

UNCLASSIFIED

AD NUMBER
AD892611
NEW LIMITATION CHANGE
TO Approved for public release, distribution unlimited
FROM Distribution authorized to U.S. Gov't. agencies only; Administrative/Operational Use; Dec 1971. Other requests shall be referred to Air Force Armament Lab., Eglin AFB, FL.
AUTHORITY
AFATL ltr, 24 Jun 1974

THIS PAGE IS UNCLASSIFIED

AFATL-TR-71-169
VOLUME I

AD892611

CLOSE AIR SUPPORT MISSILE
GUIDANCE AND CONTROL STUDY
VOLUME I. SIX-DEGREE-OF-FREEDOM SIMULATION

DEPARTMENT OF MECHANICAL ENGINEERING
THE UNIVERSITY OF FLORIDA

TECHNICAL REPORT AFATL-TR-71-169, VOLUME I

DECEMBER 1971

Test & Evaluation

Distribution limited to U. S. Government agencies only;
~~this report documents the close air support missile guid-~~
~~ance and control study;~~ distribution limitation applied
December 1971. Other requests for this document must be
referred to the Air Force Armament Laboratory (DLWG),
Eglin Air Force Base, Florida 32542.

AIR FORCE ARMAMENT LABORATORY

AIR FORCE SYSTEMS COMMAND • UNITED STATES AIR FORCE

EGLIN AIR FORCE BASE, FLORIDA

D D C
RECEIVED
MAR 24 1972
B

**Close Air Support Missile
Guidance And Control Study**

Volume I. Six-Degree-Of-Freedom Simulation

J. Mahig

Test & Evaluation

Distribution limited to U. S. Government agencies only;
~~this report documents the close air support missile guid-~~
~~ance and control study;~~ distribution limitation applied
December 1971. Other requests for this document must be
referred to the Air Force Armament Laboratory (DLNG),
Eglin Air Force Base, Florida 32542.

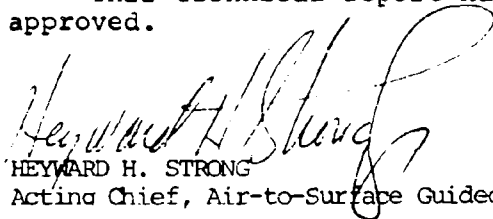
FOREWORD

This report was prepared by the Industrial and Experiment Station, Department of Mechanical Engineering, University of Florida, Gainesville, Florida, under Contract No. F08635-71-C-0073 with the Air Force Armament Laboratory, Eglin Air Force Base, Florida, during the period from 9 December 1970 to 9 December 1971. Lieutenant Robert J. Karner (DLWG) monitored the project for the Armament Laboratory.

The principal investigator for the contractor was Dr. J. Mahig.

This report consists of two volumes. Volume I is devoted to the Six-Degree-of-Freedom Simulation while Volume II is concerned with the Three-Degree-of-Freedom Simulation. This is Volume I.

This technical report has been reviewed and is approved.



HEYWARD H. STRONG

Acting Chief, Air-to-Surface Guided Weapons Div.

ABSTRACT

This report describes a six-degree-of-freedom program which can be used to determine the trajectory and miss distance of a missile system. The options for the program are such as to permit variation of the aerodynamics, seeker, autopilot, actuator, and missile motor performance for the purpose of accurately simulating a given missile design and evaluating the effects of any changes in system parameters. Sufficient detail has been included in the text in order to minimize the users' effort needed to know how to update or modify the program for his purposes.

TEST + EVALUATION

Distribution limited to U. S. Government agencies only; ~~this report documents the close air support missile guidance and control study~~ distribution limitation applied December 1971. Other requests for this document must be referred to the Air Force Armament Laboratory (DLWG), Eglin Air Force Base, Florida 32542.

TABLE OF CONTENTS

Section		Page
I	INTRODUCTION	1
II	PROGRAM DESCRIPTION	2
	Subroutines, Modules, and Tables	2
	A2 - Aero Forces and Moments	5
	D1, D2 Translational and Ro- tational Dynamics Module	9
	List of Symbols	9
	Equations of Motion	11
	Subroutine G2	13
	Subroutine C10	17
	Subroutine QUADET	17
	Subroutine S1 (Module)	19
	C1 - Autopilot Module	22
	C4 - Actuator Module	22
	Fin Deflection	31
	A3 - Engine Module	31
III	VARIABLE LOCATIONS	34
	Variable Names, Block Locations, Definitions	34
	Subroutine Call Sequence	50
	State Variables	50

TABLE OF CONTENTS (CONCLUDED)

Section		Page
IV	INPUT REQUIREMENTS	56
	Initial Conditions	56
V	PROGRAM LISTINGS	58
	Complete Six-Degree-of-Freedom Program Listing with Example	58
Appendix		
I	COORDINATE TRANSFORMATION FROM BODY TO GIMBAL AXIS SYSTEM	121
II	HIGH FREQUENCY ACTUATOR PROGRAM LISTING	124
III	HIGH FREQUENCY AUTOPILOT PROGRAM LISTING	128

LIST OF FIGURES

Figure	Title	Page
1.	Flow Chart for State Variable Calculations	3
2.	Wind Axis System	6
3.	Definitions of Angles and Co-ordinate Systems	12
4.	Euler Angles Between Body Axis and Inertial Axis	14
5.	Six-Degree-of-Freedom Equations of Motion	15
6.	Coordinate System Associated with Wind and Gust Module	16
7.	Quadrant Detector Geometry	18
8.	Coordinate Relations Between Body and Gimbal Axis System and the Line of Sight	20
9.	Schematic Diagram of Platform Gimbal Angles	21
10.	Autopilot High Frequency Model	23
11.	Pitch Rate Gyro	24
12.	Yaw Rate Gyro	25
13.	Autopilot Low Frequency Model	26
14.	High Frequency Actuator	27
15.	Actuator Torque Balance System	28
16.	Actuator Position Loop Block Diagram	29
17.	Fin Sign Conventions	32

LIST OF FIGURES (CONCLUDED)

Figure	Title	Page
18.	Offset Thrust Vector Coordinate System	33
19.	Data Card Formats and Deck Setup	57
I-1.	Angles Between Gimbal and Body Axes	122

LIST OF TABLES

Table	Title	Page
I	Subroutine and Module List Six-Degree-of-Freedom Digital Program	4
II	Correspondence Between Variable Names, Aero Symbols, and their Common Location	8
III	Blank Common Assignments	35
IV	Common Location, Variable Name, and Definition	36
V	Initialization Subroutine Call Sequence	51
VI	State Variables and Derivatives Names and Location Codes	52

SECTION I

INTRODUCTION

The purpose of this report is to provide a reference which will enable ready access to the use of a six-degree-of-freedom program which is capable of accurately determining the trajectory and miss distance of a semi-active or passive guided missile. The program is divided into convenient blocks, called modules or subroutines, which do specific tasks: e.g., determine aerodynamic forces, seeker output, state of autopilot, current value of thrust, etc. As a result, the user will be able to easily locate the section of the program where specific calculations are performed and modify them or, if necessary, to add modules to achieve other purposes.

This program has been derived from a program in the library of the North American Rockwell Company, Columbus Division, and is described in NR 70H-232-1 and -2. The purpose of the program was to determine trajectory and miss distance of an air-to-air or air-to-surface missile. This manual goes into somewhat greater detail in identifying the variables and defining coordinate systems than heretofore. This has been possible because of the extensive work carried out with the program by the author in satisfying the requirements of this contract and information supplied by Mr. R. D. Ehrich and P. D. Capcara of the North American Rockwell (NAR) Corporation. The program described below has been modified from the original version supplied to USAF by North American Rockwell Corporation by personnel at the Air Force Armament Laboratory to permit the consideration of the effect of a random spot motion on the miss distance of a laser guided missile. Incorporated into the version presented in this report are additional capabilities which provide an accurate simulation of the quadrant detector, range closure, proportional lead guidance, simplified program reset mechanism for multiple runs, greater target maneuverability in air-to-air simulations, and a more general high frequency actuator routine which will accept either experimental or theoretically derived transfer functions.

SECTION II

PROGRAM DESCRIPTION

2.1 Subroutines, Modules, and Tables

A complete listing of this program appears in Section V. The program consists of three types of subprograms:

- (a) Tables of aerodynamic coefficients in block data form.
- (b) Modules describing missile subsystems.
- (c) Executive subroutines and the main program.

The block data subroutines must be physically located at the front of the program deck after the main program for proper operation. Data is extracted from these tables in the module A1 which makes use of the table look-up subroutines TABL1, TABL2, and TABL3 which form a part of the executive routines.

For each module (e.g., A1, C4) the programmer has the option of using an associated initialization module (e.g., C4I). These initialization modules may be used to compute initial conditions from input data or add to the list of state variables to be integrated. The initialization modules are executed only once at the start of each simulated mission. It is in the modules themselves (e.g., C4) that the derivatives of the state variables are computed. Time is incremented by a fixed amount (Δt) after every other pass since a predictor-corrector integration algorithm is used.

A large block common array, called C, allows the communication of certain variables between modules and subroutines for input/output, integration, and control purposes.

The mathematical relationship of various modules and subroutines are shown in Figure 1, and a corresponding list of the modules is given in Table I.

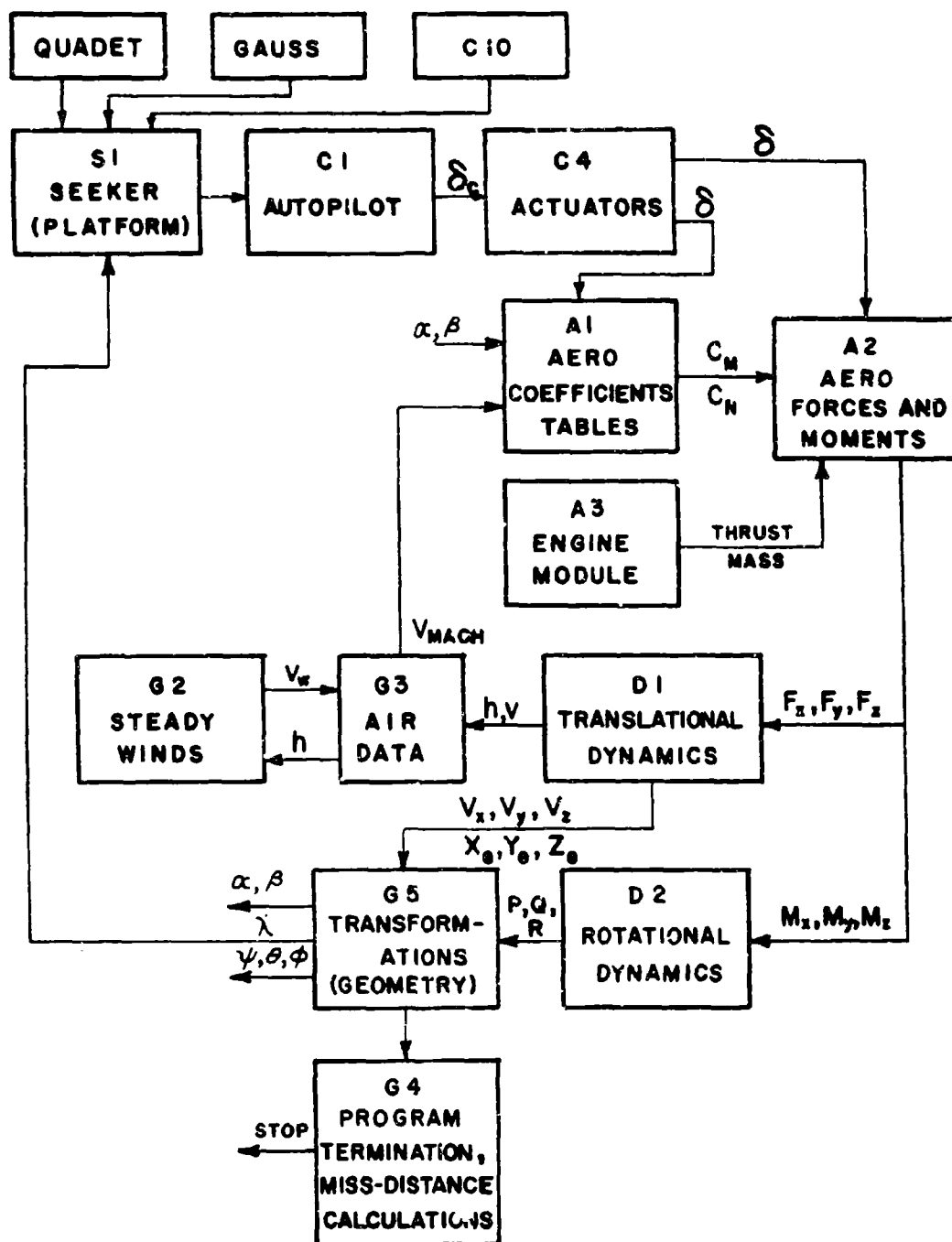


Figure 1. Flow Chart for State Variable Calculations

TABLE I. SUBROUTINE AND MODULE LIST SIX-DEGREE-
OF-FREEDOM DIGITAL PROGRAM

I	GEOPHYSICAL AND EXTERNAL ENVIRONMENT
	G2 - Steady winds
	G3 - Air data - including dynamic pressure, density, speed of sound
	G4 - Terminal geometry - computes miss distance
	G5 - Transformations of position and velocity between various coordinate systems
II	SENSORS
	C10 - Spot motion - including boresight error, aiming error, hotspot motion, etc.
	S1 - Seeker - Seeker performance and platform motion
	QUADET - Quadrant detector simulator
III	COMPUTERS
	C1 - Autopilot - computes steering commands from seeker output
	C4 - Actuators - includes flap motion and limits
IV	AIRFRAME
	A1 - Aerodynamics coefficients - table look-up
	A2 - Aerodynamic forces and moments - in wind axis, includes forces and moments on lugs while missile is on rail
	A3 - Engine - computes thrust forces as well as c.g. shifts and mass changes.
V	DYNAMICS
	D1 - Translation dynamics of missile - accelera- tions in body axes are transformed into earth coordinates and integrated into velocities and positions.
	D2 - Rotational dynamics of missile - computes rotational accelerations and velocities referred to missile body axes.

2.2 A2 - Aero Forces and Moments

Figure 2 shows the relationship between the body axis and wind axis coordinate system. In addition, the coordinate directions are shown for the positive direction of the dimensionless aerodynamics' coefficients in both the body axis system and the wind or primed axis system. The body axis system and the wind axis system are related by the following system of equations:

$$\begin{bmatrix} X_B \\ Y_B \\ Z_B \end{bmatrix} = \begin{bmatrix} \phi' \end{bmatrix} \begin{bmatrix} X'_B \\ Y'_B \\ Z'_B \end{bmatrix}$$

The aerodynamic coefficients are functions of the aerodynamic roll angle (ϕ') and angle of attack α' . It can be seen that the angles ϕ' and α' locate the wind vector in much the same way that a magnitude r and angle θ locate a vector in polar coordinates. With reference to Figure 2, it is apparent that the plane containing the wind vector is obtained by rotating the $X_B Z_B$ plane through ϕ' about the missile centerline (X_B axis). The wind vector is located in this plane by the angle α' measured from the X_B axis. The angles ϕ' and α' are related to the angle of attack α and sideslip β by the following equations:

$$\cos \alpha' = \cos \alpha \cos \beta$$

$$\sin \phi' = \sin \beta / (1 - \cos^2 \alpha \cos^2 \beta)^{1/2}$$

where if α and β are small, one finds

$$\alpha'^2 = \alpha^2 + \beta^2.$$

Since if α and β are small, α' will similarly be small and it will be found that

$$\beta = \alpha' \sin \phi'$$

$$\alpha = \alpha' \cos \phi'$$

$$\tan \phi' = \beta / \alpha$$

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

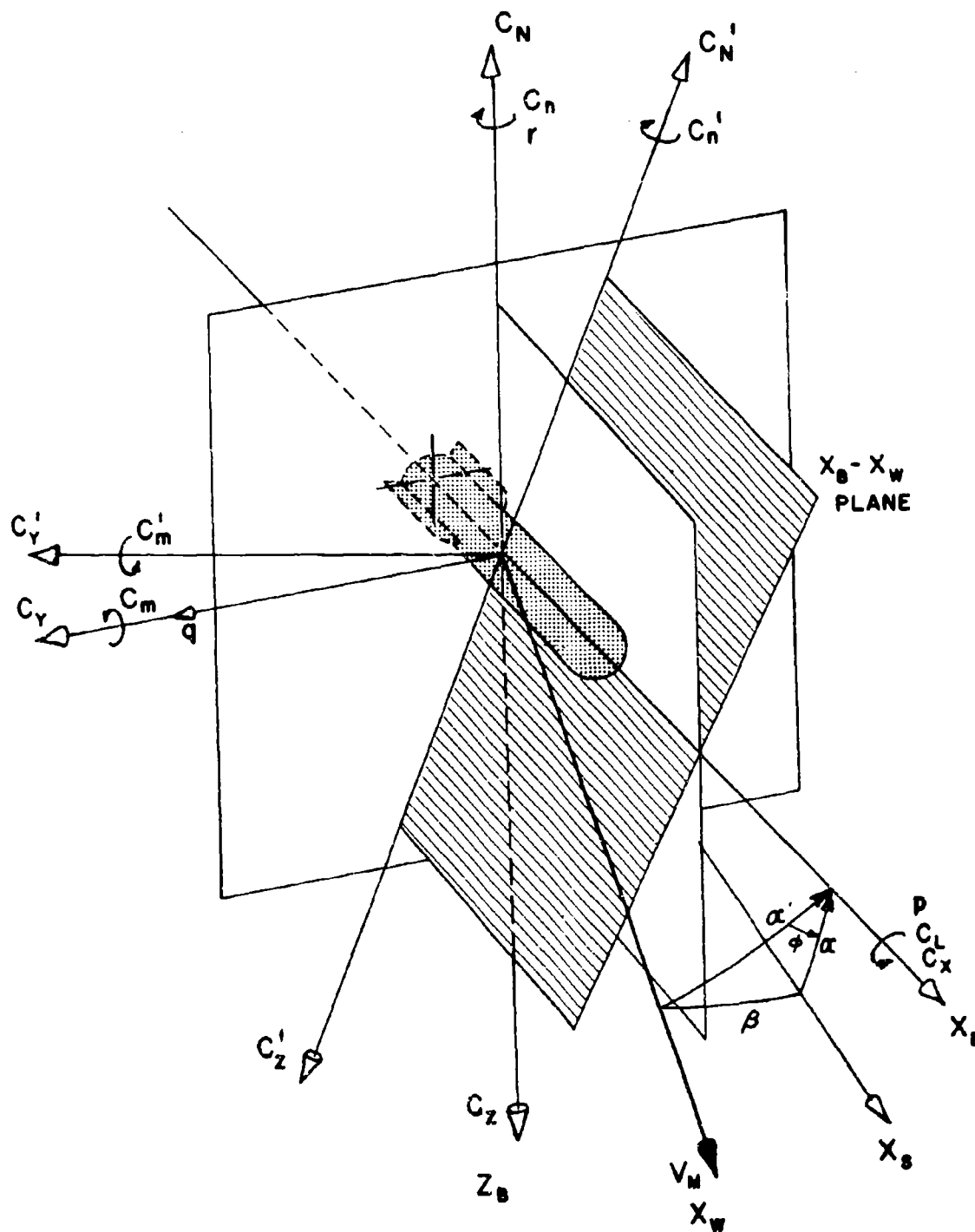


Figure 2. Wind Axis System

ϕ'_A is the aerodynamic roll angle referenced to zero with the missile flying in the + configuration. If the missile is intended to fly in the X configuration, ϕ'_A equals 45° with $\beta = 0$. Thus, $\phi' = \phi'_A - 45^\circ$.

It will be found that the following relationships hold with respect to ϕ and ϕ'_A :

$$\cos 4\phi'_A = -\cos 4\phi'$$

$$\sin 4\phi'_A = -\sin 4\phi'$$

$$\cos (\phi_A - 45^\circ) = \cos \phi'$$

$$\sin (\phi_A - 45^\circ) = \sin \phi'.$$

Some of the above relations can be experienced in terms of the angle ϕ' .

In order to facilitate application to the program, Table II lists the correspondence between variable names, commonly used aero symbols, and their COMMON location in the program.

TABLE II. CORRESPONDENCE BETWEEN VARIABLE NAMES,
AERO SYMBOLS, AND THEIR COMMON LOCATION

Name	Symbol	Common Location	Name	Symbol	Common Location
CX	C_A	1203	CLDRP	$C_n' / \delta r$	1225
CY	C_Y	1204	CNQ	$C_n' / \delta q$	1226
CZ	C_Z	1205	CLD	$C_{\ell}' / \delta p$	1227
CLP	$C_{\ell p}$	1206	CLMP	C_m'	1228
CMQ	C_{mq}	1207	CLNP	C_n'	1229
CNR	C_{nr}	1208	BDEFL	$ \delta $	1230
CL	C_{ℓ}'	1209	CDCM	$C_m'(\phi')$	1231
CM	C_M	1210	BDL	δ_p	1232
CN	C_N	1211	BDM	δ_q	1233
CXO	C_A	1212	BDN	δ_r	1234
CXC	$C_A'(\text{trim})$	1213	CDCN	$C_N'(\phi')$	1235
CNPT	$C_N'(\alpha')$	1214	CL2	$C_{\ell}'(\phi)$	1240
CY2	$C_Y'(\phi')$	1215	CL3	$C_{\ell}'(\phi) \text{ lug}$	1241
CMO	$C_M'(\alpha')$	1217	CNPU	$C_N'(\phi', \alpha')$	1244
CN2	$C_N'(\phi')$	1218	CYPU	$C_Y'(\phi)$	1245
CZQ	$C_N' / \delta q$	1219	CMP	$C_m'(\alpha', \phi')$	1247
CZR	$C_N' / \delta r$	1220	CNP	$C_n'(\phi')$	1248
CHDQP	$C_M' / \delta q$	1221	CLR	$C_{\ell}'(\phi')$	1249
CMR	$C_M' / \delta r$	1222	CZP	C_N'	1250
CYR	$C_Y' / \delta r$	1223	CYP	C_Y'	1251
CYQ	$C_Y' / \delta q$	1224			

2.3.1 D1,D2 - Translational and Rotational Dynamics Module

The following list of symbols applies to the equations of motion which are developed in paragraph 2.3.3 for modules D1 and D2.

2.3.2 List of Symbols

- m - F(t) instantaneous mass (slugs)
- p - Rolling velocity = angular velocity along X axis (rad/sec)
- q - Pitching velocity = angular velocity along Y axis (rad/sec)
- r - Yawing velocity = angular velocity along Z axis (rad/sec)
- I_X - Moment of inertia about X axis (slug-ft)
- I_Y - Moment of inertia about Y axis (slug-ft)
- I_Z - Moment of inertia about Z axis (slug-ft)
($I_Z = I_Y$ for a perfectly symmetrical missile)
- U - Linear velocity along the X body (X_E) axis (ft/sec)
- $V = U_\alpha$ - Linear velocity along the Y body (Y_E) axis (ft/sec)
- $W = U_\beta$ - Linear velocity along the Z body (Z_B) axis (ft/sec)
- X_B, Y_B, Z_B - Airframe axis system that moves with airframes
- X_e, Y_e, Z_e - Earth coordinates
- α - Angle of attack = angle between a fuselage reference line and the relative wind in the X_B, Z_B plane (rad)

$$\tan \alpha = W/U; \alpha = W/U$$

- β - Angle of sideslip (rad)

$$\tan \beta = V / \sqrt{U^2 + W^2}; \beta = V/U$$

Euler angles

- ψ, θ , and ϕ - ψ is the rotation about Z_B , θ is the rotation about Y_B , and ϕ is the rotation about X_B in that order (rad)
- g - Acceleration due to gravity (ft/sec²)
- T - Thrust along X_B
- C_N, C_Y, C_C, C_R
 C_m, C_n - Dimensionless aerodynamics coefficients (body axes)
- C'_N, C'_Y, C'_C
 C'_R, C'_m, C'_n - Dimensionless aerodynamics coefficients (primed axes system - Figure 2)
- Density (slug/ft³)
- q_0 - Dynamic pressure = $\frac{1}{2}\rho U^2$ (lb/ft²)
- S - Body reference cross sectional area (ft²)
- l - Reference body length (ft)
- ΔX - Shift of center of gravity from a reference point along the X_B axis (ft) - negative aft
- δ - Control surface deflection (rad)
- δ_q - Control surface deflection to give pitching motion (rad)
- δ_p - Control surface deflection to give rolling motion (rad)
- δ_r - Control surface deflection to give yawing motion (rad)
- C_{ij} - Dimensionless aerodynamic derivatives
- d' - Aerodynamic or wind angles of attack (rad) - Figure 2
- ϕ' - Aerodynamic roll angle (rad) - Figure 2
- ϕ'_A - Aerodynamic roll angle (rad) referenced to zero when flying in the + configuration.
 $\phi'_A = \phi' + 45^\circ$

2.3.3 Equations of Motion

The six-degree-of-freedom equations of motion implemented in the computer program in terms of the body axes are given below*. (See Figure 3 for coordinate system orientation.)

Longitudinal Force

$$\Sigma F_X = m[U + U_\alpha q - U_\beta r] = q_0 S_\pi C_C - mg \sin \theta + T$$

Lateral Force

$$\Sigma F_Y = m[d/dt(U_\beta) + U r - U_\alpha p] = q_0 S_\pi [\cos \phi 'C'_Y - \sin \phi 'C'_N] + mg \cos \theta \sin \phi$$

Vertical Force

$$\Sigma F_Z = m[d/dt(U_\alpha) + U_\beta p - U q] = -q_0 S_\pi [\cos \phi 'C'_N + \sin \phi 'C'_Y] + mg \cos \theta \cos \phi$$

Rolling Moment

$$\Sigma M_X = I_X p = q_0 S_\pi \ell [C_\ell + \ell/2 U C_{\ell p} P]$$

Pitching Moment

$$\Sigma M_Y = I_Y q + (I_X - I_Z) p r = q_0 S_\pi \ell [\cos \phi 'C'_m + \sin \phi 'C'_n + \ell/2 U C_{mq} q - \Delta X/\ell (\sin \phi 'C'_Y + \cos \phi 'C'_N)]$$

Yawing Moment

$$\Sigma M_Z = I_Z r + (I_Y - I_X) p q = q_0 S_\pi \ell [\cos \phi 'C'_n - \sin \phi 'C'_m + \ell/2 U C_{nr} r - \Delta X/\ell (\cos \phi 'C'_Y - \sin \phi 'C'_N)]$$

- * $U \sim U$ velocity in X direction
 $V \sim U_\alpha$ velocity in Y direction
 $W \sim U_\beta$ velocity in Z direction

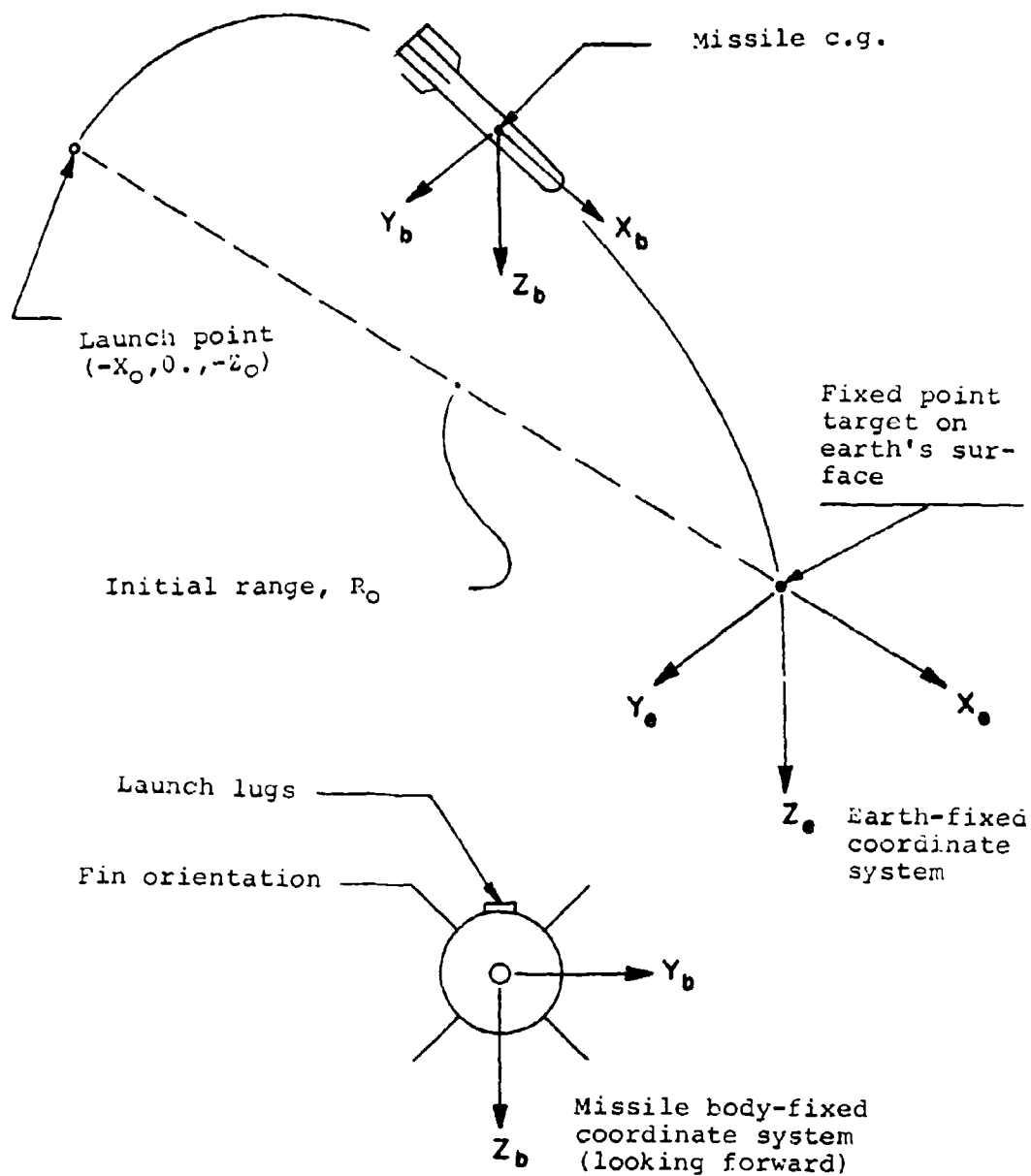


Figure 3. Definitions of Angles and Coordinate Systems

The Euler transformations of

$$\dot{\psi} = p = r \cos \phi / \cos \theta + q \sin \phi / \cos \theta$$

$$\dot{\theta} = q = q \cos \phi - r \sin \phi$$

$$\dot{\phi} = r = p + (r \cos \phi + q \sin \phi) \tan \theta$$

The velocity, in terms of the earth axes, can be obtained as:

$$\begin{aligned} \dot{X} = & U \cos \theta \cos \psi + U_{\beta} (\sin \phi \sin \theta \cos \psi - \cos \phi \sin \psi) \\ & + U_{\alpha} (\cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi) \end{aligned}$$

$$\begin{aligned} \dot{Y} = & U \cos \theta \sin \psi + U_{\beta} (\cos \phi \cos \psi + \sin \psi \sin \theta \sin \phi) \\ & + U_{\alpha} (\cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi) \end{aligned}$$

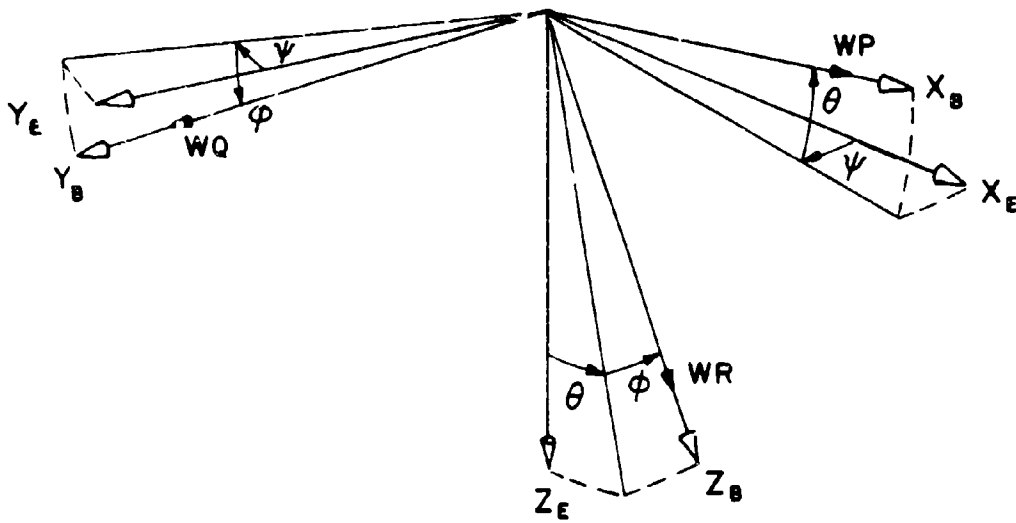
$$\dot{Z} = -U \sin \theta + U_{\beta} \sin \phi \cos \theta + U_{\alpha} \cos \theta \cos \phi$$

The Euler angles, shown in Figure 4, and the position of the missile in earth coordinates can be obtained through an integration of the above equations.

The block diagram of the implementation of the equations of motion and the Euler transformations are shown in Figure 5.

2.4 Subroutine G2

This subroutine is called the wind and gust module. This module determines the velocity and direction of the wind. The module assumes that there is no wind above an altitude RHW. Below that altitude the wind direction and magnitude are assumed to be constant throughout a layer RWINC in depth. (It should be noted that RWINC is measured along the line of sight. Since most missiles fly with only small deviation from the original line of sight, the altitude increments, if needed, may be readily estimated.) Two random variables are associated with the wind in each layer: the magnitude and angular orientation which are considered constant in each layer. The mean value of the wind magnitude is VWTE, and its standard deviation is given as SW. The mean value of the angular orientation of the wind in a layer is BPSIW, and the standard deviation of the angular variation is SW1. The current value of the wind magnitude and direction is given by VWTEV and BPSIWV, respectively. The relationship between these mean values and the inertial coordinate system is shown in Figure 6.



