
G        -     2EQ     4EQ     5EQ     6EQ     7EQ     8EQ     9EQ    10EQ    11EQ    12EQ
              13EQ    14EQ    15EQ    16EQ    17EQ    18EQ    19EQ    20EQ    21EQ    22EQ
              23EQ    24EQ    25EQ    26EQ    27EQ    28EQ    29EQ    30EQ    31EQ    32EQ
              33EQ    34EQ    35EQ    36EQ    37EQ    38EQ    39EQ    40EQ    41EQ    42EQ
              43EQ    44EQ    45EQ    46EQ    47EQ    48EQ    49EQ    50EQ    51EQ    52EQ
              53EQ    54EQ    55EQ    56EQ    57EQ    58EQ    59EQ    60EQ    61EQ    62EQ
              63EQ    64EQ    65EQ    66EQ    67EQ    68EQ    69EQ    70EQ    71EQ    72EQ
              73EQ    74EQ    75EQ    76EQ    77EQ    78EQ    79EQ    80EQ    81EQ    82EQ
              83EQ    84EQ
GAGE     -    33EQ    94      98
C1       -     1
DJK      -    18EQ   129     130
DXX      -    16EQ   128
GBIAS    -     5EQ    89      93      94
```

```
I N D E X                                     FIRE AUTOPILOT MODULE

GJK        -     20EQ    132     133
GN         -      6EQ     89      90
GXX        -     19EQ    131
HJK        -     14EQ    134     135
HXX        -     24EQ    136
OPTM       -     36EQ    115
QBTAS      -     22EQ     93
QDIAS      -     23EQ     93      97
RFS        -     21EQ    107     111     112     113
RETURN     -    141
SIGN       -    134      135     136
SMPH       -     74EQ    111=    113
SMPS       -     72EQ    109=    112
SMTH       -     70EQ    109=    111
SPHI       -     25EQ    110
SPHP       -     75EQ    110
SPSI       -     27EQ    109
SPSP       -     73EQ    109
STHP       -     71EQ    108
STHT       -     26EQ    108
T          -     40EQ     91      94
TPY        -      4EQ     91      94
TKRY       -     35EQ     95
TKRZ       -     34EQ     91
UPHI       -     37EQ    118
UPSI       -     38EQ    117
UTHT       -     39EQ    116
VAR        -     30I
WJK        -     17EQ    129     133
WL         -      9EQ     89      93
WLAMD      -     31EQ     87      94
WLAMP      -     32EQ     88      98
WLJK1      -     12EQ    132     133
WLJK2      -     13EQ    132     133
WLOS       -     44EQ     87      89
WLOSD      -     43EQ     85=     87      89
WLOSDD     -     41EQ     87=     92=
WLOSP      -     42EQ     85
WLOSS      -     46EQ    100     102
WLOSSD     -     45EQ    100=
WLPS       -     50EQ     88      90
WLPSD      -     49EQ     86=     88      90
WLPSDD     -     47EQ     88=     96=
WLPSP      -     48EQ     86
WLPSS      -     52EQ    101     103
WLPSSD     -     51EQ    101=
WLXX1      -     10EQ    131
WLXY2      -     11EQ    131
WMS        -     69EQ    108     109     110
WM1        -      8EQ     10     101
```

```
I N D E X                                                             PAGE  73
                                  FIRE AUTOPILOT MODULE

WN2      -     7E0      87       88
WCC      -     89=      93=      94=      100
WEC      -     90=      97=      98=      101
WXX      -     15E0     12*

-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
      SUBROUTINE C2I
C**HELFIRE AUTOPILOT INITIATION MODULE
C**LOW  FREQ. MODEL
      COMMON C(3510)
      DIMENSION IPL(100)
      EQUIVALENCE (C(2561),N      )
      EQUIVALENCE (C(2562),IPL    )
C
      IPL(N  ) = 800
      IPL(N+1) = 804
      IPL(N+2) = 808
      IPL(N+3) = 812
      IPL(N+4) = 816
      IPL(N+5) = 820
      IPL(N+6) = 824
      IPL(N+7) = 828
      N = N+8
      C(2662) = .005
      C(2663) = .005
      C(2664) = .005
      C(803) = 0.
      C(807) = 0.
      C(811) = 0.
      C(815) = 0.
      C(819) = 0.
      C(823) = 0.
      C(827) = 0.
      C(831) = 0.
      C( 811) = C( 861)
      IF (C( 462) .GT. 0.) RETURN
      C( 811) = C( 878) + C( 861)
      C( 823) = C( 879)
      RETURN
      END
```

```
                                  SUBROUTINE C2I

SYMBOL                             REFERENCES

C         -   2EQ    4EQ    5EQ    15=    16=    17=    18=    19=    20=    21=
              22=    23=    24=    25=    26=    27     28=    29=
C2I       -   1
IPL       -   3DI    5EQ    6=     7=     8=     9=     10=    11=    12=    13=
N         -   4EQ    6      7      8      9      10     11     12     13     14=
RETURN    -   27     30
```

```
   1            SUBROUTINE C7
          C**HELFIRE AUTOPILOT MODULE
          C**LOW  FREQ MODEL (USE DER = .005)
   2            COMMON C(3510)
   3            DIMENSION DDELTC(4),VAR(101)
          C
          C**INPUT DATA
   4            EQUIVALENCE (C( 860),TDY    )
   5            EQUIVALENCE (C( 861),GBIAS  )
   6            EQUIVALENCE (C( 862),GM     )
   7            EQUIVALENCE (C( 863),WN2    )
   8            EQUIVALENCE (C( 864),WN1    )
   9            EQUIVALENCE (C( 865),WL     )
  10            EQUIVALENCE (C( 866),WLXX1  )
  11            EQUIVALENCE (C( 867),WLXX2  )
  12            EQUIVALENCE (C( 868),WLJK1  )
  13            EQUIVALENCE (C( 869),WLJK2  )
  14            EQUIVALENCE (C( 870),WJK    )
  15            EQUIVALENCE (C( 875),GXX    )
  16            EQUIVALENCE (C( 876),GJK    )
  17            EQUIVALENCE (C( 874),QBIAS  )
  18            EQUIVALENCE (C( 879),RBIAS  )
  19            EQUIVALENCE (C(3564),OPTN4  )
          C
          C**INPUTS FROM OTHER MODULES
  20            EQUIVALENCE (C( 353),BPH1   )
  21            EQUIVALENCE (C( 354),BTH2   )
  22            EQUIVALENCE (C( 355),BPS1   )
  23            EQUIVALENCE (C( 393),BPH2   )
  24            EQUIVALENCE (C( 403),WLAMQ  )
  25            EQUIVALENCE (C( 407),WLAMR  )
  26            EQUIVALENCE (C( 461),CAGE   )
  27            EQUIVALENCE (C( 462),TKRZ   )
  28            EQUIVALENCE (C( 463),TKRY   )
  29            EQUIVALENCE (C(1739),WP     )
  30            EQUIVALENCE (C(1743),WQ     )
  31            EQUIVALENCE (C(1747),WR     )
  32            EQUIVALENCE (C(1736),WPD    )
  33            EQUIVALENCE (C(1740),WQD    )
  34            EQUIVALENCE (C(1744),WRD    )
          C
          C**INPUTS FROM MAIN PROGRAM
  35            EQUIVALENCE (C(2030),T      )
          C
          C** STATE VARIABLE OUTPUTS
  36            EQUIVALENCE (C( 800),WLOSDP)
  37            EQUIVALENCE (C( 803),WLOSP )
  38            EQUIVALENCE (C( 804),WLOSD )
  39            EQUIVALENCE (C( 807),WLOS  )
  40            EQUIVALENCE (C( 808),WLOSSD)
```

```
      EQUIVALENCE (C( 811),WLOSS )
      EQUIVALENCE (C( 812),WLRSPD)
      EQUIVALENCE (C( 815),WLRSP )
      EQUIVALENCE (C( 816),WLRSD )
      EQUIVALENCE (C( 819),WLRS  )
      EQUIVALENCE (C( 820),WLRSSD)
      EQUIVALENCE (C( 823),WLRSS )
      EQUIVALENCE (C( 824),RLOSSD)
      EQUIVALENCE (C( 827),RLOSS )
      EQUIVALENCE (C( 828),RLRSSD)
      EQUIVALENCE (C( 831),RLRSS )
C
C**OUTPUTS
      EQUIVALENCE (C( 856),RDELTC)
C
C**OTHER OUTPUTS
      EQUIVALENCE (C( 880),BXXS  )
      EQUIVALENCE (C( 881),BJJS  )
      EQUIVALENCE (C( 882),BKKS  )
      EQUIVALENCE (C( 883),BXXSS )
      EQUIVALENCE (C( 884),BJJSS )
      EQUIVALENCE (C( 885),BKKSS )
      EQUIVALENCE (C(890),WPERP)
      EQUIVALENCE (C(891),WPTIME)
C
      IF(T .LT. WPTIME) WP=WP+WPERP
C**GUIDANCE SIGNAL SHAPING
      WLOSD = WLOSP
      WLRSD = WLRSP
      WLOSDD = WN2*(WN2*(WLAMQ - WLOS) - 2.*WLOSD)
      WLRSDD = WN2*(WN2*(WLAMR - WLRS) - 2.*WLRSD)
      WQC = GN*(WLOSD/WL+WLOS) + GBIAS
      WRC = GN*(WLRSD/WL+WLRS)
C**GUIDANCE SWITCHING
      IF (OPTN4 .LE. 1.) GO TO 5
      IF (TKRZ.GT.0. .AND. T.GT.TOY) GO TO 4
      WLOSDD = 0.
      WQC = QBIAS + GBIAS
      IF (CAGE.GT.0. .AND. T.GT.TOY) WQC = WLAMQ + GBIAS
    4 IF (TKRY .GT. 0.) GO TO 5
      WLRSDD = 0.
      WRC = RBIAS
      IF (CAGE .GT. 0.) WRC = WLAMR
    5 CONTINUE
      WLOSSD = WN1*(WQC - WLOSS)
      WLRSSD = WN1*(WRC - WLRSS)
      RLOSSD = WLOSS
      BLRSSD = WLRSS
C
C**RATE GYRO DYNAMICS AND LIMITING
```

```
      CTH2 = COSD(BTH2)
      STH2 = SIND(BTH2)
      SPH1 = SIND(BPH1)
      CPH1 = COSD(BPH1)
      TPH2 = SIND(PH2)/COSD(PH2)
      TPS1 = SIND(PS1)/COSD(PS1)
      BTH    = -BTH2
      BXXS   = -BPH1
      BPS    = -BPS1
      BTHD   = -(-WQ   -(CTH2*WR   + STH2*WP  )*TPH2)
      BTHDD  = -(-WQD  -(CTH2*WRD  + STH2*WPD)*TPH2)
      BXXSD  = -(-WP   -(SPH1*WR   + CPH1*WP  )*TPS1)
      BXXSDD = -(-WPD  -(SPH1*WRD  + CPH1*WPD)*TPS1)
      BPSD   = -(       - CPH1*WR  + SPH1*WP  )
      BPSDD  = -(       - CPH1*WRD + SPH1*WPD)
C
C**SUMMATION OF RATE DAMPING AND GUIDANCE SIGNALS AND THEIR DERIVATIVES
      BJJS   =  (BLRSS -BPS   ) - (BLQSS -BTH   )
      BJJSD  =  (WLPSS -BPSD  ) - (WLQSS -BTHD  )
      BJJSDD =  (WLRSSD-BPSDD) - (WLQSSD-BTHDD)
      BKKS   = -(BLRSS -BPS   ) - (BLQSS -BTH   )
      BKKSD  = -(WLPSS -BPSD  ) - (WLQSS -BTHD  )
      BKKSDD = -(WLRSSD-BPSDD) - (WLQSSD-BTHDD)
C
C**GUIDANCE SIGNAL SHAPING AND LIMITING
      BXXSS = GXX*((BXXSDD + (WLXX1+WLXX2)*BXXSD)/(WLXX1*WLXX2) + BXXS)
      BJJSS = GJK*((BJJSDD + (WLJK1+WLJK2)*BJJSD)/(WLJK1*WLJK2) + BJJS)
      BKKSS = GJK*((BKKSDD + (WLJK1+WLJK2)*BKKSD)/(WLJK1*WLJK2) + BKKS)
      IF (ABS(BJJSS) .GT. HJK) BJJSS = SIGN(HJK,BJJSS)
      IF (ABS(BKKSS) .GT. HJK) BKKSS = SIGN(HJK,BKKSS)
C
C**COMMANDS TO ACTUATORS
      BDELTC(1) = BJJSS + BXXSS
      BDELTC(2) = BKKSS + BXYSS
      BDELTC(3) = BJJSS - BXXSS
      BDELTC(4) = BKKSS - BXXSS
      RETURN
      END
```

```
                                  SUBROUTINE C2

     SYMBOL  - - - - - - - - - - - - - - -  REFERENCES - - - - - - - - - - - - - - -

     4        -    69      73*
     5        -    68      73      77*
     ABS      -   106     107
     BDELTC   -    3EQ    52EQ    108=    109=    110=    111=
     BJJS     -   54EQ     97=    104
     BJJSD    -    98=    104
     BJJSDD   -    99=    104
     BJJSS    -   57EQ    104=    106     108     110
     BKKS     -   55EQ    100=    105
     BKKSD    -   101=    105
     BKKSDD   -   102=    105
     BKKSS    -   58EQ    105=    107     109     111
     BLQSS    -   49EQ     97     100
     BLQSSD   -   49EQ     98=
     BLPSS    -   51EQ     97     100
     BLPSSD   -   50EQ     81=
     BPH1     -   20EQ     84      85      89
     BPH2     -   23EQ
     BPS      -    90=     97     100
     BPSD     -    95=     98     101
     BPSDD    -    96=     99     102
     BPS1     -   22EQ     90
     BTH      -    88=     97     100
     BTHD     -    91=     98     101
     BTHDD    -    92=     99     102
     BTH2     -   21EQ     82      83      88
     BXXS     -   53EQ     89=    103
     BXXSD    -    93=    103
     BXXSDD   -    94=    103
     BXXSS    -   56EQ    103=    108     109     110     111
     C        -    2CO     4EQ     5EQ     6EQ     7EQ     8EQ     9EQ    10EQ    11EQ    12EQ
                  13EQ    14EQ    15EQ    16EQ    17EQ    18EQ    19EQ    20EQ    21EQ    22EQ
                  23EQ    24EQ    25EQ    26EQ    27EQ    28EQ    29EQ    30EQ    31EQ    32EQ
                  33EQ    34EQ    35EQ    36EQ    37EQ    38EQ    39EQ    40EQ    41EQ    42EQ
                  43EQ    44EQ    45EQ    46EQ    47EQ    48EQ    49EQ    50EQ    51EQ    52EQ
                  53EQ    54EQ    55EQ    56EQ    57EQ    58EQ    59EQ    60EQ
     CAGE     -   26EQ     72      76
     COSD     -    82      85      86      87
     CPH1     -    85=     93      94      95      96
     CTH2     -    82=     91      92
     G2       -     1
     GBIAS    -    5EQ     66      71      72
     GJK      -   16EQ    104     105
     GN       -    4EQ     66      67
     GXX      -   15EQ    103
     HJK      -   14EQ    106     107
     OPTN4    -   19EQ     68
     PH2      -    86
```

```
PS1      -   87
QRIAS    -   17EQ    71
RRIAS    -   18EQ    75
RETURN   -   112
SIGN     -   106     107
STND     -   83      84      86      87
SPH1     -   84=     93      94      95      96
STH2     -   83=     91      92
T        -   35EQ    61      69      72
TOY      -   4EQ     69      72
TKRY     -   28EQ    73
TKRZ     -   27EQ    69
TPH2     -   86=     91      92
TPS1     -   87=     93      94
VAP      -   70I
WL       -   9EQ     66      67
WLAMQ    -   24EQ    64      72
WLAMR    -   25EQ    65      76
WLJK1    -   12EQ    104     105
WLJK2    -   13EQ    104     105
WLOS     -   39EQ    64      66
WLOSD    -   38EQ    62=     64      66
WLOSDD   -   36EQ    64=     70=
WLOSP    -   37EQ    62
WLOSS    -   41EQ    78      86      98      101
WLOSSD   -   40EQ    78=     99      102
WLRS     -   45EQ    65      67
WLRSD    -   44EQ    63=     65      67
WLRSDD   -   42EQ    65=     74=
WLRSP    -   43EQ    63
WLRSS    -   47EQ    79      81      98      101
WLRSSD   -   46EQ    79=     99      102
WLXX1    -   10EQ    103
WLXX2    -   11EQ    103
WN1      -   8EQ     78      79
WN2      -   7EQ     64      65
WP       -   29EQ    61=     91      93      95
WPD      -   32EQ    92      94      96
WPERP    -   59EQ    61
WPTIME   -   60EQ    61
WQ       -   30EQ    91
WQC      -   66=     71=     72=     78
WQD      -   33EQ    92
WR       -   31EQ    91      93      95
WRC      -   67=     75=     76=     79
WRD      -   34EQ    92      94      96