Body Coordinate System

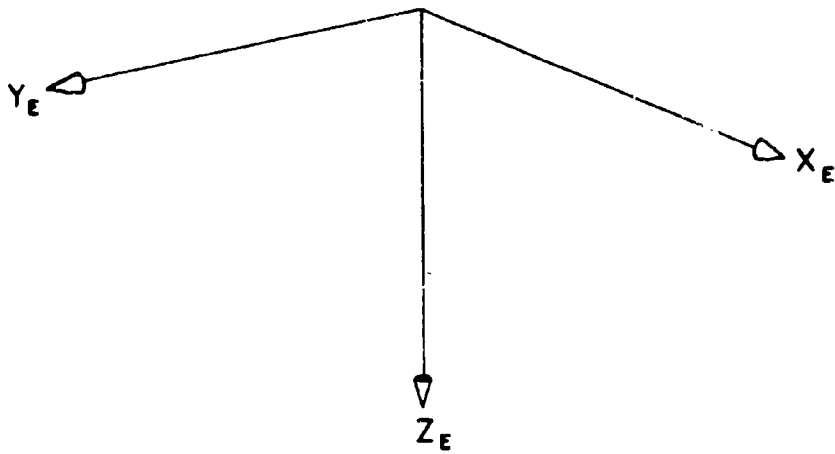


Figure 4. Euler Angles Between Body Axis and Inertial Axis

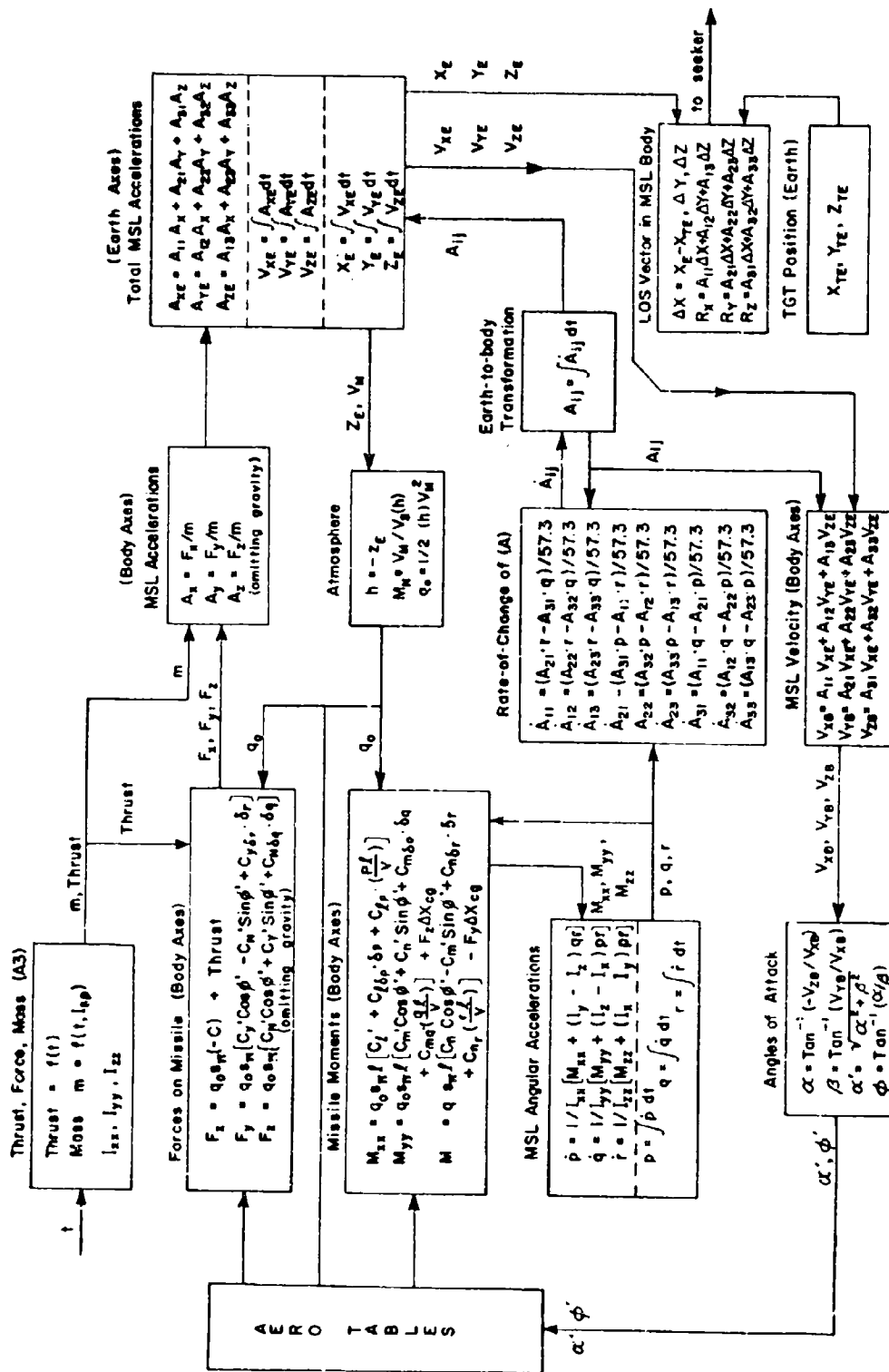


FIGURE 5. SIX-DEGREE-OF-FREEDOM EQUATIONS OF MOTION

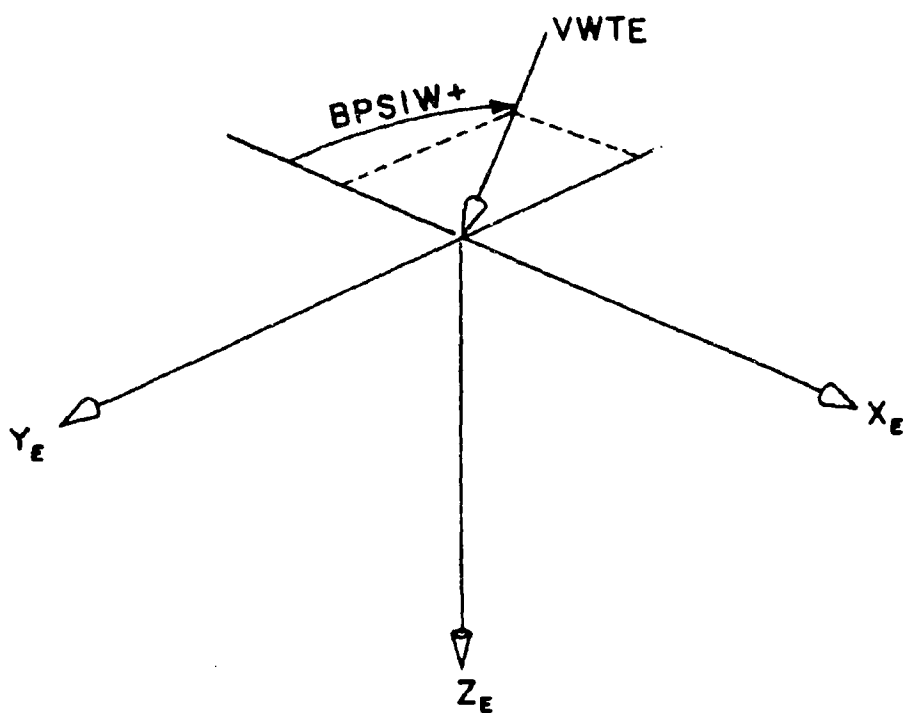


Figure 6. Coordinate System Associated with Wind and Gust Module

2.5 Subroutine C10

This subroutine determines the ground plane coordinates of that point in the area illuminated by the laser beam which the missile seeker physics causes the autopilot to consider the designated target. This distinction is necessary since some seekers are hotspot trackers while others are centroidal trackers. The procedure used to develop this apparent target is accomplished first through the designation of the coordinates of the illuminator (XIL, HILL) which may either be on the forward air controller or on the launch aircraft. The maximum errors generated on the ground are considered to be made up of three parts: the maximum boresight error (BORE), the maximum pointing error (WAND), and the maximum deviation of the hotspot from the resulting beam centroid, which is designated as (RADIUS). Each of these variables is considered a random variable with a uniform distribution. The resulting random variables generated are, respectively, BOREF, WANDF, and SPWID. The variables are considered to vary independently in the XE and YE direction and are then appropriately summed in order to determine the apparent target location. The coordinates of this point are designated as the variables ZLASR and YLASR. The location in earth coordinates may be found by equating ZLASR to XE and YLASR to YE and setting ZE equal to zero.

2.6 Subroutine QUADET

Subroutine QUADET is called by S1 for the determination of the signal generated to the autopilot by the quadrant detector (Figure 7). The quadrant detector is oriented such that the dead zone is in the same direction as the fins, assuming the missile flies in the X configuration. The subroutine determines the current size of the circular image through the assumption that the image size increases inversely proportional to the range of the missile from the laser spot. RT1 is the variable designating the ratio of the size of the current spot to its size at infinity. The laser image on the detector is assumed to be circular. In order to determine the portion of the area of each quadrant covered by the laser image, the area of the detector is subdivided into LT segments. (In the current program LT is set equal to 16.) In order to effect a dead zone, an area round the axis of the coordinate system equal to half the segment width is not included in the area of the image which cover these segments. If a portion of the laser image falls off the assumed circular detector's surface, it is not considered. The variable DETRID is half the instantaneous field of view of the detector in degrees. DEFICS is half the

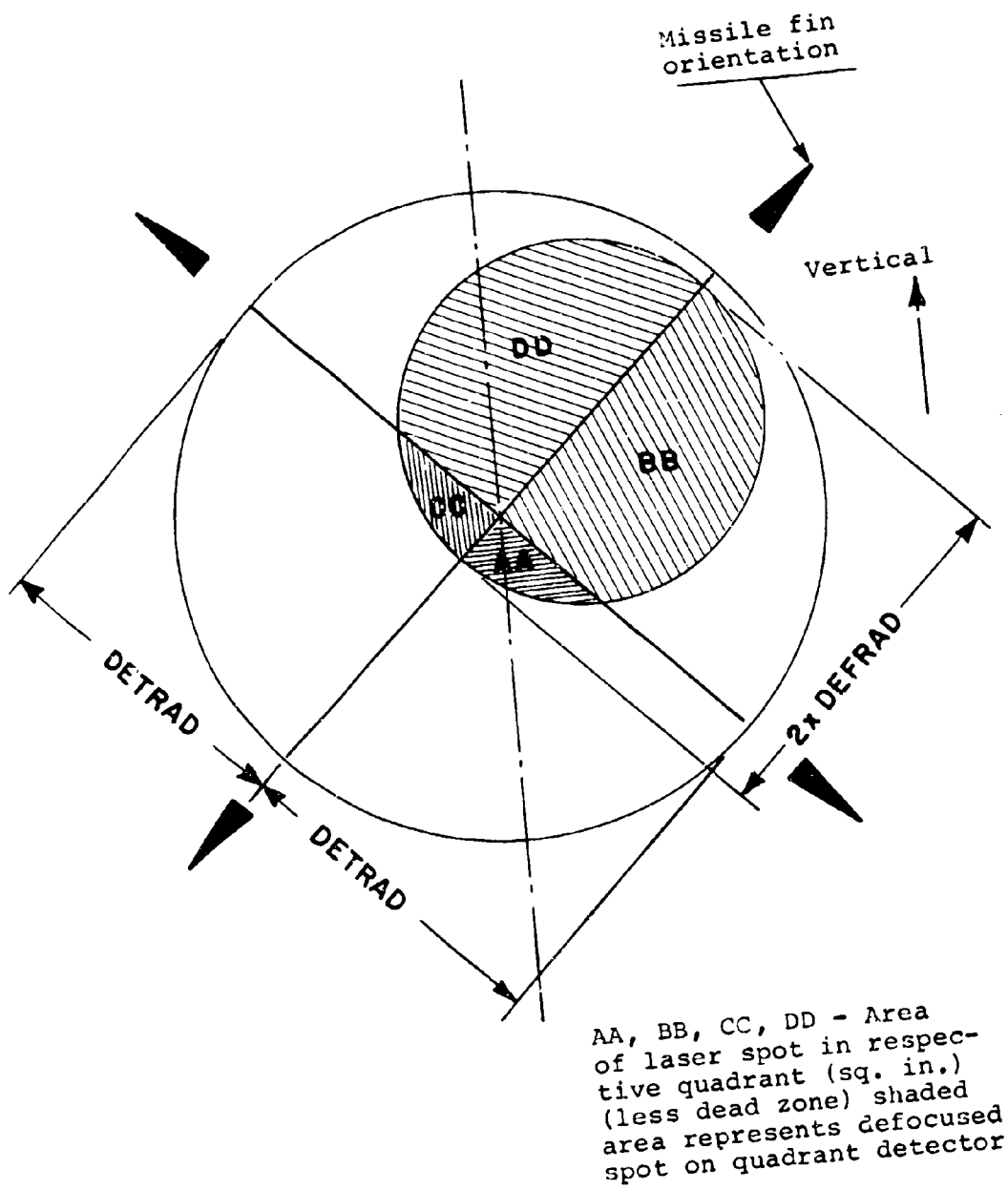


Figure 7. Quadrant Detector Geometry

instantaneous field of view intercepted by the image of the laser spot on the detector. The subroutine will determine if the following blind range and breaklock criteria are met and print this information on the line printer. The breaklock criteria is met if there is no portion of the laser image on any of the four quadrants. The blind range criteria is met if the image of the laser spot on the detector exceeds 90 percent of the total area of the detector.

2.7 Subroutine S1 (Module)

The purpose of the S1 Module is to simulate the response of several types of seeker models and to generate the commands which are transmitted to the autopilot.

The subroutine will simulate the seeker response to either a continuous information source or a sampled data source. This is accomplished by setting the variable OPTKR either to zero or one, respectively. If operating from a continuous information source, the seeker is assumed by the module to be a proportional seeker. In the sampled data mode the seeker can be programmed as either a proportional or a bang-bang seeker by the choice of the magnitude of the variable DEFOCS. If this variable is chosen so that it is equivalent to DELF [detector radius/LT (in current program)], the seeker will simulate a bang-bang laser seeker; whereas, if this variable is chosen so that it is larger than DELF, it will produce a proportional laser processor.

In the sample data mode the seeker will simulate either a pursuit or a proportional navigation system by setting the variable CAGE equal to zero or one, respectively. In the continuous information mode, corresponding changes in the guidance law will occur. In either mode of operation the PLG option may be implemented. This is done by removing the C from the two cards following the PLG OPTION card.

The mode of operation of the subroutine in either mode is to initially determine the true location of the target in the gimbal axis coordinate system (RXG, RYG, and RZG) and then determine the angles the lines of sight make with the RYG, RXG plane and the RZG, RXG planes (BEPSY and BEPSZ, respectively, shown in Figure 8 and Figure 9).

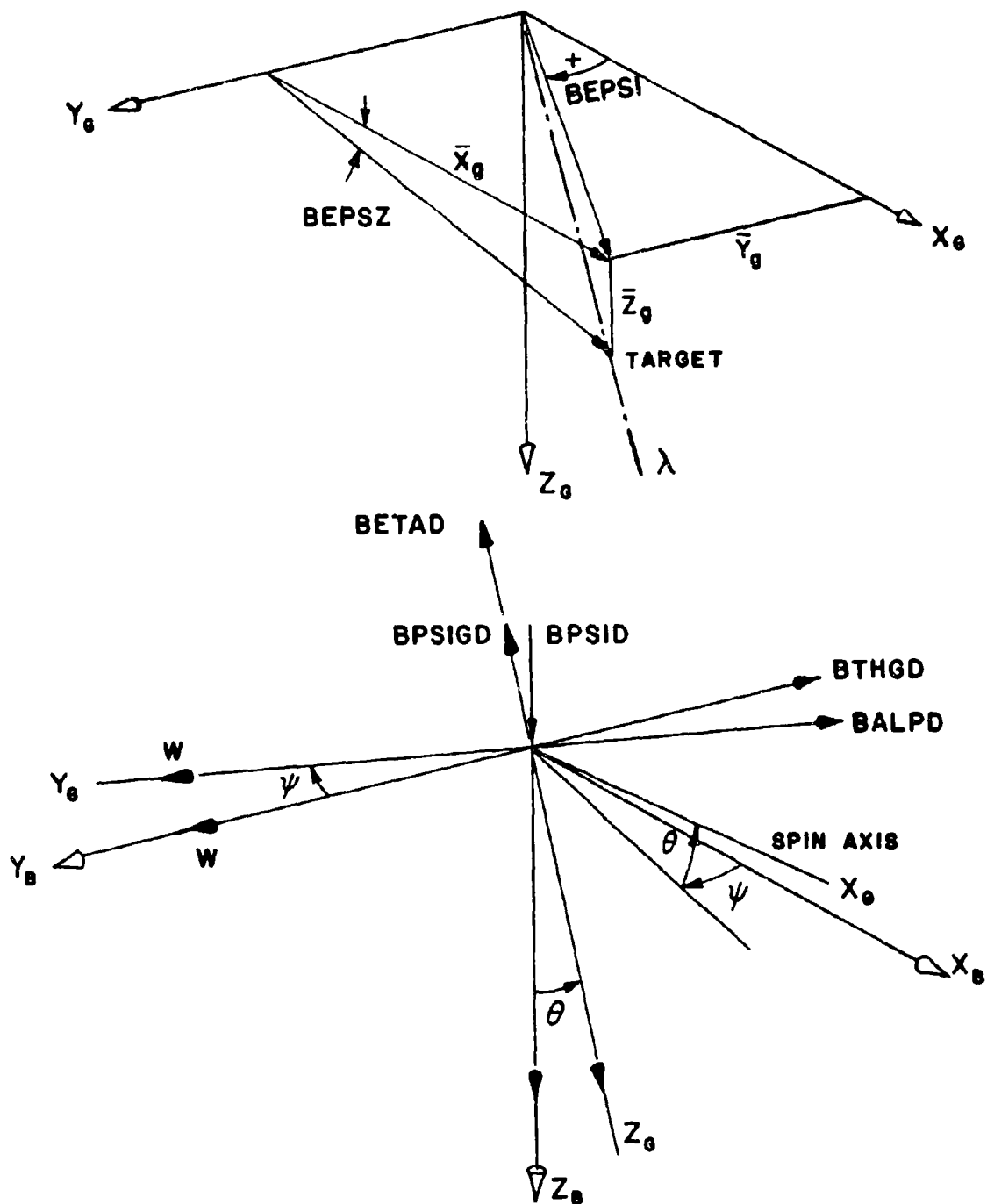
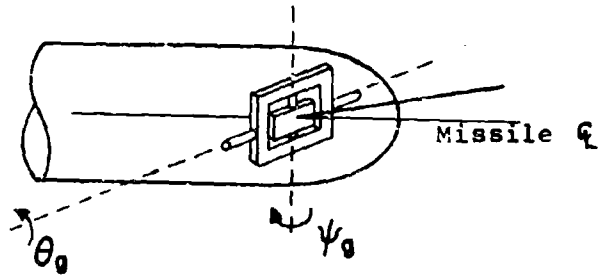


Figure 8. Coordinate Relations Between Body and Gimbal Axis System and the Line of Sight



Gyroscope Platform Gimbal Angles

θ_g outer gimbal - pitch

ψ_g inner gimbal - yaw

Figure 9. Schematic Diagram of Platform Gimbal Angles

The rate of pulse loss is determined by the value, between zero and one, initially given the variable VLAZRP. This is done by comparing a uniformly distributed random variable [C(103)] whose range is also between zero and one with VLAZRP. If it is greater, it is assumed that the information in the pulse is lost. If pulse loss has not occurred, then the apparent location of the target is determined in the gimbal axis coordinate system which has resulted from boresight errors, wander, etc. Subroutine QUADDET is then called to determine the output of the quadrant detector. This output is used to generate the required gimbal rate and autopilot commands. If pulse loss has occurred, previously generated commands (e.g., gimbal rate, autopilot signals) are maintained.

In addition, Appendix I shows the mechanics of the coordinate transformation from the body axis to the gimbal axis system for easy reference.

2.8 C1 Autopilot Module

The following high and low frequency autopilot block diagrams are suitable representations for an autopilot that would prove to be consistent with either a proportional or bang-bang seeker. The block diagrams for each of those autopilots are given in Figures 10, 11, 12, and 13. These systems correspond to those mechanized in the program listing found in Section V for the low frequency autopilot and in Appendix III for the high frequency autopilot.

2.9 C4 - Actuator Module

The actuator module simulates the action of the actuator up to a third order system, as shown in Figure 14, which corresponds to a high frequency actuator. Under these conditions it is capable of simulating the dynamics of either a torque balance system whose block diagram is shown in Figure 15, or that of the position loop-controlled actuator shown in Figure 16. It will also simulate the dynamics of an actuator whose transfer function has been determined from hardware test data up to the third order.

The transfer function, given in general form as expressed in this module, is shown below:

$$\frac{\delta}{\delta_c} = \frac{K}{A1 * S^3 + A2 * S^2 + A3 * S + A4}$$

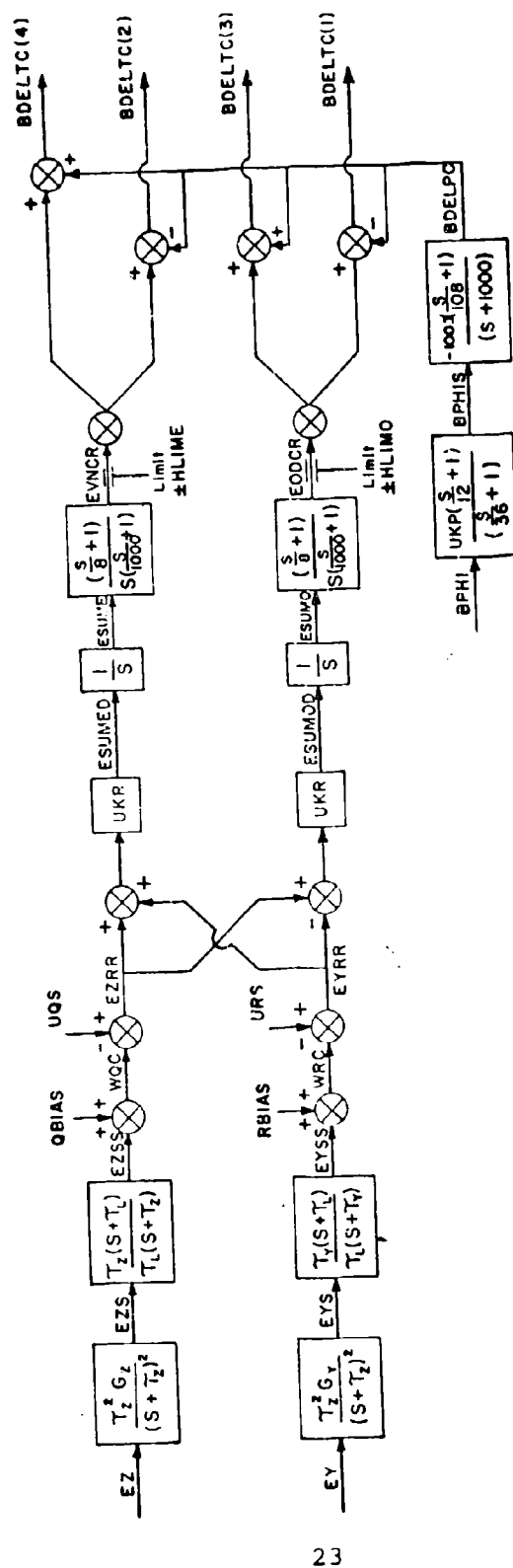


FIGURE 10. AUTOPILOT
HIGH FREQUENCY MODEL

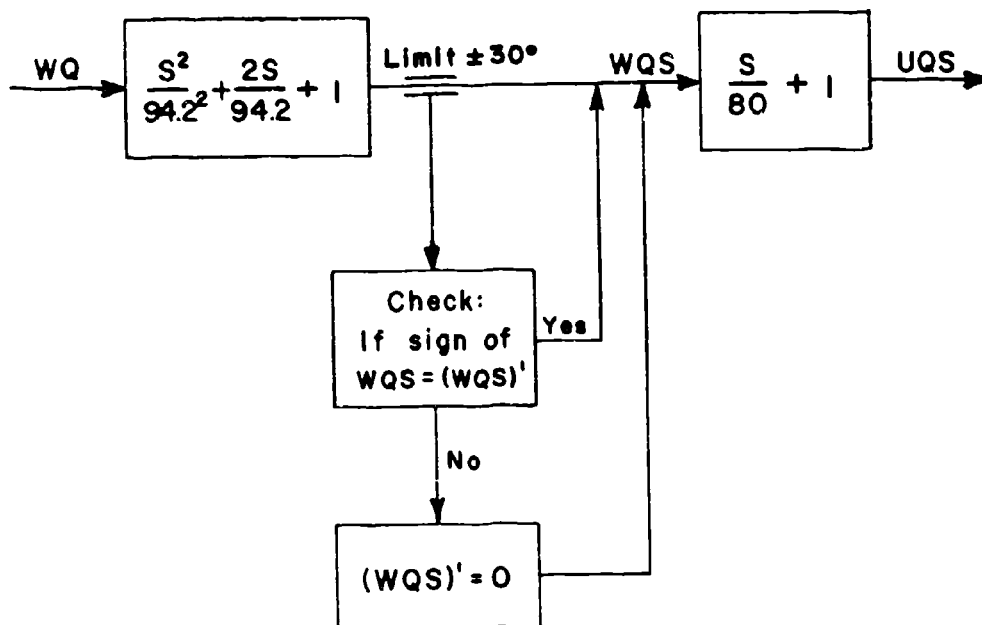


Figure 11. Pitch Rate Gyro

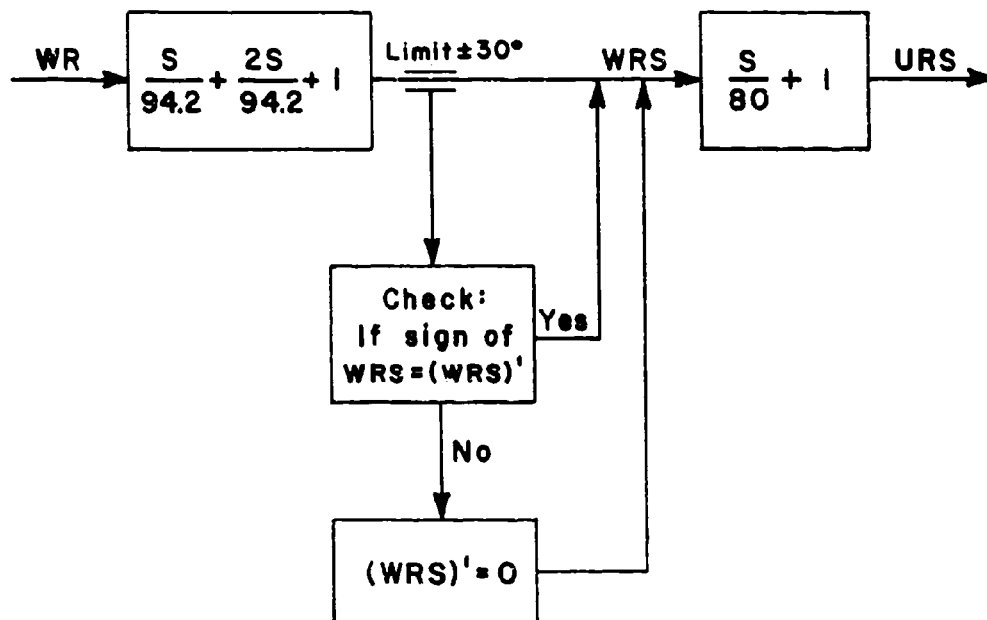


Figure 12. Yaw Rate Gyro

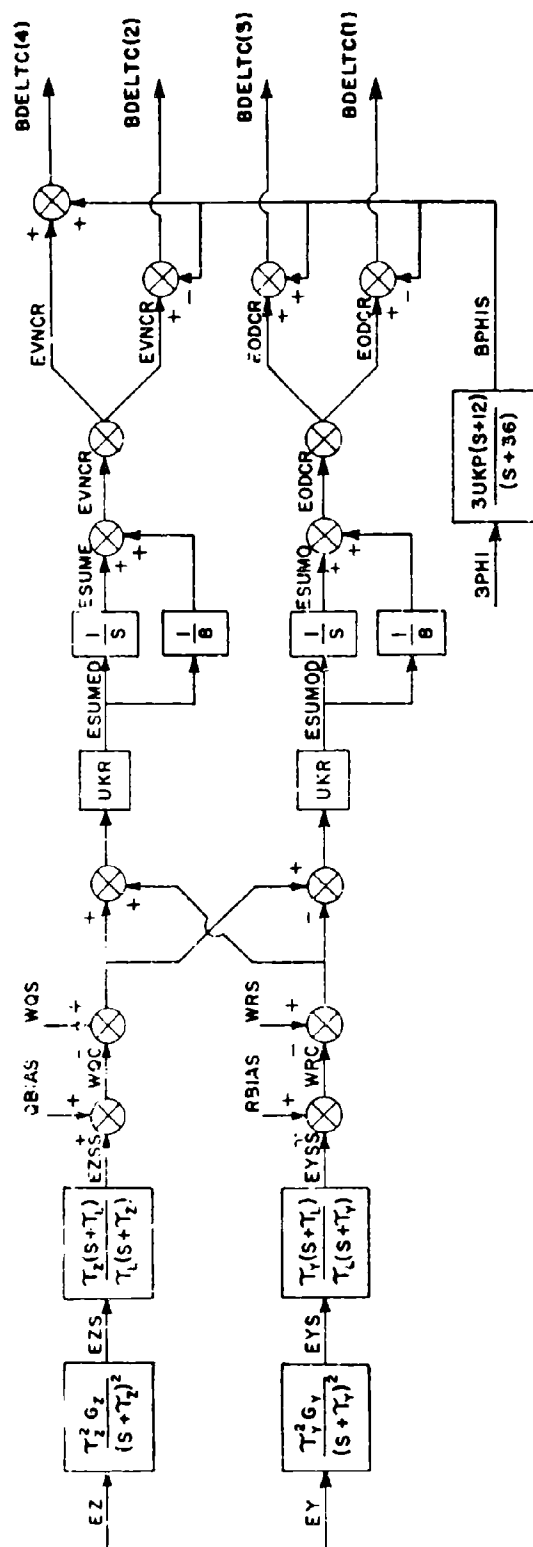


FIGURE 13. AUTOPILOT

LOW FREQUENCY MODEL

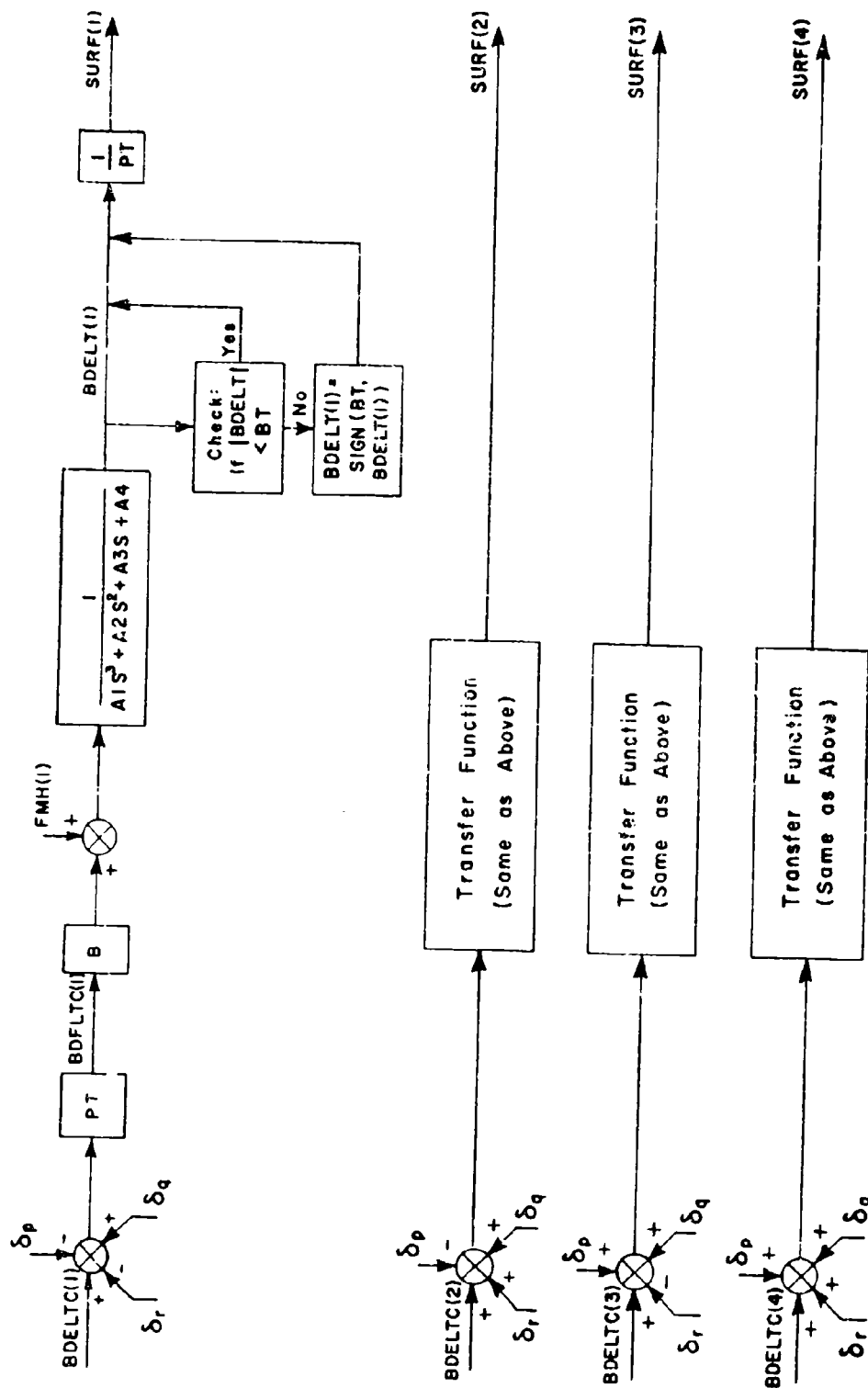


FIGURE 14. HIGH FREQUENCY ACTUATOR

$$\frac{\delta}{\delta_c} = \frac{1}{\left(\frac{s}{16.3} + 1\right) \left(\frac{s^2}{180^2} + \frac{95}{180}s + 1\right)}$$

[worst case model from hardware test data]