-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

ITIALIZATION MODULE FOR HELFIRE SIMPLIFIED ACTUATOR

```
C**-INITIALIZATION MODULE FOR HELFIRE  SIMPLIFIED ACTUATOR
C******* NON - LINEAR MODEL ******
C
      SUBROUTINE C4I
      COMMON C(3510)
      DIMENSION IPL(100)
      DIMENSION BDELT(4)
      EQUIVALENCE (C(1103),BDELT1)
      EQUIVALENCE (C(1107),BDELT2)
      EQUIVALENCE (C(1111),BDELT3)
      EQUIVALENCE (C(1115),BDELT4)
      EQUIVALENCE (C(1140),OPTACT)
      EQUIVALENCE (C(1231),BDP    )
      EQUIVALENCE (C(1232),BDQ    )
      EQUIVALENCE (C(1233),BDR    )
      EQUIVALENCE (C(2561),N      )
      EQUIVALENCE (C(2562),IPL    )
      IPL(N) = 1103
      IPL(N+1) = 1104
      IPL(N+2) = 1108
      IPL(N+3) = 1112
      N = N+4
      IF (OPTACT .LE. 0.) RETURN
      IPL(N  ) = 1160
      IPL(N+1) = 1167
      IPL(N+2) = 1174
      IPL(N+3) = 1181
      IPL(N+4) = 1163
      IPL(N+5) = 1170
      IPL(N+6) = 1177
      IPL(N+7) = 1184
      N = N+8.
      C(1163) = 0.
      C(1170) = 0.
      C(1177) = 0.
      C(1184) = 0.
      C(1166) = 0.
      C(1173) = 0.
      C(1190) = 0.
      C(1197) = 0.
      IPL(N  ) = 1116
      IPL(N+1) = 1120
      IPL(N+2) = 1124
      IPL(N+3) = 1128
      N = N+4
      C(1119) = 0.
      C(1123) = 0.
      C(1127) = 0.
      C(1131) = 0.
      RETURN
```

```
      ENTRY AII
      ADELT1 = -ADP + ADQ - ADR
      ADELT2 = -ADP + ADQ + ADR
      ADELT3 =  ADP + ADQ - ADR
      ADELT4 =  ADP + ADQ + ADR
      RETURN
      END
```

```
I N D E X                                                                              PAGE  8C
                         ITIALIZATION MODULE FOR HELFIRE SIMPLIFIED ACTUATOR
      SYMBOL--- - - - - - - - - - - - - - - - - - ---REFERENCES- - - - - - - - - - - - - - - - -

      A1I        -      40
      BDELT      -      40I
      ADELT1     -       5EQ     49=
      ADELT2     -       6EQ     50=
      ADELT3     -       7EQ     51=
      ADELT4     -       8EQ     52=
      BDP        -      10EQ     49        50        51        52
      ADQ        -      11EQ     49        50        51        52
      ADR        -      12EQ     49        50        51        52
      C          -       2EQ      5EQ       6EQ       7EQ       8EQ       9EQ      10EQ      11EQ      12EQ      13EQ
                        14EQ     30=       31=       32=       33=       34=       35=       36=       37=       43=
                        44=      45=       46=
      C4T        -       1
      IPL        -      38I      14EQ      15=       16=       17=       18=       21=       22=       23=       24=
                        25=      26=       27=       28=       38=       39=       40=       41=
      N          -      13EQ     15        16        17        18        19=       21        22        23        24
                        25       26        27        28        29=       30        39        46        41        42=
      OPTACT     -       9EQ     20
      RETURN     -      20       47        53

      -+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
I N D E X
                              ELFIRE SIMPLIFIED ACTUATOR MODEL

                C** ELFIRE SIMPLIFIED ACTUATOR MODEL
                C****** NON - LINEAR MODEL ******
                C
       1              SUBROUTINE C4
                C
       2              COMMON C(3510)
       3              DIMENSION RDELTD(4),RDELT(4),DDELTC(4),VAR(101)
       4              DIMENSION  WDS(4), WDSD(4),  WDSDD(4), BDS(4), BDSD(4)
       5              DIMENSION IPL(101)
       6              DIMENSION NC2(2), CB2(6), CHDF(6)
                C
                C**INPUT DATA
       7              EQUIVALENCE (C(1140),OPTACT)
       8              EQUIVALENCE (C(1145),CR     )
       9              EQUIVALENCE (C(1146),DPELH  )
      10              EQUIVALENCE (C(1147),HDEL   )
      11              EQUIVALENCE (C(1148),W1     )
      12              EQUIVALENCE (C(1149),ZN     )
      13              EQUIVALENCE (C(1150),FMHD   )
      14              EQUIVALENCE (C(1151),G1     )
      15              EQUIVALENCE (C(1152),RH     )
      16              EQUIVALENCE (C(1153),WN     )
      17              EQUIVALENCE (C(1154),G2     )
      18              EQUIVALENCE (C(1155),H1     )
      19              EQUIVALENCE (C(1156),H2     )
      20              EQUIVALENCE (C(1306),RFAREA)
      21              EQUIVALENCE (C(1307),RFLGTH)
                C
                C**INPUTS FROM OTHER MODULES
      22              EQUIVALENCE (C(0203),PDYNMC)
      23              EQUIVALENCE (C(0204),VMACH )
      24              EQUIVALENCE (C( 856),DDELTC)
      25              EQUIVALENCE (C(1309),FMH1  )
      26              EQUIVALENCE (C(1310),FMH2  )
      27              EQUIVALENCE (C(1311),FMH3  )
      28              EQUIVALENCE (C(1312),FMH4  )
                C
                C**INPUTS FROM MAIN PROGRAM
      29              EQUIVALENCE (C(2000),T     )
      30              EQUIVALENCE (C(2013),DCG   )
      31              EQUIVALENCE (C(2561),N     )
      32              EQUIVALENCE (C(2562),IPL   )
                C
      33              DATA NC2/6.0/
      34              DATA  CB2/  .00,  .43,  .80,  .95, 1.05, 1.40/
      35              DATA CHDF/.0013,.0014,.0018,.0022,.0032,.0023/
                C**STATE VARIABLE OUTPUTS
      36              RDELT(1)=C(1103)
      37              RDELT(2)=C(1107)
      38              RDELT(3)=C(1111)
```

ELFIRE SIMPLIFIED ACTUATOR MODEL

```
      BDELT(4)=C(1115)
      WDSD (1) = C(1156)
      WDSD (2) = C(1170)
      WDSD (3) = C(1177)
      WDSD (4) = C(1184)
      WDS  (1) = C(1166)
      WDS  (2) = C(1173)
      WDS  (3) = C(1180)
      WDS  (4) = C(1187)
      BDS(1) = C(1119)
      BDS(2) = C(1123)
      BDS(3) = C(1127)
      BDS(4) = C(1131)
C
C**ACTUATOR DYNAMICS
      XF=0.
      CHD=FINTP1(VMACH,CB2,CHDF,NC2,XF,3HCHD)
      DO 30  I=1,4
      UDS = PDYNAC*REAREA
      UDSL = UDS*RFLGTH
      FMHD = CHD*UDSL*12.
      H1 = HDEL
      H2 = H1*2.
      IF (I.GE.2 .AND. I.LE.3) GO TO 5
      H2 = H1
      H1 = H1*2.
    5 CONTINUE
      A1 =        (BDELTC(I) - BDELT(I))
      A1S= A1 - SIGN(BK,A1)
      IF(ABS(A1) .LE. BK) A1S= 0.
      A2 = G1/GR*A1S
      IF(A2 .LT. -H2) A2 = -H2
      IF(A2 .GT.  H1) A2 =  H1
      BDSD(I) = H1*(A2 - BDS(I))
      BDE = BDS(I) - G2*FMHD*BDELT(I)
      WDSDD(I) = WN*(WN*(BDE - WDS(I)) - 2.*ZN*WDSD(I))
      BDELTD(I) = WDSD(I)/H1 + WDS(I)
      IF (OPTACT .LE. 0.)
     * BDELTD(I) = (A2 - G2*FMHD*BDELT(I))/(1.+G2*FMHD/H1)
C** RATE LIMIT
C**SURFACE POSITION LIMITED
      IF((ABS(BDELT(I)).GT.19).AND.((BDELTD(I).GT.0.).AND.
     *(BDELT(I).GT.0.))) BDELTD(I)=0.
   30 CONTINUE
C
      C(1103) = BDELT(1)
      C(1107) = BDELT(2)
      C(1111) = BDELT(3)
      C(1115) = BDELT(4)
C
```

```
C**OUTPUT DERIVATIVES OF STATE VARIABLES TO INTEGRATION
      C(1100) = RDELTD(1)
      C(1104) = RDELTD(2)
      C(1108) = RDELTD(3)
      C(1112) = RDELTD(4)
      C(1160) = WDSDD(1)
      C(1167) = WDSDD(2)
      C(1174) = WDSDD(3)
      C(1181) = WDSDD(4)
      C(1116) = RDSD(1)
      C(1120) = RDSD(2)
      C(1124) = RDSD(3)
      C(1128) = RDSD(4)
C
      RETURN
      END
```

```
I N D E X                                                                                PAGE   84
                                ELFIRE SIMPLIFIED ACTUATOR MODEL

SYMBOL    - - - - - - - - - - - - - - - -   REFERENCES   - - - - - - - - - - - - - - - -

S         -   62      63=
3D        -   54DO    74
APS       -   66      75
A1        -   64=     65      66
A15       -   65=     66=     67
A2        -   67=     68      69      70      74
RDF       -   71=     72
RDELH     -   9EQ
RDELT     -   39I     36=     37=     38=     39=     64      71      74      75      77
              78      79      80
RDELTG    -   30I     24EQ    64
RDELTD    -   30I     73=     74=     75      81      82      83      84
RDS       -   49I     48=     49=     50=     51=     70      71
RDSD      -   48I     70=     89      90      91      92
BM        -   15EQ    65      66
C         -   2CD     7EQ     8EQ     9EQ     10EQ    11EQ    12EQ    13EQ    14EQ    15EQ
              16EQ    17EQ    18EQ    19EQ    20EQ    21EQ    22EQ    23EQ    24EQ    25EQ
              26EQ    27EQ    28EQ    29EQ    30EQ    31EQ    32EQ    36      37      38
              39      40      41      42      43      44      45      46      47      48
              49      50      51      77=     78=     79=     80=     81=     82=     83=
              84=     85=     86=     87=     88=     89=     90=     91=     92=
CB2       -   60I     34DA    53
CHD       -   53=     57
CHDF      -   60I     35DA    53
CP        -   8EQ     67
C4        -   1
DOC       -   30EQ
FINTP1    -   53
FMHO      -   13EQ    57=     71      74
FMH1      -   25EQ
FMH2      -   26EQ
FMH3      -   27EQ
FMH4      -   28EQ
G1        -   14EQ    67
G2        -   17EQ    71      74
HDEL      -   16EQ    58
H1        -   18EQ    58=     59      61      62=     69
H2        -   19EQ    59=     61=     68
I         -   54DO    60      64      70      71      72      73      74      75
IPL       -   50I     32EQ
N         -   31EQ
NC2       -   60I     33DA    53
OPTACT    -   7EQ     74
PDYNMC    -   22EQ    55
RETURN    -   93
RFAREA    -   20EQ    55
RFLGTH    -   21EQ    56
SIGN      -   59
```

```
I N D E X                          ELFIRE SIMPLIFIED ACTUATOR MODEL

T        -    20EQ
UOS      -    55=     56
UOSL     -    56=     57
VAP      -    50I
VMACH    -    23EQ    53
WDS      -    48I     44=     45=     46=     47=     72      73
WDSD     -    40I     40=     41=     42=     43=     72      73
WDSDD    -    40I     77=     85      86      87      89
WK       -    16EQ    72
W1       -    11EQ    72      73      74
XF       -    52=     53
ZN       -    12EQ    72
  -+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
  1          SUBROUTINE G4                                                  G4A00010
           C*************************************************************  G4A00020
           C** THIS IS A SUBROUTINE (NOT A MODULE) CALLED BY STAGE 3 **    G4A00030
           C** STOPS PROGRAM AND COMPUTES MISS DISTANCE               **   G4A00040
           C*************************************************************  G4A00050
  2          COMMON C(3510)                                                 G4A00070
  3      100 FORMAT(1H0,17H MISS DISTANCE = ,1PE15.7/                       G4A00080
            *        1H0,17H FLIGHT   TIME = ,1PE15.7)                      G4A00090
  4      200 FORMAT(1H0, 9X,8HRDELX = ,1PE15.7, 8X,8HRDELY = ,1PE15.7,      G4A00100
            *        8X,8HRDELZ = ,1PE15.7)                                 G4A00110
  5      300 FORMAT(1H0,40X,8HRYFP  = ,1PE15.7, 8X,8HRZFP  = ,1PE15.7)      G4A00120
  6          EQUIVALENCE (C( 357),BGAMH )                                   G4A00130
            *.           (C( 358),BGAMV )                                   G4A00140
            *.           (C( 371),RANGE )                                   G4A00150
            *.           (C(1635),RDELX )                                   G4A00160
            *.           (C(1636),RDELY )                                   G4A00170
            *.           (C(1637),RDELZ )                                   G4A00180
  7          EQUIVALENCE (C(2003),T     )                                   G4A00190
            *.           (C(2320),LCONV )                                   G4A00200
  8          EQUIVALENCE (C( 300),RMISS )                                   G4A00210
            *.           (C( 301),RXF   )                                   G4A00220
            *.           (C( 302),RYF   )                                   G4A00230
            *.           (C( 303),RZF   )                                   G4A00240
  9          IF (RANGE .GT. 500.) GO TO 20                                  G4A00250
 10          UC13 =-SIND(BGAMV)                                             G4A00260
 11          UC33 = COSD(BGAMV)                                             G4A00270
 12          UC21 =-SIND(BGAMH)                                             G4A00280
 13          UC22 = COSD(BGAMH)                                             G4A00290
 14          UC11 =  UC22*UC33                                              G4A00300
 15          UC12 = -UC21*UC33                                              G4A00310
 16          UC31 = -UC22*UC13                                              G4A00320
 17          UC32 =  UC21*UC13                                              G4A00330
 18          RXFP = UC11*RDELX + UC12*RDELY + UC13*RDELZ                    G4A00340
 19          RYFP = UC21*RDELX + UC22*RDELY                                 G4A00350
 20          RZFP = UC31*RDELX + UC32*RDELY + UC33*RDELZ                    G4A00360
 21          IF (RXFP .GT. 0.) GO TO 10                                     G4A00370
 22          PCT = UXFP/(RXFP - UXFP)                                       G4A00380
 23          RDX = UDELX - PCT*(RDELX - UDELY)                              G4A00390
 24          RDY = UDELY - PCT*(RDELY - UDELY)                              G4A00400
 25          RDZ = UDELZ - PCT*(RDELZ - UDELZ)                              G4A00410
 26          RYF = UYFP - PCT*(RYFP - UYFP)                                 G4A00420
 27          RZF = UZFP - PCT*(RZFP - UZFP)                                 G4A00430
 28          TZERO = UT - PCT*(T - UT)                                      G4A00440
 29          RMISS = SQRT(RYF**2 + RZF**2)                                  G4A00450
 30          WRITE(6,100) RMISS, TZERO                                      G4A00460
 31          WRITE(6,200) RDX, RDY, RDZ                                     G4A00470
 32          WRITE(6,300) RYF, RZF                                          G4A00480
 33          LCONV = 2                                                      G4A00490
 34          RETURN                                                         G4A00500
 35       10 UT = T                                                         G4A00510
```

```
  36         ROELY = RDFLY                                          G4A00520
  37         ROELY = PDFLY                                          G4A00530
  38         ROFLZ = RDFLZ                                          G4A00540
  39         RXFP = RXFP                                            G4A00550
  40         RYFP = RYFP                                            G4A00560
  41         RZFP = RZFP                                            G4A00570
  42         RETURN                                                 G4A00580
  43      20 IF (ROFLZ .LT. 0.) LCONV = 2                           G4A00590
  44         RETURN                                                 G4A00600
  45         END                                                    G4A00610
```

```
I N D E X
                                   SUBROUTINE G4

          SYMBOL                                    REFERENCES

          10        -    21      35*
          20        -     9      43*
          100       -     3*     30WR
          200       -     4*     31WR
          300       -     5*     32WR
          RGAMH     -     6EQ    12      13
          RGAMV     -     6EQ    10      11
          C         -   200       6EQ     7EQ     8EQ
          COSD      -    11      13
          G4        -     1
          LCONV     -     7EQ    33=     43=
          .PCT      -    22=     23      24      25      26      27      28
          RANGE     -     6EQ     9
          RDELX     -     6EQ    18      19      20      23      36
          RDELY     -     6EQ    18      19      20      24      37
          RDELZ     -     6EQ    19      20      25      38      43
          RDX       -    23=     31WR
          RDY       -    24=     31WR
          RDZ       -    25=     31WR
          RETURN    -    34      42      44
          RMISS     -     8EQ    29=     30WR
          RXF       -     8EQ
          RXFP      -    18=     21      22      39
          RYF       -     8EQ    26=     29      32WR
          RYFP      -    19=     26      43
          RZF       -     6EQ    27=     29      32WR
          RZFP      -    20=     27      41
          SIND      -    10      12
          SQRT      -    29
          T         -     7EQ    28      35
          TZFPO     -    28=     30WR
          UC11      -    14=     18
          UC12      -    15=     18
          UC13      -    16=     16      17      18
          UC21      -    12=     15      17      19
          UC22      -    13=     14      16      19
          UC31      -    16=     20
          UC32      -    17=     20
          UC33      -    11=     14      15      20
          UDELX     -    23      36=
          UDELY     -    24      37=
          UDELZ     -    25      39=
          UT        -    28      35=
          UXFP      -    22      39=
          UYFP      -    26      40=
          UZFP      -    27      41=

          -+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

SUBROUTINE DUMPO

```
      SUBROUTINE DUMPO
      COMMON C(3510)
      DO 100 I=1, 1800, 7
      N = 0
      DO 200 J=1, 7
      K = I + J - 1
  200 IF (ABS(C(K)) .GT. 1.E-10) N = 1
  100 IF (N .GT. 0) WRITE(6,300)
     *    I,C(I),C(I+1),C(I+2),C(I+3),C(I+4),C(I+5),C(I+6)
  300 FORMAT(1H ,I5,1P7E15.7)
      RETURN
      END
```

```
I N D E X                                                                         PAGE  90
                                        SUBROUTINE DUMPO

SYMBOL    - - - - - - - - - - - - - - - -   REFERENCES  - - - - - - - - - - - - - - - - -

100         -      300     8*
250         -      500     7*
300         -      8WR     9*
ABS         -      7
G           -      260     7          8WR
DUMPO       -      1
I           -      300     6          8WR
J           -      500     6
K           -      6=      7
N           -      4=      7=         8
RETURN      -      10

-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

SUBROUTINE OINPT1

```
      SUBROUTINE OINPT1
C     BASIC INPUT SUBROUTINE OINPT1
      COMMON       C(3510)
      EQUIVALENCE (C(3214),ONAME1), (C(3264),ONAME2), (C(3319),ONAME3),
     C            (C(3329),ONAME4), (C(2361),NOMOD ), (C(2362),MODNO ),
     C            (C(3440),NORNOM), (C(3441),RNUMNO), (C(3167),NCOUT ),
     C            (C(3168),OUTNO ), (C(2461),NOSUB ), (C(2462),SUBNO ),
     C            (C(2355),TR    ), (C(2357),VR    ), (C(3339),NOSTAT),
     C            (C(3339),LOSTAT), (C(3340),STATNO), (C(3?6F),NOLIST),
     C            (C(3447),LISTNO), (C(3117),VALUE ), (C(2008),PLOTNO),
     C            (C(2339),NOPLOT), (C(2325),VLABLE), (C,K)
      EQUIVALENCE (C(1984),NPLOT )
      EQUIVALENCE (C(1985),OUTPLT)
      DIMENSION ONAME3(10),ONAME4(10)
      DIMENSION LISTNO(50), VALUE(50)
      DIMENSION SUBNO(99),IR(2),VR(2)
      DIMENSION PROMNO(50)
      DIMENSION ALPHA(3),ONAME1(50),ONAME2(55),OUTNO(50) ,MODNO(99)
      DIMENSION K(3510)
      DIMENSION STATNO (100)
      DIMENSION VLABLE(2,16)
      DIMENSION OUTPLT(15)
      REAL MODNO
      INTEGER OUTNO
      INTEGER PROMNO
      INTEGER STATNO
      INTEGER OUTPLT
      JAP = 0
      WRITE(6,31)
   31 FORMAT(11H1INPUT DATA/)
    1 READ(5,2) IR(1),ALPHA(1),ALPHA(2),ALPHA(3),IR(2),VR(1),VR(2)
      IF(EOF(5)) 50,55
   55 CONTINUE
      WRITE(6,30)IR(1),ALPHA(1),ALPHA(2),ALPHA(3),IR(2),VR(1),VR(2)
   30 FORMAT(I2,3A6,I5,5X,1P2E15.7)
    2 FORMAT(I2,3A6,I5,5X,2E15.9)
    7 IF( IR(1) .NE. 1 )  GO TO 3
      NOSUB = NOSUB + 1
      SUBNO(NOSUB) = IR(2)
      GO TO 1
    3 IF( IR(1) .NE. 2 )  GO TO 4
      NOMOD = NOMOD + 1
      MODNO(NOMOD) = IR(2)
      GO TO 1
    4 IF(IR(1) .NE. 3) GO TO 5
      L = IR(2)
      C(L) = VR(1)
      IF (VR(2) .EQ. 0.) GO TO 1
      NOLIST = NOLIST + 1
      LISTNO(NOLIST) = L
```