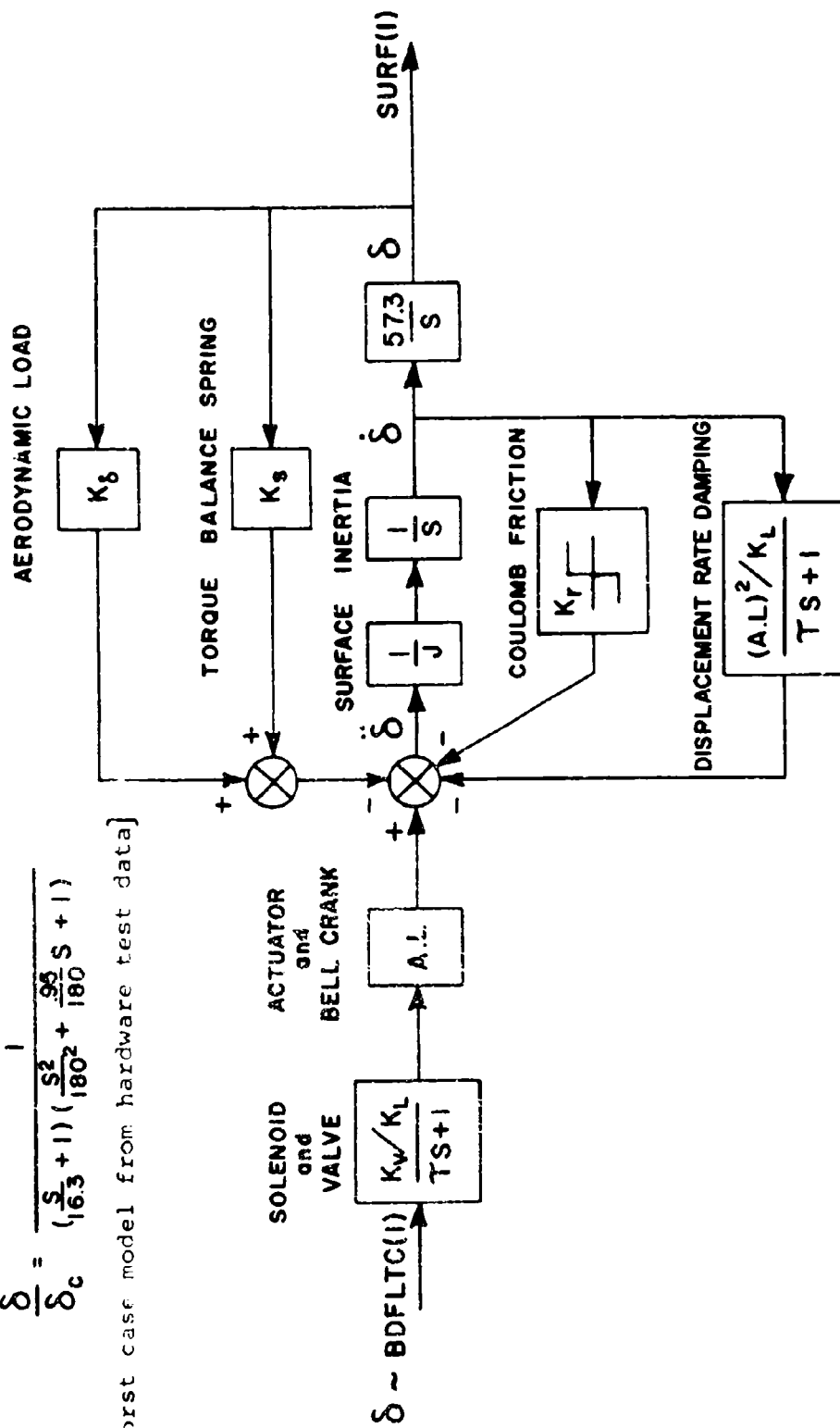


Figure 15. Actuator Torque Balance System

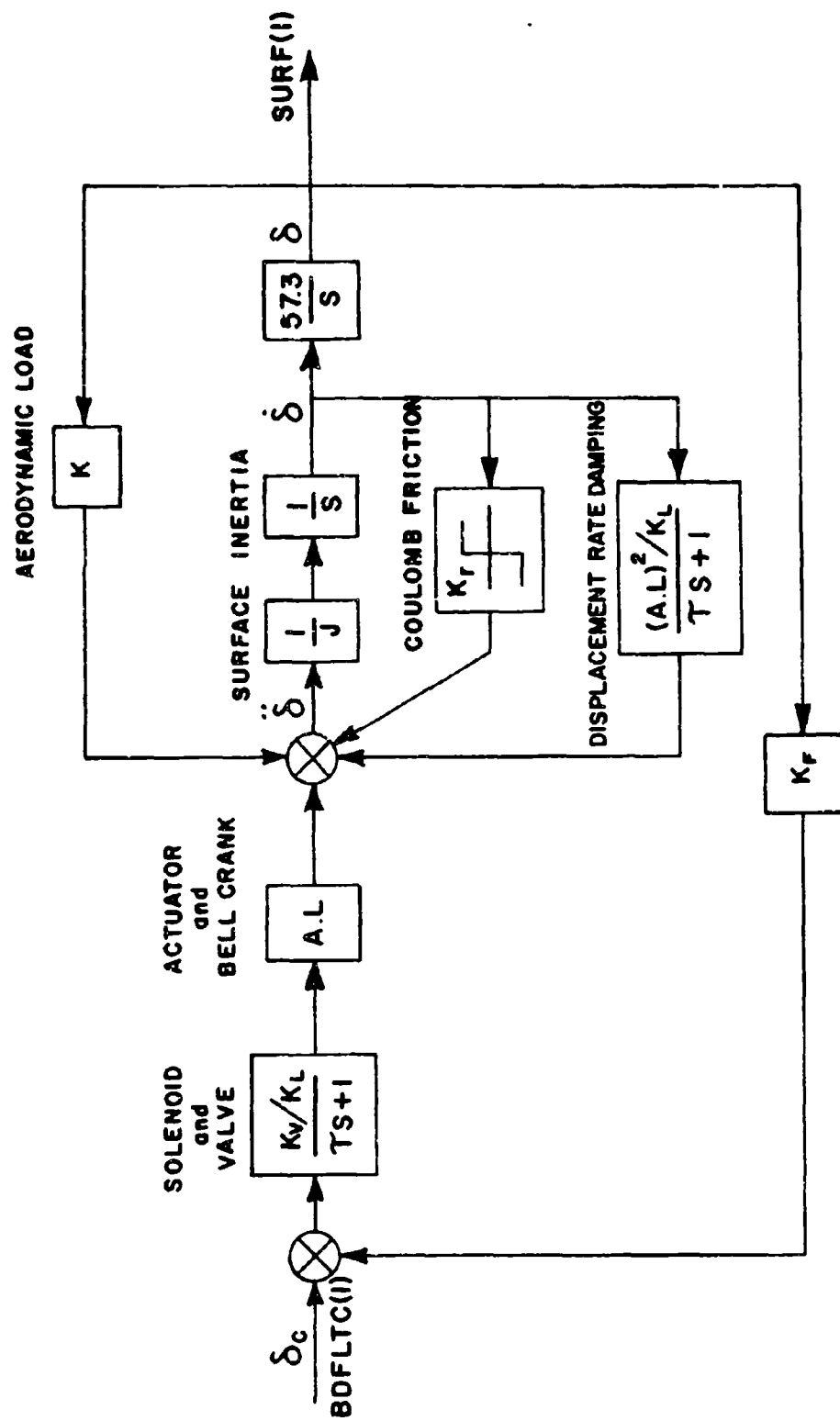


Figure 16. Actuator Position Loop Block Diagram

The transfer function for either the position feedback system or the torque balance system can be brought into the following form:

$$\frac{\delta}{\delta_C} = \frac{K_A \Lambda^* \ell (57.3)}{K_L [S(K_L J S(\tau S+1) + K_R(\tau S+1) + (A \ell)^2 / K_L) + K_S K_\delta 57.3(\tau S+1)]}$$

and similarly for the position loop block diagram.

If either the torque balance system or the position loop control system is to be activated, then CKACT should be either set equal to one or zero, depending on whether the aerodynamic tables for FMH1, FMH2, FMH3, and FMH4 are included in the data tables. The variable BDMAX limits the maximum amplitude of the fin motion. The low frequency actuator equations are developed below.

Low Frequency Actuator

$$BDELT(1) = BDELT(1) - \delta p + \delta q - \delta r$$

$$BDELT(2) = BDELT(2) - \delta p + \delta q + \delta r$$

$$BDELT(3) = BDELT(3) + \delta p + \delta q - \delta r$$

$$BDELT(4) = BDELT(4) + \delta p + \delta q + \delta r$$

$$\delta_1 = BDELT(1)$$

$$\delta_2 = BDELT(2)$$

$$\delta_3 = BDELT(3)$$

$$\delta_4 = BDELT(4)$$

where

$$\delta_p = DELTPB$$

$$\delta_q = DELTQB$$

$$\delta_r = DELTRB$$

The mechanization of these equations may be found in the Program Listing (Section V) for the low frequency actuator. The high frequency actuator program listing may be found in Appendix II.

2.9.1 Fin Deflection

A positive pitch rate (motion up) is obtained with a negative δq , where

$$\delta q = \frac{\delta_1 + \delta_2 + \delta_3 + \delta_4}{4}.$$

A positive roll rate (motion clockwise about the X body axis) is obtained with a positive δp , where

$$\delta p = \frac{(\delta_3 - \delta_2 + \delta_4 - \delta_1)}{4}.$$

A positive yaw rate (motion clockwise about the Z body axis) is obtained with a negative δr .

A positive surface deflection is defined as a trailing edge down. The surfaces are labeled by looking at the missile tail-on, with δ_1 being the upper right surface, δ_2 the lower right surface, δ_3 the lower left surface, and δ_4 the upper left surface, as shown in Figure 17.

It is assumed that the surface effectiveness will be given in terms of δq , δp , and δr as a function of α' and ϕ' . These terms will be considered as a part of the aerodynamic coefficients given in the primed axis system.

2.10 A3 - Engine Module

As a result of various sources of error occurring in the manufacture and assembly of a solid propellant motor, the thrust alignment is not perfect. The coordinate system used in determining the misalignment the user wishes to simulate is shown in Figure 18.

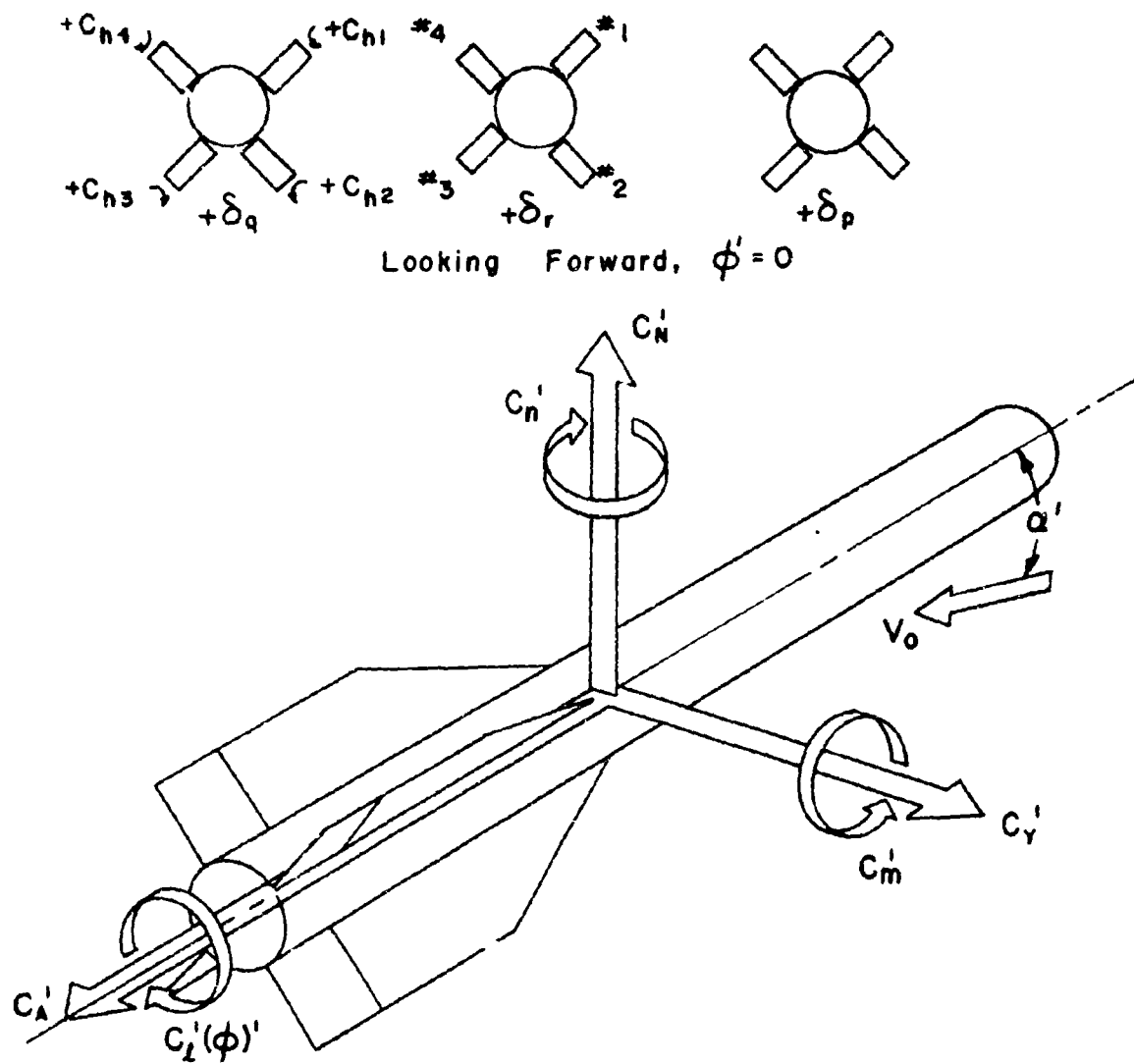
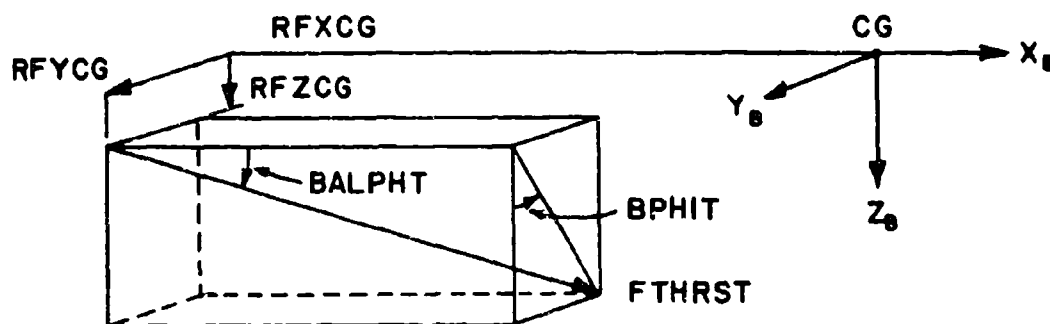


Figure 17. Fin Sign Conventions



$RFXCG$ - X Component of thrust vector with respect to body axis in the X direction

$RFYCG$ - Y Component of thrust vector with respect to body axis in the Y direction

$RFZCG$ - Z Component of thrust vector with respect to body axis in the Z direction

$FTHRST$ - Missile Thrust

Figure 18. Offset Thrust Vector Coordinate System

SECTION III

VARIABLE LOCATIONS

3.1 Variable Names, Block Locations, and Definitions

Since the proper use of this program requires that the definition of upwards of five hundred singly dimensioned variables as well as many multidimensioned be made, it is clear that some order must be maintained in the allocation of storage or serious programming difficulties could arise. Therefore, blocks of common location have been allocated to specific subroutines as shown in Table III. This procedure should be continued in the event it is necessary to add variables as a result of program modifications.

Of the large number of variables actually listed by the program, only two hundred and fifty appear to be significant in the preparation of the input or of an aid in understanding the output. Therefore, it was felt that they should be separately defined. This is done in Table IV. It should be noted that the units of the variables listed in that table should be considered to be in feet, seconds, pounds, or degrees unless otherwise specified.

TABLE III. BLANK COMMON ASSIGNMENTS

Array Index C ()	Module Name	Module Description
0 - 50	C10	Unsteady Illuminator
50 - 102	G2	Steady Winds, Variable Winds
200 - 299	G3	Air Data
350 - 399	G5	Coordinate Conversion
400 - 499	S1, S1I	Seeker - Platform
800 - 899	C1, C1I	Autopilot
1100 - 1149	C4, C4I	Actuators
1200 - 1299	A1	Aero Table Look-Up
1300 - 1399	A2	Forces and Moments
1400 - 1499	A3, A3I	Engine
1600 - 1699	D1, D1I	Translational Dynamics
1700 - 1799	D2, D2I	Rotational Dynamics

Note: Locations 1950 - 4310 are reserved for Executive Subroutines, Initial Conditions, and Input-Output Instructions.

TABLE IV. COMMON LOCATION, VARIABLE NAME, AND DEFINITION

Common Location	Variable Name	Definition
C(1)	BORE	Maximum boresight error
C(2)	WAND	Maximum pointing error
C(3)	RADIUS	Maximum deviation of hotspot from beam centroid
C(4)	HILL	Height of illuminator
C(5)	RILL	Ground range of illuminator
C(6)	AISPOT	{ 0. - Centroid tracker 1. - Hotspot tracker
C(7)	AILL	{ 0. - Stationary illuminator* 1. - Moving illuminator
C(8)	SPOTMO	{ 0. - No spot motion 1. - Spot motion
C(9)	XSPOT	X - Coordinate of centroid or hotspot
C(10)	YSPOT	Y - Coordinate of centroid or hotspot
C(11)	AIFAC	{ 0. - Tracker on ground or on launch aircraft 1. - Tracker on separate aircraft
C(12)	VILM	Velocity of illuminator, Mach number
C(18)	XILL	Ground range in X direction of illuminator

*Must give HILL, XILL for input

TABLE IV (Continued)

Common Location	Variable Name	Definition
C(51)	BPSIW	Mean angle of wind velocity vector
C(52)	VWTE	Mean wind velocity
C(53)	RHW	Altitude above which all the winds are zero
C(55)	SW	Standard deviation from mean angle of wind velocity vector BPSIW
C(56)	RWINC	Shear layer of wind. Depth of wind layers at which wind velocity and angle remain constant.
C(58)	SW1	Standard deviation from mean wind velocity VWTE
C(100)	VWXE	Wind velocity (X component with reference to the earth-fixed coordinate system)
C(101)	VWYE	Wind velocity (Y component with reference to the earth-fixed coordinate system)
C(102)	VWZE	Wind velocity (Z component with reference to the earth-fixed coordinate system)
C(203)	PDYNMC	Dynamic pressure
C(204)	VMACH	Mach number
C(205)	DRHO	Air density
C(206)	VSOUND	Speed of sound
C(207)	VAIRSP	Missile velocity with respect to air mass in earth axes

TABLE IV (Continued)

Common Location	Variable Name	Definition
C(208)	RHZRO	Initial altitude of the missile
C(209)	RI	Present altitude of the missile
C(350)	BTHT	Pitch angle in degrees - θ
C(351)	BPSI	Roll angle in degrees - ψ
C(352)	BPHI	Yaw angle in degrees - ϕ
C(356)	VTOTE	Total missile velocity
C(357)	BGAMH	Horizontal proportional navigation angle (degrees)
C(358)	BGAMV	Vertical proportional navigation angle (degrees)
C(367)	BALPHA	Vertical component of angle of attack
C(368)	BALPHY	Horizontal component of angle of attack
C(369)	BALPHP	$(\alpha' = \sqrt{BALPHA^2 + BALPHY^2})$ total angle of attack
C(370)	BPHIP	ϕ' orientation of wind vector in roll axis
C(371)	RANGE	Range
C(372)	RXBA	Range (X component in body coordinate system)
C(373)	RYBA	Range (Y component in body coordinate system)
C(374)	RZBA	Range (Z component in body coordinate system)

TABLE IV (Continued)

Common Location	Variable Name	Definition
C(377)	BALPD	$d(\text{BALPHA})/dt$
C(378)	BALYD	$d(\text{BALPHY})/dt$
C(379)	BALPPD	$d(\text{BALPHP})/dt$
C(380)	RANGO	The distance of missile from the launch point
C(403)	EZ	Seeker output to autopilot (pitch)
C(407)	EY	Seeker output to autopilot (yaw)
C(427)	BHTG	Platform position (θ_g)
C(431)	BPSIG	Platform position yaw gimbal angle (ψ_g)
C(432)	RXG	Range X in gimbal axes
C(433)	RYG	Range Y in gimbal axes
C(434)	RZG	Range Z in gimbal axes
C(435)	BEPSZ	Angular position of the line of sight in gimbal axes (see Figure 8)
C(435)	BEPSY	
C(437)	WZ	Missile body rate W_z
C(438)	WY	Missile body rate W_y
C(441)	SZGBIS	Pitch gimbal torque bias (deg/sec)
C(442)	SYGBIS	Yaw gimbal torque bias (deg/sec)
C(443)	OPTKR	Optics routine
C(444)	OPTBKL	Optical breaklock
C(445)	UT	Time at which next pulse expected
C(446)	CDT	Pulse rate, sampling period

TABLE IV (Continued)

Common Location	Variable Name	Definition
C(450)	CSW	Acquisition gain seeker constant
C(451)	CAGE	[(>0) uncaged gimbals] [(<0) remain in caged position]
C(452)	QBREAK	(Breaklock has occurred due to loss of signal) (automatically parameterized)
C(453)	RENLOK	Range at breaklock (maximum range at which lock-on can take place)
C(454)	BEDGE	Half the field of view
C(455)	WEPSMX	Breaklock drift rate
C(456)	CKSKR	Seeker gain
C(457)	CFOSTP	Pitch to yaw friction coupling
C(458)	CROSPT	Yaw to pitch friction coupling
C(460)	GUIDE	(=1 missile guidance system in effect) (=0 missile guidance system not in effect)
C(461)	SAMP	Preprogrammed guidance trajectory (cutoff check automatically parameterized) (0 - missile uses preprogrammed flight path) (1 - missile uses preprogrammed flight path until seeker acquires target)
C(464)	CGATVS	Vertical trajectory programming constant
C(465)	CGATHS	Horizontal trajectory programming constant
C(466)	ZLAZR	Location of laser spot on target in X direction due to ground or air-borne FAC

TABLE IV (Continued)

Common Location	Variable Name	Definition
C(467)	YLAZR	Location of laser spot on target in Y direction due to ground or airborne FAC
C(468)	DEFOCS	Half angle in degrees of angle intercepted by image of laser spot on detector surface
C(469)	DETRAD	Half angle in degrees of angle intercepted by quadrant detector
C(472)	CKSK1	Seeker driver constant
C(473)	VLAZRP	Used in pulse loss calculation
C(850)	HLIMO	Limit on δ_c from pitch and yaw plane (deg) (fins 1 and 3)
C(851)	HLIME	Limit on δ_c from pitch and yaw plane (deg) (fins 2 and 4)
C(852)	QBIAS	Pitch body rate bias (deg/sec) (used as "g" bias)
C(853)	RBIAS	Yaw body rate bias (deg/sec)
C(855)	GZ	Navigation ratio for pitch plane
C(856)	GY	Navigation ratio for yaw plane
C(863)	TAUZ	Pitch guidance lag filter (rad/sec)
C(864)	TAUY	Yaw guidance lag filter (rad/sec)
C(865)	TDY1	Rate loop gain switch 1 (sec)
C(866)	TDY2	Rate loop gain switch 2 (sec)
C(877)	TAUL	Guidance lead filter (rad/sec)
C(888)	CKSK2	Seeker gain constant

TABLE IV (Continued)

Common Location	Variable Name	Definition
C(1117)	BSURF 1	δ_1
C(1118)	BSURF 2	δ_2
C(1119)	BSURF 3	δ_3
C(1120)	BSURF 4	δ_4
		} fin deflection output for fins 1, 2, 3, 4
C(1160)	DELTPB	$\delta_p = (-\delta_1 - \delta_2 + \delta_3 + \delta_4)/4$
C(1161)	DELTOB	$\delta_q = (\delta_1 + \delta_2 + \delta_3 + \delta_4)/4$
C(1162)	DELTRB	$\delta_r = (-\delta_1 - \delta_2 + \delta_3 + \delta_4)/4$
C(1260)	CXERR	Drag coefficient error
C(1261)	CZERR	Normal force (C'_z) coefficient error
C(1262)	CYERR	Side force (C'_y) coefficient error
C(1263)	CLERR	Roll moment (C'_L) coefficient error
C(1264)	CMERR	Pitch moment (C'_M) coefficient error
C(1265)	CNERR	Yaw moment (C'_N) coefficient error
C(1300)	FXBA	The X component of aero force in body coordinate system
C(1301)	FYBA	The Y component of aero force in body coordinate system
C(1302)	FZBA	The Z component of aero force in body coordinate system
C(1303)	FMXBA	The X component of aero moment in body coordinate system
C(1304)	FMYBA	The Y component of aero moment in body coordinate system

TABLE IV (Continued)

Common Location	Variable Name	Definition
C(1305)	FMZBA	The Z component of aero moment in body coordinate system
C(1306)	RFAREA	Missile reference area (ft ²)
C(1307)	RFLGTH	Missile reference length (ft)
C(1308)	RDELCG	Center of gravity shift (ft)
C(1309)	FMH1	Hinge moments
C(1310)	FMH2	
C(1311)	FMH3	
C(1312)	FMH4	
C(1313)	RFXCG	Thrust vector displacements (ft)
C(1314)	RFYCG	
C(1315)	RFZCG	
C(1316)	RLUG	Distance between lugs (ft)
C(1317)	RAIL	Rail length (ft) (between rear of front lug and end of rail)
C(1320)	FMXTH	X component of moment caused by thrust misalignments
C(1321)	FMYTH	Y component of moment caused by thrust misalignments
C(1322)	FMZTH	Z component of moment caused by thrust misalignments
C(1323)	FMXLUG	X component of moment due to lugs
C(1324)	FMYLUG	Y component of moment due to lugs
C(1325)	FMZLUG	Z component of moment due to lugs

TABLE IV (Continued)

Common Location	Variable Name	Definition
C(1401)	BALPHT	The angles as indicated in Figure 18
C(1402)	LPHT	
C(1403)	QNALGN	(> 0; include thrust misalignment angles)
C(1404)	FCPTH	Fractional increase in total thrust
C(1405)	QBURN	Parameterized by program
C(1410)	FTHRST	Missile thrust
C(1411)	FTHX	X component of missile thrust
C(1412)	FTHY	Y component of missile thrust
C(1413)	FTHZ	Z component of missile thrust
C(1414)	CISP	Specific impulse (lb/sec)
C(1415)	DWT	Total missile plus propellant wt (lb) initial
C(1416)	DWP	Propellant weight (lb)
C(1417)	RDCGO	Initial value of c.g. shift (ft)
C(1418)	RDCGF	Burnout value of c.g. shift (ft)
C(1419)	FMIYO	Initial value of moment of inertia about the roll axis (slugs ft)
C(1420)	FMIYO	Initial value of moment of inertia about the pitch axis (slugs ft)
C(1421)	FLCGA	Distance between launch c.g. and rear lug (ft)
C(1422)	RLCG	Present position of c.g. of missile

TABLE IV (Continued)

Common Location	Variable Name	Definition
C(1496)	UIMPD	Thrust of the motor
C(1499)	UIMP	Impulse of the motor
C(1603)	VXE	X,Y,Z coordinates of missile velocity with respect to the earth fixed coordinate system
C(1607)	VYE	
C(1611)	VZE	
C(1615)	RXE	X,Y,Z coordinates of missile c.g. with respect to the earth fixed coordinate system
C(1619)	RYE	
C(1623)	RZE	
C(1624)	AXBA	X component of acceleration in body coordinate axis
C(1625)	AYBA	Y component of acceleration in body coordinate axis
C(1626)	AZBA	Z component of acceleration in body coordinate axis
C(1627)	AGRAV	Gravitational constant
C(1628)	DMASS	Current mass
C(1629)	ATHRST	Target thrust
C(1630)	ATURNT	Maximum transverse acceleration of target in terms of g
C(1632)	VDELX	Relative velocity of missile to target in X direction
C(1633)	VDELY	Relative velocity of missile to target in Y direction
C(1634)	VDELZ	Relative velocity of missile to target in Z direction

TABLE IV (Continued)

Common Location	Variable Name	Definition
C(1635)	RDELX	Range difference between target and missile in the X direction of the earth fixed coordinate system
C(1636)	RDELY	Range difference between target and missile in the Y direction of the earth fixed coordinate system
C(1637)	RDELZ	Range difference between target and missile in the Z direction of the earth fixed coordinate system
C(1644)	ATARG	Acceleration of the target
C(1647)	VTARG	Velocity of the target
C(1648)	RTXED	The X component of the velocity of the target in the earth fixed coordinate system
C(1651)	RTXE	The X coordinate of the position of the target in the earth fixed coordinate system
C(1652)	RTYED	The Y component of the velocity of the target in the earth fixed coordinate system
C(1655)	RTYE	The Y coordinate of the position of the target in the earth fixed coordinate system
C(1656)	RTZED	The Z component of the velocity of the target in the earth fixed coordinate system
C(1659)	RTZE	The Z coordinate of the position of the target in the earth fixed coordinate system
C(1660)	VTXE	The X component of the velocity of the target in the earth fixed coordinate system

TABLE IV (Continued)

Common Location	Variable Name	Definition
C(1661)	VTYE	The Y component of the velocity of the target in the earth fixed coordinate system
C(1662)	VTZE	The Z component of the velocity of the target in the earth fixed coordinate system
C(1663)	VDXB	The X component of the acceleration of missile in the body coordinate axes
C(1664)	VDYB	The Y component of the acceleration of missile in the body coordinate axes
C(1665)	VDZB	The Z component of the acceleration of missile in the body coordinate axes
C(1666)	BDIVE	Initial pitch orientation of the aircraft (missile assumed oriented parallel to aircraft)
C(1667)	RSLANT	Initial slant range
C(1668)	RXO	The X component of the original launch point of the missile in the earth fixed coordinate system
C(1669)	RYO	The Y component of the original launch point of the missile in the earth fixed coordinate system
C(1670)	RZO	The Z component of the original launch point of the missile in the earth fixed coordinate system
C(1672)	BPSITD	The angular rate of turn of target
C(1675)	BPSIT	The total angle through which the target has turned in degrees

TABLE IV (Continued)

Common Location	Variable Name	Definition
C(1676)	ANGX	The X component of the acceleration of the missile in terms of g with respect to body axes
C(1677)	ANGY	The Y component of the acceleration of the missile in terms of g with respect to body axes
C(1678)	ANGZ	The Z component of the acceleration of the missile in terms of g with respect to body axes
C(1700)	CFA11D	Derivative of CFA11
C(1703)	CFA11	$\cos\psi \cos\theta$
C(1704)	CFA12D	Derivative of CFA12
C(1707)	CFA12	$\sin\psi \cos\theta$
C(1708)	CFA13D	Derivative of CFA13
C(1711)	CFA13	$-\sin\theta$
C(1712)	CFA21D	Derivative of CFA21
C(1715)	CFA21	$\sin\psi \cos\theta + \cos\psi \sin\theta \sin\phi$
C(1716)	CFA22D	Derivative of CFA22
C(1719)	CFA22	$\cos\psi \cos\phi + \sin\psi \sin\theta \sin\phi$
C(1720)	CFA23D	Derivative of CFA 23
C(1723)	CFA23	$\cos\theta \sin\phi$
C(1724)	CFA31D	Derivative of CFA31
C(1727)	CFA31	$\cos\psi \sin\theta \cos\phi + \sin\psi \sin\phi$
C(1728)	CFA32D	Derivative of CFA32

TABLE IV (Concluded)

Common Location	Variable Name	Definition
C(1731)	CFA32	$\sin\psi \sin\theta \cos\phi - \cos\psi \sin\phi$
C(1732)	CFA33D	Derivative of CFA33
C(1735)	CFA33	$\cos\theta \cos\phi$
C(1736)	WPD	$d(WP)/dt$
C(1739)	WP	Roll rate of missile
C(1740)	WQD	$d(WQ)/dt$
C(1743)	WQ	Pitch rate of missile
C(1744)	WRD	$d(WR)/dt$
C(1747)	WR	Yaw rate of missile
C(1748)	FMIX	Missile moments of inertia about the X,Y, and Z missile body axes in slug-feet ²
C(1749)	FMIY	
C(1750)	FMIZ	
C(1751)	CRAD	Conversion factor (from radians to degrees)
C(1752)	BPHIO	Initial roll angle of missile
C(1753)	BTHTO	Initial pitch angle of missile
C(1754)	BPSIO	Initial yaw angle of missile

3.2 Subroutine Call Sequence

The subroutine call sequence is determined by the order in which these subroutines are identified in the data card assembly. A data card is identified by the program as a subroutine call by the number 2 located in column 2. The identification number of the subroutine may be called as either [MODNO(NOMOD)] or [XMODNO(NOMOD)] by the program. This integer must be right adjusted to column 25 on the card. Table V shows the identification number and the subroutines called in the example problem. If other routines are required, they will be found listed with their identification numbers in subroutine AUXSUB.

3.3 State Variables

The state variables within this six-degree-of-freedom simulation program are defined in the initialization subroutines (modules). These variables are identified through the IPL table which also defines the location of the state variables. Only these variables are integrated by the integration routine AMRK. Other variables found in the program which are derivatives are not state variables by this definition. A listing of the sequence number, IPL numbers, and variable names are found in Table VI. The listing is for the program when it contains the high frequency autopilot and actuator.

In the event a location is defined as a state variable, the following convention must be observed:

C(J + 3) State variable

then

C(J) is the derivative of that state variable.

TABLE V. INITIALIZATION SUBROUTINE CALL SEQUENCE
 [(By Subroutine AUXI) (As defined by current program listing)]

NOMOD	MODNO (NOMOD) XMODNO (NOMOL)	SUBROUTINE CALL
1.	23	G2I
2.	24	G3I
3.	26	G5I
4.	28	S1I
5.	7	C1I
6.	10	C4I
7.	2	A1I
8.	4	A3I
9.	3	A2I
10.	17	D1I
11.	18	D2I

TABLE VI. STATE VARIABLES AND DERIVATIVES
NAMES AND LOCATION CODES

Sequence No.	N	IPL(N)	C(IPL(N))	Module Location
1.	1	424	BTHTGD	S1, S1I
2.		427	BTHTG	SURFACE AND PLAT- FORM
3.	2	428	BPSIGD	
4.		431	BPSIG	
5.	3	800	BPHISD	C1, C1I
6.		803	BPHIS	HIGH FREQUENCY AUTO- PILOT INITIALIZATION MODULE
7.	4	804	WQSDD	
8.		807	WQSP	
9.	5	808	WQSD	
10.		811	WQS	
11.	6	812	WRSDD	
12.		815	WRSP	
13.	7	816	WRSD	
14.		819	WRS	
15.	8	820	ESUMOD	
16.		823	ESUMO	
17.	9	824	ESUMED	
18.		827	ESUME	
19.	10	828	EZSDD	
20.		831	EZSP	
21.	11	832	EZSD	
22.		835	EZS	

TABLE VI (Continued)

Sequence No.	N	IPL(N)	C(IPL(N))	Module Location
23.	12	836	EYSDD	C1, C11 HIGH FREQUENCY AUTOPILOT INITIALIZATION MODULE
24.		839	EYSP	
25.	13	840	EYSD	
26.		843	EYS	
27.	14	880	EZSSD	
28.		883	EZSS	
29.	15	884	EYSSD	
30.		887	EYSS	
31.	16	1100	BDELTD(1)	C4, C41 HIGH FREQUENCY MODULE (ACTUATORS)
32.		1103	BDELT(1)	
33.	17	1104	BDELTD(2)	
34.		1107	BDELT(2)	
35.	18	1108	BDELTD(3)	
36.		1111	BDELT(3)	
37.	19	1112	BDELTD(4)	
38.		1115	BDELT(4)	
39.	20	1124	BDLTDD(1)	
40.		1127	BDELTP(1)	
41.	21	1128	BDLTDD(2)	
42.		1131	BDELTP(2)	
43.	22	1132	BDLTDD(3)	
44.		1135	BDELTP(3)	
45.	23	1136	BDLTDD(4)	
46.		1139	BDELTP(4)	

TABLE VI (Continued)

Sequence No.	N	IFL(N)	C(IFL(N))	Module Location
47.	24	1496	UIMPD	A3I
48.		1499	UIMP	ENGINE
49.	25	1600	VXED	D1, D11
50.		1603	VXE	TRANSLATIONAL DYNAMICS
51.	26	1604	VYED	
52.		1607	VYE	
53.	27	1612	RXED	
54.		1615	RXE	
55.	28	1616	RYED	
56.		1619	RYE	
57.	29	1620	RZED	
58.		1623	RZE	
59.	30	1640	VTARGD	
60.		1643	VTARG	
61.	31	1644	BPSITD	
62.		1647	BPSIT	
63.	32	1648	RTXED	
64.		1651	RTXE	
65.	33	1652	RTYED	
66.		1655	RTYE	
67.	34	1656	RTZED	
68.		1659	RTZE	

TABLE VI (Concluded)

Sequence No.	N	IPL(N)	C(IPL(N))	Module Location
69.	35	1700	CFAIID	D2, D2I
70.		1703	CFAII	ROTATIONAL DYNAMICS
71.	36	1704	CFA12D	
72.		1707	CFA12	
73.	37	1708	CFA13D	
74.		1711	CFA13	
75.	38	1712	CFA14D	
76.		1715	CFA14	
77.	39	1716	CFA22D	
78.		1719	CFA22	
79.	40	1720	CFA23D	
80.		1723	CFA23	
81.	41	1724	CFA310	
82.		1727	CFA31	
83.	42	1728	CFA32D	
84.		1731	CFA32	
85.	43	1732	CFA33D	
86.		1735	CFA33	
87.	44	1736	WPD	
88.		1739	WP	
89.	45	1740	WQD	
90.		1743	WQ	
91.	46	1744	WRD	
92.		1757	WR	

SECTION IV

INPUT REQUIREMENTS

4.1 Initial Conditions

In order to simplify the input data, the options which had existed in the original program have been eliminated. It is assumed by the program that all variables not initialized are automatically set equal to zero. Input data and initial conditions are entered into the program by entering a number 3 in column 2 of the data card which identifies the type of information. The name of the variable may be entered in column 3 to 20. Common location of the variable must be entered right adjusted in columns 21 to 25 and the numerical data in columns 31 to 45. Figure 19 shows the position of the data card in the completed program deck which is ready for submission, as well as the actual data card format.