PLOT

SUBROUTINE OINPT1

```
      VALUE(NOLIST) = VR(1)
      GO TO 1
    5 IF(IR(1) .NE. 4)GO TO 6
      NOOUT = NOOUT + 1
      IF (NOOUT.GT.50) GO TO 1
      NAME1(NOOUT)=ALPHA(2)
      NAME2(NOOUT) = ALPHA(3)
      STNO(NOOUT) = IR(2)
      GO TO 1
    6 IF (IR(1) .NE. 5) GO TO 16
      IF (VR(1) .EQ. 0.) GO TO 17
      LOSTAT = LOSTAT + 1
   17 NOSTAT = NOSTAT + 1
      STATNO(NOSTAT) = IR(2)
      NAME3(NOSTAT)=ALPHA(2)
      NAME4(NOSTAT)=ALPHA(3)
      GO TO 1
   16 IF (IR(1).NE.7) GO TO 19
      NPLOT=NPLOT+1
      IF (NPLOT.GT.15) GO TO 1
      DO 20 I=1,2
   20 LABLE (1,NPLOT)=ALPHA(I+1)
      OUTPLT(NPLOT)=IR(2)
      GO TO 1
   19 IF (IR(2) .EQ. 0) RETURN
      N = IR(2)
      DO 12 I = 1, N
      READ (5,13) J,Y,MAND,MIER,SIGNO,BETA
      WRITE (6,13) J,Y,MAND,MIER,SIGNO,BETA
   13 FORMAT (I5,F5.0,2(3X,O12),2E15.6)
      NORNOM = NORNOM + 1
      NORNO(I) = J
      C(J) = Y
      C(J+1) = MAND
      C(J+2) = MIER
      C(J+3) = SIGNO
   12 C(J+4) = BETA
      RETURN
   50 STOP
      END
```

```
                                   SUBROUTINE OINPT1

SYMBOL         ------------------  REFERENCES  ------------------

1        -   23*     32      36      40      44      47      51      59      62      66
2        -   23RD    28*
3        -   29      33*
4        -   33      37*
5        -   37      45*
6        -   45      52*
7        -   29*
12       -   69RD    79*
13       -   70RD    71WR    72*
16       -   52      60*
17       -   53      55*
19       -   66      67*
20       -   63DO    64*
30       -   26WR    27*
31       -   21WR    22*
50       -   24      81*
55       -   24      25*
ALPHA    -   10DI    23RD    26WR    48      49      57      58      64
BETA     -   70RD    71WR    79
G        -   2EQ     3EQ     4EQ     5EQ     39=     75=     78=     79=
EOF      -   24
I        -   63DO    64      69DO    74
IR       -   3EQ     9DI     23RD    26WR    29      31      33      35      37      38
             45      50      52      56      61      65      67      68
J        -   72RD    71WR    74      75      76      77      78      79
JAR      -   25=
K        -    3EQ    11DI    76=     77=
L        -   38=     39      42
LISTNO   -    3EQ     7DI    42=
LOSTAT   -    3EQ    54=
MAND     -   70RD    71WR    76
MIER     -   70RD    71WR    77
MONNO    -    3EQ    10DI    15RL    35=
N        -   68=     69DO
NOLIST   -    3EQ    41=     42      43
NOMOD    -    3EQ    34=     35
NOOUT    -    3EQ    46=     47      48      49      50
NOPLOT   -    3EQ
NOPMOM   -    3EQ    73=
NOSTAT   -    3EQ    55=     56      57      58
NOSUB    -    3EQ    30=     31
NPLOT    -    4EQ    61=     62      64      65
OINPT1   -    1
ONAME1   -    3EQ    15DI    48=
ONAME2   -    3EQ    12DI    49=
ONAME3   -    3EQ     6DI    57=
ONAME4   -    3EQ     6DI    58=
OUTNO    -    3EQ    13DI    16IN    50=
```

```
I N D E X
                              SUBROUTINE OINPT1

OUTPLT   -    5EQ     14DI    19IN    65=
PLOTNO   -    3EQ
RETURN   -   67
RPGMNO   -    3EQ      9DI    17IN    74=
SIGNO    -   70RD     71WR    78
STATNO   -    3EQ     12DI    18IN    56=
STOP     -   81
SUBNO    -    3EQ      8DI    31=
VALUE    -    3EQ      7DI    43=
VLABLE   -    3EQ     13DI    64=
VR       -    3EQ      8DI    23RD    26WR    39      40      43      53
Y        -   70RD     71WR    75

-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

SUBROUTINE OUPT2

```
      1            SUBROUTINE OUPT2
              C    OUTPUT INITIALIZATION SUBROUTINE OUPT2
      2            COMMON        C(3510)             , GRAPH
      3            EQUIVALENCE ✓(C(2017),OTCNT ),✓(C(3167),NOOUT ),✓(C(2016),PGCNT ),
                 C             ✓(C(2014),ITCNT ),✓(C(2003),PCNT  ),✓(C(2015),CPP   ),
                 C             ✓(C(2019),TAPE  ),✓(C(2011),TAPENO),✓(C(2013),DOC   ),
                 C             ✓(C(2000),T     ),✓(C(2021),KCONV ),✓(C(2025),TIME  ),
                 C             ✓(C(2008),PLOTNO),✓(C(2009),NOPLOT),✓(C(3158),OUTNO ),
                 C             ✓(C(2024),OPNT  ),✓(C(2027),OPOINT)
      4            DIMENSION   GRAPH(300,15)   , TIME(300)         , OUTNO(50)
      5            INTEGER     PGCNT , OTCNT , OUTNO , OPOINT  .
      6            EQUIVALENCE (C(1385),OUTPLT)
      7            INTEGER OUTPLT
      8            DIMENSION OUTPLT(15)
      9            KCONV=5
     10            ITCNT = DOC + 1.0
     11            PCNT = T-0.000001
     12            PENT=PCNT                                                     PLOT  20
     13            PGCNT = 1
     14            OTCNT = (NOOUT + 4)/5
     15            IF ( ITCNT .GE. 7) GO TO 2
     16            ITCNT = ITCNT + 1
     17            CALL DUMPC
              C
     18          2 TIME(1)=T                                                     PLOT  60
     19            OPOINT =1
     20            DO 10 J=1,NOPLOT                                              PLOT  70
     21            K=OUTPLT(J)
     22         10 GRAPH(1,J)=C(K)
     23            RETURN
     24            END
```

```
I N D E X                                                                PAGE  96
                                     SUBROUTINE OUPT2
         SYMBOL   - - - - - - - - - - - - - - - - - -  REFERENCES  - - - - - - - - - - - - - - -

         2          -     15       19*
         10         -     2000     22*
         C          -      2CO      3EQ      6EQ      22
         CPP        -      3EQ
         ROG        -      3EQ     18
         DTCNT      -      3EQ      5IN      14=
         DUMPO      -     17
         GRAPH      -      2CO      4DI      22*
         ITCNT      -      3EQ     13=      15       16=
         J          -     2000     21       22
         K          -     21*      22
         KCONV      -      3EQ      9=
         NOCUT      -      3EQ     14
         NOPLOT     -      3EQ     2000
         OPOINT     -      3EQ      5IN      19=
         OUPT2      -      1
         OUTNO      -      3EQ      4DI      5IN
         OUTPLT     -      6EQ      7IN      8DI      21
         PCNT       -      3EQ     11=      12
         PGCNT      -      3EQ      5IN      13=
         PLOTNO     -      3EQ
         PPNT       -      3EQ     12=
         RETURN     -     23
         T          -      3EQ     11       18
         TAPE       -      3EQ
         TAPEND     -      3EQ
         TIME       -      3EQ      4DI      18=
```

SUBROUTINE OUPT3

```
    1          SUBROUTINE OUPT3
         C     OUTPUT SUBROUTINE OUPT3
    2          COMMON        C(3510)            , GRAPH
    3          EQUIVALENCE (C(3169),OUTNO ), (C(3218),ONAME1), (C(3268),ONAME2),
            C              (C(2017),DTCNT ), (C(3167),NOOUT ), (C(2016),PGCNT ),
            C              (C(2014),ITCNT ), (C(2003),PCNT  ), (C(2015),CPP   ),
            C              (C(2000),T     ), (C(2664),DER   ), (C(2018),TAPE  ),
            C              (C(2019),TAPEND), (C(2009),PLOTNO), (C(2009),NOPLOT),
            C              (C(2005),PPP   ), (C(2004),PPNT  ), (C(2025),TIME  ),
            C              (C(2023),OPOINT)
    4          EQUIVALENCE (C(1985),OUTPLT)
    5          DIMENSION B(50),OUTNO(50),ONAME1(50),ONAME2(50)
    6          DIMENSION TIME(300),GRAPH(300,15)                          PLOT
    7          DIMENSION OUTPLT(15)
    8          INTEGER DTCNT,PGCNT,OUTNO
    9          INTEGER OPOINT                                              PLOT
   10          INTEGER OUTPLT
   11          IF (ITCNT. GT. 6) GO TO 7
   12          ITCNT = ITCNT + 1
   13          CALL DUMPC
   14          PGCNT = 1
         C
   15        7 IF (DER. EQ. DER1) GO TO 8
   16          DER1 = DER
   17          WRITE(6,23)T,DER                                           STEPSIZE
   18       23 FORMAT(1H ,5HTIME=F14.7,2X,10HSTEP SIZE=1PE19.7)           STEPSIZE
   19        8 IF (T .LT. PCNT)GOTO15                                      PLOT
   20        9 PCNT = PCNT + CPP
   21          IF (PGCNT. NE. 1) GO TO 3
   22        1 WRITE(6,2) (ONAME1(I),ONAME2(I), I=1.NOOUT)
   23        2 FORMAT (1H1,3X,4HTIME,5X,5(7X,2A6)//(25X,2A6,7X,2A6,7X,2A6,7X,
              12A6,7X,2A6)/)
   24          PGCNT = 2*DTCNT + 4
   25        3 IF(PGCNT .GE. 66) GO TO 1
   26          DO 4 I = 1,NOOUT
   27          J = OUTNO(I)
   28        4 B(I) = C(J)
   29          WRITE (6,5) T,(B(I), I = 1,NOOUT)
   30        5 FORMAT (///,F14.7,1P5E19.7/(14X,1P5E19.7))
   31          PGCNT = PGCNT + DTCNT + 4
   32       15 IF(T.LT.PPNT.OR.NOPLOT.EQ.0)RETURN
   33          PPNT=PPNT+PPP                                               PLOT
   34          KPOINT =OPOINT +1                                           PLOT
   35          IF (KPOINT-300) 16,13,18
   36       13 WRITE (6,14)
   37       14 FORMAT (///1H **** WARNING-PLOTTING ARRAY FILLED-ONLY FIRST 300 P
              OINTS PLOTTED ****,//)
   38       16 OPOINT=KPOINT
   39          TIME (OPOINT)=T
   40          DO 10 J=1,NOPLOT                                            PLOT
```

SUBROUTINE OUPT3

```
         K=OUTPLT(J)
      1F GRAPH(OPOINT   ,J)=C(K)                                PLOT
      10 RETURN
         END
```

```
I N D E X                                                                        PAGE  99
                                       SUBROUTINE OUPT3

SYMBOL     - - - - - - - - - - - - - - - - - - REFERENCES - - - - - - - - - - - - - - - - - - -

1          -    22*     25
2          -    22WR    23*
3          -    21      25*
4          -    2600    28*
5          -    29WR    30*
7          -    11      15*
8          -    15      19*
9          -    28*
10         -    4000    42*
13         -    35      36*
14         -    36WR    37*
15         -    19      32*
16         -    35      38*
18         -    35      43*
20         -    17WR    18*
B          -     5DI    28=     29WR
C          -     2CO    3EQ     4EQ     28      42
CPP        -     3EQ    20
DER        -     3EQ    15      16      17WR
GFR1       -    15      16*
DTCNT      -     3EQ     8IN    24      31
DUMPO      -    13
GRAPH      -     2CO     6DI    42=
I          -    22WR    2600    27      28      29WR
ITCNT      -     3EQ    11      12=
J          -    27=     28      4000    41      42
K          -    41=     42
KPOINT     -    34=     35      38
NOOUT      -     3EQ    22WR    2600    29WR
NOPLOT     -     3EQ    32      4000
ONAME1     -     3EQ     5DI    22WR
ONAME2     -     3EQ     5DI    22WR
OPOINT     -     3EQ     9IN    34      38=     39      42
OUPT3      -     1
OUTNO      -     3EQ     5DI     8IN    27
OUTPLT     -     4EQ     7DI    10IN    41
PCNT       -     3EQ    19      20=
PGCNT      -     3EQ     8IN    14=     21      24=     [illegible]      31=
PLOTNO     -     3EQ
PPNT       -     3EQ    32      33=
PPP        -     3EQ    33
RETURN     -    32      43
T          -     3EQ    17WR    19      29WR    32      39
TAPE       -     3EQ
TAPEND     -     3EQ
TIME       -     3EQ     6DI    39=

-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

SUBROUTINE ZERO

```
      SUBROUTINE ZERO
      COMMON C(3501.)
      EQUIVALENCE (C(1984),NPLOT )
      EQUIVALENCE (C(2223),OPOINT)
      EQUIVALENCE (C(2361),NOMOU )
      EQUIVALENCE (C(2461),NOSUB )
      EQUIVALENCE (C(3066),NOLIST)
      EQUIVALENCE (C(3167),NOOUT )
      EQUIVALENCE (C(3339),LOSTAT)
      EQUIVALENCE (C(3439),NOSTAT)
      EQUIVALENCE (C(3440),NORNOM)
      INTEGER OPOINT
      LOSTAT = 0
      NOSTAT = 0
      NOSUB = 0
      NOMOD = 0
      NOOUT = 0
      NOPNOM = 0
      NOLIST = 0
      OPOINT=0
      NPLOT=0
      RETURN
      END
```

```
I N D E X
                                        SUBROUTINE ZERO

  SYMBOL  - - - - - - - - - - - - - - - - - -REFERENCES- - - - - - - - - - - - - - - -

  C         -     2C0    3EQ    4EQ    5EQ    6EQ    7EQ    8EQ    9EQ    10EQ   11EQ
  LOSTAT    -     9EQ    13=
  NOLIST    -     7EQ    19=
  NOPOD     -     5EQ    16=
  NOPUT     -     8EQ    17=
  NOPNCH    -     11EQ   18=
  NOSTAT    -     10EQ   14=
  NOSUB     -     6EQ    15=
  NPLOT     -     3EQ    21=
  OPOINT    -     4EQ    12IN   20=
  RETURN    -     22
  ZERO      -     1
```

SUBROUTINE SUBL1

```
      SUBROUTINE   SUBL1
      COMMON       C(3512)
      EQUIVALENCE (C(2461),NOSUB ), (C(2462),SUBNO )
      DIMENSION    SUBNO(99)
      DO 1 I = 1, NOSUB                                        SUBL1***
      J = SUBNO(I)                                             SUBL1***
      GO TO ( 1, 2, 3, 4, 5, 6, 7, 8, 9 ), J                   SUBL1***
    2 CALL INPT1                                               SUBL1***
      GO TO 1                                                  SUBL1***
    3 CALL OUPT1                                               SUBL1***
      GO TO 1                                                  SUBL1***
    4 CALL STGE1                                               SUBL1***
      GO TO 1                                                  SUBL1***
    5 CALL CNTR1                                               SUBL1***
      GO TO 1                                                  SUBL1***
    6 CALL RNDM1                                               SUBL1***
      GO TO 1                                                  SUBL1***
    7 CALL AUXA1                                               SUBL1***
      GO TO 1                                                  SUBL1***
    8 CALL AUXB1                                               SUBL1***
      GO TO 1                                                  SUBL1***
    9 CALL AUXC1                                               SUBL1***
    1 CONTINUE                                                 SUBL1***
      RETURN                                                   SUBL1***
      END                                                      SUBL1***
```

```
I N D E X                                                                      PAGF 103
                                       SUBROUTINE SUBL1

      SYMBOL  - - - - - - - - - - - - - - - REFERENCES - - - - - - - - - - - - - - - - - -

      1          -     5DO     7         9        11        13        15        17        19        21        23*
      2          -     7       8*
      3          -     7       10*
      4          -     7       12*
      5          -     7       14*
      6          -     7       16*
      7          -     7       18*
      8          -     7       20*
      9          -     7       22*
      AUXA1      -    18
      AUXB1      -    26
      AUYC1      -    22
      C          -     2C0     3EC
      CNTR1      -    14
      I          -     5D0     6
      INPT1      -     8
      J          -     6=      7
      MOSUB      -     3EQ     500
      OUPT1      -    10
      RETURN     -    24
      RNDM1      -    16
      STGE1      -    12
      SUPL1      -     1
      SUBNO      -     3EQ     4DI       6

      -+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```

SUBROUTINE SUBL2

```
      SUBROUTINE   SUBL2
                                       ( B L A N K   C A R D )
      COMMON        C(3510)
      EQUIVALENCE  (C(2461),NOSUB ), (C(2462),SUBNO )
      DIMENSION     SUBNO(99)
      DO 1 I = 1, NOSUB                                                 SUBL2***
      J = SUBNO(I)                                                      SUBL2***
      GO TO ( 1, 2, 3, 4, 5, 6, 7, 8, 9 ), J                            SUBL2***
    2 CALL INPT2                                                        SUBL2***
      GO TO 1                                                           SUBL2***
    3 CALL OUPT2                                                        SUBL2***
      GO TO 1                                                           SUBL2***
    4 CALL STGE2                                                        SUBL2***
      GO TO 1                                                           SUBL2***
    5 CALL CNTR2                                                        SUBL2***
      GO TO 1                                                           SUBL2***
    6 CALL RNDM2                                                        SUBL2***
      GO TO 1                                                           SUBL2***
    7 CALL AUXA2                                                        SUBL2***
      GO TO 1                                                           SUBL2***
    8 CALL AUXB2                                                        SUBL2***
      GO TO 1                                                           SUBL2***
    9 CALL AUXC2                                                        SUBL2***
    1 CONTINUE                                                          SUBL2***
      RETURN                                                            SUBL2***
       END                                                              SUBL2***
```

```
I N D E X                                                                              PAGE 105
                                              SUBROUTINE SUBL2
SYMBOL        - - - - - - - - - - - - - - - -  REFERENCES  - - - - - - - - - - - - - - - -

1          -    500     7       9       11      13      15      17      19      21      23*
2          -    7       8*
3          -    7       10*
4          -    7       12*
5          -    7       14*
6          -    7       16*
7          -    7       18*
8          -    7       20*
9          -    7       22*
AUXA2      -   18
AUXB2      -   20
AUXC2      -   22
C          -    200     3EQ
CNTR2      -   14
I          -    500     6
INPT2      -    8
J          -    6*      7
NOSUB      -    3EQ     500
OUPT2      -   10
RETURN     -   24
RNDM2      -   16
STKE2      -   12
SUBL2      -    1
SUBNO      -    3EQ     4DI     6
-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```

SUBROUTINE SUBL3

```
      SUBROUTINE   SUBL3
      COMMON       C(3512)
      EQUIVALENCE  (C(2461),NOSUB ), (C(2462),SUBNO )
      DIMENSION    SUBNO(99)
      DO 1 I = 1, NOSUB                                       SUBL3***
      J = SUBNO(I)                                            SUBL3***
      GO TO ( 1, 2, 3, 4, 5, 6, 7, 8, 9 ), J                  SUBL3***
    2 CALL INPT3                                              SUBL3***
      GO TO 1                                                 SUBL3***
    3 CALL OUPT3                                              SUBL3***
      GO TO 1                                                 SUBL3***
    4 CALL STGF3                                              SUBL3***
      GO TO 1                                                 SUBL3***
    5 CALL CNTR3                                              SUBL3***
      GO TO 1                                                 SUBL3***
    6 CALL RNDM3                                              SUBL3***
      GO TO 1                                                 SUBL3***
    7 CALL AUXA3                                              SUBL3***
      GO TO 1                                                 SUBL3***
    8 CALL AUXB3                                              SUBL3***
      GO TO 1                                                 SUBL3***
    9 CALL AUXC3                                              SUBL3***
    1 CONTINUE                                                SUBL3***
      RETURN                                                  SUBL3***
       END                                                    SUBL3***
```

```
I N D E X
                              SUBROUTINE SUBL3

SYMBOL   * * * * * * * * * * * * * * * *  REFERENCES  * * * * * * * * * * * * * *

1         -     500    7      9      11      13      15      17      19      21      23*
2         -     7      9*
3         -     7      10*
4         -     7      12*
5         -     7      14*
6         -     7      16*
7         -     7      18*
8         -     7      20*
9         -     7      22*
AUXA3     -     18
AUXB3     -     25
AUXC3     -     22
C         -     200    3EQ
CNTR3     -     14
I         -     500    6
INPT3     -     8
J         -     6*     7
NOSUB     -     3EQ    500
OUPT3     -     10
RETURN    -     24
RNDM3     -     16
STGE3     -     12
SUBL3     -     1
SUBNO     -     3EQ    4DI    6
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

SUBROUTINE STGE2