In addition, since the seeker is generally assumed to lock on before launch, the gimbal angles are automatically initialized to this position. However, in the cases where gimbal angles must be chosen in any other position, the transformations and angular displacements between gimbal axes and body axes coordinate systems are given in Appendix I and Figure I-1, respectively.

Initial position and velocity can only be specified in one manner for simplicity. They are specified in terms of the following variables:

BDIVE (in degrees, negative when orientation below horizontal)
RSLANT (1007) (feet)
BALPHA (367) (degrees)
BALPHY (368) (degrees)
VMACH (204) (Mach number)

It should be noted that the program when used to simulate many missions requires only that subsequent changes in data be added since the program will only update the last data set for the next run. (See "Program Description.")

SECTION V
PROGRAM LISTING

5.1 Complete Six-Degree-of-Freedom Program Listing with
Example

```

C
C*****DIPOCS TO BE USED WITH PORTAAM AMRK INTEGRATION ROUTINE
C
      CCGPON C143101,GRAPH
      EQUIVALENCE IC(2662),MMIN  ), IC(2663),MMAX  ), IC(2664),DER  ),
C      IC(2661),N    ), IC(2662),IPL  ), IC(2663),VAR  ),
C      IC(2000),T    ), IC(2011),KSTEP ), IC(2010),STEP ),
C      IC(2012),LSTEP ), IC(2008),PLDINO), IC(2009),NCPLOT),
C      IC(2021),OPOINT), IC(2025),TIME  ), IC(2025),VLABLE1),
C      IC(3167),NOOUT ), IC(2022),OPTN10), IC(2006),REPPLT),
C      IC(2665),EU    ), IC(2765),EL    ), IC(2007),PTLESS)
      EQUIVALENCE IC(1973),KASE ), IC(1971),RITE ), IC(1972),RAUTIA)
      EQUIVALENCE IC(1974),NJ   ), IC(1975),NPT   )
      DIMENSION GRAPH(2,2),TIME(2)
      DIMENSION VLABLE(2,15) , IPL(100) , DER(101)
      DIMENSION VAR(101) , EL(100) , EU(100)
      EQUIVALENCE IC(1980),RM    )
      EQUIVALENCE IC(1981),RNT   )
      EQUIVALENCE IC(1982),PLOTN4)
      EQUIVALENCE IC(1983),PLOTN2)
      EQUIVALENCE IC(1984),NPLOT )
      INTEGER CPOINT
      INTEGER PLOTNO
      INTEGER CPT
      EXTERNAL AUXSUB
      DO 22 I=1,4310
22  C(I)=0.
      NPT=2
      CALL CCOUNTY
      CALL ZERO
1000 CONTINUE
1001 IF (PLOTNO-LE=0.) GO TO 7
      IF (REPPLT-GT.0.) GO TO 7
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)
C
      IF (REPPLT-GT.-1.0) NOOUT = 0
      NPLOT=0
      7 CALL CIAPT1
      KASE=0
      IF (RKUTTA-GT.0.0) NPT=1
      LSTEP = STEP
      NPLOT4=PLDINO
      NPLOT2=PLDINO
      ACPLCT=PLDINO
1002 CALL SUBL1
1003 CALL AUX1
1004 CALL SUBL2
1005 DO 60 I = 2, N
      J = I-1
      EL(I) = (J+1)
      EU(I-1) = C(J+2)
      VAR(I) = C(J+3)
      60 DER(I) = C(J)
      VAR(I) = T
1006 CALL AUXSUB
1007 NJ=N-1
      CALL AMRK(AUXSUB)
1008 DO 50 I = 2, N
      J = IPL(I)-1
      50 C(J+3) = VAR(I)

```

```

      T = VAR(1)
1009 CALL SUB13
      IF (LSTEP.EQ. 1) GO TO 1007
      DO 155 JV=2,N
155  VAR(JV)=0.
      CALL S8
      CALL PROCES
      CALL RESET
      IF (LSTEP.EQ.3.OR.LSTEP.EQ.7.OR.NOPILOT.EQ.0)GOTO5
      CALL TIMEVIDELT)
      WRITE(6,96)ICELT
96  FORPAT(1H,17HSTART PLOTTING AT F14.7)
      LESSPT=FTLESS
      OPCINT=OPOINT-LESSPT
      CALL PLOT4 (GRAPH,CPOINT,VARIABLE,TIME,NPLOT4)
      CALL PLOT2 (NPLOT2)
      CALL PLOTN (NPLOT)
      CALL TIMEVICELT)
      WRITE(6,97)ICELT
97  FORPAT(1H,18HPLOTTING ENDED AT F14.7)
      IF (RNT.GT.0.1 .AND. RN.EQ.RNT).AND. LSTEP.EQ.2)GO TO 70
5  GC TO (1000,1001,1002,1003,1004,1005,1006,1007,1008,1009,1010),
      1 LSTEP
1010 IF (ICPTN10.GT.0.)
      1WRITE(6,6)11,C(11),C(1+1),C(1+2),C(1+3),C(1+4),C(1+5),C(1+6),1=1.35
      *10.7)
6  FORPAT(1H1/(15,1P7E15.7))
70  CALL S8
      CALL EXIT
      ENC
      BLOCK DATA
      COMMON /NCXCP/NCX(6)
      * /CXARG/ALP(7),AM(8)
      * /CXFUN/CX(56)
      DATA NCX/1,8,0,0,0,0/
      DATA ALP/0.,.4.,.6.,.8.,.10.,.15.,.20./
      DATA AM/0.,.7.,.9,1.025,1.2,1.5,2.0,2.3/
      DATA CX/
      1 0.0,0.0.,.18.,.32.,.66,1.56,2.66,
      2 0.0,0.0.,.18.,.32.,.66,1.56,2.66,
      3 0.0,0.0.,.22,1.35.,.71,1.7,3.27,
      4 0.0,0.0.,.23,1.37.,.79,1.8,3.2,
      5 0.0,0.0.,.21,1.34.,.78,1.65,3.26,
      6 0.0,0.0.,.13,1.25.,.51,1.11,2.08,
      7 0.0,0.0.,.10,1.21.,.38,1.95,1.67,
      8 0.0,0.0.,.09,1.19.,.35,1.82,1.57/
      ENC
      BLOCK DATA
      COMMON /NCC/NCC(4)
      * /CZARG/ALP(8),AM(9)
      * /CZFUN/CN(72)
      COMMON /NCCZ/NCC(4)
      * /CCARG/ALP(6),AM(6)
      * /CCIFUN/DCN(36)
      COMMON /NCF/NCF(4)
      * /CFARG/ ALP(8),BM(9 )
      * /CFUN/CN(72)
      COMMON /NCCM/NCCM(4)
      * /CCMARG/ALP(6),BMD(6)
      * /CCMFUN/CCM(36)
      DATA NCF/8,9,0,0/
      DATA ALP/0.,.2.,.4.,.6.,.8.,.10.,.15.,.20./

```

```

DATA AM/0.,.7,.9,1.05,1.1,1.5,1.7,2.0,2.3/
DATA CM/
* 0.0,.65,1.38,2.21,3.12,4.13,7.08,10.71,
* 0.0,.66,1.41,2.25,3.17,4.19,7.18,10.84,
* 0.0,.69,1.46,2.32,3.27,4.31,7.36,11.11,
* 0.0,.77,1.62,2.56,3.59,4.72,7.98,12.02,
* 0.0,.77,1.62,2.57,3.60,4.74,8.02,12.11,
* 0.0,.71,1.5,2.37,3.36,4.43,7.6,11.60,
* 0.0,.7,1.49,2.37,3.34,4.40,7.59,11.61,
* 0.0,.69,1.47,2.34,3.30,4.36,7.56,11.61,
* 0.0,.67,1.43,2.27,3.21,4.26,7.42,11.47/
DATA NCC/6,610.0/
DATA ALPC/0.,4.,8.,12.,16.,20./
DATA AMC/0.0,.7,.9,1.1,1.5,2.3/
DATA CCN/
* 0.0,.06,.30,.75,1.40,2.20,
* 0.0,.06,.30,.75,1.40,2.20,
* 0.0,.06,.30,.79,1.45,2.26,
* 0.0,.06,.30,.79,1.45,2.26,
* 0.0,.06,.30,.77,1.43,2.24,
* 0.0,.06,.30,.76,1.41,2.22/
DATA NCM/8,910.0/
DATA ALF/0.,2.,4.,6.,8.,10.,15.,20./
DATA BM/0.0,.7,.9,1.05,1.1,1.5,1.,2.0,2.3/
DATA CM/
* 0.0,-1.16,-2.48,-3.97,-5.62,-7.51,-12.83,-19.28,
* 0.0,-1.19,-2.54,-4.06,-5.76,-7.68,-13.10,-19.64,
* 0.0,-1.25,-2.66,-4.23,-5.98,-7.97,-13.54,-20.26,
* 0.0,-1.59,-3.35,-5.29,-7.40,-9.75,-16.24,-23.87,
* 0.0,-1.66,-3.48,-5.48,-7.66,-10.09,-16.75,-24.36,
* 0.0,-1.07,-2.31,-3.74,-5.35,-7.22,-12.48,-18.91,
* 0.0,-1.11,-2.39,-3.65,-5.50,-7.40,-12.70,-19.12,
* 0.0,-1.15,-2.47,-3.97,-5.65,-7.57,-12.90,-19.29,
* 0.0,-1.14,-2.46,-3.96,-5.63,-7.54,-12.81,-19.09/
DATA NCM/6,610.0/
DATA ALFO/0.,4.,8.,12.,16.,20./
DATA BMC/0.,.7,.9,1.1,1.5,2.3/
DATA CCP/
* 0.0,-.13,-.64,-1.82,-3.45,-5.40,
* 0.0,-.13,-.64,-1.82,-3.45,-5.40,
* 0.0,-.13,-.73,-1.93,-3.65,-5.65,
* 0.0,-.14,-.77,-1.99,-3.70,-5.60,
* 0.0,-.13,-.73,-1.93,-3.65,-5.65,
* 0.0,-.13,-.73,-1.93,-3.65,-5.65/
ENC
BLCK DATA
CCPPCN /NCXO/NCXO12/
* /CXGARG/AM18/
* /CXOFUN/CXO18/
DATA NCXC/8,1/
DATA AM /0.,.7,.9,1.025,1.2,1.5,2.0,2.3/
DATA CXG/.24,.24,.23,.68,.76,.91,1.0,.94/
ENC
BLCK DATA
CCPPCN /NCN2/NCN214/
* /CN2ARG/ALP16, AM13/
* /CN2FUN/CN2118/
DATA NCN2/6,3,0.0/
DATA ALP/0.,4.,8.,12.,16.,20./
DATA AM /0.,.7,2.3/
DATA CN2/
* 0.0,.17,168,1.50,2.60,3.90,

```

```

* 0.0, .17, .68, 1.50, 2.60, 3.90,
* 0.0, .17, .68, 1.50, 2.60, 3.90/
END
BLCK DATA
COMPEN /NCY2/NCY2(14)
* /CY2ARG/ALP(6), AM(3)
* /CY2FUN/CY2(18)
DATA NCY2/6,3,0,0/
DATA ALP /0., 4., 8., 12., 16., 20./
DATA AM/0.0, .7, 2.3/
DATA CY2 /
* 0.0, -.07, -.30, -.65, -1.1, -1.6,
* 0.0, -.07, -.30, -.65, -1.1, -1.6,
* 0.0, -.07, -.30, -.65, -1.1, -1.6/
END
BLCK DATA
COMPEN /NCL2/ NCL2(14)
* /CL2ARG/ALP(6), AM(4)
* /CL2FUN/CL2(24)
DATA NCL2/6,4,0,0/
DATA ALP/0.0, 4., 8., 12., 16., 20./
DATA AM/0., .7, .9, 2.3/
DATA CL2/
* 0.0, .02, .07, .14, .25, .33,
* 0.0, .02, .07, .14, .25, .33,
* 0.0, .03, .12, .25, .515, .35,
* 0.0, .02, .07, .14, .25, .33/
END
BLCK DATA
COMPEN /NCL3/ NCL3(14)
* /CL3ARG/ALP(6), AM(3)
* /CL3FUN/CL3(18)
DATA NCL3/6,3,0,0/
DATA ALP/0., 4., 8., 12., 16., 20./
DATA AM/0., .7, 2.3/
DATA CL3/
* 0.0, .022, .043, .08, .143, .215,
* 0.0, .022, .043, .08, .143, .215,
* 0.0, .022, .043, .08, .143, .215/
END
BLCK DATA
COMPEN /NCZD/ NCN(6)
* /CZCFUN/CN1(32), CN2(32), CN3(32), CN4(32), CN5(32), CN6(32),
* CN7(32), CN8(32)
* /CZCARG/ALP(8), DEF(4), AM(8)
DATA NCN/8,4,8,0,0,0/
DATA ALP/0., 2., 4., 6., 8., 10., 15., 20./
DATA DEF /-20., -10., 10., 20./
DATA AM/0., .57, .9, 1.05, 1.2, 1.5, 2., 2.3/
DATA CN1/
* -.222, -.218, -.215, -.212, -.208, -.204, -.194, -.181,
* -.205, -.204, -.203, -.201, -.198, -.196, -.190, -.181,
* .206, .207, .208, .208, .208, .207, .208, .207,
* .222, .224, .227, .229, .231, .233, .235, .237/
DATA CN2/
* -.222, -.218, -.215, -.212, -.208, -.204, -.194, -.181,
* -.206, -.204, -.203, -.201, -.198, -.196, -.190, -.181,
* .206, .207, .208, .208, .208, .209, .208, .207,
* .222, .224, .227, .229, .231, .233, .235, .237/
DATA CN3/
* -.252, -.249, -.245, -.241, -.237, -.232, -.220, -.207,
* -.236, -.235, -.233, -.231, -.228, -.225, -.217, -.209,

```

```

* .236, .237, .238, .239, .240, .240, .240, .236,
* .252, .255, .258, .260, .262, .265, .268, .269/
DATA CN4/
* -.276, -.273, -.269, -.264, -.260, -.255, -.241, -.226,
* -.260, -.259, -.257, -.254, -.251, -.249, -.239, -.229,
* .260, .261, .262, .263, .264, .264, .264, .261,
* .276, .279, .282, .285, .288, .289, .294, .295/
DATA CN5/
* -.246, -.243, -.240, -.238, -.232, -.227, -.215, -.202,
* -.230, -.229, -.227, -.225, -.222, -.220, -.212, -.203,
* .230, .232, .232, .234, .234, .234, .234, .231,
* .246, .249, .252, .255, .257, .258, .262, .263/
DATA CN6/
* -.168, -.165, -.163, -.161, -.158, -.155, -.146, -.138,
* -.152, -.151, -.149, -.149, -.147, -.145, -.139, -.134,
* .152, .152, .153, .153, .154, .154, .154, .153,
* .168, .170, .172, .173, .175, .176, .179, .180/
DATA CN7/
* -.131, -.129, -.127, -.125, -.123, -.121, -.115, -.108,
* -.115, -.114, -.113, -.112, -.111, -.109, -.106, -.102,
* .115, .116, .116, .116, .116, .117, .116, .115,
* .131, .132, .134, .135, .136, .137, .139, .140/
DATA CN8/
* -.123, -.122, -.120, -.118, -.116, -.114, -.108, -.106,
* -.107, -.106, -.106, -.105, -.104, -.103, -.099, -.095,
* .107, .108, .108, .109, .109, .109, .109, .107,
* .123, .125, .126, .127, .128, .129, .131, .132/
END
BLCKK DATA
CCPPON /NCLC/NCL16/
* /CLCARG/ALF(6), DEF(4), AM(3)
* /CLCFUN/CL1(24), CL2(24), CL3(24)
DATA NCL /8,4,3,0,0,0/
DATA ALP /0., 4., 8., 12., 16., 20./
DATA DEF /-20., -10., 10., 20./
DATA AM /0.0, 17, 2, 3/
DATA CL1 /
* -.144, -.149, -.141, -.138, -.138, -.134,
* -.131, -.128, -.125, -.124, -.123, -.121,
* .131, .128, .125, .124, .123, .121,
* .144, .149, .141, .138, .138, .134/
DATA CL2 /
* -.144, -.149, -.141, -.138, -.138, -.134,
* -.131, -.128, -.125, -.124, -.123, -.121,
* .131, .128, .125, .124, .123, .121,
* .144, .149, .141, .138, .138, .134/
DATA CL3 /
* -.148, -.145, -.143, -.140, -.138, -.136,
* -.144, -.138, -.136, -.133, -.130, -.130,
* .144, .138, .136, .133, .130, .130,
* .148, .145, .143, .140, .138, .136/
END
BLCKK DATA
CCPPON /NCHC/NCH16/
* /CHCARG/ALP(8), DEF(4), AM(8)
* /CHCFUN/CM1(32), CM2(32), CM3(32), CM4(32), CM5(32), CM6(32), CM7(32),
* CM8(32)
DATA NCH /8,4,8,0,0,0,0/
DATA ALP /0., 2., 4., 6., 8., 10., 15., 20./
DATA DEF /-20., -10., 10., 20./
DATA AM /0., .7, .9, 1.05, 1.2, 1.5, 2., 2.3/
DATA CH1/

```

```

1 .938, .937, .937, .938, .938, .939, .944, .948,
2 .864, .864, .864, .865, .864, .865, .870, .874,
3 -.854, -.854, -.864, -.865, -.864, -.865, -.870, -.874,
4 -.938, -.937, -.937, -.938, -.938, -.939, -.944, -.948/
DATA CP2/
1 -.938, -.937, -.937, -.938, -.938, -.939, -.944, -.948,
2 -.864, -.864, -.864, -.865, -.864, -.865, -.870, -.874,
3 .864, .864, .864, .865, .864, .865, .870, .874,
4 .938, .937, .937, .938, .938, .939, .944, .948/
DATA CP3/
1 -1.123, -1.137, -1.137, -1.138, -1.138, -1.14, -1.144, -1.152,
2 -1.064, -1.064, -1.064, -1.064, -1.064, -1.065, -1.072, -1.077,
3 1.054, 1.064, 1.064, 1.064, 1.064, 1.065, 1.072, 1.077,
4 1.123, 1.137, 1.137, 1.138, 1.138, 1.14, 1.144, 1.152/
DATA CP4/
1 -1.277, -1.277, -1.277, -1.277, -1.277, -1.279, -1.285, -1.29,
2 -1.204, -1.203, -1.203, -1.203, -1.204, -1.205, -1.211, -1.216,
3 1.204, 1.203, 1.203, 1.203, 1.204, 1.205, 1.211, 1.216,
4 1.277, 1.277, 1.277, 1.277, 1.277, 1.279, 1.285, 1.29/
DATA CP5/
1 -.796, -.793, -.791, -.789, -.788, -.787, -.787, -.786,
2 -.720, -.718, -.715, -.715, -.713, -.712, -.711, -.711,
3 .720, .718, .715, .715, .713, .712, .711, .711,
4 .796, .793, .791, .789, .788, .787, .787, .786/
DATA CP6/
1 -.621, -.620, -.619, -.617, -.615, -.614, -.612, -.613,
2 -.545, -.544, -.543, -.541, -.540, -.539, -.537, -.534,
3 .545, .544, .543, .541, .540, .539, .537, .534,
4 .621, .620, .619, .617, .615, .614, .612, .613/
DATA CP7/
1 -.586, -.584, -.583, -.582, -.580, -.579, -.577, -.578,
2 -.510, -.508, -.507, -.506, -.505, -.504, -.502, -.503,
3 .510, .508, .507, .506, .505, .504, .502, .503,
4 .586, .584, .583, .582, .580, .579, .577, .578/
DATA CP8/
1 -.586, -.584, -.583, -.581, -.580, -.579, -.578, -.575,
2 -.510, -.508, -.507, -.506, -.505, -.504, -.502, -.503,
3 .510, .508, .507, .506, .505, .504, .502, .503,
4 .586, .584, .583, .581, .580, .579, .578, .575/
END
BLCK DATA
COPPCN /NTX/NTX(2)
* /THARG/THA(20)
* /THFUN/THF(20)
DATA NTH/13,1/
DATA THA/
* 0., .05, .1, .2, .4, .8, 1.3, 1.5, 1.6, 1.8, 2.0, 2.8, 100./
DATA THF/
* 1., 500., 4000., 3800., 3850., 3800., 3600., 2000., 1100.,
* 700., 250., 0.0, 0.0/
END
BLCK DATA
COPPCN /NCLP/NL(4)
* /CLPARG/ALP(8),AMP(7)
* /CLPFUN/CLP(56)
COPPCN /NMQ/NQ(4)
* /CMQARG/ALQ(6),AMQ(4)
* /CMCFUN/CMQ(24)
DATA NL/8,7,0,0/
DATA ALP/0.,2.,4.,6.,8.,10.,15.,20./
DATA AMP/0.,.7,.9,1.05,1.5,2.0,2.3/
DATA CLP/

```

```

1 -6.07,-6.40,-7.07,-7.73,-8.38,-9.50,-10.58,-11.31,
2 -6.07,-6.40,-7.07,-7.73,-8.38,-9.50,-10.58,-11.31,
3 -6.53,-6.86,-7.52,-8.19,-8.84,-9.95,-11.03,-11.79,
4 -7.31,-7.65,-8.32,-8.99,-9.65,-10.79,-12.0,-12.87,
5 -5.01,-5.35,-6.03,-6.72,-7.42,-8.75,-10.16,-11.18,
6 -4.54,-4.88,-5.55,-6.22,-6.96,-8.39,-9.84,-10.89,
7 -4.45,-4.79,-5.46,-6.15,-6.92,-8.37,-9.86,-10.94/
DATA NC/6.4,0.0/
DATA ALC/0.4,0.12,10.20./
DATA APC/0.17,9.2,3/
DATA CPC/
1-.24,-.39,-.53,-.62,-.67,-.67,
2-.24,-.39,-.53,-.62,-.67,-.67,
3-.25,-.41,-.55,-.66,-.72,-.75,
4-.25,-.41,-.55,-.66,-.72,-.75/
ENC
SUBROUTINE CUADET(CETRIC,DEFICS,AA,BB,CC,DD,BEPSZ,SEPSY)
COMMON(4310)
EQUIVALENCE(C(371),RT)
DIMENSION(20),A(20),B(20)
DETRAC=HALF FOV OF DETECTOR
DEFAC=HALF FOV OF DEFOCUS SPOT
AA=LOWER LEFT QUADRANT
BB=LOWER RT QUADRANT
CC=UPPER LEFT QUADRANT
DD=UPPER RT QUADRANT
DEFCCS=HALF SIZE OF DEFOCUS SPOT AT INFINITY
LT=10
DEFCCS=DEFICS/57.29578
RT=(1.+100./RT)
3 XCF=BEPST/57.29578
YCF=BEPST/57.29578
VCF=.7071*(XCF+YCF)
UCF=.7071*(YCF-XCF)
XCF=UCF
YCF=VCF
DETRAC=CETRIC/57.29578
ZCF=SCAT(XCF**2+YCF**2)
DELRA=CETRAC*DEFOCS*RT1
IF(IZCF,CE,DELRA)GOTO42
C2TR=CEPAD**2
DEFRA=DEFOCS*RT1
D2FRA=CEFRAC**2
LN=LT/2
DELF=CETRAC/LN
DZCH=CELF/4.
LC1=LT+1
CC101=1,LC1
A(1)=0.
10 B(1)=0.
CC301=1,LC1
CIP=DETRAC-(1-1)*DELF
EXDET=SCAT(L2TR-(CIP)**2)
IF(102FRA-(CIP+YCF)**2).LT.0.1GOTO30
XX=SCAT(102FRA-(CIP+YCF)**2)
DO20NT=1,2
X1=XCF-(1-1)*NT*XX
IF(ABS(X1).GT.EXDETIDINT)=SIGNIEXDET,X1)
20 IF(ABS(X1).LE.EXDETIDINT)=(X1)
IF(SIGN(1.,C(1)).EQ.SIGN(1.,D(2)))GOTO22
IF(C(1).LE.0.)GOTO35
A(1)=C(1)-DZCH

```

```

      (F1A(1),LT,0.1A(1))=0.
      B(1)=C(2)+DZCN
      IF(B(1),GT,0.1B(1))=0.
      GOTO 30
35  A(1)=D(2)-DZCN
      IF(A(1),LT,0.1A(1))=0.
      B(1)=C(1)+CZCN
      IF(B(1),GT,0.1B(1))=0.
      GO TO 30
22  IF(C(1)+C(2)23,24,25
23  B(1)=ABS(D(1)-D(2))
      A(1)=0.
      GOTO 30
24  A(1)=0.
      B(1)=0.
      GOTO 30
25  A(1)=ABS(D(1)-D(2))
      B(1)=0.
30  CC=CONTINUE
      AA=0.
      BB=0.
      CC=0.
      CC=0.
      A(LN+1)=0.
      B(LN+1)=0.
      DO 401=1, LN
      AA=ABS((A(1)+A(1+1)))=.5*DELFI+AA
      BB=ABS((B(1)+B(1+1)))=.5*DELF1+BB
      CC=ABS((A(1+LN)+A(1+LN+1)))=.5*DELF1+CC
40  DD=ABS((B(1+LN)+B(1+LN+1)))=.5*DELF1+DD
      ADCT=3.14159*CDTR*.9
      TOTL=AA+BB+CC+DD
      IF(TOTL,GE,ADCT)WRITE(6,51)RT
51  FORMAT(15H BLIND RANGE ,1PE12.4)
      GOTO 60
42  WRITE(6,50)RT
      CBREAK=-1.
      AA=0.
      BB=0.
      CC=0.
      DD=0.
50  FORMAT(32H BREAKDOWN AT RANGE EQUAL TO ,1PE12.4)
60  RETURN
      END
      SUBROUTINE GAUSS(S,AM,V)
      COMMON C(4310)
      EQUIVALENCE(C(103),YFL)
      EQUIVALENCE(C(2000),T)
C     S=THE REQUIRED STANDARD DEVIATION
C     AM=IS THE REQUIRED MEAN
C     V=VALUE OF COMPUTED NORMALLY DISTRIBUTED RANDOM NUMBER
      A=0.
      B=1.
      DO 50 I=1,12
      YFL=RCNG(8)
50  A=A+YFL
      V=(A-6)*S+AM
      RETURN
      END
      SUBROUTINE C101
      COMMON C(4310)
      EQUIVALENCE(C(17),A1LL)

```

```

EQUIVALENCE(C(11),AIFAC)
EQUIVALENCE(C(12),VILM)
EQUIVALENCE(C(204),VMACH)
EQUIVALENCE(C(18),XIL)
EQUIVALENCE(C(1615),RXE)
EQUIVALENCE(C(14),HILL)
EQUIVALENCE(C(1623),RZE)
EQUIVALENCE(C(15),RILL)
EQUIVALENCE(C(13),VILF)
EQUIVALENCE(C(14),UT)
EQUIVALENCE(C(2000),T)
EQUIVALENCE(C(1666),BDIVE)
EQUIVALENCE(C(1667),RSLANT)
EQUIVALENCE(C(8),SPOTNO)
IF(SPCTMC.EQ.0.0) RETURN
UT = 0.0
IF(AILL.EQ.0) GO TO 10
IF(AIFAC.EQ.1) GO TO 9
VILM = VMACH
XIL = RSLANT * COSC(BDIVE)
HILL = RSLANT * SINC(BDIVE)
HILL = ABS(HILL)
GO TO 9
10 CONTINUE
VILF = 0.0
9 CONTINUE
RILL = SCRT(XIL**2 + HILL**2)
VSCUNC = -0.00392 * HILL * 1117.3
VILF = VSCUNC * VILM
1234 FORMAT(2X,9E13.4)
WRITE(6,1234) VILM,VMACH,XIL,T,HILL,RZE,RILL,VSCUNC,VILF
RETURN
END
SUBROUTINE C10
COMMON C14310)
EQUIVALENCE(C(446),ZLASR), (C(447),YLASR)
EQUIVALENCE(C(17),AILL)
EQUIVALENCE(C(13),VILF)
EQUIVALENCE(C(18),XIL)
EQUIVALENCE(C(2662),HMIN)
EQUIVALENCE(C(15),RILL)
EQUIVALENCE(C(14),HILL)
EQUIVALENCE(C(11),BCRE)
EQUIVALENCE(C(2),WAND)
EQUIVALENCE(C(3),RADIUS)
EQUIVALENCE(C(6),AISPOT)
EQUIVALENCE(C(9),XSPOT)
EQUIVALENCE(C(10),YSPOT)
EQUIVALENCE(C(14),UT)
EQUIVALENCE(C(2000),T)
EQUIVALENCE(C(446),CKDT)
EQUIVALENCE(C(1651),RTXE)
EQUIVALENCE(C(1655),RTYE)
EQUIVALENCE(C(8),SPCTMO)
IF(SPCTMC.EQ.0.0) RETURN
IF(T.EC.(PHIN/2.).AND.UT.EQ.0.0) GO TO 1
IF(T.EC.(PHIN/2.).AND.UT.EQ.0.1) GO TO 3
IF(T.LT.(UT-PHIN)) GO TO 2
IF(T.EC.0) GO TO 4
1 CONTINUE
140
UT = UT + CKDT

```



```

      IF(RH.GT.RHW)GOTO10
      IF (RANCC .LT. RAIL) GO TO 10
      IF(K2X.EQ.1969432)COT05
      R2K=1969432
      RANGTR=RANG0
5     CONTINUE
      IF(RANGTR+RWINC.GT.RANG0)GOTO7
      RANGTR=RANGTR+RWINC
      CALLGAUSS(SW,DPS17,DPS1WV)
      CALLGAUSS(SW1,VWTE,VWTEV)
7     VWXE=-VWTEV+CUSC(DPS1WV)
      VWYE=-VWTEV+SIND(BPS1WV)
      VWZE = 0.
      RETURN
10    VWXE = 0.
      VWYE = 0.
      VWZE = 0.
      RETURN
      END
C**AIR DATA MODULE G3
      SUBROUTINE G3
      COMMON C(4310)
C**INPUT DATA
      EQUIVALENCE (C(0208),RHZRO )
C**INPUTS FROM OTHER MODULES
      EQUIVALENCE (C(0100),VWXE )
      EQUIVALENCE (C(0101),VWYE )
      EQUIVALENCE (C(0102),VWZE )
      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),VWVXE )
      EQUIVALENCE (C(0201),VWVYE )
      EQUIVALENCE (C(0202),VWVZE )
      EQUIVALENCE (C(0203),PDYHMC)
      EQUIVALENCE (C(0204),VWACH )
      EQUIVALENCE (C(0205),DRHO )
      EQUIVALENCE (C(0206),VSCUND)
      EQUIVALENCE (C(0207),VAIRSP)
      EQUIVALENCE (C(0209),RH )
C**CALCULATE PRESENT ALTITUDE
      RH= -HZE+RHZRO
C**CALCULATE MISSILE VELOCITY WRT AIR MASS IN EARTH AXES
      VWXE = VXE-VWXE
      VWYE = VYE-VWYE
      VWZE = VZE-VWZE
      VAIRSP = SQRT(VWXE*VWXE+VWYE*VWYE+VWZE*VWZE)
C**AIR DENSITY, SPEED OF SOUND, DYNAMIC PRESSURE, AND MACH
      DRHO=(1.076475)/(1+.3325E-04*(RH+RH+RH+RH+.02315E-12))
      VSCUND = -.00392*(RH+.1117.3
      PDYHMC = (DRHO*VAIRSP*VAIRSP)/64.344
      VWACH = VAIRSP/VSCUND
      RETURN
      END
      SUBROUTINE G4
C** END-OF-RUN CALCULATIONS SUBROUTINE G4
C** THIS IS A SUBROUTINE, NOT A MODULE.