```
      SUBROUTINE STGE2
      COMMON      C(3510)
      EQUIVALENCE (C(2011),KSTEP ), (C(2020),LCONV ), (C(2021),KCONV )
      KCONV = 0
      LCONV = 0
      KSTEP = 1
      RETURN
      END
```

```
I N D E X                                                                          PAGE 109
                                          SUBROUTINE STGE2

SYMBOL ------------------------------- REFERENCES ---------------------------------

C          -       2GO       3EQ
KCONV      -       3EQ       4=
KSTEP      -       3EQ       6=
LCONV      -       3EQ       5=
RETURN     -       7
STGE2      -       1
```

SUBROUTINE STGE3

```
      SUBROUTINE STGE3
      COMMON C(3510),GRAPH,TEMPS(1000)
      EQUIVALENCE (C(2001),T     ),(C(2002),TF    ),(C(2003),PCNT  )
      EQUIVALENCE (C(2010),STEP  ),(C(2011),KSTEP ),(C(2020),LCONV )
      EQUIVALENCE (C(2021),KCONV ),(C(2661),N     ),(C(2662),HMIN  )
      EQUIVALENCE (C(2663),HMAX  ),(C(2664),DLR   ),(C(2765),EL    )
      EQUIVALENCE (C(2865),FU    ),(C(2965),VAR   )
      EQUIVALENCE (C(1973),KASE  ),(C(1974),NJ    ),(C(1975),NPT   )
      DIMENSION   DER(100)       , VAR(100)       , EL(100)
      DIMENSION   FU(100)        , GRAPH(300,15)
      EXTERNAL AUXSUB
      CALL G4
      IF (ABS( T-TF) .LE. 0.01 ) GO TO 20
      IF ( (TF-T) .LT. 0.) GO TO 10
      IF (LCONV .EQ. 2) GO TO 20
      IF (LCONV .EQ. 1) GO TO 10
      IF(DER(1) .LT. 0.) DER(1)=-DER(1)*.5
      RETURN
   10 IF(DER(1) .GT. 0.) DER(1)=-DER(1)*.5
      KCONV = KCONV + 1
      IF(KCONV .GE. 10) GO TO 20
      RETURN
   20 PCNT = 1.0
      IF(STEP .EQ.11.)GOTO 40
      PREDER = DER(1)
      DER(1) = 0.
      NJ=N-1
      NPT=9
      CALL AMRK(AUXSUB)
      DER(1) = PREDER
   40 CALL OUPT3
      KSTEP = 2
      RETURN
      END
```

```
I N D E X                                                                  PAGE 111
                                 SUBROUTINE STGE3

          SYMBOL    - - - - - - - - - - - - - - - -  REFERENCES - - - - - - - - - - - - - - - -

          10       -   14      16      19*
          20       -   13      15      21      23*
          40       -   24      31*
          ABS      -   13
          AMPK     -   29
          AUXSUB   -   11EX    20AG
          C        -    2EQ     3EQ     4EQ     5EQ     6EQ     7EQ     8EQ
          DFP      -    4EQ     9DI    17      19      25      26=     30=
          EL       -    6EQ     9DI
          EU       -    7EQ    10DI
          GRAPH    -    2EQ    10DI
          G4       -   12
          HMAX     -    6EQ
          HMIN     -    5EQ
          KASE     -    8EQ
          KCONV    -    5EQ    20=     21
          KSTEP    -    4EQ    32=
          LCONV    -    4EQ    15      16
          N        -    5EQ    27
          NJ       -    4EQ    27=
          NPT      -    8EQ    28=
          OUPT3    -   31
          PCNT     -    3EQ    23=
          PREDER   -   25=     30
          RETURN   -   18      22      33
          STEP     -    4EQ    24
          STGE3    -    1
          T        -    3EQ    13      14
          TEMPS    -    2EQ
          TF       -    3EQ    13      14
          VAR      -    7EQ     9DI

          -+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
   1         SUBROUTINE AMRK(AUXSUB)                                                AMRKI002
       C*SINGLE PRECISION VERSION* INDEPENDENT VARIABLE IN DOUBLE PRECISION         AMRKI003
   2         COMMON      C(3510)          , GRAPH            , T(1000)
   3         DIMENSION D(101)            ,FL(100)           ,FU(100)                AMRKI006
   4         DIMENSION   V(101)           , GRAPH(300,15)
   5         EQUIVALENCE (C(2662),HMIN  ), (C(2663),HMAX  ), (C(2664),D     )       AMRKI010
   6         EQUIVALENCE (C(2765),FL    ), (C(2865),FU    ), (C(2965),V     )       AMRKI012
   7         EQUIVALENCE (C(1971),RITE  ), (J1,N1)                                  AMRKI014
   8         EQUIVALENCE (C(1973),KASE  ), (C(1974),NJ    ), (C(1975),NPT   )
   9         DOUBLE PRECISION DELT,TME                                              AMRKI021
  10         DOUBLE PRECISION NEWC(200),NEWP(200),OLD(200)                          AMRKI022
  11         DATA KOUNT/0/                                                          AMRKI024
  12         DATA P1,P2,P3,P4/2.29166667,2.4583333,1.5416667,0.375/                 AMRKI026
  13         DATA C2,C3,C4/0.7916667,0.2083333,0.0416667/                           AMRKI028
  14         IF(KASE.GT.0)GO TO 20                                                  AMRKI032
  15         N1=NJ
  16         J2=J1+N1                                                               AMRKI038
  17         J3=J2+N1                                                               AMRKI041
  18         J4=J3+N1                                                               AMRKI044
  19         J5=J4+N1                                                               AMRKI047
  20         J6=J5+N1                                                               AMRKI050
  21         J7=J6+N1                                                               AMRKI053
  22         J8=J7+N1                                                               AMRKI056
  23         J9=J8+N1                                                               AMRKI059
  24         KASE=KASE+1                                                            AMRKI073
       C*NPT.EQ.0  ADAMS-MOULTON INTEGRATION MODE                                   AMRKI075
       C*NPT.EQ.1  RUNGE-KUTTA INTEGRATION MODE                                     AMRKI079
       C*NPT.EQ.2  BEGINNING ADAMS-MOULTON WITH RUNGE-KUTTA START                   AMRKI083
  25      20 IF(NPT.EQ.1)GO TO 40                                                   AMRKI087
  26         IF(NPT.EQ.2)GO TO 30                                                   AMRKI091
  27         IF(SNGL(DELT).NE.(0.5*D(1)))GO TO 30                                   AMRKI093
  28         IF(KOUNT.LT.3)GO TO 40                                                 AMRKI095
  29         GO TO 200                                                              AMRKI097
  30      30 KOUNT=0                                                                AMRKI099
  31         NPT=3                                                                  AMRKI103
       C*START RUNGE-KUTTA INTEGRATION                                              AMRKI107
       C*COMPUTE K1                                                                 AMRKI111
  32      40 DO 50 I=1,N1                                                           AMRKI115
  33         IF(SNGL(NEWP(I)).NE.V(I+1))NEWP(I)=V(I+1)                              AMRKI119
  34      50 CONTINUE                                                               AMRKI123
  35         IF(SNGL(TME).NE.V(1))TME=V(1)                                          AMRKI127
  36         KOUNT=KOUNT+1                                                          AMRKI131
  37         DO 60 I=1,N1                                                           AMRKI135
  38      60 OLD(I)=D(1)*D(I+1)                                                     AMRKI139
       C COMPUTE K2                                                                 AMRKI143
  39         DELT=0.5*D(1)                                                          AMRKI147
  40         TME=TME+DELT                                                           AMRKI151
  41         V(1)=TME                                                               AMRKI155
  42         DO 70 I=1,N1                                                           AMRKI159
  43         IF(KOUNT.NE.2)GO TO 65                                                 AMRKI161
```

```
 44           K1=J9+I                                      AMRKI163
 45           T(K1)=NEWP(I)                                AMRKI171
 46        65 T(I)=D(I+1)                                  AMRKI175
 47           NEWP(I)=NEWP(I)+0.5*OLD(I)                   AMRKI187
 48        70 V(I+1)=NEWP(I)                               AMRKI195
 49           CALL AUXSUB                                  AMRKI199
 50           DO 80 I=1,N1                                 AMRKI203
 51        80 NEWC(I)=D(1)*D(I+1)                          AMRKI207
       C COMPUTE K3                                        AMRKI211
 52           DO 90 I=1,N1                                 AMRKI215
 53           NEWP(I)=NEWP(I)+0.5*(NEWC(I)-OLD(I))         AMRKI219
 54        90 V(I+1)=NEWP(I)                               AMRKI223
 55           CALL AUXSUB                                  AMRKI227
 56           DO 100 I=1,N1                                AMRKI231
 57           K2=J7+I                                      AMRKI235
 58       100 T(K2)=D(1)*D(I+1)                            AMRKI239
       C COMPUTE K4                                        AMRKI243
 59           TME=TME+DELT                                 AMRKI247
 60           V(1)=TME                                     AMRKI251
 61           DO 110 I=1,N1                                AMRKI255
 62           K2=J7+I                                      AMRKI259
 63           NEWP(I)=NEWP(I)+T(K2)-0.5*NEWC(I)            AMRKI263
 64       110 V(I+1)=NEWP(I)                               AMRKI267
 65           CALL AUXSUB                                  AMRKI271
 66           DO 120 I=1,N1                                AMRKI275
 67           K3=J8+I                                      AMRKI279
 68       120 T(K3)=D(1)*D(I+1)                            AMRKI283
       C COMPUTE VALUE OF FUNCTION                         AMRKI287
 69           DO 130 I=1,N1                                AMRKI291
 70           K2=J7+I                                      AMRKI295
 71           K3=J8+I                                      AMRKI299
 72           NEWP(I)=NEWP(I)-T(K2)+0.16666667*            AMRKI303
             X(OLD(I)+NEWC(I)+NEWC(I)+T(K2)+T(K2)+T(K3))   AMRKI307
 73       130 V(I+1)=NEWP(I)                               AMRKI311
 74       140 CALL AUXSUB                                  AMRKI315
 75           DO 150 I=1,N1                                AMRKI319
 76           K5=J1+I                                      AMRKI323
 77           K6=J2+I                                      AMRKI327
 78           K1=J3+I                                      AMRKI331
 79           K2=J4+I                                      AMRKI335
 80           K3=J5+I                                      AMRKI339
 81           K4=J6+I                                      AMRKI343
 82           T(K4)=T(K3)                                  AMRKI347
 83           T(K3)=T(K2)                                  AMRKI351
 84           T(K2)=T(K1)                                  AMRKI355
 85           T(K1)=T(K6)                                  AMRKI359
 86           T(K6)=T(K5)                                  AMRKI363
 87           T(K5)=T(I)                                   AMRKI367
 88       150 T(I)=D(I+1)                                  AMRKI371
 89           RETURN                                       AMRKI375
```

SUBROUTINE AMRK(AUXSUB)

```
C ADAMS-MOULTON INTEGRATION                                             AMRKI379
  200 KOUNT=KOUNT+1                                                     AMRKI387
      DELT=0.5*D(1)                                                     AMRKI391
      DO 210 I=1,N1                                                     AMRKI395
      K1=J2+I                                                           AMRKI399
      K2=J3+I                                                           AMRKI403
      K4=J1+I                                                           AMRKI407
C COMPUTE Y-PREDICTED                                                   AMRKI411
      OLD(I)=NEWP(I)                                                    AMRKI415
      NEWP(I)=OLD(I)+D(1)*(P1*T(I)-P2*T(K4)+P3*T(K1)-P4*T(K2))          AMRKI419
  210 Y(I+1)=NEWP(I)                                                    AMRKI423
      TME=TME+D(1)                                                      AMRKI427
      Y(1)=TME                                                          AMRKI431
      CALL AUXSUB                                                       AMRKI435
      K5=0                                                              AMRKI439
      DO 250 I=1,N1                                                     AMRKI443
      K2=J2+I                                                           AMRKI447
      K4=J1+I                                                           AMRKI451
C COMPUTE Y-CORRECTED                                                   AMRKI455
      NEWC(I)=OLD(I)+D(1)*(C1*D(I+1)+C2*T(I)-C3*T(K4)+C4*T(K2))         AMRKI459
      IF(HMIN.EQ.HMAX)GO TO 250                                         AMRKI463
      TEMP=ABS(SNGL(NEWC(I)-NEWP(I)))                                   AMRKI467
      IF(TEMP.LT.EU(I))GO TO 240                                        AMRKI471
      IF(RITE.LE.0.0)GO TO 230                                          AMRKI475
      SPTIME=SNGL(TME)                                                  AMRKI479
      WRITE(6,220)I,SPTIME,TEMP                                         AMRKI483
  220 FORMAT(1H ,14HSTATE VARIABLE,I3,26H  EXCEEDED TOLERANCE ERROR.    AMRKI487
     *7H  TIME=,F14.7,9H  TEMP=  ,1PE17.9)                              AMRKI491
  230 IF(ABS(SNGL(DELT)).GE.HMIN)GO TO 270                              AMRKI495
  240 IF(TEMP.LT.EL(I))K5=K5+1                                          AMRKI499
  250 CONTINUE                                                          AMRKI503
      IF(K5.LT.N1)GO TO 290                                             AMRKI507
      IF(ABS(D(1)+D(1)).GT.HMAX)GO TO 290                               AMRKI511
C SET-UP FOR DOUBLING STEP SIZE                                         AMRKI515
      IF(KOUNT.LE.5)GO TO 290                                           AMRKI519
      DO 260 I=1,N1                                                     AMRKI523
      K1=J1+I                                                           AMRKI527
      K2=J2+I                                                           AMRKI531
      K3=J3+I                                                           AMRKI535
      K5=J5+I                                                           AMRKI539
      T(I)=T(K1)                                                        AMRKI543
      T(K1)=T(K3)                                                       AMRKI547
  260 T(K2)=T(K5)                                                       AMRKI551
      D(1)=D(1)+D(1)                                                    AMRKI555
      KOUNT=4                                                           AMRKI559
      DELT=0.5*D(1)                                                     AMRKI563
      GO TO 290                                                         AMRKI567
C SET-UP FOR HALVING STEP SIZE                                          AMRKI571
  270 IF(KOUNT.LE.4)GO TO 310                                           AMRKI575
      TME=TME-D(1)                                                      AMRKI579
```

SUBROUTINE AMRK(AUXSUB)

```
      V(1)=TME                                                        AMRKI583
      D(1)=DELT                                                       AMRKI587
      DELT=0.5*D(1)                                                   AMRKI591
      DO 280 I=1,N1                                                   AMRKI595
      K1=J1+I                                                         AMRKI599
      K2=J2+I                                                         AMRKI603
      K3=J3+I                                                         AMRKI607
      NEWP(I)=OLD(I)                                                  AMRKI611
      V(I+1)=OLD(I)                                                   AMRKI615
      T(K3)=T(K2)+0.5*(T(K1)-T(K2))                                   AMRKI619
      T(K2)=T(K1)                                                     AMRKI623
  280 T(K1)=T(K1)+0.5*(T(I)-T(K1))                                    AMRKI627
      KOUNT=4                                                         AMRKI631
      GO TO 290                                                       AMRKI635
C INTEGRATION IS FINISHED. SET UP DERIVATIVES AND EXIT.                AMRKI639
  290 DO 300 I=1,N1                                                   AMRKI643
      NEWP(I)=NEWC(I)                                                 AMRKI647
  300 V(I+1)=NEWC(I)                                                  AMRKI651
      GO TO 140                                                       AMRKI655
C RETURN TO 3RD PRECEDING POINT AND RESTART RK                         AMRKI659
  310 DO 320 I=1,N1                                                   AMRKI663
      K1=J9+I                                                         AMRKI667
      NEWP(I)=T(K1)                                                   AMRKI671
  320 V(I+1)=T(K1)                                                    AMRKI675
      TME=TME-4.5*D(1)                                                AMRKI679
      V(1)=TME                                                        AMRKI683
      D(1)=DELT                                                       AMRKI687
      CALL AUXSUB                                                     AMRKI691
      GO TO 30                                                        AMRKI695
      END                                                             AMRKI750
```

```
I N D E X                                                                          PAGE 116
                                        SUBROUTINE AMPK(AUXSUB)

      SYMBOL    - - - - - - - - - - - - - - - -    REFERENCES  - - - - - - - - - - - - - - - - -

      20        -       14       25*
      30        -       26       27       30*      160
      40        -       25       28       32*
      50        -       3200     34*
      60        -       3700     38*
      65        -       43       46*
      70        -       4200     48*
      80        -       5000     51*
      90        -       5200     54*
      100       -       5600     58*
      110       -       6100     64*
      120       -       6600     68*
      130       -       6900     73*
      140       -       74*      151
      150       -       7500     83*
      200       -       29       94*      147
      210       -       9200     98*
      220       -       11200    113*
      230       -       111      114*
      240       -       109      115*
      250       -       10300    107      116*
      260       -       12500    127*
      270       -       114      132*
      280       -       13700    145*
      290       -       117      118      119      131      148*
      300       -       14800    150*
      310       -       132      152*
      320       -       15200    155*
      ABS       -       108      114      118
      AMPK      -        1
      AUXSUB    -        1AG     49       55       65       74       101      159
      G         -       2CO      5EQ      6EQ      7EQ      8EQ
      C2        -       130A     106
      C3        -       130A     136
      C4        -       130A     106
      D         -        30I     5EQ      27       38       39       46       51       58       68       88
                        91       97       99       106      118      128=     130      133      135=     136
                        156      158=
      DELT      -       900      27       39=      40       59       91=      114      130=     135      136=
                       158
      EL        -       30I      6EQ      115
      EU        -       30I      6EQ      109
      GRAPH     -       2CO      40I
      HMAX      -       5EQ      107      118
      HMIN      -       5EQ      107      114
      I         -       3200     33       3700     38       4200     44       45       46       47       48
                        5600     51       5200     53       54       5600     57       58       6100     62
                        63       64       6600     67       68       6900     70       71       72       73
```

```
I N D E X                                                                        PAGE 117
                                     SUBROUTINE APPK(AUXSUP)

                 7500     76      77      78      79      80      81      87      88      9200
                 93       94      95      96      97      98      10300   104     105     106
                 108      109     112WR   113     12000   121     122     123     124     125
                 13700    138     139     140     141     142     145     14800   149     150
                 15200    153     154     155
J1        -      7EQ      16      76      95      105     121     138
J2        -      16=      17      77      93      104     122     139
J3        -      17=      18      78      94      123     140
J4        -      18=      19      79
J5        -      19=      20      80      124
J6        -      20=      21      81
J7        -      21=      22      57      62      73
J8        -      22=      23      67      71
J9        -      23=      44      153
KASE      -      8EQ      14      24=
KOUNT     -      110A     28      30=     36=     43      90=     119     129=    132     146=
K0        -      77=      85      86
K1        -      44=      45      78=     84      85      93=     97      121=    125     126
                 138=     143     144     145     153=    154     155
K2        -      57=      58      62=     63      72=     72      79=     83      84      94=
                 97       104=    106     122=    127     139=    143     144
K3        -      67=      68      71=     72      80=     82      83      123=    126     140=
                 143
K4        -      81=      82      95=     97      105=    106
K5        -      76=      86      87      102=    115=    117     124=    127
NEWC      -      109R     51=     53      63      72      106=    108     149     150
NEWP      -      109R     33      45      47=     48      53=     54      63=     64      72=
                 73       96      97=     98      103     141=    149=    154=
NJ        -      8EQ      15
NPT       -      8EQ      25      26      31=
N1        -      7EQ      15=     16      17      18      19      20      21      22      23
                 3200     3700    4200    5000    5200    5600    6100    6600    6900    7500
                 9200     10300   117     12000   13700   14800   15200
OLD       -      109R     38=     47      53      72      96=     97      106     141     143
P1        -      120A     97
P2        -      120A     97
P3        -      120A     97
P4        -      120A     97      106
RETURN    -      89
RITE      -      7EQ      110
SNGL      -      27       33      35      108     111     114
SPTIME    -      111=     112WR
T         -      20D      45=     46=     58=     63      68=     72      82=     83=     84=
                 85=      86=     87=     88=     97      106     125=    126=    127=    143=
                 144=     145=    154     155
TEMP      -      108=     109     112WR   115
TPE       -      90B      35      40=     41      59=     60      99=     100     111     133=
                 134      156=    157
V         -      40F      6EQ     33      35      41=     48=     54=     62=     64=     73=
                 98=      100=    134=    142=    150=    155=    157=
```

SUBROUTINE AMRK(AUXSUB)