```

```

C** IT IS CALLED BY STAGE 3 TO COMPUTE MISS DISTANCE AND STOP THE
C** PROGRAM IF RANGE IS ZERO.
C*****
COMMON C(4310)
C**INPUT DATA
C**NCNE
C**INPUTS FROM OTHER MODULES
EQUIVALENCE (C(10357),BGAMH )
EQUIVALENCE (C(10358),BGAMV )
EQUIVALENCE (C(10371),RANGE )
EQUIVALENCE (C(1603),VXE )
EQUIVALENCE (C(1607),VYE )
EQUIVALENCE (C(1611),VZE )
EQUIVALENCE (C(1660),VTXE )
EQUIVALENCE (C(1661),VTYE )
EQUIVALENCE (C(1662),VTZE )
EQUIVALENCE (C(1615),RXE )
EQUIVALENCE (C(1619),RYE )
EQUIVALENCE (C(1623),RZE )
EQUIVALENCE (C(1651),RTAE )
EQUIVALENCE (C(1655),RTYE )
EQUIVALENCE (C(1659),RTZE )
C**INPUTS FROM MAIN PROGRAM
EQUIVALENCE (C(2000),T )
C**STATE VARIABLE OUTPUTS
C**NCNE
C**OTHER OUTPUTS
EQUIVALENCE (C(2020),LCGMV )
C** MISS DISTANCE PARAMETERS ARE OUTPUT DIRECTLY AND ARE NOT IN COMMON
C** TEST FOR INCREASING RANGE AND SOLVE FOR TIME AT WHICH RANGE IS ZERO
5 FORMAT (1H0, 16H MISS DISTANCE= 1PE17.8/1H0,13H TIME FINAL=
1PE17.8)
6 FORMAT (1H0,10X,10FXH EARTH=1PE17.8,3X,10HYM EARTH=1PE17.8,3X,
10HZM EARTH=1PE17.8)
7 FORMAT (1H0,40X,10FY FLTPATH=1PE17.8,3X,10HZ FLTPATH=1PE17.8)
C** TEST FOR INCREASING RANGE AND SOLVE FOR TIME AT WHICH RANGE IS ZERO
IF (RANGE.GT.500.) GO TO 20
IF (RANGE-URANGE.LT.0.) GO TO 10
TDEL = -(UVXT*(UXT-UXE)+UVYT*(UYT-UYE)+UVZT*(UZT-UZE)+UVXE*(UXE-
C UXT)+UVYE*(UYE-UYT)+UVZE*(UZE-UZT))/(UVXT*UVXT-2.*UVXT*UVXE
C +UVXE*UVXE+UVYT*UVYT-2.*UVYT*UVYE+UVYE*UVYE+UVZT*UVZT-2.*
C UVZT*UVZE+UVZE*UVZE)
UXTC = UVXT+TDEL*UXT
UYTO = UVYT+TDEL*UYT
UZTC = UVZT+TDEL*UZT
UXPO = UVXE+TDEL*UXE
UYTC = UVEE+TDEL*UYE
UZMO = UVZE+TDEL*UZE
RCX = UXTC-UXTO
RCY = UYTC-UYTO
RCZ = UZTC-UZTO
RMIS = SQRT(RCX*RCX+RCY*RCY+RCZ*RCZ)
TZERO = TDEL+UT
USTHT = SIN(BGAMV)
UCPSI = COS(BGAMV)
USPSI = SIN(BGAMH)
UCPSI = COS(BGAMH)
UC11 = UCSTHT*UCPSI
UC12 = -USTHT*UCPSI
UC13 = USPSI
UC21 = USTHT
UC22 = UCSTHT

```

```

UC31= -UCTHT*USPSI
UC32= USTHT*USPSI
UC33= UCPSI
RXFP = UC11*ROX+UC12*ROY+UC13*ROZ
RYFP = UC21*ROX+UC22*ROY
RZFP = UC31*ROX+UC32*ROY+UC33*ROZ
WRITE (6,5) PMISS, IZERO
WRITE (6,6) ROX, ROY, ROZ
WRITE (6,7) RYFP, RZFP
LCONV=2
RETURN
10 UT = 1
UXE = RAE
UYE = RYE
UZE = RZE
UXT = RTXE
UYT = RTYE
UZT = RTZE
UVXE = VXE
UVYE = VYE
UVZE = VZE
UVXT = VIXI
UVYT = VTYE
UVZT = VTZE
20 URANGE = RANGE
IF (RZE .GT. 100.) LCONV = 2
RETURN
END
C***COORDINATE CONVERSION MODULE
SUBROUTINE C5
COMMON C14310)
C
C***INPUTS FROM OTHER MODULES
EQUIVALENCE (C10200),VMWYE )
EQUIVALENCE (C10201),VMWYE )
EQUIVALENCE (C10202),VMWZE )
EQUIVALENCE (C10207),VAIRSP)
EQUIVALENCE (C11603),VXE )
EQUIVALENCE (C11607),VYE )
EQUIVALENCE (C11611),VZE )
EQUIVALENCE (C11615),RXE )
EQUIVALENCE (C11619),RYE )
EQUIVALENCE (C11623),RZE )
EQUIVALENCE (C11635),RO LX )
EQUIVALENCE (C11639),ROELY )
EQUIVALENCE (C11637),ROELZ )
EQUIVALENCE (C11661),VDXB )
EQUIVALENCE (C11664),VLYB )
EQUIVALENCE (C11665),VDZB )
EQUIVALENCE (C11668),RXO )
EQUIVALENCE (C11669),RYO )
EQUIVALENCE (C11670),RZO )
EQUIVALENCE (C11703),CFA11 )
EQUIVALENCE (C11707),CFA12 )
EQUIVALENCE (C11711),CFA13 )
EQUIVALENCE (C11715),CFA21 )
EQUIVALENCE (C11719),CFA22 )
EQUIVALENCE (C11723),CFA23 )
EQUIVALENCE (C11727),CFA31 )
EQUIVALENCE (C11731),CFA32 )
EQUIVALENCE (C11735),CFA33 )
EQUIVALENCE (C11739),WP )

```

```

EQUIVALENCE IC(1743),WQ      )
EQUIVALENCE IC(1747),WR      )
EQUIVALENCE IC(1751),CRAD    )
EQUIVALENCE IC(2000),T       )
EQUIVALENCE IC(2664),DER      )
EQUIVALENCE IC(3504),OPTNA    )

C
C**OTHER OUTPUTS
EQUIVALENCE IC(3501),BTBT    )
EQUIVALENCE IC(3511),BPSI    )
EQUIVALENCE IC(3521),BPHI    )
EQUIVALENCE IC(3531),BTHTO   )
EQUIVALENCE IC(3541),BPSID   )
EQUIVALENCE IC(3551),BPHID   )
EQUIVALENCE IC(3561),VTOTE    )
EQUIVALENCE IC(3571),BGAMM    )
EQUIVALENCE IC(3581),BGAMV    )
EQUIVALENCE IC(3601),VMWU     )
EQUIVALENCE IC(3611),VMWV     )
EQUIVALENCE IC(3621),VMWY     )
EQUIVALENCE IC(3631),BTPLV    )
EQUIVALENCE IC(3641),BPSLV    )
EQUIVALENCE IC(3651),BLAMV    )
EQUIVALENCE IC(3661),BLAMH    )
EQUIVALENCE IC(3671),BALPHA    )
EQUIVALENCE IC(3681),BALPHY    )
EQUIVALENCE IC(3691),BALPHY    )
EQUIVALENCE IC(3701),BPHIP    )
EQUIVALENCE IC(3711),RANGE    )
EQUIVALENCE IC(3721),RXBA     )
EQUIVALENCE IC(3731),RYBA     )
EQUIVALENCE IC(3741),RZBA     )
EQUIVALENCE IC(3751),BLCSBP    )
EQUIVALENCE IC(3761),BLOSBY    )
EQUIVALENCE IC(3771),BALPD     )
EQUIVALENCE IC(3781),BALYO     )
EQUIVALENCE IC(3791),BALPPC    )
EQUIVALENCE IC(3801),RANGO     )

C
C**CALCULATION OF HEADING, PITCH, ROLL EULER ANGLES IN DEGREES
BPHI = ATAND(CFA23,CFA33)
BTHT = ATAND(1-CFA13,SQRT(CFA11*CFA11+CFA12*CFA12))
BPSI = ATAND(CFA12,CFA11)

C
IF(COSD(BTHT).EQ.0.0) GO TO 5
BPSID = (WQ*SIND(BPHI)*WR*COSD(BPHI))/COSD(BTHT)
CONTINUE
BPHID = WP*BPSID*SIND(BTHT)
BTHTO = WQ*CCSD(CPHI)-WR*SIND(BPHI)

C
C**CALCULATION OF TOTAL VELOCITY
VTOTE = SQRT(VXE*VXE+VYE*VYE+VZE*VZE)

C
RANGO = SQRT((RXE-RXC)**2 + (RYE-RYO)**2 + (RZE-RZO)**2)

C
C**TRANSFORM MISSILE LOS FROM EARTH TO BODY AXES
RXBA = RDELX*CFA11 + RDEL2*CFA12 + RDEL3*CFA13
RYBA = RDELX*CFA21 + RDEL2*CFA22 + RDEL3*CFA23
RZBA = RDELX*CFA31 + RDEL2*CFA32 + RDEL3*CFA33

C
C**MISSILE-TGT LOS IN BODY AXES
BLCSBP = ATAND(1-RZBP,SQRT(RXBA*RXBA+RYBA*RYBA))

```

```

      BLC8BY = ATANC(RYBA,RXBA)
C
      UVP1 = VXE*RDELX+VYE*RDELY
      UVP2 = RDELX*RDELX+RDELY*RDELY
      UVP3 = VZE*RDELZ
      UVP4 = SQR(UVP2)
      RANGE = SQR(UVP2+RDELZ**2)
C**VERTICAL AND HORIZONTAL LINE OF SIGHT ANGLES (EARTH AXES)
C
      BLAMH = ATANC(-RDELY,RDELX)
      BLAMV = ATANC(-RDELZ,UVP4)
C
C**VERTICAL AND HORIZONTAL PROPORTIONAL NAVIGATION ANGLES
      IF(UT,CE)GOTO30
      IF(RANGE,EQ,0.0) GO TO 2
      VXP=(UVP1+UVP3)/RANGE
2     CONTINUE
      IF(UVP4,EQ,0.0) GO TO 1
      VYP = (VYE*RDELX-VXE*RDELY)/UVP4
      VZP = (VZE+UVP2-RDELZ*UVP1)/(RANGE*UVP4)
1     CONTINUE
      BTHLV = ATAND(VZP,VXP)
      BPSLV = ATAND(VYP,VXP)
C
      BGAMV = ATAND(-VZE,SQR(VXE*VXE+VYE*VYE))
      BGAMH = ATANC(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
      BALPHA = ATAND(VMWV,VMWU)
      BALPHY = ATAND(VMWV,VMWU)
C
      USC = VMWU**2
      IF(USC,EQ,0.0,AND,VMWW,EQ,0.0) GO TO 3
      BALPD = (VMWU*VDZB - VMWW*VDXB)/(USC+VMWW**2)*CRAD
3     CONTINUE
      IF(USC,EQ,0.0,AND,VMWV,EQ,0.0) GO TO 4
      BALYD = (VMWU*VDYD - VMWV*VDXB)/(USC+VMWV**2)*CRAD
4     CONTINUE
      BALFPE = 0.
      IF (BALPH,GT,0.) BALPPC= (BALPHA+BALPD + BALPHY+BALYD)/BALPH
C
C**ALPHA PRIME AND PHI PRIME (WIND TUNNEL AXES)
      IF (1/BALPHA-BALPHY),EQ,0.1 GO TO 30
      BPHIP=ATAND(VMWV,VMWU)
30  BALPHP=SQR(BALPHA**2+BALPHY**2)
      IF(ABS(1/BALPHP),GT,20.1)BALPHP=20.
      RETURN
      END
C**SEEKER AND PLATFORM INIT MODULE
      SUBROUTINE S11
      COMMON C(4310)
      DIMENSION IFL(100)
      EQUIVALENCE (C(445),UT )
      EQUIVALENCE (C(451),CAGE )
      EQUIVALENCE (C(460),GUIDE )
      EQUIVALENCE (C(461),SAMP )
      EQUIVALENCE (C(2541),N )

```

```

EQUIVALENCE (C(2562),IPL )
SZ=0.
SY=0.
UT = 0.
GUIDE=1.
CAGE=0.
SAMP = 0.
C(452) = 0.
IPLIN ) = 424
IPLIN+1) = 428
N=N+2
RETURN
END
C      TIGER PLATFORM AND TRACKER MODULE
C
SUBROUTINE S1
COMMON C(4310)
C
C**INPUT DATA
EQUIVALENCE (C(10441),SZGBIS)
EQUIVALENCE (C(10442),SYGBIS)
EQUIVALENCE (C( 443),OPTKR )
EQUIVALENCE (C( 444),OPTBKL)
EQUIVALENCE (C( 445),UT )
EQUIVALENCE (C( 446),COT )
EQUIVALENCE (C( 447),CKCB )
EQUIVALENCE (C( 448),CFGVZ )
EQUIVALENCE (C( 449),CFGVY )
EQUIVALENCE (C(11623),ZMSL)
EQUIVALENCE (C( 450),GSM )
EQUIVALENCE (C( 451),CAGE )
EQUIVALENCE (C(10452),ODREAR)
EQUIVALENCE (C(10453),RBKLOK)
EQUIVALENCE (C(10454),DECGE )
EQUIVALENCE (C(1405),RY)
EQUIVALENCE (C(1404),RZ)
EQUIVALENCE (C(10455),WEPSMX)
EQUIVALENCE (C(1470),SINCL)
EQUIVALENCE (C(1455),VLASR), (C(467),VLASR)
EQUIVALENCE (C(1463),CFVCS)
EQUIVALENCE (C(1449),CBTRAD)
EQUIVALENCE (C(10456),CKSKR )
EQUIVALENCE (C(1473),VLAIIRP)
EQUIVALENCE (C(1472),CKSK1)
EQUIVALENCE (C(10457),CRGSTP)
EQUIVALENCE (C(10458),CROSPT)
EQUIVALENCE (C( 460),GUIDE )
EQUIVALENCE (C( 461),SAMP )
EQUIVALENCE (C( 464),CGAMVS)
EQUIVALENCE (C( 465),CGAMMS)
EQUIVALENCE (C(10855),GZ )
EQUIVALENCE (C(10856),GY )
EQUIVALENCE (C(13504),OPTN4 )
C
C**INPUTS FROM OTHER MODULES
EQUIVALENCE (C(10371),RANGE )
EQUIVALENCE (C(10372),RXBA )
EQUIVALENCE (C(10373),RYBA )
EQUIVALENCE (C(10374),RZBA )
EQUIVALENCE (C(11615),RXE )
EQUIVALENCE (C(11616),RYE )
EQUIVALENCE (C(11623),RZE )

```

```

EQUIVALENCE IC(1739),WP      )
EQUIVALENCE IC(1740),WQD    )
EQUIVALENCE IC(1743),WQ      )
EQUIVALENCE IC(1744),WRC     )
EQUIVALENCE IC(1747),WR      )
EQUIVALENCE IC(2000),T       )
C
C**STATE VARIABLE OUTPUTS
EQUIVALENCE IC(0424),BTHTG01
EQUIVALENCE IC(0427),BTHTG  )
EQUIVALENCE IC(0428),BPSIG01
EQUIVALENCE IC(0431),BPSIG  )
C
C**OTHER OUTPUTS
EQUIVALENCE IC(0403),EZ      )
EQUIVALENCE IC(0407),EY      )
EQUIVALENCE IC(0432),RXG     )
EQUIVALENCE IC(0433),RYG     )
EQUIVALENCE IC(0434),RZG     )
EQUIVALENCE IC(0435),BEPSTZ )
EQUIVALENCE IC(0436),BEPSTY )
EQUIVALENCE IC(0437),WZ      )
EQUIVALENCE IC(0438),WY      )
EQUIVALENCE IC(0459),BGCEFL )
EQUIVALENCE IC( 462),UZ      )
EQUIVALENCE IC( 463),UY      )
C
C**DIRECTION COSINES FOR BODY TO PLATFORM TRANSFORMATION
IF(1,GT,0.)ICCTO30
SLTS=0.
SZ=0.
SY=0.
30 CONTINUE
UB31 = SIN0(BTHTG)
UB33 = COS0(BTHTG)
UB12 = SIN0(BPSIG)
UB22 = COS0(BPSIG)
UB11 = UB22*UB33
UB13 = -UB31*UB22
UB21 = -UB33*UB12
UB23 = UB31*UB12
UB32 = 0.
C
C** CALCULATE TOTAL DEFLECTION OF GIMBALS
BGCEFL=SQRT(BTHTG**2*BPSIG**2)
C
C**TRANSFORM LCS FROM BODY TO GIMBAL AXES
RXG = UB11*RXDA+UB12*RYDA+UB13*RZBA
RYG = UB21*RXDA+UB22*RYDA+UB23*RZBA
RZG = UB31*RXDA+UB33*RZBA
PCI=ATAND(-ZMSL,RANGE)
C
C**CHECK FOR MISSILE AT SEEKER BREAK-LOCK RANGE
IF (RANGE,GT,REKLGK) GO TO 40
IF (ICBREAK.LE,0.) GO TO 35
IF (ICICLK,GT,0.) GO TO 49
C
C**LINE OF SIGHT RATES AFTER BREAK-LOCK
UEPS = (T-UTIME)*WEP$MX
IF (UEPS,GT,BEDGE) UEPS = BEDGE
BEPSTZ = UEPSZ*UEPS
BEPSTY = UEPSY*UEPS

```

```

      QICKEK = 1.
      GO TO 50
C
C**INITIALIZATION OF BREAK LOCK VARIABLES
35 UTIME = T
   UEPSZ = BEPSZ
   UEPSY = BEPSY
   CBREAK = 1.
   WRITE (6,200) T,RANGE
200 FORMAT (30HOBREAKLOCK HAS OCCURRED, TIME=,F6.4,8H, RANGE=,F12.6)
C
C**LOS ERRORS IN PLATFORM COORDINATES
40 BEPSZ = ATAND(-RZG,RXG)
   BEPSY = ATAND(RYG,RXG)
   GO TO 50
49 QICKEK = -1.
50 CONTINUE
C
   IF (CPTKR .GT. 0.) GO TO 80
   SZ = CKSKR*BEPSZ
   SY = CKSKR*BEPSY
   CAGE = 1.
   GUICE = 1.
   GO TO 92
80 IF (T .LE. UT) GO TO 92
   UT = UT + CDT
   SZT=0.
   AAA=0.
   CALL GAUSS(SZT,AAA,AAV)
   IF (C103) .GT. VLZRP GO TO 91
   SST=0.
   SSZ=ZLASR*SIGN(PC1)/RANGE
   SSY=YLASR/RANGE
60 PZG=RZG+SSZ*RANGE
   PXG=RXG
   BEPSZ=ATAND(-PZG,PXG)
   PYG=RYG+SSY*RANGE
   BEPSY=ATAND(PYG,PXG)
   CALL GADETC(CTRAC,CEFOCS,AA,BB,CC,DD,BEPSZ,BEPSY)
   EF=AA+BB+CC+DD
   IF (EF.LT.-1.E-5) CF=10.
   SY= (CC-BB)*CKSKR/CF
   SZ= (CC-AA)*CKSKR/CF
   CAGE=1.
   GOTO 92
91 SZ=0.
   SY=0.
92 UZ = SZ
   UY = SY
   IF (CF.GT.0.) QICHECK=1.
C
C**PITCH PROGRAMMING AND SEEKER GAIN SWITCHING
   IF (GUICE.GT.0) GOTO 20
   IF (SAMP .GT. 0.) GO TO 19
   IF (CAGE .LE. 0.) GO TO 21
   UEZ = UZ
   SAMP = 1.
19 IF (SIGN(1., UZ) .EQ. SIGN(1., UEZ)) GO TO 21
   GUICE = 1.
20 DAFOCS=57.6*.00667/CDT/CDT
   EZ= UZ/CKSKR*DAFOCS
   EY= UY/CKSKR*DAFOCS

```

```

      UZ=UZ
      UY=UY
      GO TO 22
21  EZ = CGAMVS/GZ
      EY = CGAMHS/GY
      UZ = GSW*UZ
      UY = GSW*UY
22  CONTINUE
C
C***NON-LINEAR FRICTIONAL COUPLING OF GIMBALS
      UZK=SIGN(CRCSP, BPSICD)
      UYK=SIGN(CRCSTP, BTHTGD)
C
C***MISSILE BODY RATES IN GIMBAL AXES
      WY = UB21*WP+UB22*WC+UB23*WR
      WZ = UB31*WP+UB32*WC+UB33*WR
C
      BALPD = (UZ + SZGBIS + UZK)
      BETAD = (UY + SYGBIS - UYK)/UB22
C
C***GIMBAL ANGLE DERIVATIVES
      IF (CAGB .LE. 0.) GO TO 99
      BTHTGD = (BALPD - WY)/UB22
      BPSIGD = (BETAD - WZ)
C
      PLG OPTION
      BTHTGD=-CKSK3*BTHTGD
      BPSIGD=-CKSK3*BPSIGD
C
      RETURN
99  BTHTGD = 0.
      BPSIGD = 0.
      RETURN
      END
C** TIGER AUTOPILOT INITIALIZATION MODULE
C*****LCW FREQUENCY MODEL*****
      SUBROUTINE C11
      COMMON C14310
      DIMENSION IPL(100)
      EQUIVALENCE (C( 835),E2S )
      EQUIVALENCE (C( 843),EYS )
      EQUIVALENCE (C( 883),E2S5 )
      EQUIVALENCE (C( 887),EYSS )
      EQUIVALENCE (C( 464),CGAMVS)
      EQUIVALENCE (C( 465),CGAMHS)
      EQUIVALENCE (C(3504),OPTK4 )
      EQUIVALENCE (C(2561),N )
      EQUIVALENCE (C(2562),IPL )
C
      NLSUM = N
      IPL(N) = 800
      IPL(N+1) = 820
      IPL(N+2) = 824
      IPL(N+3) = 828
      IPL(N+4) = 832
      IPL(N+5) = 836
      IPL(N+6) = 840
      IPL(N+7) = 880
      IPL(N+8) = 884
      N = N+9
      E2S = 0.
      EYS = 0.
      GO TO 22
21  E2S = CGAMVS

```

```

EYS = CGAMHS
22 EZSS = EZS
EYSS = EYS
CI 803) = 0.
CI 823) = 0.
CI 827) = 0.
CI 831) = 0.
CI 839) = 0.
RETURN
END
C** TIGER AUTOPILOT MODULE
C*****LOW FREQUENCY MODEL*****
SUBROUTINE CI
COMMON C(4310)
DIMENSION BUELTC(4),VAR(101)

C
C**INPUT DATA
EQUIVALENCE (CI(0850),MLIMO )
EQUIVALENCE (CI(0851),MLIME )
EQUIVALENCE (CI 852),OBIAS )
EQUIVALENCE (CI 853),RBIAS )
EQUIVALENCE (CI(0855),GZ )
EQUIVALENCE (CI(0856),GY )
C
CI 857) THRU CI 860) ARE USED BY ECNTRL(1)
EQUIVALENCE (CI(0863),TAUZ )
EQUIVALENCE (CI(0864),TAUY )
EQUIVALENCE (CI 865),TCY1 )
EQUIVALENCE (CI 866),TCY2 )
EQUIVALENCE (CI 877),TAUL )

C
C**INPUTS FROM OTHER MODULES
EQUIVALENCE (CI(0352),BPH1 )
EQUIVALENCE (CI(0353),BPH10 )
EQUIVALENCE (CI(0403),EZ )
EQUIVALENCE (CI(0407),EY )
EQUIVALENCE (CI(0848),NH-SUM )
EQUIVALENCE (CI(1739),WP )
EQUIVALENCE (CI(1740),WGC )
EQUIVALENCE (CI(1743),WCC )
EQUIVALENCE (CI(1744),WRD )
EQUIVALENCE (CI(1747),WR )
EQUIVALENCE (CI(1751),CRAD )

C
C**INPUTS FROM MAIN PROGRAM
EQUIVALENCE (CI(2000),T )
EQUIVALENCE (CI(2965),VAR )
EQUIVALENCE (CI(2664),DER )

C
C** STATE VARIABLE OUTPUTS
EQUIVALENCE (CI 800),BPHISC)
EQUIVALENCE (CI 803),BPHIS )
EQUIVALENCE (CI 820),ESUMOD )
EQUIVALENCE (CI 823),ESUMD )
EQUIVALENCE (CI 824),ESUMEC )
EQUIVALENCE (CI 827),ESUME )
EQUIVALENCE (CI 828),EZSDO )
EQUIVALENCE (CI 831),EZSP )
EQUIVALENCE (CI 832),EZSD )
EQUIVALENCE (CI 835),EYS )
EQUIVALENCE (CI 836),EYSDO )
EQUIVALENCE (CI 839),EYSP )
EQUIVALENCE (CI 840),EYSD )

```

```

EQUIVALENCE IC( 843),EY8  )
EQUIVALENCE IC( 880),EZSSO )
EQUIVALENCE IC( 883),EZ6S  )
EQUIVALENCE IC( 884),EY8SO )
EQUIVALENCE IC( 887),EY6S  )

C
C**OUTPUTS
EQUIVALENCE IC( 857),BDELTC)

C
C**OTHER OUTPUTS
EQUIVALENCE IC( 876),EZRR )
C**PLATFORM RATES IN INERTIAL SPACE
EQUIVALENCE IC( 868),EYRR )
EQUIVALENCE IC( 869),WGC  )
EQUIVALENCE IC( 870),WRC  )
EQUIVALENCE IC( 873),ECCCR )
EQUIVALENCE IC( 874),EVNCR )
EQUIVALENCE IC( 875),BDELP )

C
C
C**GUIDANCE SIGNAL SHAPING
EZSC = EZSP
EYSD = EYSP
EZSCD = TAUZ*(TAUZ*(GZ+EZ - EZS) - 2.*EZSD)
EYSCD = TAUZ*(TAUZ*(GY+EY - EYS) - 2.*EYSD)
EZSSC = TAUZ*(EZSU/TAUL + EZS - EZSS)
EYSSD = TAUZ*(EYSC/TAUL + EYS - EYSS)

C
C**GRAVITY AND RATE BIAS
WCC = EZSS + QBIA5
WRC = EYSS + RBIA5

C
C**BODY RATE SHAPING AND CYRO DYNAMICS
WQS = WQ
WRS = WR
IF (ABS(WQS) .GT. 30.) WQS = SIGN(30., WQS)
IF (ABS(WRS) .GT. 30.) WRS = SIGN(30., WRS)

C
C**SUMMATION OF RATE DAMPING AND GUIDANCE SIGNALS AND THEIR DERIVATIVES
EZRR = WCS - WCC
EYRR = WRS - WRC

C
UKR = .85
IF (T.LT.TCY2) UKR = 4.25
IF (T.LT.TCY1) UKR = 0.
ESUMCD = UKR*(EZRR - EYRR)
ESUMED = UKR*(EZRR + EYRR)

C
C**TOTAL GUIDANCE SIGNAL SHAPING AND LIMITING
ECCCR = (ESUMCD/8. + ESUMD)
EVNCR = (ESUMED/8. + ESUME)
IF (ABS(ECCCR) .GT. HLIME) ECCCR = SIGN(HLIME, ECCCR)
IF (ABS(EVNCR) .GT. FLIME) EVNCR = SIGN(FLIME, EVNCR)

C
C**ROLL SIGNAL SHAPING
UKP = .33
IF (T.LT.TCY2) UKP = 1.65
IF (T.LT.TCY1) UKP = 0.
UPHIS = UKP*(BPHIC/12. + BPHI)
BPHISC = 16.*UPHIS - OPHIS
IF (ABS(BPHIS) .GT. 2.) BPHIS = SIGN(2., BPHIS)
BDELP = -BPHIS

```

```

C
C**AUTOPILLOT OUTPUT CURRENTS TO EACH ACTUATOR (FROM SUMMATION AMPS)
  BDELT(1) = EDCR - BDELPC
  BDELT(2) = EVCR - BDELPC
  BDELT(3) = EDCR + BDELPC
  BDELT(4) = EVCR + BDELPC
  RETURN
END
C** TIGER SIMPLIFIED ACTUATOR MODEL
C*****LCM FREQUENCY MODEL*****
C
C  SUBROUTINE C4
C
C  COMMON C(4310)
C  DIMENSION BDELT(4),BDELT(4),BDELT(4),VAR(101)
C  DIMENSION BDELT(4),BDELT(4)
C
C**INPUT DATA
  EQUIVALENCE(C(1121),BDMAX)
  EQUIVALENCE(C(1140),DELTPB)
  EQUIVALENCE(C(1141),DELTOB)
  EQUIVALENCE(C(1142),DELTRB)
C
C**INPUTS FROM OTHER MODULES
  EQUIVALENCE(C(1116),NDELTP)
  EQUIVALENCE(C(1157),BDELT)
  EQUIVALENCE(C(1117),BSURF1)
  EQUIVALENCE(C(1118),BSURF2)
  EQUIVALENCE(C(1119),BSURF3)
  EQUIVALENCE(C(1120),BSURF4)
C
C**FLAP DEFLECTION RATES
  BDELT(1) = BDELT(1) - DELTPB * DELTOB - DELTRB
  BDELT(2) = BDELT(2) - DELTPB * DELTOB * DELTRB
  BDELT(3) = BDELT(3) + DELTPB * DELTOB - DELTRB
  BDELT(4) = BDELT(4) + DELTPB * DELTOB * DELTRB
C
C**ACTUATOR DYNAMICS
  DO 30 I=1,4
    BDELT(I) = BDELT(I)
C
C**SURFACE POSITION LIMITER
  IF (ABS(BDELT(1)).LT.BDMAX)GOTO30
  BDELT(1) = SIGN(BDMAX,BDELT(1))
  30 CONTINUE
C
  BSURF1 = BDELT(1)
  BSURF2 = BDELT(2)
  BSURF3 = BDELT(3)
  BSURF4 = BDELT(4)
C
  C(1103) = BDELT(1)
  C(1107) = BDELT(2)
  C(1111) = BDELT(3)
  C(1115) = BDELT(4)
  RETURN
END
SUBROUTINE A1
C
C  COMMON C(4310)
C
C**TABLE LOOKUP FOR BGCY FORCE COEFFICIENTS

```

```

COPPEN /NCKO /NCKO /CXDARG/ CXOA /CXOFUN/ CXOF
1 /NCKOP / NCK /CKARG /CXA /CXFUN /CXF
2 /NCK2 /NCKN /CZARG /CNA /CZFUN /CNF
3 /NCK2 /NCKN /CZARG /CNA /CZFUN /CNF
4 /NCK2 /NCKN /CZARG /CNA /CZFUN /CNF
5 /NCK2 /NCKN /CZARG /CNA /CZFUN /CNF

C
C**TABLE LOOKUP FOR BODY MOMENT COEFFICIENTS
1 /NCL2/NCL2 /CL2ARG/ CL2A /CL2FUN/ CL2F
2 /NCL3/NCL3 /CL3ARG/ CL3A /CL3FUN/ CL3F
3 /NCLP/NCLP /CLPARG/ CLPA /CLPFUN/ CLPF
4 /NCLP/NCLP /CLPARG/ CLPA /CLPFUN/ CLPF
5 /NCLP/NCLP /CLPARG/ CLPA /CLPFUN/ CLPF
6 /NCLP/NCLP /CLPARG/ CLPA /CLPFUN/ CLPF
7 /NCLP/NCLP /CLPARG/ CLPA /CLPFUN/ CLPF
8 /NCLP/NCLP /CLPARG/ CLPA /CLPFUN/ CLPF
9 /NCLP/NCLP /CLPARG/ CLPA /CLPFUN/ CLPF
10 /NCLP/NCLP /CLPARG/ CLPA /CLPFUN/ CLPF

C
C**TABLE LOOKUP FOR SURFACE COEFFICIENTS
COMMON /NCL0/ NCL0 /CZCARG/ CZCA /CZDFUN/ CZDF
2 /NCL0/ NCL0 /CZCARG/ CZCA /CZDFUN/ CZDF
3 /NCL0/ NCL0 /CZCARG/ CZCA /CZDFUN/ CZDF

C
C** INPUT DATA
EQUIVALENCE (C11252),XINTER)
EQUIVALENCE (C11260),CXERR )
EQUIVALENCE (C11261),CZERR )
EQUIVALENCE (C11262),CYERR )
EQUIVALENCE (C11263),CLERR )
EQUIVALENCE (C11264),CMERR )
EQUIVALENCE (C11265),CENRR )

C
C**INPUTS FROM OTHER MODULES
EQUIVALENCE (C10204),VMACH )
EQUIVALENCE (C10367),BALPHA)
EQUIVALENCE (C10202),LCONV)
EQUIVALENCE (C10359),BALPHY)
EQUIVALENCE (C10359),BALPH)
EQUIVALENCE (C10370),BPHIP )
EQUIVALENCE (C11117),BSURF1)
EQUIVALENCE (C11118),BSURF2)
EQUIVALENCE (C11119),BSURF3)
EQUIVALENCE (C11120),BSURF4)

C
C**INPUTS FROM MAIN PROGRAM
EQUIVALENCE (C12000),T )
EQUIVALENCE (C12664),DER )

C
C**OUTPUTS - COEFFICIENTS FOR BODY FORCES
EQUIVALENCE (C11203),CX )
EQUIVALENCE (C11212),CXO )
EQUIVALENCE (C11213),CXG )
EQUIVALENCE (C11214),CNPT )
EQUIVALENCE (C11215),CY2 )
EQUIVALENCE (C11230),BOEFL )
EQUIVALENCE (C11735),CCGN )
EQUIVALENCE (C11244),CNPU )
EQUIVALENCE (C11245),CYPV )

C
C**OUTPUTS - COEFFICIENTS FOR BODY MOMENTS
EQUIVALENCE (C11206),CLP )
EQUIVALENCE (C11207),CMQ )
EQUIVALENCE (C11208),CNA )
EQUIVALENCE (C11209),CL )
EQUIVALENCE (C11210),CM )

```

```

EQUIVALENCE IC(1211),CN      )
EQUIVALENCE IC(1217),CM0     )
EQUIVALENCE IC(1218),CM2     )
EQUIVALENCE IC(1231),CCGM    )
EQUIVALENCE IC(1240),CL2     )
EQUIVALENCE IC(1241),CL3     )
EQUIVALENCE IC(1247),CMP     )
EQUIVALENCE IC(1248),CNP     )
EQUIVALENCE IC(1249),CLR     )

C
C**OUTPUTS - COEFFICIENTS FOR SURFACE EFFECTS, AND TOTAL EFFECTS
EQUIVALENCE IC(1204),CY      )
EQUIVALENCE IC(1205),CZ      )
EQUIVALENCE IC(1209),CL      )
EQUIVALENCE IC(1210),CM      )
EQUIVALENCE IC(1211),CN      )
EQUIVALENCE IC(1219),CZO     )
EQUIVALENCE IC(1220),CZR     )
EQUIVALENCE IC(1221),CMGCP   )
EQUIVALENCE IC(1222),CMR     )
EQUIVALENCE IC(1223),CYR     )
EQUIVALENCE IC(1224),CYQ     )
EQUIVALENCE IC(1225),CLDRP   )
EQUIVALENCE IC(1226),CNO     )
EQUIVALENCE IC(1227),CLD     )
EQUIVALENCE IC(1228),CLMP    )
EQUIVALENCE IC(1229),CLNP    )
EQUIVALENCE IC(1232),BCI     )
EQUIVALENCE IC(1233),BCM     )
EQUIVALENCE IC(1234),BCN     )
EQUIVALENCE IC(1250),CZP     )
EQUIVALENCE IC(1251),CYP     )

C
C INPUT VARIABLE XINTER IS THE INTERPOLATION CONTROL
C LESS THAN ZERO - STRAIGHT LINE INTERPOLATION
C POSITIVE - PARABOLIC INTERPOLATION, WITH END INTERVAL
C INTERPOLATION (0. TO 1.)
C 0.0 - STRAIGHT LINE
C 1.0 - FULL PARABOLIC
C
C
C IF (T.LE.CER) UTIME= 0.
C IF (T-UTIME .LE. 0.) RETURN
C UTIME= T

C
C MULTIPLE ANGLE FORMULAE AND ABSOLUTE VALUES OF ANGLE OF ATTACK
USPH1 = SIN(I3PHIP)
UCPH1 = COS(I3PHIP)
US2PH1 = SIN ( 2. * BPHIP )
US2PH2 = US2PH1 ** 2
US4PH1 = SIN ( 4. * BPHIP )
IF (ABS(BALPHY).GT.20.)BALPHY=SIGN(20.,BALPHY)
IF (ABS(BALPHA).GT.20.)BALPHA=SIGN(20.,BALPHA)
UALPHA = ABS(BALPHA)
UALPHY = ABS(BALPHY)

C
C**CALCULATION OF BODY FORCE COEFFICIENTS
BDEF1 = (ABS(SURF1)+ABS(SURF2)+ABS(SURF3)+ABS(SURF4))/4.
CXO=CCCIN2(VPACH,CXO,CXOF,NCXD,XINTER,3HCXO)
CALL TABL2 (BALPMP, VMACH,CXA,QXF,NCX,XINTER,3HCXC,CXC)
IF (LCNV.EQ.2)CALLCUPT2
IF (LCNV.EQ.2)C(12000)=C(12001)