```
-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     1            SUBROUTINE AUXI
     2            COMMON      C(3510)
     3            EQUIVALENCE (C(2361),NOMOD ), (C(2362),XMODNO), (C(2561),N      )
     4            DIMENSION  XMODNO(99)
     5            N = 1                                                           AUXIX***
     6            DO 1 I=1,NOMOD                                                  AUXIX***
     7            L =XMODNO(I)
     8            GO TO (1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23
                1      ,24,25,26,27,28,29,30,31,32,33,34,35,36,37),L
     9          2 CALL  A1I                                                       AUXIX***
    10            GO TO 1                                                         AUXIX***
    11          3 CALL  A2I                                                       AUXIX***
    12            GO TO 1                                                         AUXIX***
    13          4 CALL  A3I                                                       AUXIX***
    14            GO TO 1                                                         AUXIX***
    15          5 CALL  A4I                                                       AUXIX***
    16            GO TO 1                                                         AUXIX***
    17          6 CALL  A5I                                                       AUXIX***
    18            GO TO 1                                                         AUXIX***
    19          7 CALL  C1I                                                       AUXIX***
    20            GO TO 1                                                         AUXIX***
    21          8 CALL  C2I                                                       AUXIX***
    22            GO TO 1                                                         AUXIX***
    23          9 CALL  C3I                                                       AUXIX***
    24            GO TO 1                                                         AUXIX***
    25         10 CALL  C4I                                                       AUXIX***
    26            GO TO 1                                                         AUXIX***
    27         11 CALL  C5I                                                       AUXIX***
    28            GO TO 1                                                         AUXIX***
    29         12 CALL  C6I                                                       AUXIX***
    30            GO TO 1                                                         AUXIX***
    31         13 CALL  C7I                                                       AUXIX***
    32            GO TO 1                                                         AUXIX***
    33         14 CALL  C8I                                                       AUXIX***
    34            GO TO 1                                                         AUXIX***
    35         15 CALL  C9I                                                       AUXIX***
    36            GO TO 1                                                         AUXIX***
    37         16 CALL  C10I                                                      AUXIX***
    38            GO TO 1                                                         AUXIX***
    39         17 CALL  D1I                                                       AUXIX***
    40            GO TO 1                                                         AUXIX***
    41         18 CALL  D2I                                                       AUXIX***
    42            GO TO 1                                                         AUXIX***
    43         19 CALL  D3I                                                       AUXIX***
    44            GO TO 1                                                         AUXIX***
    45         20 CALL  D4I                                                       AUXIX***
    46            GO TO 1                                                         AUXIX***
    47         21 CALL  D5I                                                       AUXIX***
    48            GO TO 1                                                         AUXIX***
    49         22 CALL  G1I                                                       AUXIX***
```

```
I N D E X                                   SUBROUTINE AUXI

   50                GO TO 1                                             AUXIX***
   51             23 CALL G2I                                            AUXIX***
   52                GO TO 1                                             AUXIX***
   53             24 CALL G3I                                            AUXIX***
   54                GO TO 1                                             AUXIX***
   55             25 CALL G4I                                            AUXIX***
   56                GO TO 1                                             AUXIX***
   57             26 CALL G5I                                            AUXIX***
   58                GO TO 1                                             AUXIX***
   59             27 CALL G6I                                            AUXIX***
   60                GO TO 1                                             AUXIX***
   61             28 CALL S1I                                            AUXIX***
   62                GO TO 1                                             AUXIX***
   63             29 CALL S2I                                            AUXIX***
   64                GO TO 1                                             AUXIX***
   65             30 CALL S3I                                            AUXIX***
   66                GO TO 1                                             AUXIX***
   67             31 CALL S4I                                            AUXIX***
   68                GO TO 1                                             AUXIX***
   69             32 CALL S5I                                            AUXIX***
   70                GO TO 1                                             AUXIX***
   71             33 CALL S6I                                            AUXIX***
   72                GO TO 1                                             AUXIX***
   73             34 CALL S7I                                            AUXIX***
   74                GO TO 1                                             AUXIX***
   75             35 CALL S8I                                            AUXIX***
   76                GO TO 1                                             AUXIX***
   77             36 CALL S9I                                            AUXIX***
   78                GO TO 1                                             AUXIX***
   79             37 CALL S10I                                           AUXIX***
   80              1 CONTINUE                                            AUXIX***
   81                RETURN                                              AUXIX***
   82                END                                                 AUXIX***
```

```
                                              SUBROUTINE AUXI

    SYMBOL        - - - - - - - - - - - - - - - REFERENCES - - - - - - - - - - - - - - -

    1        -     6T0     8      10      12      14      16      18      20      22      24
                   26      28     30      32      34      36      38      40      42      44
                   46      48     50      52      54      56      58      60      62      64
                   66      68     70      72      74      76      78      80*
    2        -     8       9*
    3        -     8      11*
    4        -     8      13*
    5        -     8      15*
    6        -     8      17*
    7        -     8      19*
    8        -     8      21*
    9        -     8      23*
    10       -     8      25*
    11       -     8      27*
    12       -     8      29*
    13       -     8      31*
    14       -     8      33*
    15       -     8      35*
    16       -     8      37*
    17       -     8      39*
    18       -     8      41*
    19       -     8      43*
    20       -     8      45*
    21       -     8      47*
    22       -     8      49*
    23       -     8      51*
    24       -     8      53*
    25       -     8      55*
    26       -     8      57*
    27       -     8      59*
    28       -     8      61*
    29       -     8      63*
    30       -     8      65*
    31       -     8      67*
    32       -     8      69*
    33       -     8      71*
    34       -     8      73*
    35       -     8      75*
    36       -     8      77*
    37       -     8      79*
    AUXI     -     1
    A1I      -     9
    A2I      -    11
    A3I      -    13
    A4I      -    15
    A5I      -    17
    C        -    2C9     3E0
    C1I      -    19
```

```
I N D E X                                    SUBROUTINE AUXI

    C10I       -     37
    C2I        -     21
    C3I        -     23
    C4I        -     25
    C5I        -     27
    C6I        -     29
    C7I        -     31
    C8I        -     33
    C9I        -     35
    D1I        -     39
    D2I        -     41
    D3I        -     43
    D4I        -     45
    D5I        -     47
    G1I        -     49
    G2I        -     51
    G3I        -     53
    G4I        -     55
    G5I        -     57
    G6I        -     59
    I          -      600        7
    L          -      7=         8
    N          -      3EQ        5=
    NOMOD      -      3EQ        600
    RETURN     -     81
    S1I        -     61
    S10I       -     79
    S2I        -     63
    S3I        -     65
    S4I        -     67
    S5I        -     69
    S6I        -     71
    S7I        -     73
    S8I        -     75
    S9I        -     77
    XMODNO     -      3EQ        40I        7

-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

SUBROUTINE AUXSUB

```
   1          SUBROUTINE AUXSUB
   2          COMMON       C(5120)
   3          EQUIVALENCE (C(2760),T      ), (C(2761),NOMOD ), (C(2362),XPCONO)
   4          EQUIVALENCE (C(2561),N      ), (C(2563),IPL   ), (C(2664),DER   )
   5          EQUIVALENCE (C(2965),VAR    )
   6          EQUIVALENCE (C(2020),LCONV)
   7          DIMENSION   DER(101)        , VAR(101)         , IPL(100)
   8          DIMENSION   XMODNO(99)
   9          EXTERNAL A1
  10          EXTERNAL A2
  11          EXTERNAL A3
  12          DO 50 I = 2, N                                                AUXSU***
  13          J = IPL(I-1)                                                  AUXSU***
  14       50 C(J+3) = VAR(I)                                               AUXSU***
  15          T = VAR(1)                                                    AUXSU***
  16          DO 1 I=1,NOMOD                                                AUXSU***
  17          IF(LCONV.EQ.2)RETURN
  18          L =XMODNO(I)
  19          GO TO (1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,  AUXSU***
             123,24,25,26,27,28,29,30,31,32,33,34,35,36,37),L                AUXSU***
  20        2 CALL A1                                                       AUXSU***
  21          GO TO 1                                                       AUXSU***
  22        3 CALL A2                                                       AUXSU***
  23          GO TO 1                                                       AUXSU***
  24        4 CALL A3                                                       AUXSU***
  25          GO TO 1                                                       AUXSU***
  26        5 CALL A4                                                       AUXSU***
  27          GO TO 1                                                       AUXSU***
  28        6 CALL A5                                                       AUXSU***
  29          GO TO 1                                                       AUXSU***
  30        7 CALL C1                                                       AUXSU***
  31          GO TO 1                                                       AUXSU***
  32        8 CALL C2                                                       AUXSU***
  33          GO TO 1                                                       AUXSU***
  34        9 CALL C3                                                       AUXSU***
  35          GO TO 1                                                       AUXSU***
  36       10 CALL C4                                                       AUXSU***
  37          GO TO 1                                                       AUXSU***
  38       11 CALL C5                                                       AUXSU***
  39          GO TO 1                                                       AUXSU***
  40       12 CALL C6                                                       AUXSU***
  41          GO TO 1                                                       AUXSU***
  42       13 CALL C7                                                       AUXSU***
  43          GO TO 1                                                       AUXSU***
  44       14 CALL C8                                                       AUXSU***
  45          GO TO 1                                                       AUXSU***
  46       15 CALL C9                                                       AUXSU***
  47          GO TO 1                                                       AUXSU***
  48       16 CALL C10                                                      AUXSU***
  49          GO TO 1                                                       AUXSU***
```

SUBROUTINE AUXSU7

```
   17 CALL D1                                                    AUXSU***
      GO TO 1                                                    AUXSU***
   18 CALL D2                                                    AUXSU***
      GO TO 1                                                    AUXSU***
   19 CALL D3                                                    AUXSU***
      GO TO 1                                                    AUXSU***
   20 CALL D4                                                    AUXSU***
      GO TO 1                                                    AUXSU***
   21 CALL D5                                                    AUXSU***
      GO TO 1                                                    AUXSU***
   22 CALL G1                                                    AUXSU***
      GO TO 1                                                    AUXSU***
   23 CALL G2                                                    AUXSU***
      GO TO 1                                                    AUXSU***
   24 CALL G3                                                    AUXSU***
      GO TO 1                                                    AUXSU***
   25 CALL G4                                                    AUXSU***
      GO TO 1                                                    AUXSU***
   26 CALL G5                                                    AUXSU***
      GO TO 1                                                    AUXSU***
   27 CALL G6                                                    AUXSU***
      GO TO 1                                                    AUXSU***
   28 CALL S1                                                    AUXSU***
      GO TO 1                                                    AUXSU***
   29 CALL S2                                                    AUXSU***
      GO TO 1                                                    AUXSU***
   30 CALL S3                                                    AUXSU***
      GO TO 1                                                    AUXSU***
   31 CALL S4                                                    AUXSU***
      GO TO 1                                                    AUXSU***
   32 CALL S5                                                    AUXSU***
      GO TO 1                                                    AUXSU***
   33 CALL S6                                                    AUXSU***
      GO TO 1                                                    AUXSU***
   34 CALL S7                                                    AUXSU***
      GO TO 1                                                    AUXSU***
   35 CALL S8                                                    AUXSU***
      GO TO 1                                                    AUXSU***
   36 CALL S9                                                    AUXSU***
      GO TO 1                                                    AUXSU***
   37 CALL S10                                                   AUXSU***
    1 CONTINUE                                                   AUXSU***
      DO 60 I = 2, N                                             AUXSU***
      J = IPL(I-1)                                               AUXSU***
   60 DER(I) = C(J)                                              AUXSU***
      RETURN                                                     AUXSU***
      END                                                        AUXSU***
```

```
                              SUBROUTINE AUXSUB

SYMBOL   - - - - - - - - - - - - - - REFERENCES - - - - - - - - - - - - - - - -

1        -    1600    19    21    23    25    27    29    31    33    35
              37      39    41    43    45    47    49    51    53    55
              57      59    61    63    65    67    69    71    73    75
              77      79    81    83    85    87    89    91*
2        -    19      20*
3        -    19      22*
4        -    19      24*
5        -    19      26*
6        -    19      28*
7        -    19      30*
8        -    19      32*
9        -    19      34*
10       -    19      36*
11       -    19      38*
12       -    19      40*
13       -    19      42*
14       -    19      44*
15       -    19      46*
16       -    19      48*
17       -    19      50*
18       -    19      52*
19       -    19      54*
20       -    19      56*
21       -    19      58*
22       -    19      60*
23       -    19      62*
24       -    19      64*
25       -    19      66*
26       -    19      68*
27       -    19      70*
28       -    19      72*
29       -    19      74*
30       -    19      76*
31       -    19      78*
32       -    19      80*
33       -    19      82*
34       -    19      84*
35       -    19      86*
36       -    19      88*
37       -    19      90*
50       -    1200    14*
60       -    9200    94*
AUXSUB   -    1
A1       -    9EX     20
A2       -    10EX    22
A3       -    11EX    24
A4       -    26
A5       -    28
```

```
I N D E X                                                            PAGE 125
                              SUBROUTINE AUXSUB

     C           -    2EQ     3EQ     4EQ     5EQ     6EQ     14=     94
     C1          -    30
     C10         -    48
     C2          -    32
     C3          -    34
     C4          -    36
     C5          -    38
     C6          -    40
     C7          -    42
     C8          -    44
     C9          -    46
     DER         -     4EQ    7DI     94=
     D1          -    50
     D2          -    52
     D3          -    54
     D4          -    56
     D5          -    58
     G1          -    60
     G2          -    62
     G3          -    64
     G4          -    66
     G5          -    68
     G6          -    70
     I           -    1200    13      14      1600    18      9200   -93      94
     IPL         -     4EQ    7DI     13      93
     J           -    13=     14      93=     94
     L           -    18=     19
     LCONV       -     6EQ    17
     N           -     4EQ    1200    9200
     NOMOD       -     3EQ    1600
     RETURN      -    17      95
     S1          -    72
     S10         -    90
     S2          -    74
     S3          -    76
     S4          -    78
     S5          -    80
     S6          -    82
     S7          -    84
     S8          -    86
     S9          -    88
     T           -     3EQ    15=
     VAR         -     5EQ    7DI     14      15
     XMOONO      -     3EQ    8DI     18

     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
I N D E X
                                   SUBROUTINE RESET

        1          SUBROUTINE RESET
        2          COMMON      C(3517)
        3          EQUIVALENCE (C(3166),NOLIST), (C(3167),LISTNO), (C(3117),VALUE )
        4          DIMENSION   LISTNO(50)        , VALUE(50)
        5          IF (NOLIST .EQ. 0) RETURN
        6          DO 1 I = 1, NOLIST
        7          J = LISTNO(I)
        8        1 C(J) = VALUE(I)
        9          RETURN
       10          END
```

```
                               SUBROUTINE RESET

        SYMBOL                        REFERENCES

        1          -     6DM     8*
        C          -     2CM     3EQ     8=
        I          -     6DM     7       8
        J          -     7=      8
        LISTNO     -     3EQ     4DI     7
        NOLIST     -     3EQ     5       6DO
        RESET      -     1
        RETURN     -     5       9
        VALUE      -     3EQ     4DI     8
```

SUBROUTINE TABLE (X,XI,YI,NX,XK,XLABEL,Y)

```
      SUBROUTINE TABLE (X,XI,YI,NX,XK,XLABEL,Y)
      DIMENSION XLABEL (2)
      XK = 0.
      Y = PINTP1 (X,XI,YI,NX,XK,XLABEL)
      RETURN
      END
```

```
I N D E X                                                                        PAGE 129
                              SUBROUTINE TABLE (X,XI,YI,NX,XK,XLABEL,Y)
SYMBOL                                    REFERENCES

FINTP1    -      4
NX        -      1AG      4
RETURN    -      5
TABLE     -      1
X         -      1AG      4
XI        -      1AG      4
XK        -      1AG      3=        4
XLABEL    -      1AG      2DI       4
Y         -      1AG      4=
YI        -      1AG      4

 -+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
      SUBROUTINE TABL2(X,Y,XYI,ZI,NXY,XINTER,XLABEL,Z)
      DIMENSION XLABEL(2)
      DIMENSION XYI(2),NXY(2)
      Z = FINTP2 (X,Y,XYI,XYI(NXY+1),ZI,NXY,NXY(2),NXY,XINTER,XLABEL)
      RETURN
      END
```

```
I N D E X                                                                    PAGE 131
                    SUBROUTINE TABL2(X,Y,XYI,ZI,NXY,XINTER,XLABEL,Z)

     SYMBOL                                    REFERENCES

     FINTP2   -    4
     NXY      -    146     30I     4
     RETURN   -    5
     TABL2    -    1
     X        -    146     4
     XINTER   -    146     4
     XLABEL   -    146     20I     4
     XYI      -    146     30I     4
     Y        -    145     4
     Z        -    145     4=
     ZI       -    146     4

     -+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
      SUBROUTINE TABL3(X,Y,Z,XYZI,WI,NXYZ,XINTER,XLABEL,W)
      DIMENSION XLABEL(2)
      DIMENSION XYZI(1),NXYZ(1)
      NZI= NXYZ(1) + NXYZ(2) + 1
      XINTER = 0.
      W  =   FINIP3 (X,Y,Z,XYZI,XYZI(NXYZ+1),XYZI(NZI), WI,NXYZ(3),
     8 NXYZ(2),NXYZ,XINTER,XLABEL)
      RETURN
      END
```

```
I N D E X
                    SUBROUTINE TABL3(X,Y,Z,XYZI,WI,NXYZ,XINTER,XLABEL,W)

SYMBOL                              REFERENCES

FINTP3   -     6
NXYZ     -     1AG     3DI     4       6
NZI      -     4=      6
RETURN   -     7
TABL3    -     1
W        -     1AG     6=
WI       -     1AG     6
X        -     1AG     6
XINTER   -     1AG     5=      6
XLABEL   -     1AG     20I     6
XYZI     -     1AG     30I     6
Y        -     1AG     6
Z        -     1AG     6

-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

FUNCTION FINTP1(X,XI,YI,N,F,XL)

```
      FUNCTION FINTP1(X,XI,YI,N,F,XL)
      DIMENSION XI(N), YI(N) , XL(2)
      IF(F .GT. 0.)GO TO 30
      DO 10 I=2, N
      IF(X .LE. XI(I)) GO TO 20
   10 CONTINUE
      I = N
   20 PCT = (X-XI(I-1))/(XI(I)-XI(I-1))
      F = 1.
   30 FINTP1 = YI(I-1) + PCT*(YI(I)-YI(I-1))
      RETURN
      END
```

```
I N D E X                                                                               PAGE 135
                                          FUNCTION FINTP1(X,XI,YI,N,F,XL)

SYMBOL          - - - - - - - - - - - - - - -       REFERENCES - - - - - - - - - - - - - - -

10         -     4DO    6*
20         -     5      8*
30         -     7      13*
F          -     1AG    3      9*
FINTP1     -     1      12*
I          -     4DO    5      7*     8      10
N          -     1AG    20I    4DO    7
PCT        -     8*     13
RETURN     -    11
X          -     1AG    5      8
XI         -     1AG    20I    5      6
XL         -     1AG    20I
YI         -     1AG    20I    10

-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```

FUNCTION FINTP2(X,Y,XI,YI,ZI,NXD,NY,NX,F,XL)

```
         FUNCTION FINTP2(X,Y,XI,YI,ZI,NXD,NY,NX,F,XL)
         DIMENSION XI(1),YI(1), ZI(NXD,1), T(2), XL(2)
         IF(F .GT. 1.) GO TO 30
         DO 10 I=2, NY
         IF(Y .LE. YI(I)) GO TO 20
      10 CONTINUE
         I = NY
      20 PCT = (Y-YI(I-1))/(YI(I)-YI(I-1))
      30 DO 40 J=1,2
         L = I + J - 2
      40 T(J) = FINTP1(X,XI,ZI(1,L),NX,F,XL)
         FINTP2 = T(1) + PCT*(T(2)-T(1))
         RETURN
         END
```

```
I N D E X
                         FUNCTION FINTP2(X,Y,XI,YI,ZI,NXD,NY,NX,F,XL)

  SYMBOL     - - - - - - - - - - - - - - - REFERENCES - - - - - - - - - - - - - -

  10        -      400     6*
  20        -      5       8*
  30        -      3       9*
  40        -      900     11*
  F         -      1AG     3        11
  FINTP1    -     11
  FINTP2    -      1       12=
  I         -      400     5        7=       8        10
  J         -      900     10       11
  L         -     10=      11
  NX        -      1AG     11
  NXD       -      1AG      20I
  NY        -      1AG      400     7
  PCT       -      8=      12
  RETURN    -     13
  T         -      20T     11=      12
  X         -      1AG     11
  XI        -      1AG      20I     11
  XL        -      1AG      20I     11
  Y         -      1AG     5        8
  YI        -      1AG      20I     5        8
  ZI        -      1AG      20I     11

  -+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

FUNCTION FINTP3(X,Y,Z,XI,YI,ZI,WI,NZ,NY,NX,F,XL)

```
      FUNCTION FINTP3(X,Y,Z,XI,YI,ZI,WI,NZ,NY,NX,F,XL)
      DIMENSION XI(1), YI(1), ZI(1), WI(NX,NY,1), T(2), XL(2)
      DO 10 I=2, NZ
      IF(Z .LE. ZI(I)) GO TO 20
   10 CONTINUE
      I = NZ
   20 PCT = (Z-ZI(I-1))/(ZI(I)-ZI(I-1))
   30 DO 40 J=1,2
      L = I + J - 2
   40 T(J) = FINTP2(X,Y,XI,YI,WI(1,1,L),NX,NY,NX,F,XL)
      FINTP3 = T(1) + PCT*(T(2)-T(1))
      RETURN
      END
```

```
I N D E X
                    FUNCTION FINTP3(X,Y,Z,XI,YI,ZI,WI,NZ,NY,NX,F,XL)

SYMBOL                                      REFERENCES

10          -     300     5*
20          -     4       7*
30          -     8*
40          -     800     10*
F           -     1AG     10
FINTP2      -    10
FINTP3      -     1       11=
I           -     300     4        6=       7        9
J           -     800     9        10
L           -     9=      10
NX          -     1AG     20I      10
NY          -     1AG     20I      10
NZ          -     1AG     300      6
PCT         -     7=      11
RETURN      -    12
T           -     20T     10=      11
WI          -     1AG     20I      10
X           -     1AG     10
XI          -     1AG     20I      10
XL          -     1AG     20I      10
Y           -     1AG     10
YI          -     1AG     20I      10
Z           -     1AG     4        7
ZI          -     1AG     20I      4        7
```

```
      SUBROUTINE PLOT4 (GRAPH,NP,YL,T,NPLOT4,NPLOT2,NOPLOT)
C**PLOT SUBROUTINE
      DIMENSION GRAPH(300,15),YL(2,4),T(300)
      DIMENSION IXP(4),IYP(4),MRKPT(4)
      DATA (MRKPT(I),I=1,4)/42,16,38,63/
      DATA IXP/4,24,4,24/
      DATA IYP/776,776,411,411/
      IF (NPLOT4.EQ.0) RETURN
      KK = 1
      XM1 = GRAPH(1,1)
      YM1 = GRAPH(1,2)
      XT1 = GRAPH(1,3)
      YT1 = GRAPH(1,4)
      XM2 = XM1
      YM2 = YM1
      XT2 = XT1
      YT2 = YT1
      DO 1 I=1,NP
      XM1 = AMIN1(GRAPH(I,1),XM1)
      YM1 = AMIN1(GRAPH(I,2),YM1)
      XT1 = AMIN1(GRAPH(I,3),XT1)
      YT1 = AMIN1(GRAPH(I,4),YT1)
      XM2 = AMAX1(GRAPH(I,1),XM2)
      YM2 = AMAX1(GRAPH(I,2),YM2)
      XT2 = AMAX1(GRAPH(I,3),XT2)
    1 YT2 = AMAX1(GRAPH(I,4),YT2)
      XMIN = AMIN1(XM1,XT1)
      YMIN = AMIN1(YM1,YT1)
      XMAX = AMAX1(XM2,XT2)
      YMAX = AMAX1(YM2,YT2)
      DELX = ABS(XMAX-XMIN)
      DELY = ABS(YMAX-YMIN)
      DEL = AMAX1(DELX,DELY)
      X1 = XMIN
      Y1 = YMIN-(DEL-DELY)/2.
      X2 = X1+DEL
      Y2 = Y1+DEL
      CALL CAMRAV (9)
      CALL DXDYV(1,X1,X2,DX,N,I,NX,25.,IERR)
      CALL DXDYV(2,Y2,Y1,DY,M,J,NY,25.,IERR)
      CALL SETMIV (24,0,24,24)
      CALL GRID1V(KK,X1,X2,Y2,Y1,DX,DY,N,M,I,J,NX,NY)
      DO 2 J=1,3,2
      K = J+1
      UTIME = 0.
      IX1 = NXV(GRAPH(1,J) )
      IY1 = NYV(GRAPH(1,K) )
      DO 2 IJ=2,NP
      IX2 = NXV(GRAPH(IJ,J))
      IY2 = NYV(GRAPH(IJ,K))
```

```
      SUBROUTINE PLOT4 (GRAPH,NP,YL,T,NPLOT4,NPLOT2,NOPLOT)
      IF(T(IJ)-(.5+UTIME)) 7,3,3
    3 UTIME = T(IJ)
      CALL POINTV(IX2,IY2,-17,2)
    7 IF(J-2) 4,5,5
    5 CALL POINTV(IX2,IY2,0,2)
      GO TO 6
    4 CALL LINEV(IX1,IY1,IX2,IY2)
    6 IX1 = IX2
    2 IY1 = IY2
      CALL PRINTV(12,YL(1,1)     ,524,12)
      CALL APRNTV(0,-14,12,YL(1,2)     ,12,524)
      RETURN
C
      ENTRY PLOT2
      IF (NPLOT2.EQ.0) RETURN
      JX= NPLOT4+1
      JY1= JY+1
      JYN= NPLOT4+NPLOT2
      X1= GRAPH(1,JX)
      X2= X1
      DO 110 I=2,NP
      X1= AMIN1 (GRAPH(I,JX),X1)
  110 X2= AMAX1 (GRAPH(I,JX),X2)
      Y1= GRAPH(1,JY1)
      Y2= Y1
      DO 120 JY=JY1,JYN
      DO 120 I=1,NP
      Y1= AMIN1 (GRAPH(I,JY),Y1)
  120 Y2= AMAX1 (GRAPH(I,JY),Y2)
      CALL CAMRAV (9)
      CALL DXDYV (1,X1,X2,DX,N,I,NX,14,6,IERR)
      CALL DXDYV (2,Y1,Y2,DY,M,J,NY,14.0,IERR)
      CALL SETMIV (36,24,24,24)
      CALL GRID1V (1,X1,X2,Y1,Y2,DX,DY,N,M,I,J,-3,-3)
      IMARK= 1
      DO 140 JY=JY1,JYN
      IX1= NXV (GRAPH(1,JX))
      IY1= NYV (GRAPH(1,JY))
C
      DO 130 IJ=2,NP
      IX2= NXV (GRAPH(IJ,JX))
      IY2= NYV (GRAPH(IJ,JY))
      CALL LINEV (IX1,IY1,IX2,IY2)
      CALL LINEV (IX1,IY1,IX2,IY2)
      IX1= IX2
  130 IY1= IY2
      IF (IMARK.GT.4) GO TO 140
      CALL APLCTV (NP,GRAPH(1,JX),GRAPH(1,JY),20,20,1,MRKPT(IMARK),IRR)
  140 IMARK= IMARK + 1
      CALL PRINTV (12,     YL(1,JX),468,8)
```

```
      I=1
      DO 150 JY=JY1,JYN
      IF (I.GT.4) GO TO 150
      IYO= IYP(I) + 29
      CALL PLOTV (IXP(I),IYO,MRKPT(I))
      CALL APRNTV (0,-14,12,    YL(1,JY),IXP(I),IYP(I))
  150 I=I+1
      RETURN
C
      ENTRY PLOTN
      NPLOT3=NOPLOT-NPLOT2-NPLOT4
      IF (NPLOT3.LE.0) RETURN
      DO 105 NM=1,NPLOT3
      JY=NPLOT4+NPLOT2+NM
      IX=MOD(NM,3)
      IF(IX  .EQ.   0)IX=3
      II=712-344*(IX-1)
      JJ=29+344*(IX-1)
      KK=1
      IF(IX  .GT.   1)KK=2
      X1=T(1)
      X2=T(NP)
      Y1=GRAPH(1,JY)
      Y2=Y1
      DO 50 I=1,NP
      Y1=AMIN1(GRAPH(I,JY),Y1)
   50 Y2=AMAX1(GRAPH(I,JY),Y2)
      CALL CAMRAV(9)
      CALL DXDYV(1,X1,X2,DX,N,I,NX,14.,IERR)
      CALL DXDYV(2,Y1,Y2,DY,M,J,NY,14.,IERR)
      CALL SETMIV(24,9,II,JJ)
      CALLGRID1V(KK,X1,X2,Y1,Y2,DX,DY,N,M,I,J,NX,-3)
      IX1=NXV(X1)
      IY1=NYV(GRAPH(1,JY))
      DO 55 IJ=2,NP
      IX2=NXV(T(IJ))
      IY2=NYV(GRAPH(IJ,JY))
      CALL LINEV(IX1,IY1,IX2,IY2)
      IY1=IY2
   55 IX1=IX2
      CALL PRINTV (-11,10HTIME (SEC),468,696-344*(IX-1))
  105 CALL APRNTV (0,-14,12,    YL(1,JY),4,890-344*(IX-1))
      RETURN
      END
```

```
                    SUBROUTINE PLOT4 (GRAPH,NP,YL,T,NPLOT4,NPLOT2,NOPLOT)

SYMBOL              - - - - - - - - - - - - - - - - REFERENCES - - - - - - - - - - - - - - - - - -

1         -     1700     20*
2         -     4200     4700     58*
3         -     50       51*
4         -     53       56*
5         -     53       54*
6         -     55       57*
7         -     52       53*
50        -     12100    123*
55        -     13100    136*
100       -     10900    138*
110       -     6900     71*
120       -     7400     7500     77*
130       -     8700     93*
140       -     8400     94       96*
150       -     9900     100      104*
ABS       -     30       31
AMAX1     -     22       23       24       25       28       29       32       71       77       123
AMIN1     -     18       19       20       21       26       27       70       76       122
APLOTV    -     95
APPNTV    -     65       103      138
CAMRAV    -     37       78       124
CEL       -     32=      34       35       36
DELX      -     30=      32
DELY      -     31=      32       34
DX        -     38AG     41AG     79AG     82AG     125AG    128AG
DXDYV     -     38       39       79       80       125      126
DY        -     39AG     41AG     80AG     82AG     126AG    128AG
GRAPH     -      1AG     20I      9        10       11       12       18       19       20       21
                22       23       24       25       45       46       48       49       67       70
                71       72       76       77       85       86       88       89       95AG     119
                122      123      130      133
GRIDIV    -     41       82       128
I         -      40A     1700     18       19       20       21       22       23       24       25
                38AG     41AG     6900     70       71       7500     76       77       79AG     82AG
                98=      100      101      102AG    103AG    104=     12100    122      123      125AG
                128AG
IERR      -     38AG     39AG     79AG     80AG     125AG    126AG
II        -     113=     127AG
IJ        -     4700     48       49       50       51       8700     88       89       13100    132
                133
IMARK     -     83=      94       95AG     96=
IRR       -     95AG
IX        -     111=     112      113      114      116      137AG    138AG
IXP       -      30I     50A      102AG    103AG
IX1       -     45=      56AG     57=      85=      90AG     91AG     92=      129=     134AG    136=
IX2       -     48=      52AG     54AG     56AG     57       88=      90AG     91AG     92       132=
                134AG    136
IYP       -      30I     60A      101      103AG
```

```
I N D E X
                    SUBROUTINE PLOT4 (GRAPH,NP,YL,T,NPLOT4,NPLOT2,NOPLOT)

IY0      -   101=     102AG
IY1      -   46=      56AG     58=      96=      9_AG     91AG     93=      13_=     134AG    135=
IY2      -   49=      52AG     54AG     56AG     58       69=      90AG     91AG     93       133=
             134AG    135
J        -   39AG     41AG     4200     43       45       48       53       80AG     82AG     126AG
             128AG
JJ       -   114=     127AG
JX       -   64=      65       67       70       71       85       88       95AG     97AG
JY       -   7400     76       77       8400     86       89       95AG     9900     103AG    113=
             119      122      123      130      133      138AG
JYN      -   66=      7400     8400     9900
JY1      -   65=      72       7400     8400     9900
K        -   43=      46       49
KK       -   8=       41AG     115=     116=     128AG
LINEV    -   56       90       91       134
M        -   39AG     41AG     80AG     82AG     126AG    128AG
MOD      -   111
MRKPT    -   30T      40A      95AG     102AG
N        -   38AG     41AG     79AG     82AG     125AG    128AG
NM       -   10900    113      111
NOPLOT   -   1AG      107
NP       -   1AG      1700     4700     6900     7500     8700     95AG     110      12100    13100
NPLOT2   -   1AG      63       66       107      110
NPLOT3   -   107=     108      10900
NPLOT4   -   1AG      7        64       66       107      110
NX       -   38AG     41AG     79AG     125AG    128AG
NXV      -   45       48       85       88       129      132
NY       -   39AG     41AG     86AG     126AG
NYV      -   46       49       86       89       130      133
PLOTN    -   106
PLOTV    -   102
PLOT2    -   62
PLOT4    -   1
POINTV   -   52       54
PRINTV   -   59       97       137
RETURN   -   7        61       63       105      108      139
SETMIV   -   40       81       127
T        -   1AG      20I      53       51       117      118      132
UTIME    -   44=      50       51=
XMAX     -   28=      33
XMIN     -   26=      30       33
XM1      -   9=       13       18=      26
XM2      -   13=      22=      28
XT1      -   11=      15       23=      26
XT2      -   15=      24=      28
X1       -   32=      35       38AG     41AG     67=      68       70=      79AG     82AG     117=
             125AG    128AG    129
X2       -   35=      38AG     41AG     68=      71=      79AG     82AG     118=     125AG    128AG
YL       -   1AG      20I      59AG     60AG     97AG     103AG    138AG
YMAX     -   29=      31
```

```
I N D E X                                                                              PAGE 145
                              SUBROUTINE FLOT4 (GRAPH,NP,YL,T,NPLOT4,NPLOT2,NOPLOT)

YMIN          -     27=      31       34
YM1           -     10=      14       19=      27
YM2           -     14=      27=      20
YT1           -     12=      16       21=      27
YT2           -     16=      25=      29
Y1            -     34=      36       39AG     41AG     72=      73       76=      80AG     82AG     119=
                    126      122=     126AG    128AG
Y2            -     36=      39AG     41AG     73=      77=      80AG     82AG     120=     123=     126AG
                    128AG

-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
      1          SUBROUTINE DUMMY
             C   DUMMY SUBROUTINE
      2          ENTRY A1
      3          ENTRY A1I
      4          ENTRY A2
      5          ENTRY A2I
      6          ENTRY A3
      7          ENTRY A3I
      8          ENTRY A4
      9          ENTRY A4I
     10          ENTRY A5
     11          ENTRY A5I
     12          ENTRY C1
     13          ENTRY C1I
     14          ENTRY C2
     15          ENTRY C2I
     16          ENTRY C3
     17          ENTRY C3I
     18          ENTRY C4
     19          ENTRY C4I
     20          ENTRY C5
     21          ENTRY C5I
     22          ENTRY C6
     23          ENTRY C6I
     24          ENTRY C7
     25          ENTRY C7I
     26          ENTRY C8
     27          ENTRY C8I
     28          ENTRY C9
     29          ENTRY C9I
     30          ENTRY C10
     31          ENTRY C10I
     32          ENTRY D1
     33          ENTRY D1I
     34          ENTRY D2
     35          ENTRY D2I
     36          ENTRY D3
     37          ENTRY D3I
     38          ENTRY D4
     39          ENTRY D4I
     40          ENTRY D5
     41          ENTRY D5I
     42          ENTRY G1
     43          ENTRY G1I
     44          ENTRY G2
     45          ENTRY G2I
     46          ENTRY G3
     47          ENTRY G3I
     48          ENTRY G4
     49          ENTRY G4I
```

```
     50            ENTRY G5
     51            ENTRY G5I
     52            ENTRY G6
     53            ENTRY G6I
     54            ENTRY S1
     55            ENTRY S1I
     56            ENTRY S2
     57            ENTRY S2I
     58            ENTRY S3
     59            ENTRY S3I
     60            ENTRY S4
     61            ENTRY S4I
     62            ENTRY S5
     63            ENTRY S5I
     64            ENTRY S6
     65            ENTRY S6I
     66            ENTRY S7
     67            ENTRY S7I
     68            ENTRY S8
     69            ENTRY S8I
     70            ENTRY S9
     71            ENTRY S9I
     72            ENTRY S10
     73            ENTRY S10I
     74            ENTRY AUXA1
     75            ENTRY AUXA2
     76            ENTRY AUXA3
     77            ENTRY AUXB1
     78            ENTRY AUXB2
     79            ENTRY AUXB3
     80            ENTRY AUXC1
     81            ENTRY AUXC2
     82            ENTRY AUXC3
     83            ENTRY CNTR1
     84            ENTRY CNTR2
     85            ENTRY CNTR3
     86            ENTRY INPT1
     87            ENTRY INPT2
     88            ENTRY INPT3
     89            ENTRY OUPT1
     90            ENTRY OUPT2
     91            ENTRY OUPT3
     92            ENTRY PROCES
     93            ENTRY RESET
     94            ENTRY RNDM1
     95            ENTRY RNDM2
     96            ENTRY RNDM3
     97            ENTRY STGE1
     98            ENTRY STGE2
     99            ENTRY STGE3
```

```
                    SUBROUTINE DUMMY

  100          ENTRY SUML1
  101          ENTRY SUML2
  102          ENTRY KIKSET
  103          ENTRY COUNTV
  104          ENTRY TIMEV
  105          ENTRY WRITE
  106          ENTRY PLOT4
  107          ENTRY PLOT2
  108          ENTRY PLOTN
  109          RETURN
  110          END
```

```
I N D E X                                                                    PAGE 149
                             SUBROUTINE DUMMY
          SYMBOL  - - - - - - - - - - - - - - REFERENCES - - - - - - - - - - - - - -

          AUXA1    -    74
          AUXA2    -    75
          AUXA3    -    76
          AUXB1    -    77
          AUXB2    -    78
          AUXB3    -    79
          AUXC1    -    80
          AUXC2    -    81
          AUXC3    -    82
          A1       -     2
          A1I      -     3
          A2       -     4
          A2I      -     5
          A3       -     6
          A3I      -     7
          A4       -     8
          A4I      -     9
          A5       -    10
          A5I      -    11
          CNTR1    -    83
          CNTR2    -    84
          CNTR3    -    85
          COUNTV   -   193
          C1       -    12
          C1I      -    13
          C10      -    30
          C10I     -    31
          C2       -    14
          C2I      -    15
          C3       -    16
          C3I      -    17
          C4       -    18
          C4I      -    19
          C5       -    20
          C5I      -    21
          C6       -    22
          C6I      -    23
          C7       -    24
          C7I      -    25
          C8       -    26
          C8I      -    27
          C9       -    28
          C9I      -    29
          DUMMY    -     1
          D1       -    32
          D1I      -    33
          D2       -    34
          D2I      -    35
```