```

```

IF ILCCNV.EQ.2)RETURN
CALL TABL2 (BALPFP,VMACH,CMA,CNF,NCN,XINTER,4HCNPT,CNPT)
CALL TABL2 (BALPFP,VMACH,CDCNA,CDCNF,HCDCN,XINTER,4HCDCN,CDCN)
CALL TABL2 (BALPFP,VMACH,CY2A,CY2F,NCY2,XINTER,3HCY2,CY2)

C
CX=CX0+CXG
CAPU=CNPT+CCCN*US2PH2
CYPV = CY2+US4PHI

C
C**CALCULATION OF BODY MOMENT COEFFICIENTS
CALL TABL2 (BALPFP,VMACH,CL2A,CL2F,NCL2,XINTER,3HCL2,CL2)
CALL TABL2 (BALPFP,VMACH,CL3A,CL3F,NCL3,XINTER,3HCL3,CL3)
CALL TABL2 (BALPFP,VMACH,CMA,CMF,NCM,XINTER,3HCMO,CMO)
CALL TABL2 (BALPFP,VMACH,CUCHA,CUCHF,HCDCN,XINTER,4HCDCN,CDCN)
CALL TABL2 (BALPFP,VMACH,CN2A,CN2F,NCN2,XINTER,3HCN2,CN2)
CALL TABL2 (BALPFP,VMACH,CLPA,CLPF,NCLP,XINTER,3HCLP,CLP)
CALL TABL2 (BALPFP,VMACH,CMQA,CMQF,NCMJ,XINTER,3HCMQ,CMQ)
CALL TABL2 (BALPFP,VMACH,CMQA,CMQF,NCMJ,XINTER,3HCMQ,CMQ)

C
CLP = CL2*US4PHI + CL3*USPHI
CMP = CMQ+CCCN*US2PH2
CMP = CN2*US4PHI

C
C**CALCULATION OF SURFACE COEFFICIENTS
BDL = (1-BSURF1-BSURF2+BSURF3+BSURF4)/4.
BCM = (1-BSURF1+BSURF2+BSURF3+BSURF4)/4.
BCN = (1-BSURF1+BSURF2-BSURF3+BSURF4)/4.
BCNP = -BCM
CALL TABL3 (BALPFP,BDM,VMACH,CZDA,CZDF,NCZD,XINTER,3HCZD,CZD)
CALL TABL3 (BALPFP,BDM,VMACH,CZDA,CZDF,NCZD,XINTER,3HCZD,CZD)
CALL TABL3 (BALPFP,BDM,VMACH,CMCA,CMCF,NCMD,XINTER,3HCMDCP,CMDCP)
CALL TABL3 (BALPFP,BDM,VMACH,CMCA,CMCF,NCMD,XINTER,3HCMDCP,CMDCP)
CALL TABL3 (BALPFP,BDM,VMACH,CZDA,CZDF,NCZD,XINTER,3HCYR,CYR)
CALL TABL3 (BALPFP,BDM,VMACH,CZDA,CZDF,NCZD,XINTER,3HCYR,CYR)
CALL TABL3 (BALPFP,BDM,VMACH,CZDA,CZDF,NCZD,XINTER,3HCYR,CYR)
CALL TABL3 (BALPFP,BDM,VMACH,CMCA,CMCF,NCMD,XINTER,3HCMDCP,CMDCP)
CALL TABL3 (BALPFP,BDM,VMACH,CMCA,CMCF,NCMD,XINTER,3HCMDCP,CMDCP)
CALL TABL3 (BALPFP,BDM,VMACH,CMCA,CMCF,NCMD,XINTER,3HCMDCP,CMDCP)

C
CNC=-ABS(CNC)
CMCCP=-ABS(CMDCP)
CLC=ABS(CLO)
CYR=ABS(CYR)
CYC=ABS(CYQ)
CZR=ABS(CZR)
CZC=ABS(CZQ)
CMR=-ABS(CMR)
CLCRP=-ABS(CLDCP)
CZP=(CNCU+CZC*UCPHI+BCM+CZR*USPHI+BDN)
CYP=(CYR+CZC*UCPHI+BCN+CYC*USPHI+BDN)
CL=(CLR+CLO+BDL)
CLYF=CMP+CYC*UCPHI+BCN+CMR*USPHI+BDN)
CLNF=CMR+CLCRP*UCPHI+BDN+CNO*USPHI+BDN)

C
C**AERC COEFFICIENT ERRORS
CX = CX + CXERR
CZP = CZP + CZERR
CYP = CYP + CYERR
CL = CL + CLERR
CLMP = CLMP + CMERR
CLNP = CLNP + CNERR

C
C** TRANSFORMATION FROM WIND TO BODY AXIS

```

```

      CY = CYP*UCPHI-CZP*USPHI
      CZ = -CZP*UCPHI-CYP*USPHI
      CN = CLPP*UCPHI+CLNP*USPHI
      CN = CLNP*UCPHI-CLPP*USPHI
      RETURN
    END
C**AERG FORCE AND MOMENT MODULE   BODY AXES
SUBROUTINE A2
  COPPEN C(4310)
C
C**INPUT DATA
  EQUIVALENCE (C(1306),RFAREA)
  EQUIVALENCE (C(1307),RFLCT)
  EQUIVALENCE (C(1308),RDELCO)
  EQUIVALENCE (C(1313),RFACG)
  EQUIVALENCE (C(1314),RFVCG)
  EQUIVALENCE (C(1315),RFZCG)
  EQUIVALENCE (C(1316),RLCG)
  EQUIVALENCE (C(1317),RAIC)
  EQUIVALENCE (C(1627),AGRAV)
  EQUIVALENCE (C(3504),OPTN4)
C
C**INPUTS FROM OTHER MODULES
  EQUIVALENCE (C(1023),PDYHNC)
  EQUIVALENCE (C(1027),VAIRSP)
  EQUIVALENCE (C(1350),BIPT)
  EQUIVALENCE (C(1380),RANGD)
  EQUIVALENCE (C(1203),CX)
  EQUIVALENCE (C(1204),CY)
  EQUIVALENCE (C(1205),CZ)
  EQUIVALENCE (C(1206),CLP)
  EQUIVALENCE (C(1207),CMQ)
  EQUIVALENCE (C(1208),CNR)
  EQUIVALENCE (C(1209),CL)
  EQUIVALENCE (C(1210),CM)
  EQUIVALENCE (C(1211),CN)
  EQUIVALENCE (C(1236),CM1)
  EQUIVALENCE (C(1237),CM2)
  EQUIVALENCE (C(1238),CM3)
  EQUIVALENCE (C(1239),CM4)
  EQUIVALENCE (C(1411),FTX)
  EQUIVALENCE (C(1412),FTY)
  EQUIVALENCE (C(1413),FTZ)
  EQUIVALENCE (C(1422),RLCG)
  EQUIVALENCE (C(1628),DNASS)
  EQUIVALENCE (C(1723),CPA23)
  EQUIVALENCE (C(1735),CPA33)
  EQUIVALENCE (C(1739),WP)
  EQUIVALENCE (C(1743),WQ)
  EQUIVALENCE (C(1747),WR)
  EQUIVALENCE (C(1749),FM1Y)
  EQUIVALENCE (C(1750),FM1Z)
C
C**OTHER OUTPUTS
  EQUIVALENCE (C(1300),FXBA)
  EQUIVALENCE (C(1301),FYBA)
  EQUIVALENCE (C(1302),FZBA)
  EQUIVALENCE (C(1303),FXBA)
  EQUIVALENCE (C(1304),FYBA)
  EQUIVALENCE (C(1305),FZBA)
  EQUIVALENCE (C(1309),FM1)
  EQUIVALENCE (C(1310),FM2)

```

```

EQUIVALENCE (C(1311),FMP3) 1
EQUIVALENCE (C(1312),FMP4) 1
EQUIVALENCE (C(1320),FMXTM) 1
EQUIVALENCE (C(1321),FMYTH) 1
EQUIVALENCE (C(1322),FMZTH) 1
EQUIVALENCE (C(1323),FMXLUG) 1
EQUIVALENCE (C(1324),FMYLUG) 1
EQUIVALENCE (C(1325),FMZLUG) 1

C
C==FORCE VECTOR COMPONENTS
UCS = PLVANC*RFAREA
UCSL = UCS*REFLGM

C
FXBA=UCS*(1-CX)*FTX
FYBA=UCS*(CY+FTY
FZBA=UCS*(CZ+FTZ
IF (VAIRSP.LE.0.0) GO TO 72

C
C==AERO MOMENTS
FMXBA = (CL*(ICLP/VAIRSP)*REFLGM*WP)*UQSL
FMYBA = (CM*(ICMG/VAIRSP)*REFLGM*WQ)*UQSL+FYBA*ROELCG
FMZBA = (CN*(ICNR/VAIRSP)*REFLGM*WR)*UQSL-FYBA*ROELCG

C
C==MOMENTS CAUSED BY THRUST MISALIGNMENTS
FMXTM = -FTX*RFZCC + FTZ*RFYCC
FMYTH = FTX*RFZCC + FTZ*RFXCC
FMZTH = -FTX*RFYCC - FTY*RFXCC

C
C==MOMENTS AND FORCES DUE TO LUGS
IF (RANGC.LE.RAIL+RLUG)GOTO70
FYLUG = 0.
FZLUG = 0.
FMXLUG = 0.
FMYLUG = 0.
FMZLUG = 0.
GO TO 74

70 IF (RANGC .LE. RAIL) GO TO 72
FYLUG = -(FYBA + CMASS*AGRAV*CFA23 + (FMZBA + FMZTH)*
      RLCG*DMASS/FMIZ)/(1. + DMASS*RLCG*RLCG/FMIZ)
FZLUG = -(FZBA + DMASS*AGRAV*CFA33 + (FMYBA + FMYTH)*
      RLCG*DMASS/FMIY)/(1. + DMASS*RLCG*RLCG/FMIY)
FMXLUG = -(FMXBA + FMXTM)
FMYLUG = FZLUG*RLCG
FMZLUG = FYLUG*RLCG
GO TO 74

72 CONTINUE
FYLUG = -(FYBA + CMASS*AGRAV*CFA23)
FZLUG = -(FZBA + DMASS*AGRAV*CFA33)
FMXLUG = -(FMXBA + FMXTM)
FMYLUG = -(FMYBA + FMYTH)
FMZLUG = -(FMZBA + FMZTH)

74 CONTINUE

C
C==TOTAL PERCE AND MOMENTS
FYBA = FYBA + FYLUG
FZBA = FZBA + FZLUG
FMXBA = FMXBA + FMXTM + FMXLUG
FMYBA = FMYBA + FMYTH + FMYLUG
FMZBA = FMZBA + FMZTH + FMZLUG

C
C== CALCULATE HINGE MOMENTS
FPHI = CF1*UQSL

```

```

      FPH2 = CP2*UCSL
      FPH3 = CP3*UCSL
      FPH4 = CP4*UCSL
      RETURN
      END
C**INITIALIZATION FOR ENGINE MODULE
      SUBROUTINE A31
      COMMON C(4310)
      DIMENSION IPL(101)
      EQUIVALENCE (C(2561),N )
      EQUIVALENCE (C(2562),IPL )
      C(1499) = 0.
      IPLIN 1 = 1494
      N = N+1
      RETLPH
      END
C**ENGINE MODULE
      SUBROUTINE A3
      COMMON C(4310)
C
C**LOOK UP TABLE FOR THRUST
      COMMON /NTH/INTH /THARG/THA /TMPUN/THF
C
C** INPUT DATA
      EQUIVALENCE (C(1401),BALPHT)
      EQUIVALENCE (C(1402),BPHIT )
      EQUIVALENCE (C(1403),GNALGN)
      EQUIVALENCE (C(1404),PCFTH )
      EQUIVALENCE (C(1405),QBURN )
      EQUIVALENCE (C(1414),CISP )
      EQUIVALENCE (C(1415),CHT )
      EQUIVALENCE (C(1416),OWP )
      EQUIVALENCE (C(1417),ROCGO )
      EQUIVALENCE (C(1418),ROCGF )
      EQUIVALENCE (C(1419),FMIXO )
      EQUIVALENCE (C(1420),FMIXO )
      EQUIVALENCE (C(1421),RLCGO )
      EQUIVALENCE (C(1627),ACRAV )
C
C** INPUTS FROM OTHER MODULES
      EQUIVALENCE (C(1252),XINTER)
      EQUIVALENCE (C(2000),T )
C
C** OUTPUTS
      EQUIVALENCE (C(1308),RDELCG)
      EQUIVALENCE (C(1409),UCWP )
      EQUIVALENCE (C(1410),FTHRST)
      EQUIVALENCE (C(1411),FTMX )
      EQUIVALENCE (C(1412),FTFY )
      EQUIVALENCE (C(1413),FTFZ )
      EQUIVALENCE (C(1422),RLCG )
      EQUIVALENCE (C(1628),DMASS )
      EQUIVALENCE (C(1748),FMIX )
      EQUIVALENCE (C(1749),FMIX )
      EQUIVALENCE (C(1750),FMIX )
C
C**STATE VARIABLES AND THEIR DERIVATIVES
      EQUIVALENCE (C(1496),UIMPO )
      EQUIVALENCE (C(1499),UIMP )
C
      IF (CBURN.GT.10.) RETURN
      FTHRST=CCOIMP(T,THA,THF,INTH,XINTER,OHFTHRST)

```

```

      FTHRST = FTHRST*11. + PCFTM1
C
10  USINA=SIN(18ALPMT)
    FTHX=FTHRST*COS(18ALPMT)
    FTHY=-FTHRST*USINA*SIN(18PHIT1)
    FTHZ=FTHRST*USINA*COS(18PHIT1)
    GO TO 30
20  FTHX=FTHRST
    FTHY=0.
    FTHZ=0.
30  CCNTINUE
C
    UIMP0 = FTHRST
    UDWP = UIMP/CISP
C
    CMASS = ICWT - UDWP/ACRAV
    RDELGG = RCGG0 - (RCGG0 - RCGG1)*UDWP/DWP
C
    FPIX = FPIX0*(ICWT - UDWP)/CWT
    FPIY = FPIY0*(ICWT - UDWP)/CWT
    FPIZ = FPIY
    RLCG = RLCG0 + RDELGG
    IF (FTHRST .GT. 0.) RETURN
C
    WRITE (6,100) T
100  FORMAT (//14H BURNOUT TIME=,F8.4,5H SEC.)
    CBURN=1.0
    FTHRST=0.
    FTHX=0.
    FTHY=0.
    FTHZ=0.
    RETURN
END
C** TRANSLATIONAL DYNAMICS INITIALIZATION MODULE FOR D1
SUBROUTINE G11
COMMON C143101
EQUIVALENCE (C125611,M )
EQUIVALENCE (C125621,IPL )
DIMENSION IPL (100)
C
C** INPUT DATA
EQUIVALENCE (C1 100),VMRE )
EQUIVALENCE (C1 101),VMY6 )
EQUIVALENCE (C1 102),VMZE )
EQUIVALENCE (C1 204),VMACH )
EQUIVALENCE (C1 208),RHZRO )
EQUIVALENCE (C1 367),DALPHA1
EQUIVALENCE (C1371),PANGE)
EQUIVALENCE (C11752),RPH10)
EQUIVALENCE (C11501),EMTARG)
EQUIVALENCE (C1 368),BALPHY)
EQUIVALENCE (C1 427),BTFYG )
EQUIVALENCE (C1 431),BPSIG )
EQUIVALENCE (C11405),CBURN )
EQUIVALENCE (C11639),OPTARG)
EQUIVALENCE (C11666),BCIVE )
EQUIVALENCE (C11667),RSLANT)
EQUIVALENCE (C11739),WP )
EQUIVALENCE (C11743),WQ )
EQUIVALENCE (C11747),WR )
EQUIVALENCE (C13502),OPTN2 )
EQUIVALENCE (C13504),OPTN4 )

```

```

C
C** OUTPUT TO MODULES
EQUIVALENCE (C(1370),BPHIP )
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(1629),ATHRST)
EQUIVALENCE (C(1635),RCELX )
EQUIVALENCE (C(1636),RDELY )
EQUIVALENCE (C(1637),RCELZ )
EQUIVALENCE (C(1644),ATARG)
EQUIVALENCE (C(1647),VTARG)
EQUIVALENCE (C(1651),RTXE )
EQUIVALENCE (C(1655),RTYE )
EQUIVALENCE (C(1659),RTZE )
EQUIVALENCE (C(1663),RXD )
EQUIVALENCE (C(1669),RYD )
EQUIVALENCE (C(1670),RZD )
EQUIVALENCE (C(1753),BTHTO )
EQUIVALENCE (C(1754),BPSIO )

```

C

```

C(1647)=C(1648)
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
IPL(N+11)=1672
N=N+12
IF(EMTARG)10,9,10
10 ATARG=ATHRST/EMTARG
GETG11
9 ATARG=0.
11 CCNTINUE
BPHIC=0.
C**CALCULATE MISSILE PARAMETER INITIAL CONDITIONS
BTHTC=B0IVE*BALPHA
BPSIC=-B0ALPHY
RXE=-RSLANT*COSEC(B0CIVE)
RANCE=RSLANT
RYE=RSLANT*SINC(B0ALPHY)
RZE=RSLANT*SINC(B0CIVE)
20 RH = RHRO - RZE
C(1427)=-BALPHA
C(1431)=BALPHY
C(1431)=-BALPHY
C
USTHT = SINC(BTHTO)
UCTHT = COSC(BTHTO)
UCPSI = COSD(BPSIO)
USPSI = SINC(BPSIO)
C

```

```

RXBA = -UCPSI*UCTHT*RXE + USTHT*RZE
RYBA = USPSI*RXE
RZBA = -UCPSI*USTHT*RXE - UCTHT*RZE
C
24 VSCUND = 1117.3 - .00392*RH
VMTE = VMACH*VSOUND
VMXY = VMTE*CCSO(BALPHA - BTHTO)
VXE=VMXE+VMXY*CCSO(BALPHY)
VYE=VMYE-VMXY*SINC(BALPHY)
VZE=VMZE-VMTE*SINC(BOTIVE)
C
30 RDELX = RTXE-RXE
RDELY = RTYE-RYE
RDELE = RTZE-RZE
RXO = RXE
RYC = RYE
RZC = RZE
RETURN
END
C**TRANSLATIONAL DYNAMICS MODULE
SUBROUTINE C1
COMMON C(4310)
C
C**INPUT DATA
EQUIVALENCE (C(1621),RAIL )
EQUIVALENCE (C(1627),AGRAV )
EQUIVALENCE (C(1628),CMASS )
EQUIVALENCE (C(1629),ATHRST)
EQUIVALENCE (C(1630),ATURNY)
EQUIVALENCE (C(1631),BCAMT )
EQUIVALENCE (C(1680),OPTND )
EQUIVALENCE (C(1681),ACTIVE )
EQUIVALENCE (C(1751),CRAD )
EQUIVALENCE (C(13504),OPTN4 )
C
C**INPUTS FROM OTHER MODULES
EQUIVALENCE (C( 371),RANGE )
EQUIVALENCE (C( 380),RANGD )
EQUIVALENCE (C(1300),FXBA )
EQUIVALENCE (C(1301),FYBA )
EQUIVALENCE (C(1302),FZBA )
EQUIVALENCE (C(1667),RZ )
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(2000),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(1611),VZE )
EQUIVALENCE (C(1612),RXED )
EQUIVALENCE (C(1615),RXE )

```

```

EQUIVALENCE (C116161,RYED )
EQUIVALENCE (C116191,RYE )
EQUIVALENCE (C116201,RZED )
EQUIVALENCE (C116231,RZE )
EQUIVALENCE (C116441,ATARG)
EQUIVALENCE (C116471,VTARG)
EQUIVALENCE (C116721,B1C,TD)
EQUIVALENCE (C116751,OPSIT )
EQUIVALENCE (C1135011,FMTARG)
EQUIVALENCE (C116441,RTXED )
EQUIVALENCE (C116511,RTXE )
EQUIVALENCE (C116521,RTYED )
EQUIVALENCE (C116551,RTYE )
EQUIVALENCE (C116561,RTZED )
EQUIVALENCE (C116591,RTZE )

C
C**OTHER CLTPUTS
EQUIVALENCE (C116241,AXBA )
EQUIVALENCE (C116251,AYBA )
EQUIVALENCE (C116261,AZBA )
EQUIVALENCE (C116321,VDELX )
EQUIVALENCE (C116331,VDELY )
EQUIVALENCE (C116341,VDELZ )
EQUIVALENCE (C116351,RDELX )
EQUIVALENCE (C116361,RDELY )
EQUIVALENCE (C116371,RDELZ )
EQUIVALENCE (C116301,VCLSHG)
EQUIVALENCE (C116601,VTRE )
EQUIVALENCE (C116611,VTYE )
EQUIVALENCE (C116621,VTZE )
EQUIVALENCE (C116631,VDXB )
EQUIVALENCE (C116641,VDYB )
EQUIVALENCE (C116651,VDEB )
EQUIVALENCE (C116761,ANGX )
EQUIVALENCE (C116771,ANGY )
EQUIVALENCE (C116781,ANGZ )

C
C116471=C116481
C**ADD AERO AND THRUST FORCES TO GET TOTAL ACCELERATION IN BODY AXES
AXBA = FXBA/CMAS5
AYBA = FYBA/CMAS5
AZBA = FZBA/CMAS5

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 + AGRV

C
C** CALCULATE TOTAL MISSILE ACCELERATION IN BODY AXES
VOXB = CFA11*VXED + CFA12*VYED + CFA13*VZED
VOYB = CFA21*VXED + CFA22*VYED + CFA23*VZED
VOZB = CFA31*VXED + CFA32*VYED + CFA33*VZED
IF (AGRAV.LE10.) GO TO 10
ANGX = VOXB/AGRAV
ANGY = VOYB/AGRAV
ANGZ = VOZB/AGRAV

C

```

```

C**INTEGRATE VELOCITIES TO EARTH AXES POSITION
RVEC=VXE
RVEC = VYE
RVEC = VZE
IF (EMTARG) 10,9,10
10 ATARG=ATHRST/EMTARG
GOTO 11
9 ATARG=0.
11 CONTINUE
EPSITC= 0.
IF (VTARG.GT.0.) EPSITC= ATURNI*AGRAV*CRAD/VTARG
C
RTXED = VTXE
RTYED = VTYE
RTZED = VTZE
C
VDELX = VTXE-VXE
VDELY = VTYE-VYE
VDELZ = VTZE-VZE
C
RDELX = RTXED-RXE
RDELY = RTYED-RYE
RDELZ = RTZED-RZE
VCLNG = (RDELX*VDELX+RDELY*VDELY+RDELZ*VDELZ)/RANGE
RETURN
ENC
C**ROTATIONAL DYNAMICS INITIALIZATION MODULE DZIEUL
SUBROUTINE DZIEUL
COMMON C(4310)
DIMENSION IPL (100)
C**NAME
C**INPUTS FROM MAIN PROGRAM
EQUIVALENCE (C(2561)),N 1
EQUIVALENCE (C(2562)),IPL 1
C**STATE VARIABLE OUTPUTS
EQUIVALENCE (C(1703)),CFA11 1
EQUIVALENCE (C(1707)),CFA12 1
EQUIVALENCE (C(1711)),CFA13 1
EQUIVALENCE (C(1715)),CFA21 1
EQUIVALENCE (C(1719)),CFA22 1
EQUIVALENCE (C(1723)),CFA23 1
EQUIVALENCE (C(1727)),CFA31 1
EQUIVALENCE (C(1731)),CFA32 1
EQUIVALENCE (C(1735)),CFA33 1
C**OTHER OUTPUTS
C**NAME
C**INITIAL CALCULATION OF EULER ANGLE MATRIX OF DIRECTION COSINES (CFA)
USPHI = SIN(18PHI0)
UCPHI = COS(18PHI0)
USTHT = SIN(18HT0)
UCTHT = COS(18HT0)
USPSI = SIN(18PSI0)
UCPSI = COS(18PSI0)
CFA11 = UCPSI*UCTHT
CFA12 = USPSI*UCTHT
CFA13 = -USTHT
CFA21 = -USPSI*UCPHI+UCPSI*USTHT+USPHI
CFA22 = UCPSI*UCPHI+USPSI*USTHT+USPHI
CFA23 = UCTHT+USPHI
CFA31 = UCPSI*USTHT+UCPHI*USPSI+USPHI
CFA32 = USPSI*USTHT+UCPHI*UCPSI+USPHI
CFA33 = UCTHT+UCPHI

```

```

C
C**INTEGRATED PARAMETER LIST (IPL) FOR WPD,WOD,WRO,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

```

```

C** RESET ANGULAR RATE DERIVATIVES TO ZERO.

```

```

C(1700) = 0.
C(1704) = 0.
C(1708) = 0.
C(1712) = 0.
C(1716) = 0.
C(1720) = 0.
C(1724) = 0.
C(1728) = 0.
C(1732) = 0.
C(1736) = 0.
C(1740) = 0.
C(1744) = 0.

```

```

RETURN

```

```

END
C** ROTATIONAL DYNAMICS MODULE
SUBROUTINE D7
COMMON C(4310)

```

```

C

```

```

C**DATA INPUTS

```

```

EQUIVALENCE (C(1421),RAIL )
EQUIVALENCE (C(1748),FMIX )
EQUIVALENCE (C(1749),FM1Y )
EQUIVALENCE (C(1750),FM1Z )
EQUIVALENCE (C(1751),CRAO )
EQUIVALENCE (C(13503),OPTN3)
EQUIVALENCE (C(13504),OPTN4 )

```

```

C

```

```

C**INPUTS FROM OTHER MODULES

```

```

EQUIVALENCE (C(1380),RANGO )
EQUIVALENCE (C(1303),FMXBA )
EQUIVALENCE (C(1304),FMYBA )
EQUIVALENCE (C(1305),FMEBA )
EQUIVALENCE (C(1308),RDELGG)

```

```

C

```

```

C**INPUTS FROM MAIN PROGRAM

```

```

C**STATE VARIABLE OUTPUTS

```

```

EQUIVALENCE (C(1700),CFA11C)
EQUIVALENCE (C(1703),CFA11 )
EQUIVALENCE (C(1704),CFA12C)
EQUIVALENCE (C(1707),CFA12 )
EQUIVALENCE (C(1708),CFA13C)
EQUIVALENCE (C(1711),CFA13 )
EQUIVALENCE (C(1712),CFA21C)
EQUIVALENCE (C(1715),CFA21 )
EQUIVALENCE (C(1718),CFA22C)
EQUIVALENCE (C(1719),CFA22 )

```

```

EQUIVALENCE (C(1720),CFA230)
EQUIVALENCE (C(1723),CFA23 )
EQUIVALENCE (C(1724),CFA310)
EQUIVALENCE (C(1727),CFA31 )
EQUIVALENCE (C(1728),CFA320)
EQUIVALENCE (C(1731),CFA32 )
EQUIVALENCE (C(1732),CFA330)
EQUIVALENCE (C(1735),CFA33 )
EQUIVALENCE (C(1736),WPD )
EQUIVALENCE (C(1739),WP )
EQUIVALENCE (C(1740),WGD )
EQUIVALENCE (C(1743),WQ )
EQUIVALENCE (C(1744),WRD )
EQUIVALENCE (C(1747),WR )

C
C**INTEGRATE BODY ANGULAR RATES
WPC = CRAQ*FMYDA/FMIX
55 WQC = (CRAQ*FMYDA*(FMIZ-FMIX)*WP*WR/CRAQ)/FMIX
65 WRC = (CRAQ*FMZDA*(FMIX-FMIZ)*WP*WQ/CRAQ)/FMIZ

C
C**INTEGRATE ATTITUDE DIRECTION COSINES
49 CFA11C=(CFA21*WR-CFA31*WQ)/CRAQ
CFA12C=(CFA22*WR-CFA32*WQ)/CRAQ
CFA13C=(CFA23*WR-CFA33*WQ)/CRAQ
CFA21C = (CFA31*WP-CFA11*WR)/CRAQ
CFA22C = (CFA32*WP-CFA12*WR)/CRAQ
CFA23C = (CFA33*WP-CFA13*WR)/CRAQ
CFA31C = (CFA11*WQ-CFA21*WP)/CRAQ
CFA32C = (CFA12*WQ-CFA22*WP)/CRAQ
CFA33C = (CFA13*WQ-CFA23*WP)/CRAQ
RETURN
END

C
BASIC INPUT SUBROUTINE QINPT1
SUBROUTINE CINPT1
COMMON C(4310)
EQUIVALENCE (C(2800),B)
EQUIVALENCE (K(4310),KK)
EQUIVALENCE (C(3218),ONAME1), (C(3268),ONAME2), (C(3318),ONAME3),
C (C(3328),ONAME4), (C(2361),NOMCO ), (C(2362),MCDNO ),
C (C(3440),NORMNO), (C(3441),RNDMNO), (C(3167),NCGUT ),
C (C(3168),OUTNO ), (C(2461),NOSUB ), (C(2462),SUBNO ),
C (C(2355),IR ), (C(2357),VR ), (C(3339),NOSTAT),
C (C(3338),LCSTAT), (C(3340),STATNO), (C(3066),NCLIST),
C (C(3067),LISTNO), (C(3117),VALLE ), (C(2009),PLOTNO),
C (C(2009),NCPLOT), (C(2325),VARIABLE), (C,K)
EQUIVALENCE (C(1984),NPLOT )
EQUIVALENCE (C(1985),OUTPLT)
DOUBLE PRECISION ALPHA
DIMENSION ONAME(110),ONAME4(10)
DIMENSION LISTNO(50), VALUE(50)
DIMENSION SUBNO(99),IR(2),VR(2)
DIMENSION RNDMNO(50)
DIMENSION B(100)
DIMENSION ALPHA(3),ONAME1(50),ONAME2(50),OUTNO(50),MCDNO(99)
DIMENSION K(4310)
DIMENSION STATNO (100)
DIMENSION VTABLE(2,15)
DIMENSION OUTPLT(15)
REAL PCCAO
INTEGER OUTNO
INTEGER RNDMNO
INTEGER OUTPLT

```

```

      INTEGER $STATNO
      IF (K1+310) 56,55,56
95  KK=0
96  CONTINUE
      JAR = 0
      WRITE(6,31)
31  FORMAT(11P1 INPUT DATA/)
      1 READ(5,2) (IR(1),ALPHA(1),ALPHA(2),ALPHA(3),IR(2),VR(1),VR(2))
      WRITE(6,30) (IR(1),ALPHA(1),ALPHA(2),ALPHA(3),IR(2),VR(1),VR(2))
30  FORMAT(12,3A6,15,5X,1P2E15.7)
      2 FORMAT(12,3A6,15,5X,2E15.9)
      7 IF (IR(1) .NE. 1) GO TO 3
      NCSUB = NCSUB + 1
      SLRNC(NCSUB) = IR(2)
      K1=KK+3510
      C(K1)=IR(2)
      K(K1+1)=2461+NCSUB
      KK=KK+2
      GO TO 1
      3 IF (IR(1) .NE. 2) GO TO 4
      NOMCD = NOMCD + 1
      MCCNC(NOMCD) = IR(2)
      K1=KK+3510
      C(K1)=IR(2)
      K(K1+1)=2361+NOMCD
      KK=KK+2
      GO TO 1
      4 IF (IR(1) .NE. 3) GO TO 5
      L = IR(2)
      C(L) = VR(1)
      K1=KK+3510
      C(K1)=VR(1)
      K(K1+1)=IR(2)
      KK=KK+2
      IF (VR(2) .EQ. 0.) GO TO 1
      NCLIST = NCLIST + 1
      LISTNC(NCLIST) = L
      VALUE(NCLIST) = VR(1)
      K1=KK+3510
      K(K1)=L
      K(K1+1)=3066+NCLIST
      C(K1+2)=VR(1)
      K(K1+3)=3116+NCLIST
      KK=KK+4
      GO TO 1
      5 IF (IR(1) .NE. 4) GO TO 6
      NCCUT = NCCUT + 1
      IF (NCCUT.GT.50) GO TO 1
      CNAME1(NCCUT)=ALPHA(2)
      CNAME2(NCCUT) = ALPHA(3)
      OUTNC(NCCUT) = IR(2)
      K1=KK+3510
      C(K1)=ALPHA(2)
      K(K1+1)=3217+NCCUT
      C(K1+2)=ALPHA(3)
      K(K1+3)=3267+NCCUT
      K(K1+4)=IR(2)
      K(K1+5)=33167+NCCUT
      KK=KK+6
      GO TO 1
      6 IF (IR(1) .NE. 5) GO TO 16
      IF (VR(1) .EQ. 0.) GO TO 17

```

```

      LCSTAT = LCSTAT + 1
17  NCSTAT = NCSTAT + 1
      STATNC(NCSTAT) = IR(2)
      CNAPE3(NCSTAT) = ALPHA(2)
      CNAPE4(NCSTAT) = ALPHA(3)
      GO TO 1
16  IF (IR(1).NE.7) GO TO 19
      NPLCT = NPLCT + 1
      IF (NPLCT.GT.15) GO TO 1
      DO 20 I=1,2
20  VTABLE (I,NPLOT) = ALPHA(I+1)
      CUTPLT(NPLOT) = IR(2)
      GO TO 1
19  CONTINUE
      K(4309) = NCSUB
      K(4308) = NCWCC
      K(4307) = NCLIST
      K(4306) = NCGUT
      K(4305) = NLCS:AT
      K(4304) = NCSTAT
      K(4303) = NPLCT
      K(4302) = NORNCM
      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,I12),2E15.8)
      NORNCM = NORNCM + 1
      RNCNCG(I) = J
      C(I) = Y
      K(I+1) = MAND
      K(I+2) = MIER
      C(I+3) = SIGNO
      K(I+4) = 3510
      K(I+5) = MAND
      K(I+6) = J+1
      K(I+7) = MIER
      K(I+8) = J+2
      C(I+9) = SIGNO
      K(I+10) = J+3
      C(I+11) = Y
      K(I+12) = J
      C(I+13) = J
      K(I+14) = 3440+I
      KK = KK + 10
12  CONTINUE
      RETURN
      END
C  OUTPUT INITIALIZATION SUBROUTINE OUP2
      SUBROUTINE OUP2
      COMMON C(4310),GRAPH
      EQUIVALENCE (C(2017),DTCNT ), (C(13167),NOOUT ), (C(2016),PGCNT ),
C      (C(2014),ITCNT ), (C(2003),PGNT ), (C(2015),CPP ),
C      (C(2018),TAPE ), (C(2019),TAPEND), (C(2013),DCC ),
C      (C(2000),T ), (C(2021),KCONV ), (C(2025),TIME ),
C      (C(2009),PLOTNC), (C(2009),NOPLOT), (C(13168),OUTNO ),
C      (C(2004),PPNT ), (C(2023),OPOINT)
      DIMENSION GRAPH(2,2),TIME(2),OUTNO(50)
      INTEGER PGCNT , DTCNT , OUTNO , OPOINT
      EQUIVALENCE (C(1985),OUTPLT)
      INTEGER OUTPLT