```
I N D E X
                          SUBROUTINE DUMMY
D3       -   36
D3I      -   37
D4       -   38
D4I      -   39
D5       -   40
D5I      -   41
G1       -   42
G1I      -   43
G2       -   44
G2I      -   45
G3       -   46
G3I      -   47
G4       -   48
G4I      -   49
G5       -   50
G5I      -   51
G6       -   52
G6I      -   53
INPT1    -   86
INPT2    -   87
INPT3    -   88
KIKSFT   -  102
OUPT1    -   89
OUPT2    -   90
OUPT3    -   91
PLOTN    -  108
PLOT2    -  107
PLOT4    -  106
PROCES   -   92
RESET    -   93
RETURN   -  109
RNDM1    -   94
RNDM2    -   95
RNDM3    -   96
STGE1    -   97
STGE2    -   98
STGE3    -   99
SUBL1    -  100
SUBL2    -  101
S1       -   54
S1I      -   55
S10      -   72
S10I     -   73
S2       -   56
S2I      -   57
S3       -   58
S3I      -   59
S4       -   60
S4I      -   61
S5       -   62
```

```
I N D E X
                                   SUBROUTINE DUMMY

S5I      -     63
S6       -     64
S6I      -     65
S7       -     66
S7I      -     67
S8       -     68
S8I      -     69
S9       -     70
S9I      -     71
TIMEV    -    104
WRITE    -    105
```

SUBROUTINE TERROR (XLABEL)

```
      SUBROUTINE TERROR (XLABEL)
C**   FOR USE WITH CONIM2, FCN2, FCN3
C
      COMMON C(3510)
      EQUIVALENCE (C(2323),LCONV)
      WRITE (6,10) XLABEL
   10 FORMAT ( 43H0         NO AERO POINTS SPECIFIED FOR ARG , 5X,
     C   7HTABLE  ,A6 )
      CALL EXIT
      END
```

```
                              SUBROUTINE TERROR (XLABEL)

SYMBOL                               REFERENCES

IG       -     4WR      5*
G        -     2CQ      3EQ
EXIT     -     6
LCONV    -     3EQ
TERROR   -     1
XLABEL   -     1AG      4WR
```

SUBROUTINE AERROR (XLABEL)

```
      SUBROUTINE AERROR (XLABEL)
      COMMON C(3510)
      EQUIVALENCE (C(2020),LCONV)
      WRITE (5,20) XLABEL
   20 FORMAT  ( 43H0           OUT OF AERO TABLE ARGUMENT ARRAY , 5X,
     C   7HTABLE  ,A6 )
      DO 40 I=1202,1251,7
   40 WRITE(6,30) C(I),C(I+1),C(I+2),C(I+3),C(I+4),C(I+5),C(I+6)
   30 FORMAT(1H ,7E15.7)
      WRITE (6,30) C(2500),C(367),C(368),C(284),C(369),C(370),C(1117),
     C C(1118),C(1119),C(1120)
      LCONV=2
      RETURN
      END
```

```
I N D E X                                                                PAGE 155
                              SUBROUTINE AERROR (XLABEL)

     SYMBOL                                  REFERENCES

     20        -     4HR     5*
     30        -     7WR     8*      9WR
     40        -     600     7*
     AERROR    -     1
     G         -     2CO     3EO     7WR     9WR
     I         -     600     7WR
     LCONV     -     3EO     10=
     RETURN    -     11
     XLABEL    -     1AG     4WR
```

```
FUNCTION SIND (X)
SIND= SIN (X/57.29578)
RETURN
END
```

```
I N D E X                                                                PAGE 157
                                   FUNCTION SIND (X)
SYMBOL--- * * * * * * * * * * * * * * * * ---REFERENCES--- * * * * * * * * * * * * * * * * *

RETURN  -    3
SIN     -    2
SIND    -    1       2*
X       -    1AG     2

-*-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
FUNCTION-COSD (X)
COSD= COS (X/57.29578)
RETURN
END
```

```
I N D E X                                                             PAGE 159
                           FUNCTION COSD (X)
SYMBOL------*-*-*-*-*-*-*-*-*-*-*-*-*-*-- REFERENCES -- * * * * * * * * * * * * * * *

COS       -    2
COSD      -    1       2
RETURN    -    3
X         -    1AG     2

-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
```

```
                         FUNCTION ATAND (Y,X)

      1         FUNCTION ATAND (Y,X)
      2         ATAND= 57.29578*ATAN2 (Y,X)
      3         RETURN
      4         END
```

```
                                                                              PAGE 161
I N D E X
                                     FUNCTION ATAND (Y,X)

SYMBOL    - - - - - - - - - - - - - - - - - REFERENCES - - - - - - - - - - - - - - - - - -

ATAND     -      1        2*
ATAN2     -      2
RETURN    -      3
X         -      1AG      2
Y         -      1AG      2

-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

END

```
I N D E X                                                                    PAGE 163
                                *********** SUPER INDEX ***********

SYMBOL    ******************** ROUTINES IN WHICH THE SYMBOL APPEARS ********************
                    (AN * FOLLOWS ALL ROUTINE NAMES IN WHICH THE SYMBOL IS USED.)

ABS      -  AMRK*     A1*       C1*       C2*       C4*       DUMPO*    PLOT4*    STGE3*
            S1*
ADIVE    -  D1*
AGPAV    -  A2*       D1*
AGV      -  A2*
AINT     -  C1*
ALPHA    -  OTNFT1*
AMAX1    -  A1*       PLOT4*
AMIN1    -  PLOT4*
AMRK     -  MAIN*     STGE3*
ANGX     -  D1*
ANGY     -  D1*
ANGZ     -  D1*
APLOTV   -  PLOT4*
APRNTV   -  PLOT4*
ASIN     -  D1I*
ATAND    -  D1I*      G5*       S1*
ATAN2    -  ATAND*
ATHRST   -  D1*
ATURNT   -  D1*
AUXA1    -  DUMMY*    SUBL1*
AUXA2    -  DUMMY*    SUBL2*
AUXA3    -  DUMMY*    SUBL3*
AUXB1    -  DUMMY*    SUBL1*
AUXB2    -  DUMMY*    SUBL2*
AUXB3    -  DUMMY*    SUBL3*
AUXC1    -  DUMMY*    SUBL1*
AUXC2    -  DUMMY*    SUBL2*
AUXC3    -  DUMMY*    SUBL3*
AUXT     -  MAIN*
AUXSUB   -  AMRK*     MAIN*     STGE3*
AXRA     -  D1*
AXE      -  D1*
AYPA     -  D1*
AYE      -  D1*
AZPA     -  D1*
AZE      -  D1*
AG21     -  D2I*      G5*
AG22     -  D2I*      G5*
AG23     -  D2I*      G5*
AG31     -  D2I*      G5*
AG32     -  D2I*      G5*
AG33     -  D2I*      G5*
A1       -  AUXSUB*   C4*       DUMMY*
A1I      -  AUXI*     C4I*      DUMMY*
A1S      -  C4*
A2       -  AUXSUB*   C4*       DUMMY*
```

```
                              *********** SUPER INDEX ***********

A2I      -  AUXT*     DUMMY*
A3       -  AUXSB1*   DUMMY*
A3I      -  AUXT*     DUMMY*
A4       -  AUXSB2*   DUMMY*
A4I      -  AUXT*     DUMMY*
A5       -  AUXSB3*   DUMMY*
A5I      -  AUXT*     DUMMY*
B        -  OUTPT3*
BALPHA   -  A1*       A3I*      D1I*      G5*
BALPHP   -  A1*       G5*
BALPHT   -  A3*
BALPHY   -  A1*       A3I*      D1I*      G5*
BAP      -  A1*
BA2      -  BLDATA*
BA3      -  BLDATA*
BA5      -  BLDATA*
BDB      -  S1*
BDE      -  C4*
BDFLH    -  C4*
BDELT    -  C4*       C4I
BDELTC   -  C1*       C2*       C4*
BDELTD   -  C4*
BDELT1   -  A1*       C4I*
BDELT2   -  A1*       C4I*
BDELT3   -  A1*       C4I*
BDELT4   -  A1*       C4I*
BDL      -  A1*
BDM      -  A1*
BDN      -  A1*
BDP      -  A1*       C4I*
BDQ      -  A1*       C4I*
BDR      -  A1*       C4I*
BDS      -  C4*
BDSD     -  C4*
BEPSY    -  S1*
BEPSZ    -  S1*
BETA     -  OTNPT1*
BGAMH    -  G4*       G5*
BGAMT    -  D1*
BGAMV    -  G4*       G5*
BGDEFL   -  S1*
BH       -  C4*
BJJ      -  C1
BJJS     -  C1*       C2*
BJJSD    -  C1*       C2*
BJJSDD   -  C1*       C2*
BJJSP    -  C1*
BJJSS    -  C1*       C2*
BKK      -  C1*
BKKS     -  C1*       C2*
```

```
                    *********** SUPER INDEX ***********

BKKSD    -  C1*        C2*
BKKSDD   -  C1*        C2*
BKKSP    -  C1*
BKKSS    -  C1*        C2*
BLAMH    -  G5*
BLAMV    -  G5*
BLOSV    -  D1I*
BLCSS    -  C1*        C2*
BLCSSD   -  C1*        C2*
BLPSS    -  C1*        C2*
BLPSSD   -  C1*        C2*
BPHI     -  G5*
BPHIO    -  D1I*       D2I*
BPHIP    -  A1*        A3I*       G5*
BPHIT    -  A3*
BPH1     -  C1*        C2*        G5*
BPH2     -  C2*        G5*
BPS      -  C2*
BPSD     -  C2*
BPSDD    -  C2*
BPSI     -  G5*
BPSIG    -  D1I*       S1*
BPSIGD   -  S1*
BPSIO    -  D1I*       D2I*
BPSIS    -  C1*
BPSIT    -  D1*
BPSITD   -  D1*
BPSIH    -  G2*
BPSLV    -  G5*
BPS1     -  C1*        C2*        G5*
BTH      -  G2*
BTHD     -  C2*
BTHDD    -  C2*
BTHLV    -  G5*
BTHT     -  A2*        G5*
BTHTG    -  D1I*       S1*
BTHTGD   -  S1*
BTHTC    -  D1I*       D2I*
BTHTS    -  C1*
BTH2     -  C1*        C2*        G5*
BXX      -  C1*
BXXS     -  C1*        C2*
BXXSD    -  G1*        G2*
BXXSDD   -  C1*        C2*
BXXSP    -  C1*
BXXSS    -  G1*        G2*
BY       -  S1*
BZ       -  S1*
B21      -  G5*
B22      -  G5*
```

```
                    ********** SUPER INDEX **********

B23      -  G5*
B31      -  G5*
B32      -  G5*
B33      -  G5*
C        -  AERROR*   AHRK*     AUXI*     AUXSUB*   A1*       A2*       A3*       A3I*
            C1*       C1I*      C2*       C2I*      C4*       C4I*      DUMPO*    D1*
            D1I*      D2*       D2I*      G2*       G3*       G4*       G5*       MAIN*
            OINPT1*   OUPT2*    OUPT3*    RESET*    STGE2*    STGE3*    SUBL1*    SUBL2*
            SUBL3*    S1*       S1I*      TERROR*   ZERO*
CAGE     -  C1*       C2*       S1*
CAHRAV   -  PLOT4*
CA1      -  A1*       BLDATA*
CA2      -  A1*       BLDATA*
CA3      -  A1*       BLDAT3*
CA4      -  A1*       BLDATA*
CA5      -  A1*       BLDATA*
CB2      -  C4*
CFA11    -  D1*       D2*       D2I*      G5*
CFA11D   -  D2*
CFA12    -  D1*       D2*       D2I*      G5*
CFA12D   -  D2*
CFA13    -  D1*       D2*       D2I*      G5*
CFA13D   -  D2*
CFA21    -  D1*       D2*       D2I*      G5*
CFA21D   -  D2*
CFA22    -  D1*       D2*       D2I*      G5*
CFA22D   -  D2*
CFA23    -  A2*       D1*       D2*       D2I*      G5*
CFA23D   -  D2*
CFA31    -  D1*       D2*       D2I*      G5*
CFA31D   -  D2*
CFA32    -  D1*       D2*       D2I*      G5*
CFA32D   -  D2*
CFA33    -  A2*       D1*       D2*       D2I*      G5*
CFA33D   -  D2*
CFOVY    -  S1*
CFOVZ    -  S1*
CFREQ    -  A2*
CHD      -  C4*
CHDF     -  C4*
CH1      -  A1*       A2*
CH11     -  A1*
CH2      -  A1*       A2*
CH21     -  A1*
CH3      -  A1*       A2*
CH31     -  A1*
CH4      -  A1*       A2*
CH41     -  A1*
CISP     -  A3*
CL       -  A1*       A2*
```

```
CLDF     -  A1*        BLDATA*
CLDP     -  A1*
CLD1     -  BLDATA*
CLP      -  A1*        A2*        BLDATA*
CLPF     -  A1*        BLDATA*
CL2      -  BLDATA*
CL2F     -  A1*        BLDATA*
CL4      -  BLDATA*
CL4F     -  A1*        BLDATA*
CM       -  A1*        A2*
CMDF     -  A1*        BLDATA*
CMDQ     -  A1*
CMDR     -  A1*
CMD1     -  BLDATA*
CMO      -  A1*        BLDATA*
CMOF     -  A1*        BLDATA*
CMP      -  A1*        BLDATA*
CMPF     -  A1*        BLDATA*
CMQ      -  A1*        A2*        BLDATA*
CMQF     -  A1*        BLDATA*
CMS      -  A1*
CM2      -  BLDATA*
CM2F     -  A1*        BLDATA*
CN       -  A1*        A2*
CNP      -  A1*
CNR      -  A1*        A2*
CNTR1    -  DUMMY*     SUBL1*
CNTR2    -  DUMMY*     SUBL2*
CNTR3    -  DUMMY*     SUBL3*
CN4      -  BLDATA*
CN4F     -  A1*        BLDATA*
COS      -  COSD*
COSD     -  A1*        A3*        C2*        D1*        D1I*       D2I*       G2*        G4*
            G5*        S1*
COUNTV   -  DUMMY*     MAIN*
CPHAS    -  A2*
CPH1     -  G2*
CPP      -  OUPT2*     OUPT3*
CPSIO    -  D1I*
CR       -  C4*
CPAD     -  D1*        D1I*       D2*        G5*
CTH2     -  C2*
CX       -  A1*        A2*
CXO      -  A1*        BLDATA*
CXOF     -  A1*        BLDATA*
CY       -  A1*        A2*        S1*
CYP      -  A1*
CY4      -  BLDATA*
CY4F     -  A1*        BLDATA*
CZ       -  A1*        A2*        S1*
```

```
C2DF     -  A1*       BLDATA*
C2DQ     -  A1*
C2DR     -  A1*
C2D1     -  BLDATA*
CZO      -  A1*
CZP      -  A1*       BLDATA*
CZPF     -  A1*       BLDATA*
CZZ      -  BLDATA*
CZZF     -  A1*       BLDATA*
C1       -  AUXSUB*   DUMMY*
C1I      -  AUXI*     DUMMY*
C1R      -  AUXSUB*   DUMMY*
C1RI     -  AUXI*     DUMMY*
C2       -  AMRK*     AUXSUB*   DUMMY*
C2I      -  AUXI*     DUMMY*
C3       -  AMRK*     AUXSUB*   DUMMY*
C3I      -  AUXI*     DUMMY*
C4       -  AMRK*     AUXSUB*   DUMMY*
C4I      -  AUXI*     DUMMY*
C5       -  AUXSUB*   DUMMY*
C5I      -  AUXI*     DUMMY*
C6       -  AUXSUB*   DUMMY*
C6I      -  AUXI*     DUMMY*
C7       -  AUXSUB*   DUMMY*
C7I      -  AUXI*     DUMMY*
C8       -  AUXSUB*   DUMMY*
C8I      -  AUXI*     DUMMY*
C9       -  AUXSUB*   DUMMY*
C9I      -  AUXI*     DUMMY*
D        -  AMRK*
DCL1     -  A1*
DCL2     -  A1*
DCL4     -  A1*
DCM2     -  A1*
DCM4     -  A1*
DCY4     -  A1*
DCZ2     -  A1*
DEL      -  PLOT4*
DELT     -  AMRK*     MAIN*
DELX     -  PLOT4*
DELY     -  PLOT4*
DER      -  AUXSUB*   MAIN*     OUPT3*    STGE3*
DER1     -  OUPT3*
DJK      -  C1*
DMASS    -  A2*       A3*       A3I*      C1*
DOC      -  C4*       OUPT2*
DPHO     -  G3*
DT       -  S1*
DTCNT    -  OUPT2*    OUPT3*
DUMPO    -  MAIN*     OUPT2*    OUPT3*
```

```
                    *********** SUPER INDEX ***********

DWP      -  A3*
DWT      -  A3*       A3I*
DX       -  PLOT4*
DXDYV    -  PLOT4*
DXX      -  C1*
DY       -  PLOT4*
D1       -  AUXSUB*   DUMMY*
D1I      -  AUXI*     DUMMY*
D2       -  AUXSUB*   DUMMY*
D2I      -  AUXI*     DUMMY*
D3       -  AUXSUB*   DUMMY*
D3I      -  AUXI*     DUMMY*
D4       -  AUXSUB*   DUMMY*
D4I      -  AUXI*     DUMMY*
D5       -  AUXSUB*   DUMMY*
D5I      -  AUXI*     DUMMY*
EL       -  AMPK*     MAIN*     STGE3*
EOF      -  DINPT1*
EU       -  AMRK*     MAIN*     STGE3*
EXIT     -  MAIN*     TERRCP*
F        -  FINTP1*   FINTP2*   FINTP3*
FINTP1   -  C4*       FINTP2*   TABLE*
FINTP2   -  FINTP3*   TABL2*
FINTP3   -  TABL3*
FLG1     -  A2*
FLG2     -  A2*
FMHD     -  C4*
FMH1     -  A2*       C4*
FMH2     -  A2*       C4*
FMH3     -  A2*       C4*
FMH4     -  A2*       C4*
FMIX     -  A2*       A3*       A3I*      D2*
FMIXF    -  A3*       A3I*
FMIY     -  A2*       A3*       A3I*      D2*
FMIYF    -  A3*       A3I*
FMIZ     -  A2*       A3*       A3I*      D2*
FMXBA    -  A2*       D2*
FMXLUG   -  A2*
FMXTH    -  A2*       A3*       A3I*
FMYBA    -  A2*       D2*
FMYLUG   -  A2*
FMYTH    -  A2*       A3*       A3I*
FMZBA    -  A2*       D2*
FMZLUG   -  A2*
FMZTH    -  A2*       A3*       A3I*
FTHRST   -  A3*       A3I*
FTHX     -  A2*       A3*       A3I*
FTHY     -  A2*       A3*       A3I*
FTHZ     -  A2*       A3*       A3I*
FXBA     -  A2*       D1*
```

```
                    *********** SUPER INDEX **********

FYBA     -  A2*      D1*
FYLUG    -  A2*
F7PA     -  A2*      D1*
F7LUG    -  A2*
GBIAS    -  C1*      C2*
GCO      -  S1*
GJK      -  D1*      G2*
GN       -  C1*      C2*
GRAPH    -  AMRK     MAIN*     OUPT2*    OUPT3*    PLOT4*    STGE3
GRIDIV   -  PLOT4*
GS       -  S1*
GSX      -  S1*
GXX      -  G1*      G2*
G1       -  AUXSUB*  C4*       DUMMY*
G1I      -  AUXI*    DUMMY*
G2       -  AUXSUB*  C4*       DUMMY*
G2I      -  AUXI*    DUMMY*
G3       -  AUXSUB*  DUMMY*
G3I      -  AUXI*    DUMMY*
G4       -  AUXSUB*  DUMMY*    STGE3*
G4I      -  AUXI*    DUMMY*
G5       -  AUXSUB*  DUMMY*
G5I      -  AUXI*    DUMMY*
G6       -  AUXSUB*  DUMMY*
G6I      -  AUXI*    DUMMY*
HDEL     -  C4*
HJK      -  C1*      C2*
HMAX     -  AMRK*    MAIN*     STGE3*
HMIN     -  AMRK*    MAIN*     STGE3*
HXX      -  C1*
H1       -  C4*
H2       -  C4*
I        -  AERFOR*  AMRK*     AUXI*     AUXSUB*   C4*       DUMPO*    FINTP1*   FINTP2*
            FINTP3*  MAIN*     OINPT1*   OUPT3*    PLOT4*    RESET*    SUBL1*    SUBL2*
            SUBL3*
IA       -  A1*
IB       -  A1*
IC       -  A1*
ID       -  A1*
IE       -  A1*
IERR     -  PLOT4*
IF       -  A1*
II       -  PLOT4*
IJ       -  PLOT4*
IMARK    -  PLOT4*
INPT1    -  DUMMY*   SUBL1*
INPT2    -  DUMMY*   SUBL2*
INPT3    -  DUMMY*   SUBL3*
IPL      -  AUXSUB*  A3I*      C1I*      C2I*      C4*       C4I*      D1I*      D2I*
            MAIN*    S1I*
```

```
I N D E X
                       *********** SUPER INDEX ***********

IP       -  OINPT1*
IPR      -  PLOT4*
ITCNT    -  OUPT2*      OUPT3*
IX       -  PLOT4*
IXP      -  PLOT4*
IX1      -  PLOT4*
IX2      -  PLOT4*
IYP      -  PLOT4*
IYO      -  PLOT4*
IY1      -  PLOT4*
IY2      -  PLOT4*
J        -  AUXSUB*     DUMPO*      FINTP2*     FINTP3*     MAIN*       OINPT1*     OUPT2*      OUPT3*
            PLOT4*      RESET*      SUBL1*      SUBL2*      SUBL3*
JAP      -  OINPT1*
JJ       -  PLOT4*
JV       -  MAIN*
JX       -  PLOT4*
JY       -  PLOT4*
JYN      -  PLOT4*
JY1      -  PLOT4*
J1       -  AMRK*
J2       -  AMRK*
J3       -  AMRK*
J4       -  AMRK*
J5       -  AMRK*
J6       -  AMRK*
J7       -  AMRK*
J8       -  AMRK*
J9       -  AMRK*
K        -  DUMPO*      OINPT1*     OUPT2*      OUPT3*      PLOT4*
KASE     -  AMRK*       MAIN*       STGE3*
KCONV    -  OUPT2*      STGE2*      STGE3*
KIKSET   -  DUMMY*
KK       -  PLOT4*
KOUNT    -  AMRK*
KPOINT   -  OUPT3*
KSTEP    -  MAIN*       STGE2*      STGE3*
KC       -  AMRK*
K1       -  AMRK*
K2       -  AMRK*
K3       -  AMRK*
K4       -  AMRK*
K5       -  AMRK*
L        -  AUXI*       AUXSUB*     FINTP2*     FINTP3*     OINPT1*
LCONV    -  AERROR*     AUXSUB*     A1*         G4*         STGE2*      STGE3*      TERROR*
LESSPT   -  MAIN*
LINEV    -  PLOT4*
LISTNO   -  OINPT1*     RESET*
LOSTAT   -  OINPT1*     ZERO*
LSTEP    -  MAIN*
```

```
M        -  PLOT4*
MAND     -  OINPT1*
MIFR     -  OINPT1*
MOD      -  PLOT4*
MCDNO    -  OINPT1*
MRKPT    -  PLOT4*
N        -  AUXI*     AUXSUB*   A3I*      C1I*      C2I*      C4*       C4I*      DUMPC*
            D1I*      D2I*      FINTP1*   MAIN*     OINPT1*   PLOT4*    STGE3*    S1I*
NC1      -  A1*       BLDATA*
NC2      -  A1*       BLDATA*   C4*
NC3      -  A1*       BLDATA*
NC5      -  A1*       BLDATA*
NEWC     -  AMRK*
NEWP     -  AMRK*
NJ       -  AMRK*     MAIN*     STGE3*
NM       -  PLOT4*
NOLIST   -  OINPT1*   RESET*    ZERO*
NOMOD    -  AUXI*     AUXSUB*   OINPT1*   ZERO*
NOOUT    -  MAIN*     OINPT1*   OUPT2*    OUPT3*    ZERO*
NOPLOT   -  MAIN*     OINPT1*   OUPT2*    OUPT3*    PLOT4*
NOPNOM   -  OINPT1*   ZERO*
NOSTAT   -  OINPT1*   ZERO*
NOSUB    -  OINPT1*   SUBL1*    SUBL2*    SUBL3*    ZERO*
NP       -  PLOT4*
NPLOT    -  MAIN*     OINPT1*   ZERO*
NPLOT2   -  MAIN*     PLOT4*
NPLOT3   -  PLOT4*
NPLOT4   -  MAIN*     PLOT4*
NPT      -  AMRK*     MAIN*     STGE3*
NTH      -  A3*
NX       -  FINTP2*   FINTP3*   PLOT4*    TABLE*
NXD      -  FINTP2*
NXV      -  PLOT4*
NXY      -  TABL2*
NXYZ     -  TABL3*
NY       -  FINTP2*   FINTP3*   PLOT4*
NYV      -  PLOT4*
NZ       -  FINTP3*
NZI      -  TABL3*
N1       -  AMRK*
OINPT1   -  MAIN*
OLD      -  AMRK*
ONAME1   -  OINPT1*   OUPT3*
ONAME2   -  OINPT1*   OUPT3*
ONAME3   -  OINPT1*
ONAME4   -  OINPT1*
OPOINT   -  MAIN*     OUPT2*    OUPT3*    ZERO*
OPT      -  MAIN
OPTACT   -  C4*       C4I*
OPTARG   -  D1*       D1I*
```

```
I N D E X                                                                   PAGE 173
                              *********** SUPER INDEX ***********

     OPTHNG   -  A1*
     OPTM     -  A1*      C1*
     OPTNSK   -  S1*
     OPTNW    -  G2*
     OPTN10   -  MAIN*
     OPTN2    -  D1I*
     OPTN3    -  D2*
     OPTN4    -  A2*      C2*      D1I*     S1I*
     OPTN6    -  D1I*
     OUPT1    -  DUMMY*   SUBL1*
     OUPT2    -  DUMMY*   SUBL2*
     OUPT3    -  DUMMY*   STGE3*   SUBL3*
     OUTNG    -  OINPT1*  OUPT2*   OUPT3*
     OUTPLT   -  OINPT1*  OUPT2*   OUPT3*
     PCFTH    -  A2*
     PCNT     -  OUPT2*   OUPT3*   STGE3*
     PCT      -  FINTP1*  FINTP2*  FINTP3*  G4*
     PDYNMC   -  A2*      C4*      G3*
     PCGNT    -  OUPT2*   OUPT3*
     PH2      -  C2*
     PLOTN    -  DUMMY*   MAIN*    PLOT4*
     PLOTNO   -  MAIN*    OINPT1*  OUPT2*   OUPT3*
     PLOTN2   -  MAIN*
     PLOTN4   -  MAIN*
     PLOTV    -  PLOT4*
     PLOT2    -  DUMMY*   MAIN*    PLOT4*
     PLOT4    -  DUMMY*   MAIN*
     POINTV   -  PLOT4*
     PPNT     -  OUPT2*   OUPT3*
     PPP      -  OUPT3*
     PREDER   -  STGE3*
     PPINTV   -  PLOT4*
     PROCES   -  DUMMY*   MAIN*
     PS1      -  C2*
     PTLESS   -  MAIN*
     P1       -  AMRK*
     P2       -  AMRK*
     P3       -  AMRK*
     P4       -  AMRK*
     QBIAS    -  G1*      G2*
     QBURN    -  A2*      A3*      A3I*     G5*
     QNALGN   -  A3*
     RAIL     -  A2*      G2*      G5*
     RANGE    -  D1*      G4*      G5*      S1*
     RANGO    -  A2*      G2*      G5*
     RBIAS    -  G1*      G2*
     RRK      -  S1*
     ROCGF    -  A3*      A3I*
     ROCGO    -  A3*
     RDELCG   -  A2*      A3*      A3I*
```

```
RDFLX   -  D1*      G4*      G5*
RDFLY   -  D1*      G4*      G5*
RDFLZ   -  D1*      G4*      G5*
RDX     -  G4*
RDY     -  G4*
RDZ     -  G4*
REPPLT  -  MAIN*
RES     -  C1*
RESET   -  DUMMY*   MAIN*
RETURN  -  AERROR*  AMRK*    ATAMO*   AUXI*    AUXSUB*  A1*      A2*      A3*
           A3I*     COSD*    C1*      C1I*     C2*      C2I*     C4*      C4I*
           DUMMY*   DUMPO*   D1*      D1I*     D2*      D2I*     FINTP1*  FINTP2*
           FINTP3*  G2*      G3*      G4*      G5*      DINPT1*  OUPT2*   OUPT3*
           PLOT4*   RESET*   SINC*    STGE2*   STGE3*   SUBL1*   SUBL2*   SUBL3*
           S1*      S1I*     TABLE*   TABL2*   TABL3*   ZERO*
RFAREA  -  A2*      C4*
RFLGTH  -  A2*      C4*
RFXCG   -  A3*
RFYCG   -  A3*
RFZCG   -  A3*
RH      -  D1I*     G2*      G3*
RHW     -  G2*
RHZRO   -  D1I*     G3*
RITE    -  AMRK*    MAIN*
RKUTTA  -  MAIN*
RLCG    -  A2*      A3*
RLCGO   -  A3*
RLOCK   -  S1*
RLUG    -  A2*
RMISS   -  G4*
RN      -  MAIN*
RNDMNO  -  DINPT1*
RNDM1   -  DUMPY*   SUBL1*
RNDM2   -  DUMPY*   SUBL2*
RNDM3   -  DUMPY*   SUBL3*
RNT     -  MAIN*
RSLANT  -  D1I*
RTXE    -  D1*      D1I*     G5*
RTXED   -  D1*
RTYE    -  D1*      D1I*     G5*
RTYED   -  D1*
RTZE    -  D1*      D1I*     G5*
RTZED   -  D1*
RXDA    -  D1I*     G5*      S1*
RXE     -  D1*      D1I*     G5*
RXFD    -  D1*
RXF     -  G4*
RXFP    -  G4*
RXG     -  S1*
RXL     -  G5*
```

```
I N D E X                                                                        PAGE 175
                              *********** SUPER INDEX **********

RXO       -   D1I*      G5*
RYDA      -   D1I*      G5*       S1*
RYE       -   D1*       D1I*      G5*
RYED      -   D1*
RYF       -   G4*
RYFP      -   G4*
RYG       -   S1*
RYL       -   G5*
RYO       -   D1I*      G5*
RZPA      -   D1I*      G5*       S1*
RZCO      -   A2*
RZE       -   D1*       D1I*      G3*       G5*
RZEO      -   D1*
RZF       -   G4*
RZFP      -   G4*
RZG       -   S1*
RZL       -   G5*
RZO       -   D1I*      G5*
SEPS      -   S1*
SETMIV    -   PLCT4*
SIGN      -   C1*       C2*       C4*       S1*
SIGNO     -   OINPT1*
SIN       -   SIND*
SIND      -   A1*       A2*       A3*       C2*       D1*       D1I*      D2I*      G2*
              G4*       S1*
SNGL      -   AMRK*
SNPH      -   C1*
SNPS      -   C1*
SNTH      -   C1*
SPHI      -   C1*
SPHP      -   C1*
SPH1      -   C2*
SPSI      -   C1*
SPSP      -   C1*
SPTIME    -   AMRK*
SQRT      -   D1I*      G3*       G4*       G5*       S1*
ST        -   S1*
STATNO    -   OINPT1*
STEP      -   MAIN*     STGE3*
STGE1     -   DUMMY*    SUBL1*
STGE2     -   DUMMY*    SUBL2*
STGE3     -   DUMMY*    SUBL3*
STHP      -   C1*
STHT      -   C1*
STH2      -   C2*
STOP      -   MAIN*     OINPT1*
SUBL1     -   DUMMY*    MAIN*
SUBL2     -   DUMMY*    MAIN*
SUBL3     -   MAIN*
SUPNO     -   OINPT1*   SUBL1*    SUBL2*    SUBL3*
```

```
                          *********** $UFER INDEX ***********

SWP       -  S1*
S1        -  AUXSUB*    DUMMY*
S1I       -  AUXI*      DUMMY*
S10       -  AUXSUB*    DUMMY*
S10I      -  AUXI*      DUMMY*
S2        -  AUXSUB*    DUMMY*
S2I       -  AUXI*      DUMMY*
S3        -  AUXSUB*    DUMMY*
S3I       -  AUXI*      DUMMY*
S4        -  AUXSUB*    DUMMY*
S4I       -  AUXI*      DUMMY*
S5        -  AUXSUB*    DUMMY*
S5I       -  AUXI*      DUMMY*
S6        -  AUXSUB*    DUMMY*
S6I       -  AUXI*      DUMMY*
S7        -  AUXSUB*    DUMMY*
S7I       -  AUXI*      DUMMY*
S8        -  AUXSUB*    DUMMY*     MAIN*
S8I       -  AUXI*      DUMMY*
S9        -  AUXSUB*    DUMMY*
S9I       -  AUXI*      DUMMY*
T         -  AMRK*      AUXSUB*    A1*        A2*        A3*        C1*        C2*        C4*
             D1*        FINTP2*    FINTP3*    G2*        G4*        G5*        MAIN*      OUPT2*
             OUPT3*     PLOT4*     STGE3*     S1*
TABLE     -  A1*        A3*
TAPL2     -  A1*
TAPE      -  OUPT2*     OUPT3*
TAPEND    -  OUPT2*     OUPT3*
TDY       -  C1*        C2*
TEMP      -  AMRK*
TEMPS     -  MAIN*      STGE3*
TF        -  STGE3*
THA       -  A3*
THF       -  A3*
TIME      -  MAIN*      OUPT2*     OUPT3*
TIMEV     -  DUMMY*     MAIN*
TKOR      -  S1*
TKRY      -  C1*        C2*        S1*
TKPZ      -  C1*        C2*        S1*
TMF       -  AMRK*
TPH2      -  C2*
TPS1      -  C2*
TRKZY     -  S1*
TTHTG     -  D1I*
TZERO     -  G4*
UAL       -  A1*
UBT       -  A1*
UB11      -  S1*
UB12      -  S1*
UB13      -  S1*
```

```
                    ********** SUPER INDEX **********

UR21     -  S1*
UR22     -  S1*
UR23     -  S1*
UR31     -  S1*
UR32     -  S1*
UR33     -  S1*
UCP      -  D1I*       S1*
UCPHI    -  A1*        D2I*
UCPSI    -  D2I*
UCT      -  D1I*       S1*
UCTHT    -  D2I*
UC11     -  G4*
UC12     -  G4*
UC13     -  G4*
UC21     -  G4*
UC22     -  G4*
UC31     -  G4*
UC32     -  G4*
UC33     -  G4*
UDELX    -  G4*
UDELY    -  G4*
UDELZ    -  G4*
UDL1     -  A1*
UDL2     -  A1*
UDL3     -  A1*
UDL4     -  A1*
UDMP     -  A3*
UFZL2    -  A2*
UIMP     -  A3*
UIMPD    -  A3*
UL2V     -  A2*
UPHI     -  C1*
UPSI     -  C1*
UQS      -  A2*        C4*
UQSL     -  A2*        C4*
USINA    -  A3*
USP      -  D1I*       S1*
USPHI    -  A1*        D2I*
USPSI    -  D2I*
UST      -  D1I*       S1*
USTHT    -  D2I*
US2PHI   -  A1*
US2PH2   -  A1*
US4PHI   -  A1*
UT       -  A1*        G4*
UTHT     -  G1*
UTIME    -  PLOT4*
UVP1     -  G5*
UVP2     -  G5*
UVP3     -  G5*
```

```
                                ********** SUPER INDEX **********

UVP4     -  G5*
UXFP     -  G4*
UY       -  S1*
UYFP     -  G4*
UYK      -  S1*
UZ       -  S1*
UZFP     -  G4*
UZK      -  S1*
V        -  AIRPK*
VAIRSP   -  A2*       G3*       G5*
VALUE    -  OINPT1*   RESET*
VAP      -  AUXSUB*   C1        C2        C4        MAIN*     STGE3*
VCLSNG   -  D1*
VDELX    -  D1*
VDELY    -  D1*
VDELZ    -  D1*
VDXB     -  D1*
VDYB     -  D1*
VDZB     -  D1*
VLABLE   -  MAIN*     OINPT1*
VM       -  A1*
VMACH    -  A1*       A2*       C4*       D1I*      G3*
VMMTE    -  D1I*
VMMU     -  G5*
VMMV     -  G5*
VMMW     -  G5*
VMMXE    -  G3*       G5*
VMMXY    -  D1I*
VMMYE    -  G3*       G5*
VMMZE    -  G3*       G5*
VM1      -  BLDATA*
VM2      -  BLDATA*
VM3      -  BLDATA*
VM4      -  BLDATA*
VM5      -  BLDATA*
VR       -  OINPT1*
VSOUND   -  D1I*      G3*
VTARG    -  D1*
VTARGD   -  D1*
VTOTE    -  G5*
VTXE     -  D1*
VTYE     -  D1*
VTZE     -  D1*
VWTE     -  G2*
VWXE     -  D1I*      G2*       G3*
VWYE     -  D1I*      G2*       G3*
VWZE     -  D1I*      G2*       G3*
VXF      -  D1*       D1I*      G3*       G5*
VXFO     -  D1*
VXO      -  D1I*      G5*
```

********** SUPER INDEX **********

```
I N D E X                                                                                   PAGE 180
                                   *********** SUFER INDEX **********

WOP      -  S1*
WR       -  A2*       A3I*      C2*       D2*       S1*
WFC      -  C1*       C2*
WPD      -  C2*       D2*
WRITE    -  DUMMY*
WRP      -  S1*
WSL      -  S1*
WSN      -  S1*
WXX      -  C1*
WY       -  S1*
WZ       -  S1*
W1       -  C4*
X        -  ATAND*    COSD*     FINTP1*   FINTP2*   FINTP3*   SIND*     TABLE*    TABL2*
            TABL3*
XF       -  A1*       A3*       C4*
XI       -  FINTP1*   FINTP2*   FINTP3*   TABLE*
XINTER   -  TABL2*    TABL3*
XK       -  TABLE*
XL       -  FINTP1*   FINTP2*   FINTP3*
XLABEL   -  AERROR*   TABLE*    TABL2*    TABL3*    TERROR*
XMAX     -  PLOT4*
XMIN     -  PLOT4*
XMOOMO   -  AUXI*     AUXSUB*
XM1      -  PLOT4*
XM2      -  PLOT4*
XT1      -  PLOT4*
XT2      -  PLOT4*
XYI      -  TABL2*
XYZI     -  TABL3*
X1       -  PLOT4*
X2       -  PLOT4*
Y        -  ATAND*    FINTP2*   FINTP3*   DINPT1*   TABLE*    TABL2*    TABL3*
YI       -  FINTP1*   FINTP2*   FINTP3*   TABLE*
YL       -  PLOT4*
YMAX     -  PLOT4*
YMIN     -  PLOT4*
YM1      -  PLOT4*
YM2      -  PLOT4*
YT1      -  PLOT4*
YT2      -  PLOT4*
Y1       -  PLOT4*
Y2       -  PLOT4*
Z        -  FINTP3*   TABL2*    TABL3*
ZERO     -  MAIN*
ZI       -  FINTP2*   FINTP3*   TABL2*
ZN       -  C4*

THE FOLLOWING   2 SYMBOLS ARE NOT USED IN ANY OF THE ROUTINES LISTED ABOVE -

            OPT   *   TEMPS
```