```

```

      DIMENSION OUTPLT(15)
      KCCNV=0
      ITCNT = DCC * 1.0
      PCNT = T-0.000001
      PPNT=PCNT
      PCCAT = 1
      DTENT = INOCUT * 41/5
      IF (ITCNT .GE. 7) GO TO 2
      WRITE(6,6)(I,C(I),C(I+1),C(I+2),C(I+3),C(I+4),C(I+5),C(I+6),I=1,35
      *10,7)
      6 FORMAT(1H1/(15,1PTE15,7))
C
      2 TIME(1)=T
      CPCINT =1
      EC 10 J=1,NOPL0T
      K=CUTPLT(1)
      10 GRAPH(1,J)=C(K)
      RETURN
      END
C
      SUBROUTINE CUPT3
      COMMON C(4310),GRAPH
      EQUIVALENCE (C(3168),OUTNO 1, (C(3218),ONAME1), (C(3268),GNAME2),
      C (C(2017),DTENT 1, (C(3167),NOOUT 1, (C(2216),PCNT 1,
      C (C(2014),ITCNT 1, (C(2003),PCNT 1, (C(2215),CFF 1,
      C (C(2000),T 1, (C(2664),DER 1, (C(2216),TAPE 1,
      C (C(2019),TAPEND), (C(2008),PLOTNG), (C(2209),NPL0T),
      C (C(2005),PPP 1, (C(2004),PPNT 1, (C(2225),TIME 1,
      C (C(2023),CPCINT)
      EQUIVALENCE (C(1965),CUTPLT)
      DIMENSION B(50),CUTN(50),ONAME1(50),ONAME2(50)
      DIMENSION GRAPH(2,2),TIME(2)
      DIMENSION OUTPLT(15)
      INTEGER DTENT,PCNT,OUTNO
      INTEGER CPCINT
      INTEGER CUTPLT
      IF (ITCNT .GT. 6) GO TO 7
      ITCNT = ITCNT + 1
C
      WRITE(6,6)(I,C(I),C(I+1),C(I+2),C(I+3),C(I+4),C(I+5),C(I+6),I=1,35
      *10,7)
      6 FORMAT(1H1/(15,1PTE15,7))
      PCCAT = 1
      7 IF (DER .EQ. DER1) GO TO 8
      DER1 = DER
      WRITE(6,20)T,DER
      20 FORMAT(1H ,5)TIME=F14.7,2X,10HSTEP SIZE=1P619.7)
      8 IF (T .LT. PCNT)GO TO 15
      9 PCNT = PCNT + CFF
      IF (CPCAT .NE. 1) GO TO 3
      1 WRITE(5,2) (ONAME1(I),GNAME2(I), I=1,NOOUT)
      2 FORMAT (1H1,3X,4HTIME,5X,517X,2A6)/(20X,2A6,7X,2A6,7X,2A6,7X,
      12A6,7X,2A61/)
      PCCAT = 2+DTENT * 4
      3 IF (PCCAT .GE. 86) GO TO 1
      DC 4 1 * 1,ACOUT
      J = CUTNO(1)
      4 B(1) = C(J)
      WRITE (6,5) T,B(1), I = 1,NOOUT1
      5 FORPAT (///,F14.7,1P5E19.7/(14X,1P5E19.7))
      PCCAT = PCCAT + DTENT * 4
      13 IF (T .LT. PPNT .OR. NOPL0T .EQ. 0) RETURN

```

```

PPNT=PPNTAPPP
KPCINT =CPOINT +1
IF IKPCINT-3001 14,13,18
13 WRITE (6,14)
14 FORMAT (//71F 9.999 WARNING-PLOTTING ARRAY FILLED-ONLY FIRST 302 P
CPOINTS PLCTEC 9.999,/)
16 CPOINT=KPCINT
TIME (CPOINT)=T
DO 10 J=1,NCPLOT
KACUTPLT(J)
10 GRAPH(CPOINT ,J)=CIM)
18 RETURN
END
SUBROUTINE ZERO
COMMON C143101
EQUIVALENCE (C11984),NPLOT 1
EQUIVALENCE (C12023),CPOINT1
EQUIVALENCE (C12361),NCMOD 1
EQUIVALENCE (C12461),NCSUB 1
EQUIVALENCE (C13066),NCLIST1
EQUIVALENCE (C13167),NCGUT 1
EQUIVALENCE (C13338),LCSTAT1
EQUIVALENCE (C13339),NOSTAT1
EQUIVALENCE (C13440),NCRNOM1
EQUIVALENCE (C12008),PLCTNO1
INTEGER PLCTNO
INTEGER CPOINT
LOSTAT = 0
NCSTAT = 0
NCSUB = 0
NCMOD = 0
NCGUT = 0
NCRACH = 0
NCLIST = 0
CPOINT=0
NPLUT=0
RETURN
END
SUBROUTINE SUBL1
COMMON C143101
EQUIVALENCE (C12461),NCSUB 1, (C12462),SUBNO 1
DIMENSION SUBNO(99)
DO 1 I = 1, NCSUB
J = SUBNO(I)
GO TO (1, 2, 3, 4, 5, 6, 7, 8, 9), J
2 CALL INPT1
GO TO 1
3 CALL DUPT1
GO TO 1
4 CALL STGE1
GO TO 1
5 CALL CNTR1
GO TO 1
6 CALL RACH1
GO TO 1
7 CALL AUXAL
GO TO 1
8 CALL AUXB1
GO TO 1
9 CALL AUXC1
1 CONTINUE
RETURN

```

```

      END
      SUBROUTINE SUBL2
      COMMON C(4310)
      EQUIVALENCE (C(2461),NOSUB ), (C(2462),SUBNO )
      DIMENSION SUBNO(99)
      GO TO 1, NOSUB
      J = SUBNO(1)
      GO TO ( 1, 2, 3, 4, 5, 6, 7, 8, 9 ), J
2     CALL INPT2
      GO TO 1
3     CALL CUPT2
      GO TO 1
4     CALL STGE2
      GO TO 1
5     CALL CNTR2
      GO TO 1
6     CALL RNDM2
      GO TO 1
7     CALL AUXA2
      GO TO 1
8     CALL AUXO2
      GO TO 1
9     CALL AUXC2
1     CONTINUE
      RETURN
      END
      SUBROUTINE SUBL3
      COMMON C(4310)
      EQUIVALENCE (C(2461),NOSUB ), (C(2462),SUBNO )
      DIMENSION SUBNO(99)
      GO TO 1, NOSUB
      J = SUBNO(1)
      GO TO ( 1, 2, 3, 4, 5, 6, 7, 8, 9 ), J
2     CALL INPT3
      GO TO 1
3     CALL CUPT3
      GO TO 1
4     CALL STGE3
      GO TO 1
5     CALL CNTR3
      GO TO 1
6     CALL RNDM3
      GO TO 1
7     CALL AUXA3
      GO TO 1
8     CALL AUXB3
      GO TO 1
9     CALL AUXC3
1     CONTINUE
      RETURN
      END
      SUBROUTINE STGE2
      COMMON C(4310)
      EQUIVALENCE (C(2011),KSTEP ), (C(2020),LCONV ), (C(2021),KCONV )
      KCONV = 0
      LCONV = 0
      KSTEP = 1
      RETURN
      END
      SUBROUTINE STGE3
      COMMON C(4310),GRAPH
      EQUIVALENCE (C(2000),T 3, (C(2001),TF 3, (C(2003),PCNT 1

```

```

EQUIVALENCE JCI(2010),STEP 1, JCI(2011),KSTEP 1, JCI(2020),LCCNV 1
EQUIVALENCE JCI(2021),KCCNV 1, JCI(2561),N 1, JCI(2662),HMIN 1
EQUIVALENCE JCI(2663),HMAX 1, JCI(2664),DER 1, JCI(2765),EL 1
EQUIVALENCE JCI(2865),EU 1, JCI(2965),VAR 1
EQUIVALENCE JCI(1973),KASE 1, JCI(1974),NJ 1, JCI(1975),NPT 1
DIMENSION CER(101) , VAR(101) , EL(100)
DIMENSION EU(100),GRAPH(2,2)
EXTERNAL AUXSUB
CALL C4
IF (ABS(T-TF) .LE. 0.01) GO TO 20
IF ( (TP-T) ,LT. 0.1) GO TO 10
IF (LCCNV .EQ. 2) GO TO 20
IF (LCCNV .EQ. 1) GO TO 10
IF (CER(1),LT.0.)CER(1)=CER(1)+0.5
RETURN
10 IF (CER(1),GT.0.)CER(1)=CER(1)+0.5
KCCNV = KCCNV + 1
IF (KCCNV .GE. 10) GO TO 20
RETURN
20 PCNT = 1.0
C CUMM = MECP(LOCIX),LOC(X4)
C WRITE (6,30) X3, X4
C 30 FORMAT (38H0 RESTART INITIALIZERS, X3 AND X4, ARE 2F11.0)
IF (STEP .EQ. 11) GO TO 40
PRECER = CER(1)
DER(1) = 0.
NJ=N-1
NPT=0
CALL AMRK(AUXSUB)
DER(1) = PRECER
40 CALL LUPT3
KSTEP = 2
RETURN
END
SUBROUTINE AMRK(AUXSUB)
C=SINGLE PRECISION VERSION* INDEPENDENT VARIABLE IN DOUBLE PRECISION
COMMON C(4310),GRAPH
C DOUBLE PRECISION DELT,TME
C DOUBLE PRECISION NEWC(200),NEWPI(200),OLD(200)
C DOUBLE PRECISION DELT,TME
C DOUBLE PRECISION NEWC(200),NEWPI(200),OLD(200)
DIMENSION T(1000)
DIMENSION D(101) ,EL(100) ,EU(100)
DIMENSION V(101),GRAPH(2,2)
C DIMENSION NEWPI(200),OLD(200),NEWC(200)
EQUIVALENCE JCI(2652),HMIN 1, JCI(2663),HMAX 1, JCI(2664),O 1
EQUIVALENCE JCI(2765),EL 1, JCI(2865),EU 1, JCI(2965),V 1
EQUIVALENCE JCI(1971),RITE 1, JCI(1974),NJ 1, JCI(1975),NPT 1
EQUIVALENCE JCI(1973),KASE 1, JCI(1974),NJ 1, JCI(1975),NPT 1
DATA KCOUNT/0/
DATA P1,P2,P3,P4/2.2916667,2.4583333,1.5416667,0.375/
DATA C2,C3,C4/0.7916667,0.2083333,0.0416667/
IF (KASE-GT.0)GO TO 20
N1=NJ
J2=J1+N1
J3=J2+N1
J4=J3+N1
J5=J4+N1
J6=J5+N1
J7=J6+N1
J8=J7+N1
J9=J8+N1

```

```

      KASE=KASE+1
C=NPT,EC,0  ACAMS-MOULTON INTEGRATION MODE
C=NPT,EC,1  RUNGE-KUTTA INTEGRATION MODE
C=NPT,EC,2  BEGINNING ACAMS-MOULTON WITH RUNGE-KUTTA START
20  IF(NPT,EC,1)GO TO 40
    IF(NPT,EC,2)GO TO 30
    IF(SNGL(DEL),NE,(0.5*C(1)))GO TO 30
    IF(KCUNT,LT,3)GO TO 40
    GO TO 200
30  KCUNT=0
    NPT=0
C=START RUNGE-KUTTA INTEGRATION
C=COMPUTE K1
40  DO 50 I=1,N1
    NEWP(I)=V(I+1)
50  CONTINUE
    TME=V(1)
    KCUNT=KCUNT+1
    DO 60 I=1,N1
60  C(1)=C(1)+D(1)*1
C COMPUTE K2
    DELT=0.5*C(1)
    TME=TME+DELT
    V(1)=TME
    DO 70 I=1,N1
    IF(KCUNT,NE,2)GO TO 85
    K1=J9+1
    T(K1)=NEWP(I)
65  V(1)=C(1+1)
    NEWP(I)=NEWP(I)+0.5*OLD(I)
70  V(1+1)=NEWP(I)
    CALL AUXSUB
    DO 80 I=1,N1
80  NEWC(I)=C(1)+D(1)*1
C COMPUTE K3
    DO 90 I=1,N1
    NEWP(I)=NEWP(I)+0.5*INENC(I)-OLD(I)
90  V(1+1)=NEWP(I)
    CALL AUXSUB
    DO 100 I=1,N1
    K2=J7+1
100  T(K2)=C(1)+D(1)*1
C COMPUTE K4
    TME=TME+DELT
    V(1)=TME
    DO 110 I=1,N1
    K2=J7+1
    NEWP(I)=NEWP(I)+T(K2)-0.5*NEWC(I)
110  V(1+1)=NEWP(I)
    CALL AUXSUB
    DO 120 I=1,N1
    K3=J8+1
120  T(K3)=C(1)+D(1)*1
C COMPUTE VALUE OF FUNCTION
    DO 130 I=1,N1
    K2=J7+1
    K3=J8+1
    NEWP(I)=NEWP(I)-T(K2)+0.16666667*
    X(C(1)+NEWC(I)+INENC(I)+T(K2)+T(K2)+T(K3))
130  V(1+1)=NEWP(I)
140  CALL AUXSUB
    DO 150 I=1,N1

```

```

      K5=J1+1
      K0=J2+1
      K1=J3+1
      K2=J4+1
      K3=J5+1
      K4=J6+1
      T(K4)=T(K3)
      T(K3)=T(K2)
      T(K2)=T(K1)
      T(K1)=T(K0)
      T(K0)=T(K5)
      T(K5)=T(1)
150  T(1)=C(1+1)
      RETURN
C ADAMS-MOULTON INTEGRATION
200  KCUNT=KCUNT+1
      DELT=0.5*C(1)
      CC 210 1=1,N1
      K1=J2+1
      K2=J3+1
      K4=J1+1
C COMPUTE Y-PREDICTED
      CLC(1)=NEWP(1)
      NEWP(1)=CLC(1)+C(1)*T(1)+P2*T(K4)+P3*T(K1)+P4*T(K2)
210  V(1)=NEWP(1)
      TME=TIME+C(1)
      V(1)=TME
      CALL AUXSUB
      K5=0
      CC 250 1=1,N1
      K2=J2+1
      K4=J1+1
C COMPUTE Y-CORRECTED
      NEWC(1)=CLC(1)+C(1)*(P4+C(1)+C2+T(1)-C3+T(K4)+C4+T(K2))
      IF (HMIN.EQ.HMAX) GO TO 250
      TEMPM=ABS(SNCL(NEWC(1)-NEWP(1)))
      IF (TEMP.LT.EU(1)) GO TO 240
      IF (RITE.LE.0.0) GO TO 230
      SPTIME=SNCL(TME)
      WRITE(6,220) SPTIME,TEMP
220  FORMAT(1X,14HSTATE VARIABLE=I3.26H EXCEEDED TOLERANCE ERROR
      X7H TIME=F14.7,9H TEMP=,1PE17.0)
230  IF (ABS(SNCL(DELTA)).GE.HMIN) GO TO 270
240  IF (TEMP.LT.EL(1)) K5=K5+1
250  CONTINUE
      IF (K5.LT.N1) GO TO 290
      IF (ABS(C(1)+C(1)).GT.HMAX) GO TO 290
C SET-UP FOR COURLING STEP SIZE
      IF (KCUNT.LE.6) GO TO 290
      CC 260 1=1,N1
      K1=J1+1
      K2=J2+1
      K3=J3+1
      K5=J5+1
      T(1)=T(K1)
      T(K1)=T(K3)
      T(K2)=T(K5)
260  T(K2)=T(K5)
      C(1)=C(1)+D(1)
      KCUNT=4
      DELT=0.5*C(1)
      CC TO 290
C SET-UP FOR HALVING STEP SIZE

```

```

270 IF(KCOUNT.LE.N)GO TO 310
TME=TME-C(11)
V(11)=TME
D(11)=DEL T
DEL T=0.5*D(11)
CC 280 1=L,N1
K1=J1+1
K2=J2+1
K3=J3+1
NEW F(11)=C(11)
V(11)=C(11)
T(K3)=T(K2)+0.5*(T(K1)-T(K2))
T(K2)=T(K1)
280 T(K1)=T(K1)+0.5*(T(11)-T(K1))
KCOUNT=KCOUNT+1
CC TO 200
C INTEGRATION IS FINISHED. SET UP DERIVATIVES AND EXIT.
290 CC 300 1=L,N1
NEW P(11)=NEW C(11)
300 V(11)=NEW C(11)
CC TO 140
C RETURN TO 3RD PRECEDING POINT AND RESTART RK
310 CC 320 1=L,N1
K1=J1+1
NEW F(11)=T(K1)
320 V(11)=T(K1)
TME=TME-A.0=C(11)
V(11)=TME
D(11)=DEL T
CALL AUXSUB
CC TO 30
END
SUBROUTINE AUXI
COMMON C(4310)
EQUIVALENCE (C(2361),NOMOD 1, (C(2362),XMODNO1, (C(2561),N
DIMENSION XPOCNC(99)
N = 1
CC 1 1=L,NOMOD
L=XPOCNC(11)
CC TO 11,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23
1 24,25,26,27,28,29,30,31,32,33,34,35,36,37),L
2 CALL A11
CC TO 1
3 CALL A21
CC TO 1
4 CALL A31
CC TO 1
5 CALL A41
CC TO 1
6 CALL A51
CC TO 1
7 CALL C11
CC TO 1
8 CALL C21
CC TO 1
9 CALL C31
CC TO 1
10 CALL C41
CC TO 1
11 CALL C51
CC TO 1
12 CALL C61

```

```

      GO TO 1
13  CALL C7I
      GO TO 1
14  CALL C8I
      GO TO 1
15  CALL C9I
      GO TO 1
16  CALL C10I
      GO TO 1
17  CALL C11I
      GO TO 1
18  CALL C2I
      GO TO 1
19  CALL C3I
      GO TO 1
20  CALL C4I
      GO TO 1
21  CALL C5I
      GO TO 1
22  CALL G1I
      GO TO 1
23  CALL G2I
      GO TO 1
24  CALL G3I
      GO TO 1
25  CALL G4I
      GO TO 1
26  CALL G5I
      GO TO 1
27  CALL G6I
      GO TO 1
28  CALL S1I
      GO TO 1
29  CALL S2I
      GO TO 1
30  CALL S3I
      GO TO 1
31  CALL S4I
      GO TO 1
32  CALL S5I
      GO TO 1
33  CALL S6I
      GO TO 1
34  CALL S7I
      GO TO 1
35  CALL S8I
      GO TO 1
36  CALL S9I
      GO TO 1
37  CALL S10I
1  CCATINUE
   RETURN
   END
SUBROUTINE AUXSUB
COMMON C(4310)
EQUIVALENCE (C(2000),Y      ), (C(2361),NOMOD ), (C(2362),XMCND0)
EQUIVALENCE (C(2561),V      ), (C(2562),IPL  ), (C(2664),DER  )
EQUIVALENCE (C(2965),VAR    )
EQUIVALENCE (C(2020),LCONV)
DIMENSION  CER(101)      , VAR(101)      , IPL(100)
DIMENSION  XMOCND(99)
DO 50 I = 2, N

```

```

      J = IPL(I-1)
50 C(I,J) = VAR(I)
      T = VAR(I)
      CC 1 J=1,NOMOD
      IF (LCCNV.EQ.2) RETURN
      L = XPCNC(I)
      GO TO (142,314,506,708,910,1112,1314,1516,1718,1920,2122,
123,2425,2627,2829,3031,3233,3435,3637),L
2 CALL A1
  GO TO 1
3 CALL A2
  GO TO 1
4 CALL A3
  GO TO 1
5 CALL A4
  GO TO 1
6 CALL A5
  GO TO 1
7 CALL C1
  GO TO 1
8 CALL C2
  GO TO 1
9 CALL C3
  GO TO 1
10 CALL C4
  GO TO 1
11 CALL C5
  GO TO 1
12 CALL C6
  GO TO 1
13 CALL C7
  GO TO 1
14 CALL C8
  GO TO 1
15 CALL C9
  GO TO 1
16 CALL C10
  GO TO 1
17 CALL C1
  GO TO 1
18 CALL C2
  GO TO 1
19 CALL C3
  GO TO 1
20 CALL C4
  GO TO 1
21 CALL C5
  GO TO 1
22 CALL G1
  GO TO 1
23 CALL G2
  GO TO 1
24 CALL G3
  GO TO 1
25 CALL G4
  GO TO 1
26 CALL G5
  GO TO 1
27 CALL G6
  GO TO 1
28 CALL S1
  GO TO 1

```

```

29 CALL S2
GO TO 1
30 CALL S3
GO TO 1
31 CALL S4
GO TO 1
32 CALL S5
GO TO 1
33 CALL S6
GO TO 1
34 CALL S7
GO TO 1
35 CALL S8
GO TO 1
36 CALL S9
GO TO 1
37 CALL S10
1 CONTINUE
DO 60 I = 1, 2, N
J = IPL(I-1)
60 DER(I) = C(I)
RETURN
END
SUBROUTINE RESEF
COMMON C(4310)
EQUIVALENCE(C(4310),KK),(C,K)
EQUIVALENCE(C(2066),NOLIST),(C(3067),LISTNO),(C(3117),VALUE)
EQUIVALENCE(C(2900),NORMNO),(C(3441),RNDMNO)
EQUIVALENCE(C(2800),B)
DIMENSION K(4310)
DIMENSION LISTNO(50) , VALUE(50)
DIMENSION B(100)
DIMENSION RNDMNO(50)
C(331)=1.3509
33 C(1)=0.
JC=3504KK
DC(31)=3510,JC,2
63 K(K(1)+1)=K(1)
K(2461)=K(4309)
K(2361)=K(4308)
K(3066)=K(4307)
K(3167)=K(4306)
K(3338)=K(4305)
K(3339)=K(4304)
K(1984)=K(4303)
K(3440)=K(4302)
K(2012)=C(2010)
K(1975)=2
RETURN
END
C DUMMY SUBROUTINE
SUBROUTINE DUMMY
C ENTRY A1
ENTRY A11
C ENTRY A2
ENTRY A21
C ENTRY A3
ENTRY A31
C ENTRY A4
ENTRY A41
ENTRY A5
ENTRY A51

```

C	ENTRY C1
C	ENTRY C1I
	ENTRY C2
	ENTRY C2I
	ENTRY C3
	ENTRY C3I
C	ENTRY C4
	ENTRY C4I
	ENTRY C5
	ENTRY C5I
	ENTRY C6
	ENTRY C6I
	ENTRY C7
	ENTRY C7I
	ENTRY C8
	ENTRY C8I
	ENTRY C9
	ENTRY C9I
	ENTRY C10
	ENTRY C10I
C	ENTRY C1
C	ENTRY C1I
C	ENTRY C2
C	ENTRY C2I
	ENTRY C3
	ENTRY C3I
	ENTRY C4
	ENTRY C4I
	ENTRY C5
	ENTRY C5I
	ENTRY C6
	ENTRY C6I
	ENTRY C7
	ENTRY C7I
C	ENTRY G1
	ENTRY G1I
	ENTRY G2
	ENTRY G2I
C	ENTRY G3
	ENTRY G3I
C	ENTRY G4
	ENTRY G4I
C	ENTRY G5
	ENTRY G5I
	ENTRY G6
	ENTRY G6I
C	ENTRY S1
C	ENTRY S1I
	ENTRY S2
	ENTRY S2I
	ENTRY S3
	ENTRY S3I
	ENTRY S4
	ENTRY S4I
	ENTRY S5
	ENTRY S5I
	ENTRY S6
	ENTRY S6I
	ENTRY S7
	ENTRY S7I
	ENTRY S8
	ENTRY S8I
	ENTRY S9
	ENTRY S9I
	ENTRY S10
	ENTRY S10I


```

      FUNCTION ATAND (Y,X)
      IF (X.EQ.0.,AND.Y.EQ.0.) GO TO 1
      ATANG= 57.29578*ATAN2 (Y,X)
      CONTINUE
      RETURN
      END
      SUBROUTINE TABL2 (X,XI,YI,NX,XX,XLABEL,Y)
      DIMENSION XLABEL (2)
      Y = CODIP2 (X,XI,YI,NX,XX,XLABEL)
      RETURN
      END
      SUBROUTINE TABL2 (X,XI,YI,NX,XX,XINTER,XLABEL,Z)
      DIMENSION XYI (2),XAYI (2),ZI (2),XLABEL (2)
      Z=FCOZ2 (X,Y,XI (1),XAYI (1)+1,ZI,NXY (1),NXY (2),NXY (3),XINTER,
      XLABEL)
      RETURN
      END
      SUBROUTINE TABL2 (X,XI,YI,NX,XX,XINTER,XLABEL,W)
      DIMENSION XYI (4),XAYI (4),ZI (2),XLABEL (2)
      N2I= NXYI (1) + NXYI (2) + 1
      W=FCOZ2 (X,YI,XI (1),XAYI (1)+1,XI (2),ZI,NXYI (3),
      NXYI (4),XINTER,XLABEL)
      RETURN
      END
      SUBROUTINE TIMEVICELT)
      RETURN
      END
      SUBROUTINE WRITEIIA,P,N)
      RETURN
      END
      SUBROUTINE PLOT4 (GRAPH,OPUNT,VARIABLE,TIME,NPLOT4)
      RETURN
      END
      SUBROUTINE PLOT2 (INPLOT2)
      RETURN
      END
      SUBROUTINE PLOT4 (INPLOT4)
      RETURN
      END
      SUBROUTINE CODIN2
      C * * * * *
      C
      C SUBROUTINE CODIN2
      C
      C PURPOSE
      C TO FIT A SET OF POINTS WITH A CONTINUOUS FUNCTION THAT
      C SIMULATES A FRENCH CURVE TYPE CURVE FIT.
      C
      C USAGE
      C Y = CODIP2 (X,XI,YI,N,F,XLABEL)
      C OR
      C Y = CODIM1 (X,XI,YI,N,F,XLABEL)
      C
      C DESCRIPTION OF PARAMETERS
      C X ARGUMENT - INDEPENDENT VARIABLE
      C XI ARRAY OF INDEPENDENT VARIABLE, X
      C YI ARRAY OF DEPENDENT VARIABLE, Y
      C N NUMBER OF POINTS REPRESENTED BY XI AND YI ARRAYS
      C F INTERPOLATION CONTROL
      C LESS THAN ZERO - STRAIGHT LINE INTERPOLATION
      C POSITIVE - END INTERVAL INTERPOLATION
      C 0.0 STRAIGHT LINE
      C 1.0 FULL PARABOLIC

```

```

C          XLABEL  MOLLERITH FIELD OF UP TO 6 CHARACTERS
C
C          REMARKS
C          EXTRAPOLATION IS DONE BY PASSING A STRAIGHT LINE THRU THE
C          TWO POINTS AT THE END INTERVAL.
C          THE ARRAY OF THE INDEPENDENT VARIABLE , XI , MAY BE IN
C          EITHER INCREASING OR DECREASING ORDER.
C
C .....
C
FUNCTION COCIM2 ( X , XI , YI , N , F , XLABEL )

DIMENSION XI(N) , YI(N) , P(2) , E(2) , IS(4,2) , XLABEL(2)
LOGICAL CUT
DATA IS / -1, 0, -2, -1, 0, 1, -1, 0 /

100 OUT = .FALSE.
    N1 = N
    XX = X
    J = 1
    IF ( N1 - 2 ) 150 , 1200 , 300
150 CALL TERROR (XLABEL)
200 COCIM2 = YI(J)
    RETURN

300 KPL = 1
    KPU = 2
    IF ( XI(1) - XI(2) ) 400 , 150 , 600
400 CC = 500 J = 1 , N1
    IF ( XX - XI(J) ) 900 , 200 , 500
500 CONTINUE
    GO TO 400
600 CC = 700 J = 1 , N1
    IF ( XI(J) - XX ) 900 , 200 , 700
700 CONTINUE

800 J = N1
    CALL TERROR (XLABEL)
    GO TO 1300

900 CUT = F .LT. 0.0
    IF ( J - 2 ) 1200 , 1000 , 1100
1000 KPL = 2
    GO TO 1500
1100 IF ( J - N1 ) 1500 , 1400 , 1300
1200 J = 2
1300 CUT = .TRUE.
1400 KPU = 1
1500 AL = ( XX - XI(J-1) ) / ( XI(J) - XI(J-1) )
    COCIM2 = AL * YI(J) + ( 1.0 - AL ) * YI(J-1)
    IF ( CUT ) RETURN

    CC = 1800 KP = KPL , KPU
    P(KP) = 0.0
    CC = 1600 K = 1 , 3
    JO = J + KP + K - 4
    IF (JO) 1900, 1550, 1900
1550 XC = 0.
    YC = 0.
    GOTO 1950
1900 YC = XI(JO)

```

```

      VC=VI(J0)
1950 J1=J+IS(K,KP)
      J2 = J + IS(K+1,KP)
1600 F(KP) = PI(KP) + YII * ( XX - XI(J1) ) / ( XO - XI(J1) )
      * ( XX - XI(J2) ) / ( XO - XI(J2) )
      IF ( KPL .NE. KPU ) GO TO 1700
      J1 = 3 - KPL
      PI(J1) = COCIM2 + F * ( PI(KP) - COCIM2 )
      E(J1) = ABS ( PI(J1) - COCIM2 )
1700 E(KP) = ABS ( PI(KP) - COCIM2 )
1800 CONTINUE

      IF ( E(1) + E(2) .EQ. 0.0 ) RETURN
      COCIM2 = ( ( E(1) * AL ) + P(2) + ( E(2) * ( 1.0 - AL ) )
      * P(1) ) / ( ( E(1) * AL ) + ( E(2) * ( 1.0 - AL ) ) )

      RETURN
END
2-DIMENSIONAL INTERPOLATION SUBPROGRAM...FCCDM2
C
C CALLING SEQUENCE -
C Z = FCCDM2(X,Y,XI,YI,ZI,NXD,NY,NX,XX,XLABEL)
C
C X = ARGUMENT - 1ST VARIABLE
C Y = ARGUMENT - 2ND VARIABLE
C XI = ARRAY OF 1ST VARIABLE
C YI = ARRAY OF 2ND VARIABLE
C ZI = ARRAY OF DEPENDENT VARIABLE
C NXD = DIMENSIONED SIZE OF XI ARRAY
C NY = NUMBER OF VALUES IN ARRAY YI
C NX = NUMBER OF VALUES IN ARRAY XI
C XX = END INTERVAL INTERPOLATION CONTROL CONSTANT
C XLABEL = COLLERITH FIELD OF UP TO 6 CHARACTERS
C
C THIS ROUTINE DIFFERS FROM FCCDM2 IN THAT THE ZI ARRAY DOES NOT
C HAVE TO BE PACKED - I.E., IT DOES NOT HAVE TO OCCUPY CON-
C SECUTIVE LOCATIONS IN CORE, AND IN THAT EITHER OR BOTH THE
C XI AND YI ARRAYS MAY BE IN ASCENDING OR DESCENDING ORDER.
C
C FUNCTION FCCDM2(X,Y,XI,YI,ZI,NXD,NY,NX,XX,XLABEL)
C
C DIMENSION XI(1), YI(1), ZI(NXD,1), T(4), XLABEL(2)
C
C IF (NY.GT.4) GO TO 120
C N3=1
C N3 IS THE INDEX NUMBER OF THE FIRST Y CURVE TO BE USED
C N4=NY
C N4 IS THE COUNT OF THE NUMBER OF Y CURVES TO BE USED
C GO TO 200
C
C 120 N4=4
C IF (YI(1)-YI(2)) 130,150,133
C 130 GO 132 K=1,NY
C IF (Y-YI(K)) 150,150,132
C 132 CONTINUE
C GO TO 140
C
C 133 GO 134 K=1,NY
C IF (Y(XI)-Y) 150,150,134
C 134 CONTINUE
C 140 N3=NY-3

```

```

      GO TO 200
C
150 IF (K-31 155,155,160
155 N3=1
      GO TO 200
C
160 IF (K-NY) 165,160,140
165 N3=K-2
C
200 L=N3
      DO 300 I=1,N4
          T(I)=CCOIM2(X,XI,ZI(I,L),NX,NX,XLABEL)
300   L=L+1
C
      FCCCNZ=CCOIM2(Y,YI(N3),T,N4,NX,XLABEL)
      RETURN
      END
3-DIMENSIONAL INTERPOLATION SUBPROGRAM....FCCOON3
C
      CALLING SEQUENCE -
C
C      W = FCCCN3(X,Y,Z,XI,YI,ZI,WI,NZ,NY,NX,NX,XLABEL)
C      X = ARGUMENT - 1ST VARIABLE
C      Y = ARGUMENT - 2ND VARIABLE
C      Z = ARGUMENT - 3RD VARIABLE
C      XI = ARRAY OF 1ST VARIABLE
C      YI = ARRAY OF 2ND VARIABLE
C      ZI = ARRAY OF 3RD VARIABLE
C      WI = ARRAY OF DEPENDENT VARIABLE
C      NZ = NUMBER OF POINTS IN ZI ARRAY
C      NY = NUMBER OF POINTS IN YI ARRAY
C      NX = NUMBER OF POINTS IN XI ARRAY
C      XX = END INTERVAL INTERPOLATION CONTROL CONSTANT (0.0 TO 1.0)
C      XLABEL = COLLERITH FIELD OF UP TO 6 CHARACTERS
C
      FCCCN3 DIFFERS FROM FCCOON3 IN THAT THE WI ARRAY DOES NOT NEED
      TO BE PACKED, I.E., WI NEED NOT OCCUPY CONSECUTIVE LOCATIONS
      IN CORE, AND ANY OR ALL ARRAYS MAY BE IN EITHER ASCENDING OR
      DESCENDING ORDER.
C
      FUNCTION FCCCN3(X,Y,Z,XI,YI,ZI,WI,NZ,NY,NX,NX,XLABEL)
C
      DIMENSION XI(1), YI(1), ZI(1), WI(1,1,1), T(4), XLABEL(2)
C
      IF (NZ.GT.4) GO TO 120
      N4=1
      N5=NZ
      GO TO 200
C
120 N5=4
      IF (ZI(1)-ZI(2)) 130,150,133
130   DO 132 K=1,NZ
          IF (Z-ZI(K)) 150,150,132
132   CONTINUE
      GO TO 140
C
133   DO 134 K=1,NZ
          IF (ZEIKI-Z) 150,150,134
134   CONTINUE
140   N4=NZ-3
      GO TO 200
C

```

```

150 IF (K.GT.3) GO TO 160
    M4=1
    GO TO 200
C
160 IF (K.G6.NZ) GO TO 140
    M4=M-2
C
200 L=4
    CO 300 I=1,N5
    F(1)= FCOCN2 (X,Y,XI,YI,WI(1,1,LI,NX,NY,NX,XR,XLABEL)
300 L=L+1
C
    FCCCN3=CCOIM2(I,Z,ZI(M4),T,NZ,XR,XLABEL)
    RETURN
    END

```

1	CURT 2,3	3	
1	STGB 2,3	4	
2	G2-T	23	
2	G3-H	24	
2	G5-H	26	
2	S1-T	28	
2	C1-T	7	
2	C4-T	10	
2	A1-T	2	
2	A3-T	4	
2	A2-H	3	
2	C1-H	17	
2	C-10 M	16	
2	G2-H	18	
3	TF	2001	7.0
3	UCC	2013	7.
3	MPIN	2652	.0075
3	MPAX	2663	.0075
3	CER1	2664	.0025
3	CPIKR	443	1.
3	CRDT	446	.05
3	GSM	450	1.
3	RORLCM	453	100.
3	CRSKR	456	2.
3	VZE	1611	110.
3	VXE	1603	440.
3	BNIVE	1666	-20.
3	RSIANT	1667	9000.
3	VMACH	204	.82789
3	THETA-G	427	.1
3	BALPHA	367	2.3
3	CBIAS	852	1.
3	MLIPC	850	15.
3	MLIPC	851	15.
3	GY	856	6.0
3	GZ	855	3.0
3	TAUY	864	6.0
3	TAUZ	863	6.0
3	TAUL	877	.5
3	BCMAX	1121	20.
3	AFAREA	1306	.267
3	RFLGTH	1307	.586
3	RFXCG	1313	2.75
3	RFZCG	1315	-.00833
3	RLCGC	1421	2.54
3	FMETO	1420	15.4

3	ROCGP	1418	-0.1165	
3	ROCGC	1417	-0.3165	
3	DWP	1416	19.4	
3	DWT	1415	135.	
3	CISP	1414	155.8	
3	VAIRSPD	207	880.	
3	CKSK2	888	6.	
3	GNALGA	1403	1.	
3	RAIL	1317	7.5	
3	RLUG	1316	2.57	
3	CEFGCS	468	3.	
3	CEFRAD	469	7.5	
3	CKACT	1163	1.	
3	TOY2	866	1.	
3	TOLV2	866	9.0	
3	CKSK1	472	1.0	
3	VLAZRP	473	1.00	
3	CKSK2	888	6.	
3	CEA1	2664	0.075	
3	FHIXC	1419	0.234	
3	GZ	855	6.0	
3	AGRAV	1627	32.174	
3	CRAC	1751	57.295778	-0.
3	STEP	2010	2.	-0.
3	BCRE	00001	0.10	1.0
3	RADIUS	00003	0.250	1.0
3	WAND	00002	0.20	1.0
3	AIFAC	11	0.0	
3	AILL	7	1.0	
3	SPLTMC	8	1.0	1.0
3	CPP	2015	1.0	
4	P	1739		
4	PCLL FLAP	1232		
4	PHI MISSILE	352		
4	VX EARTH	1603		
4	X PSL	1615		
4	C	1743		
4	PITCH FLAP	1233		
4	THETA MISSILE	350		
4	VY EARTH	1607		
4	Y PSL	1619		
4	R	1747		
4	YAW FLAP	1234		
4	PSI MISSILE	351		
4	VZ EARTH	1611		
4	Z PSL	1623		
4	ALPHA DEG	367		
4	BETA DEG	368		
4	ALPHA GRIME	369		
4	PHI PRIME	370		
4	AZEA	1626		
4	MACH	204		
4	AIR SPEED	207		
4	PDYMC	203		
4	FTHRST	1410		
4	THETA LCS-V	363		
4	EPS Z	435		
4	THETA GIMBAL	427		
4	C COP	869		
4	PSI LCS-V	364		
4	EPS Y	436		
4	PSI GIMBAL	431		

4	R CCM	870		
4	CELT P COM	875		
4	GAMMAN	357		
4	GAMPAV	358		
4	AYEA	1625		
4	EZ	403		
4	EY	407		
4	RY	405		
4	RZ	404		
6		-0	-0.	-0.
3	RSLANT	1667	SC00.0	
3	CPP	2015	.1	
3	BDIVE	1666	-30.	
4		-0	-0.	-0.
3	CZ	855	5.0	
3	CV	856	5.0	
6		0	0.	
7				
7				

TIME	P	ROLL FLAP	PAI SIGNAL	VA LARM	A PUL
3		PILUM FLAP	INTEA MISSILE	VT LARM	Y DSL
4		YAN FLAP	PSI MISSILE	VT LARM	Z DSL
		BETA DEG	ALPHA PRAP	EMI PRIME	ALFA
		AIN SPLED	PJANG	FIPRST	INTEA LCS-V
EPS 2		INTEA GIMBAL	Q CUR	PSI LUS-V	EPS Y
PSI GIMBAL	R LUM	DELT P COM	GAMMA	GAMMAV	
AYUA	EZ	LT			
0.0079000	0.	9.5511131E+17	0.	8.0000117E+02	-1.0332359E+03
0.	-3.710000E-02	-2.7700000E+01	0.	4.8063052E+02	0.
0.	3.3300071E+10	0.	0.	0.	-5.7033376E+02
2.5902773E+00	0.	2.2922775E+00	0.	0.	-2.848155E+01
9.2851007E+01	9.237457E+02	9.3445328E+02	7.5085000E+02	0.	-1.7273221E+03
-4.8602221E-06	-2.3003000E+00	1.0000000E+00	0.	0.	-1.557332E+15
1.5257373E+15	7.8493975E+15	0.	0.	0.	-2.5550271E+03
0.	-2.5234260E+12	2.4411007E+13	0.	0.	0.
0.0050000	1.0672218E-02	0.4426462E-02	-1.7012325E+01	0.7664300E+04	-3.2193050E+02
2.0094733E+01	1.2103549E+01	2.3160363E+01	2.3160363E+01	0.5349373E+01	-0.5067675E+02
1.0500032E+01	9.7933105E-01	9.3740041E-01	1.3454000E+00	5.1490841E+02	-0.4570060E+02
3.7500204E+01	3.3901273E+02	3.7221272E+02	3.7221272E+02	3.7221272E+02	1.0000000E+01
4.3962432E+01	1.0473430E+03	1.2815006E+03	3.5774230E+03	0.336742E+02	0.
4.4687730E+03	0.1793895E+01	9.9040355E+01	6.2382395E+02	0.897242E+03	0.
3.1627732E+01	1.6671336E+03	0.4074061E+02	-6.5320724E+02	2.3941475E+01	0.
2.8021531E+00	-5.5973035E-02	3.4631930E-02	0.	0.	0.
0.0025000	1.6450703E+03	4.3614036E-02	-4.7510322E+01	9.5005302E+02	-2.6795000E+02
3.5650000E+00	3.2393000E+01	-2.9701793E+01	2.9411722E+00	0.	1.3390000E+02
4.6174471E+00	5.0933034E-02	1.3454000E+00	5.5305305E+02	4.3300610E+02	0.
-6.2253221E-01	-2.8877347E-01	0.8873716E+01	-1.9222346E+02	2.3634750E+01	0.
1.0173307E+00	1.1361500E+03	1.5077355E+03	3.7574305E+03	2.752310E+01	0.
7.033543E+03	-2.004074E+01	7.403133E+01	1.270942E+01	-4.4912193E+03	0.
1.164713E+00	1.7665410E+01	6.3135692E-02	1.4709140E+01	-3.0111540E+01	0.
1.2145337E+01	7.7254010E-02	-4.3077504E+02	0.	0.	0.
0.0000000	1.3020397E+01	-4.6514131E+01	-5.7020000E-02	4.0244302E+03	-7.0001300E+02
2.7611900E+00	-9.4194754E+01	-2.8500000E+01	2.9503094E+01	3.1122375E+01	0.
1.4901230E+00	1.0100000E+01	5.0001111E+01	5.0001111E+01	0.	-0.4170000E+02
3.2700030E-01	3.2697700E-01	4.0010390E+01	4.7593394E+02	-1.4445630E+02	0.
1.0950000E+00	1.2220162E+03	1.7451930E+03	3.0000000E+03	3.0047300E+01	0.
3.7072310E+02	-1.0320310E+00	0.0000000E+00	0.0000000E+00	0.0000000E+00	0.
-4.0367407E+01	-1.3133000E+02	-9.2525557E+03	1.0000000E+01	-3.0273300E+01	0.
-1.0000000E+01	7.7220200E+02	-4.0077500E+02	0.	0.	0.

1.334290000	5.724310000-02	-3.073000000-01	1.10332500003	-0.57733710000
1.334290000	-0.677340000-01	-2.34735000000	6.42377100000	6.42377100000
3.112751700-01	3.47742400000	1.22659000000	6.00047000000	3.76500020000
1.119403000-01	-2.30376000000	3.00370900000	1.64337000000	1.91772530000
1.17820000000	1.31335800003	2.30332110000	3.03053350000	0.63327100000
5.170333000-02	-1.13061000000	3.42411000000	4.35447700000	-2.34047350000
1.35090710000	1.33061300000	3.75721100000	2.45040700000	3.07070000000
1.31500910000	5.12000150000	-2.77720000000	0.	0.
-5.0250000	-3.73562770000-04	-1.13104000000	1.17700000000	-5.45775450000
-2.47700000000	3.73400900000	2.34000000000	6.37333000000	1.1349033500000
1.15700000000	3.10210000000	7.33090000000	6.56427000000	-0.11001740000
1.18374300000	2.70320400000	-2.73277000000	0.01000000000	3.51760930000
1.25570000000	1.40770000000	6.01077000000	0.00000000000	6.69000000000
1.00340000000	4.47910000000	-5.87300000000	3.72250000000	3.30331150000
-0.30007100000	-2.60093000000	-3.51330000000	4.63500700000	-0.06711700000
1.13570000000	2.50021000000	-0.13000000000	0.	0.

[illegible]

TIME	P	ROLL FLAP	PRI MISSILE	VA CARFM	X MSL
	Q	PITCH FLAP	INCL MISSILE	VA CARFM	Y PSL
	R	YAW FLAP	PRI MISSILE	VA CARFM	Z PSL
	ALPHA DEG	DELTA DEG	ALPHA PRIME	PRI PRIME	ALPHA
MACH	AIR SPEED	FOYINC	FINAST	INCL LUS-V	
EPS Z	INCL GUNBAL	Q CUA	PSI LOAN	GAPNAV	
PSI GUNBAL	R LUM	DELTA P LUM	GAPNAV		
AYUA	EZ	EY			
:600000					
	1.472208E-01	4.263014E-02	-2.745539E-01	1.245635E+03	-4.292578E+02
	-2.595331E-04	-1.440344E+00	-2.571773E+01	6.373531E+00	1.712623E+00
	1.236657E+00	6.113285E-01	1.775735E+00	7.772450E+02	-2.517533E+02
	3.113307E-03	-8.115535E-02	0.173479E+02	-8.571279E+01	3.094624E+00
	1.331843E+00	1.460312E+03	2.161447E+03	3.241417E+03	9.388014E+03
	2.101060E-01	-8.408376E-02	1.030099E+01	4.524236E-01	-3.448354E-02
	1.101074E+00	-5.270397E+00	4.300105E+02	2.521319E-01	-3.653093E+01
	7.445093E+00	1.450216E+00	-0.115533E+01	0.	0.
:705000					
	1.726042E+00	-6.414306E-03	-3.151413E-02	1.320512E+03	-2.960231E+02
	3.265730E+00	-1.500365E+00	-2.944543E+01	7.551931E+00	-2.453364E+00
	-1.262535E+00	8.650234E-01	7.506014E+01	7.342740E+02	-1.016047E+02
	3.753213E-01	1.847330E-01	3.134737E+01	2.347333E+01	1.170073E+01
	1.472169E+00	1.527476E+04	2.434906E+03	3.024531E+03	1.453455E+00
	4.521430E-01	-1.509113E-01	1.328111E+01	7.007047E-01	-2.074505E-01
	-3.763031E-01	-9.631050E+03	-6.411061E-03	3.201506E-01	-3.059095E+01
	-1.026439E+01	4.308782E+03	-1.107133E+00	0.	0.
:802900					
	-2.390036E+00	-3.245577E-02	-1.311172E-01	1.327617E+03	-1.571503E+02
	1.170747E+01	-2.934957E+00	-2.683673E+01	7.756165E+00	3.264626E+00
	1.756135E+00	1.319311E+00	7.339410E-01	2.246027E+02	-8.321474E+01
	3.754274E-01	1.537555E-01	9.073729E-01	6.957633E+00	-4.157458E+01
	1.584753E+00	1.653085E+03	3.250315E+03	3.739331E+03	-2.610531E+00
	1.209642E+00	-2.705782E+01	3.544246E+01	1.444606E+04	-7.003443E-01
	-3.702133E-01	-1.364552E+01	-3.242821E+02	3.775837E-02	-3.054291E+01
	-6.561000E+00	9.405071E+03	-5.356720E+00	0.	0.

.9.00000 7.574334e+00 -1.505544e-01 7.759598e-01 4.476333e+00 -1.741355e+01
 -3.026772e+01 -4.603110e+00 2.037353e+01 6.579709e+00 3.946342e+01
 -7.707030e+00 2.360780e+00 2.774211e-01 6.572727e+00 -1.162124e+00
 -2.807052e+00 5.214310e-01 2.718506e+00 3.709273e+01 1.475921e+00
 1.555132e+00 1.737406e+03 3.503673e+03 5.755171e+03 2.730170e+01
 -2.708000e+01 2.317748e+00 3.503673e+03 1.221077e+01 3.420957e+00
 -1.277217e+00 -3.144414e+01 -1.505544e-01 2.718506e+00 -3.026772e+01
 -3.026772e+01 1.76914e+01 -5.367270e+00 0.

OVERFLOW AT RANGE 13 1.76914e+01

BREARLOCK HAS OCCURRED, TIME = .9037, RANGE = 12.7190

MISS DISTANCE = 3.7657003e+00

TIME FINAL = 9.02146423e-01

RM EARLY = -3.144414e+01 YM EARLY = 4.0110152e+00 ZM EARLY = 6.7409462e+00

Y FALPAIR = 5.44440171e+00 Z FALPAIR = 6.75433017e+00

APPENDIX I
COORDINATE TRANSFORMATION FROM BODY TO GIMBAL AXIS SYSTEM

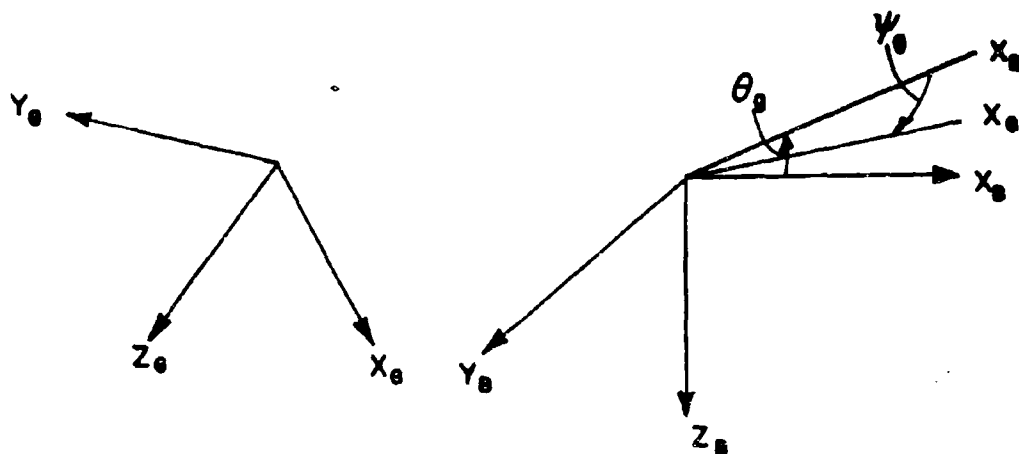


Figure I-1. Angles Between Gimbal and Body Axes

Transformation for Gimbal Pitch Angle (θ_g)

$$\begin{bmatrix} X_B' \\ Y_B' \\ Z_B' \end{bmatrix} = \begin{bmatrix} \cos\theta_g & 0 & -\sin\theta_g \\ 0 & 1 & 0 \\ \sin\theta_g & 0 & \cos\theta_g \end{bmatrix} \begin{bmatrix} X_B \\ Y_B \\ Z_B \end{bmatrix}$$

Transformation for Gimbal Yaw Angle (ψ_g)

$$\begin{bmatrix} X_G \\ Y_G \\ Z_G \end{bmatrix} = \begin{bmatrix} \cos\psi_g & \sin\psi_g & 0 \\ -\sin\psi_g & \cos\psi_g & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_B' \\ Y_B' \\ Z_B' \end{bmatrix}$$

Transformation from Body Axis to Gimbal Axes

$$\begin{bmatrix} X_G \\ Y_G \\ Z_G \end{bmatrix} = \begin{bmatrix} \cos\psi_g & \sin\psi_g & 0 \\ -\sin\psi_g & \cos\psi_g & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta_g & 0 & -\sin\theta_g \\ 0 & 1 & 0 \\ \sin\theta_g & 0 & \cos\theta_g \end{bmatrix} \begin{bmatrix} X_B \\ Y_B \\ Z_B \end{bmatrix}$$

Transformation for both Gimbal Pitch angle and Gimbal Yaw angle for Range Determination is given as:

$$\begin{bmatrix} \text{RXG} \\ \text{RYG} \\ \text{RZG} \end{bmatrix} = \begin{bmatrix} \cos\psi \cos\theta & \sin\psi & -\sin\theta \cos\psi \\ -\sin\psi \cos\theta & \cos\psi & -\sin\psi \sin\theta \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} \text{RXBA} \\ \text{RYBA} \\ \text{RZBA} \end{bmatrix}$$

APPENDIX II
HIGH FREQUENCY ACTUATOR PROGRAM LISTING

C** INITIALIZATION MODULE FOR TIGER SIMPLIFIED ACTUATOR
C*****HIGH FREQUENCY MODEL*****

```

C
  SUBROUTINE C41
  COMMON C(4310)
  DIMENSION IPL(100)
  DIMENSION UDELTA(4)
  EQUIVALENCE (C(1116),NMDELTA)
  EQUIVALENCE (C(1160),DELTPB), (C(1161),DELTOB)
  EQUIVALENCE (C(1162),DELTPB)
  EQUIVALENCE (C(12561),N )
  EQUIVALENCE (C(12562),IPL )
C
  NMDELTA=N
  BDELTA(1) = C(1103)
  BDELTA(2) = C(1107)
  BDELTA(3) = C(1111)
  BDELTA(4) = C(1115)
  BDELTA(1) = BDELTA(1) - DELTPB + DELTOB - DELTRB
  BDELTA(2) = BDELTA(2) - DELTPB + DELTOB + DELTRB
  BDELTA(3) = BDELTA(3) + DELTPB + DELTOB - DELTRB
  BDELTA(4) = BDELTA(4) + DELTPB + DELTOB + DELTRB
  C(1103) = BDELTA(1)
  C(1107) = BDELTA(2)
  C(1111) = BDELTA(3)
  C(1115) = BDELTA(4)
  IPL(N) = 1100
  IPL(N+1) = 1104
  IPL(N+2) = 1109
  IPL(N+3) = 1112
  IPL(N+4) = 1124
  IPL(N+5) = 1128
  IPL(N+6) = 1132
  IPL(N+7) = 1136
  IPL(N+8) = 1140
  IPL(N+9) = 1144
  IPL(N+10) = 1148
  IPL(N+11) = 1152
  N=N+12
  CALL KIKSEY
  RETURN
  END

```

C** TIGER SIMPLIFIED ACTUATOR MODEL
C*****HIGH FREQUENCY MODEL*****

```

C
  SUBROUTINE C4
  COMMON C(4310)
  DIMENSION ACDELTA(4),BDELTA(4),BDELTA(4),VAR(101)
  DIMENSION BDELTP(4),BDELTO(4)
  DIMENSION BDELTA(4)
  DIMENSION FPM(4),BDELTA(4)
C
C**INPUT DATA
  EQUIVALENCE (C(1163),CMAX)
  EQUIVALENCE (C(1309),FPM1), (C(1310),FPM2), (C(1311),FPM3), (C(1312),
  *FPM4)
  EQUIVALENCE (C(1121),RCMAX)
  EQUIVALENCE (C(1122),CACTO)
  EQUIVALENCE (C(1160),DELTPB), (C(1161),DELTOB)
  EQUIVALENCE (C(1162),DELTPB)
C
  BDELTA ARE MEASURED IN RADJANS

```

```

C
C**INPUTS FROM OTHER MODULES
EQUIVALENCE (C(1118),BDELT)
EQUIVALENCE (C(1119),BDELT)
EQUIVALENCE (C(1120),BSURF1)
EQUIVALENCE (C(1121),BSURF2)
EQUIVALENCE (C(1122),BSURF3)
EQUIVALENCE (C(1123),BSURF4)

C
C**STATE VARIABLE OUTPUTS
BDELT(1) = C(1127)
BDELT(2) = C(1131)
BDELT(3) = C(1135)
BDELT(4) = C(1139)
BDELT(1) = C(1103)
BDELT(2) = C(1107)
BDELT(3) = C(1111)
BDELT(4) = C(1115)
PT = 6.28318/360.
FMH(1) = FMH1
FMH(2) = FMH2
FMH(3) = FMH3
FMH(4) = FMH4

C
A1 = 1./15.3/180./180.
A4 = 1.
A3 = .6/180. + 1./16.3
A2 = 1./10./180. + .6/16.3/180.
B1 = CKACT
BT = LPMAX*PT
C**INPUTS FROM MAIN PROGRAM
EQUIVALENCE (C(2965),VAR )
C(1100) = C(1127)
C(1112) = C(1139)
C(1108) = C(1135)
C(1104) = C(1131)
C(1136) = C(1155)
C(1132) = C(1151)
C(1129) = C(1147)
C(1124) = C(1143)

C
C**FLAP DEFLECTION BIAS
BDELT(1) = BDELT(1) - DELTPB + DELTQB - DELTRB
BDELT(2) = BDELT(2) - DELTPB + DELTQB + DELTRB
BDELT(3) = BDELT(3) + DELTPB + DELTQB - DELTRB
BDELT(4) = BDELT(4) + DELTPB + DELTQB + DELTRB

C
C**ACTUATOR DYNAMICS
CC 30 1=1.4
BDELT(1) = BDELT(1)*PT
K = (1-1)/4+1
BDELT(1) = C(1142)*K
BDELT(1) = BDELT(1)
BDELT(1) = -A2/A1*BDELT(1) - A3/A1*BDELT(1) - A4/A1*BDELT(1)
+ BDELT(1)/A1 + B1*FMH(1)/A1

C
C**SURFACE POSITION LIMITER
IF (ABS(BDELT(1)) .LT. BT ) GO TO 30
BDELT(1) = SIGN(BT ,BDELT(1))
J = NDELT + 1
VAR(J) = BDELT(1)
IF (SIGN(1., BDELT(1)) .NE. SIGN(1., BDELT(1))) GO TO 30

```

```

      BCELTC(1) = 0.
      BCELTP(1) = 0.
      J = J + 4
      VAR(J) = BCELTP(1)
30 CONTINUE
C
      BSURF1=BDELT(11)/PT
      BSURF2=BDELT(12)/PT
      BSURF3=BDELT(13)/PT
      BSURF4=BDELT(14)/PT
      C(1103) = BDELT(11)
      C(1107) = BDELT(12)
      C(1111) = BDELT(13)
      C(1115) = BDELT(14)
      C(1140)=BCLTC(11)
      C(1144)=BCLTC(12)
      C(1148)=BCLTC(13)
      C(1152)=BCLTC(14)
C
C**OUTPUT DERIVATIVES OF STATE VARIABLES TO INTEGRATION
      C(1100) = BCELTC(1)
      C(1104) = BCELTC(2)
      C(1108) = BCELTC(3)
      C(1112) = BCELTC(4)
      C(1127)=C(1100)
      C(1139)=C(1112)
      C(1135)=C(1108)
      C(1131)=C(1104)
      RETURN
      END

```

APPENDIX III
HIGH FREQUENCY AUTOPILOT PROGRAM LISTING

```

C** TIGER AUTOPILOT INITIALIZATION MODULE
C*****HIGH FREQUENCY MODEL*****

```

```

SUBROUTINE C11
COMMON C(4310)
OUPPASIGN IFL1001
EQUIVALENCE (C( 839),E2S )
EQUIVALENCE (C( 843),EYS )
EQUIVALENCE (C( 883),EZSS )
EQUIVALENCE (C( 807),EYSS )
EQUIVALENCE (C( 464),CCAMVS)
EQUIVALENCE (C( 465),CCAMPS)
EQUIVALENCE (C(3504),CPTH4 )
EQUIVALENCE (C(0848),NFSUM )
EQUIVALENCE (C(0956),GY )
EQUIVALENCE (C(2561),N )
EQUIVALENCE (C(2562),IPL )

```

```

C
NFSUP = N
IPLIN) = 830
IPL(A+1) = 804
IPL(A+2) = 808
IPL(A+3) = 812
IPL(A+4) = 816
IPL(A+5) = 820
IPL(A+6) = 824
IPL(A+7) = 828
IPL(A+8) = 832
IPL(A+9) = 836
IPL(A+10) = 840
IPL(A+11) = 880
IPL(A+12) = 864
IPL(A+14) = 710
IPL(A+13) = 712
IPL(A+15) = 708
IPL(A+16) = 890
IPL(A+17) = 894
IPL(A+18) = 900
IPL(A+19) = 904
N=N+20

```

```

21 E2S = CCAMVS
EYS = CCAMPS

```

```

22 EZSS = E2S
EYSS = EYS

```

```

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( 839) = 0.

```

```

RETURN

```

```

END

```

```

C** TIGER AUTOPILOT MODULE
C*****HIGH FREQUENCY MODEL*****

```

```

SUBROUTINE C1
COMMON C(4310)
DIMENSION BCELTC(4),VAR(101)

```

```

C
C**INPUT DATA
EQUIVALENCE (C(0850),PLTMO )

```

```

EQUIVALENCE (C(10951),ML14E )
EQUIVALENCE (C( 852),CM1AS )
EQUIVALENCE (C(11211),PM424)
EQUIVALENCE (C( 853),RD1AS )
EQUIVALENCE (C(1914),UR51,(C(1916),UR5D1,(C(1908),ODELPO)
EQUIVALENCE (C(1915),UR51,(C(1912),UR5D1,(C(1911),ODELFC)
EQUIVALENCE (C(1951),EUDCR1,(C(1857),EUDUC1,(C(1503),EVMCR1)
EQUIVALENCE (C(1907),EVNUL1,(C(1890),EUDCR1,(C(1894),EUDUC1)
EQUIVALENCE (C(1900),EVMCR1,(C(1904),EVMUL1)
EQUIVALENCE (C(1005),GZ )
C
C( 857) THRU C( 860) ARE USED BY ECHRL11)
EQUIVALENCE (C(1063),TAU2 )
EQUIVALENCE (C(1064),TAU1 )
EQUIVALENCE (C( 865),TAY1 )
EQUIVALENCE (C( 866),TAY2 )
EQUIVALENCE (C( 877),TAU1 )
C
C**INPUTS FROM OTHER MODULES
EQUIVALENCE (C(10352),BPH1 )
EQUIVALENCE (C(10355),BPH1D )
EQUIVALENCE (C(10402),EZ )
EQUIVALENCE (C(10407),EY )
EQUIVALENCE (C(10048),NFSUM )
EQUIVALENCE (C(11739),WP )
EQUIVALENCE (C(11740),WQ )
EQUIVALENCE (C(11742),WQ )
EQUIVALENCE (C(11744),WRQ )
EQUIVALENCE (C(11747),WR )
EQUIVALENCE (C(11751),CRAD )
C
C**INPUTS FROM MAIN PROGRAM
EQUIVALENCE (C(12000),T )
EQUIVALENCE (C(12965),VAR )
EQUIVALENCE (C(12664),CER )
C
C** STATE VARIABLE OUTPUTS
EQUIVALENCE (C( 800),BPH1SC )
EQUIVALENCE (C( 801),BPH1S )
EQUIVALENCE (C( 804),WQSCD )
EQUIVALENCE (C( 807),WQSP )
EQUIVALENCE (C( 808),WQSC )
EQUIVALENCE (C( 811),WQSC )
EQUIVALENCE (C( 812),WQSCD )
EQUIVALENCE (C( 815),WRSP )
EQUIVALENCE (C( 816),WRSC )
EQUIVALENCE (C( 819),WYS )
EQUIVALENCE (C( 820),ESUMDC )
EQUIVALENCE (C( 821),ESUMD )
EQUIVALENCE (C( 824),ESUMEC )
EQUIVALENCE (C( 827),ESUME )
EQUIVALENCE (C( 828),EZSCD )
EQUIVALENCE (C( 831),EZSP )
EQUIVALENCE (C( 832),EZSC )
EQUIVALENCE (C( 835),EYS )
EQUIVALENCE (C( 836),EYSCD )
EQUIVALENCE (C( 839),EYSP )
EQUIVALENCE (C( 840),EYSC )
EQUIVALENCE (C( 842),EYS )
EQUIVALENCE (C( 843),EYSSD )
EQUIVALENCE (C( 844),EYSS )
EQUIVALENCE (C( 884),EYSSD )
EQUIVALENCE (C( 887),EYSS )

```

```

C
C**OUTPUTS
      EQUIVALENCE (C( 857),BCELTC)
C
C**OTHER OUTPUTS
      EQUIVALENCE (C(10876),EZRR )
      EQUIVALENCE (C(10868),EYRR )
      EQUIVALENCE (C( 855),WCC )
      EQUIVALENCE (C( 870),WRC )
C
C**GUIDANCE SIGNAL SHAPING
      ELCCR=ECCCC
      EVNLM=EVACD
      EZSC = EZSP
      EYSC = EYSP
      EZSCC = TAUZ*(TAUZ*(GZ+EZ - EZS) - 2.*EZSD)
      EYSCC = TAUZ*(TAUZ*(GY+EY - EYS) - 2.*EYSD)
      EZSSC = TAUZ*(EZSC/TAUL + EZS - EZSS)
      EYSSC = TAUZ*(EYSC/TAUL + EYS - EYSS)
C
C**GRAVITY AND RATE BIAS
      WCC = EZSS + GBIAS
      WRC = EYSS + RBIAS
C
C
C**BODY RATE SHAPING AND GYRO DYNAMICS
      WCSC = WCSP
      WRSC = WRSP
      WCCSD = 94.2*(94.2*(WQ - WCS) - 2.*WCSD)
      WRSCC = 94.2*(94.2*(WR - WRS) - 2.*WRSD)
      IF (ABS(WCS) .LE. 30.) GO TO 30
      WCS = SIGN(30., WCS)
      VAR(INFSUM+1) = WCS
      IF (SIGN(1., WCSP) .NE. SIGN(1., WCS)) GO TO 30
      WCSP = C.
      WCSC = 0.
      VAR(INFSUM+2) = WCSP
30  IF (ABS(WRS) .LE. 30.) GO TO 32
      WRS = SIGN(30., WRS)
      VAR(INFSUM+3) = WRS
      IF (SIGN(1., WRSP) .NE. SIGN(1., WRS)) GO TO 32
      WRSP = 0.
      WRSC = 0.
      VAR(INFSUM+4) = WRSP
32  CONTINUE
C
      IF (ABS(WCS).GT.30.) WCS=WCS/80.*WCS
      IF (ABS(WRS).GT.30.) WRS=WRS/80.*WRS
      WCCS=(WCS/80.*WCS)-WCS)*80.
      WRC=(WRS/80.*WRS)-WRS)*80.
C
C**SUMMATION OF RATE DAMPING AND GUIDANCE SIGNALS AND THEIR DERIVATIVES
      EZRR = UCS - WCC
      EYRR = URS - WRC
C
      UKR = .95
      IF (T .LT. TCY2) UKR = 4.25
      IF (T .LT. TCY1) UKR = 0.
C
      ESUMCC = UKR*(EZRR - EYRR)
      ESUMEC = UKR*(EZRR + EYRR)
      IF (ABS(ESUMEC).GT.60.) ESUMEC=SIGN(60.,ESUMEC)

```

```

      IF (ABS(ESUMCC),GT,.001) ESUMCC=SIGN(0.0,ESUMCC)
C
C==TOTAL GUIDANCE SIGNAL SHAPING AND LIMITING
      E1CCC=1000.+(11*ESUMCC/8.+(ESUMC)-E1DCC)
      EVACC=1000.+(11*ESUMCC/8.+(ESUMC)-EVNDR)
      IF (ABS(EVND1),GT,1000.) EVNDR=ESUMC+ESUMCC/8.
      E1DCC=1.0, EVND1
      IF (ABS(E1CCC),GT,1000.) E1DCC=ESUMC+ESUMCC/8.
      E1DCC=1.0, E1CCC
      IF (ABS(EVND1),GT,1000.) EVNDR=SIGN(1000.,EVND1)
      IF (ABS(E1CCC),GT,1000.) E1DCC=SIGN(1000.,E1CCC)
      EVACC=EVNDR
      E1DCC=E1DCC
      IF (ABS(E1DCC),GT,HLIMO) E1DCC = SIGN(HLIMO, E1DCC)
      IF (ABS(EVND1),GT,HLIME) EVNDR = SIGN(HLIME, EVNDR)
C
C==ROLL SIGNAL SHAPING
      UKP = .33
      IF (T.LT,TCY2) UKP = 1.65
      IF (T.LT,TCY1) UKP = 0.
      UPHIS = UKP*(DPHIC/12. + BPHI)
      BPHISC=36.+(UPHIS-BPHIS)
      BDELPT=-(UPHISC/10.+(BPHIS)
      BDELPC=1000.+(BDELPT-BDELPC)
      BDELPC=MAX(10.,
      IF (ABS(BPHIS),GT,BD) BPHIS=SIGN(BD,BPHIS)
      IF (ABS(BDELPC),GT,500.) BDELPC=BDELPT-SIGN(1.5,BDELPC)
      IF (ABS(BDELPC),GT,500.) BDELPC=SIGN(500.,BDELPC)
      IF (ABS(BDELPC),GT,BD) BDELPC=SIGN(BD,BDELPC)
      IF (T.GT,.9),AND(T.LT,.999)
      1=11*(16.6111,C11),C11+1,C11+2,C11+3,C11+4,C11+5,C11+6),1-1.35
      *10.7)
      6 FORMAT(1P1/(15,1P7E15.7))
C
C==AUTOPILOT OUTPUT CURRENTS TO EACH ACTUATOR (FROM SUMMATION AMPS)
      BDELTC(1) = E1DCC - BDELPC
      BDELTC(2) = EVNDR - BDELPC
      BDELTC(3) = E1DCC + BDELPC
      BDELTC(4) = EVNDR + BDELPC
      RETURN
      END

```

Distribution List
Not Filmed

Page 133 & 134

UNCLASSIFIED

Security Classification

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)		2a. REPORT SECURITY CLASSIFICATION
Department of Mechanical Engineering University of Florida Gainesville, Florida		Unclassified
		2b. GROUP
		- - -
3. REPORT TITLE		
CLOSE AIR SUPPORT MISSILE GUIDANCE AND CONTROL STUDY Volume I. Six-Degree-of-Freedom Simulation		
4. DESCRIPTIVE NOTES (Type of report and inclusive dates)		
Final Report (9 December 1970 to 9 December 1971)		
5. AUTHOR(S) (First name, middle initial, last name)		
J. Mahig		
6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
December 1971	142	None
8a. CONTRACT OR GRANT NO.		8b. ORIGINATOR'S REPORT NUMBER(S)
F08635-71-C-0073		
8. PROJECT NO.		8c. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)
670B		
c. Task No. 01		
d. Work Unit No. 010		AFATL-TR-71-169
10. DISTRIBUTION STATEMENT		
Distribution limited to U. S. Government agencies only; ~~this report documents the close air support missile guidance and control study~~ distribution limitation applied December 1971. Other requests for this document must be referred to the Air Force Armament Laboratory (DLWG), Eglin Air Force Base, Florida 32542.		
11. SUPPLEMENTARY NOTES		12. SPONSORING MILITARY ACTIVITY
Available in DDC		Air Force Armament Laboratory Air Force Systems Command Eglin Air Force Base, Florida 32542
13. ABSTRACT		
This report describes in detail a six-degree-of-freedom program which can be used to determine the trajectory and miss distance of a missile system. The options for the program are such as to permit variation of the aerodynamics, seeker, autopilot actuator and missile motor performance for the purpose of accurately simulating a given missile design and evaluating the effects of changes in system parameters. Sufficient detail has been included in the text in order to minimize the users' effort needed to know how to update or modify the program for his purposes.		

DD FORM 1 NOV 65 1473

UNCLASSIFIED

Security Classification

UNCLASSIFIED

Security Classification

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Guided Missile Missile Simulation System Six-Degree-of-Freedom Simulation						

UNCLASSIFIED

Security Classification