UNCLASSIFIED

| AD NUMBER |
| --- |
| **AD894590** |
| LIMITATION CHANGES |
| TO:<br>**Approved for public release; distribution is unlimited. Document partially illegible.** |
| FROM:<br>**Distribution authorized to U.S. Gov't. agencies only; Test and Evaluation; JAN 1971. Other requests shall be referred to Air Force Armament Laboratory (DLWS), Eglin AFB, FL 32542.** |
| AUTHORITY |
| **AFATL ltr, 17 feb 1976** |

THIS PAGE IS UNCLASSIFIED

Best
Available
Copy

AFATL-TR-71-7
VOLUME VI

AD894590

# CLOSE AIR SUPPORT WEAPON ENGINEERING DESIGN STUDY

## VOLUME VI. MISSILE SIMULATION

HUGHES AIRCRAFT COMPANY

TECHNICAL REPORT AFATL-TR-71-7



JANUARY 1971

Distribution limited to U. S. Government agencies only; this report documents test and evaluation; distribution limitation applied January 1971. Other requests for this document must be referred to the Air Force Armament Laboratory (DLWS), Eglin Air Force Base, Florida 32542.

AIR FORCE ARMAMENT LABORATORY
AIR FORCE SYSTEMS COMMAND • UNITED STATES AIR FORCE
EGLIN AIR FORCE BASE, FLORIDA

# Close Air Support Weapon

# Engineering Design Study

## Volume VI. Missile Simulation

DR. W.S. Walker

G.E. Blackshaw

R.W. Knowles

## FOREWORD

(U) This report presents the results of the engineering design study of the close air support weapon (CASW) conducted by Hughes Aircraft Company (HAC), Canoga Park, California, during the period from 23 September 1970 to 22 December 1970 under Contract F08635-71-C-0048 with the Air Force Armament Laboratory, Eglin Air Force Base, Florida. The report consists of six volumes, of which this is Volume VI: Volume I - Management Summary; Volume II - Operational Analysis and Warhead Effectiveness; Volume III - System Analysis; Volume IV - System Design; Volume V - Cost Analysis; and Volume VI - Missile Simulation. The contractor's report number is C2448.

(U) The program monitor for the Armament Laboratory was Mr. Vernon L. Reierson (DLWS). The following contractor personnel from the departments indicated were significant contributors to this report: Operational Analysis - Messrs. J. R. Green, W. N. Bragg, G. G. Latta, P. W. Lindsey, and R. H. Martin; System Analysis: Dr. E. S. Ibrahim and Messrs. J. E. Almanza, D. Berman, L. E. Butts, S. E. Milleman, J. H. Miller, J. B. Stonehouse, and L. Wong; System Design - Dr. R. A. Hubach and Messrs. S. J. Goldberg, A. L. Baker, J. C. Kern, D. N. Perper, M. T. Pett, and H. E. Recher; Cost Analysis - Messrs. A. H. Schlueter, R. C. Hendricks, D. D. Lenhart, and K. E. Rufener.

(U) This technical report has been reviewed and is approved.

RANDALL L. FETTY, Colonel, USAF
Chief, Air-to-Surface Guided Weapons Division

## UNCLASSIFIED ABSTRACT

(U) The objective of the engineering design study of the close air support weapon (CASW) was to provide design considerations for the new close air support missile (CASM). The derivation of the missile was undertaken based on the modification of an existing missile. This study incorporates operational requirement and warhead effectiveness studies for various close air support targets leading to warhead and launch envelope recommendations. A thorough analysis of the system performance and terminal accuracy was conducted. Missile simulation models and a system description, including missile, launcher, avionics, and aerospace ground equipment (AGE) are provided. A cost analysis exercise was conducted for the design, development, test and evaluation (DDT&E) and production of the candidate approach. This report consists of six volumes: Management Summary, Operational Analysis and Warhead Effectiveness, System Analysis, System Design, Cost Analysis, and Missile Simulation.





(The reverse of this page is blank.)

TABLE OF CONTENTS

## LIST OF FIGURES

## LIST OF TABLES




(The reverse of this page is blank.)

## SECTION I

## CLOSE AIR SUPPORT MISSILE SIMULATIONS

### 1.1 INTRODUCTION

(U) The basic simulation tools used for the CAS weapon system analysis included (1) a basic six-degree-of-freedom simulation digital program, (2) a modified version of the six-degree-of-freedom including a Monte Carlo version, and (3) and a simplified adjoint system model (described in Volume III, System Analysis).

### 1.2 SIX-DEGREE-OF-FREEDOM DIGITAL SIMULATION

(U) The basic simulation program used in the performance analysis evaluation of the close air support missile system was a six-degree-of-freedom digital computer program which has been constructed by modifying the simulation of the AGM-65A. The objective of the simulation was to provide a complete and intensively detailed representation of the entire missile system which could be used for final design verification, performance evaluation, and spot checks of parameter optimization results from simpler simulations.

(U) The present simulation represents a highly sophisticated and analytical model of the entire missile system. The simulation was developed using the system analysis by digital simulation using analog methods (SADSAM) programming system. This system is used with the FORTRAN IV compiler language and provides an extensive library of functions and operations which lend themselves well to handling the bookkeeping and computational problems of engineering systems. The computational speed of a simulation developed around this system is much greater than one programmed in a more conventional manner.

(U) Wide use has been made of the 6 DOF program in evaluating the system performance, especially as an analytical tool in defining the miss weighting function as affected by heading error, launch velocity, target motion, motor temperature effects, and seeker drift effects.

(U) A listing of the basic 6 DOF programs used in the study together with system nomenclature and input data coefficients is presented herein. This document represents a complete and comprehensive description of the 6 DOF program, including:

1) Program listing
2) Mathematical model description

   a) Block diagram
   b) Parameter definitions

c) Transfer functions where applicable
d) Program input requirements
e) Program flow charts

(U) Figure 1 illustrates a simplified block diagram of the entire simulation model and indicates the depth and scope that have been included in this simulation package.

(U) One of the specific laser seeker model capabilities include the ability to evaluate the effects of laser spot size image variations which cause a variation in the angle-tracking loop gain. Figure 2 shows a typical seeker gain curve varying with range and spot size that has been modeled. The compensation networks, as indicated in Figure 1, can be placed in the seeker forward loop to increase the stability margin and to reduce degradation in angle-tracking response resulting from the effects of spot size growth.

## 1.3 MONTE CARLO SIMULATION

(U) Paralleling the approach used for AGM-65A performance evaluation, a six-degree-of-freedom digital computer simulation incorporating Monte Carlo techniques has been developed for the close air support missile concept formulation study. For any given set of launch conditions against a particular target, there will be some statistical variation of the miss distance. This results from target designation errors, missile parameter tolerances, uncertainties in the launch conditions, and uncertainties in ambient flight conditions such as temperature and winds.

(U) A statistical description of each parameter is stored in the computer and sampled by a Monte Carlo process which randomly selects a value of each parameter within its own distribution. A homing encounter is then run with this set of parameters and with all error sources present, including tracker noise, seeker drift limit cycle, steering unbalance, and target motion. This results in a miss distance. The process is repeated many times, each with a new set of parameters selected by the Monte Carlo technique. As a result, a distribution of impact points about the aimpoint is obtained, as illustrated in Figure 3.

(U) The Monte Carlo simulation was primarily used to verify results obtained from the adjoint system simulation. The Monte Carlo simulation was used to check the wind disturbance effects obtained from the adjoint. It has also been used to simulate and verify random noise disturbance occurring from tracker noise or designation signal noise.

(U) The results from these simulations are reported in Volume III, System Analysis.

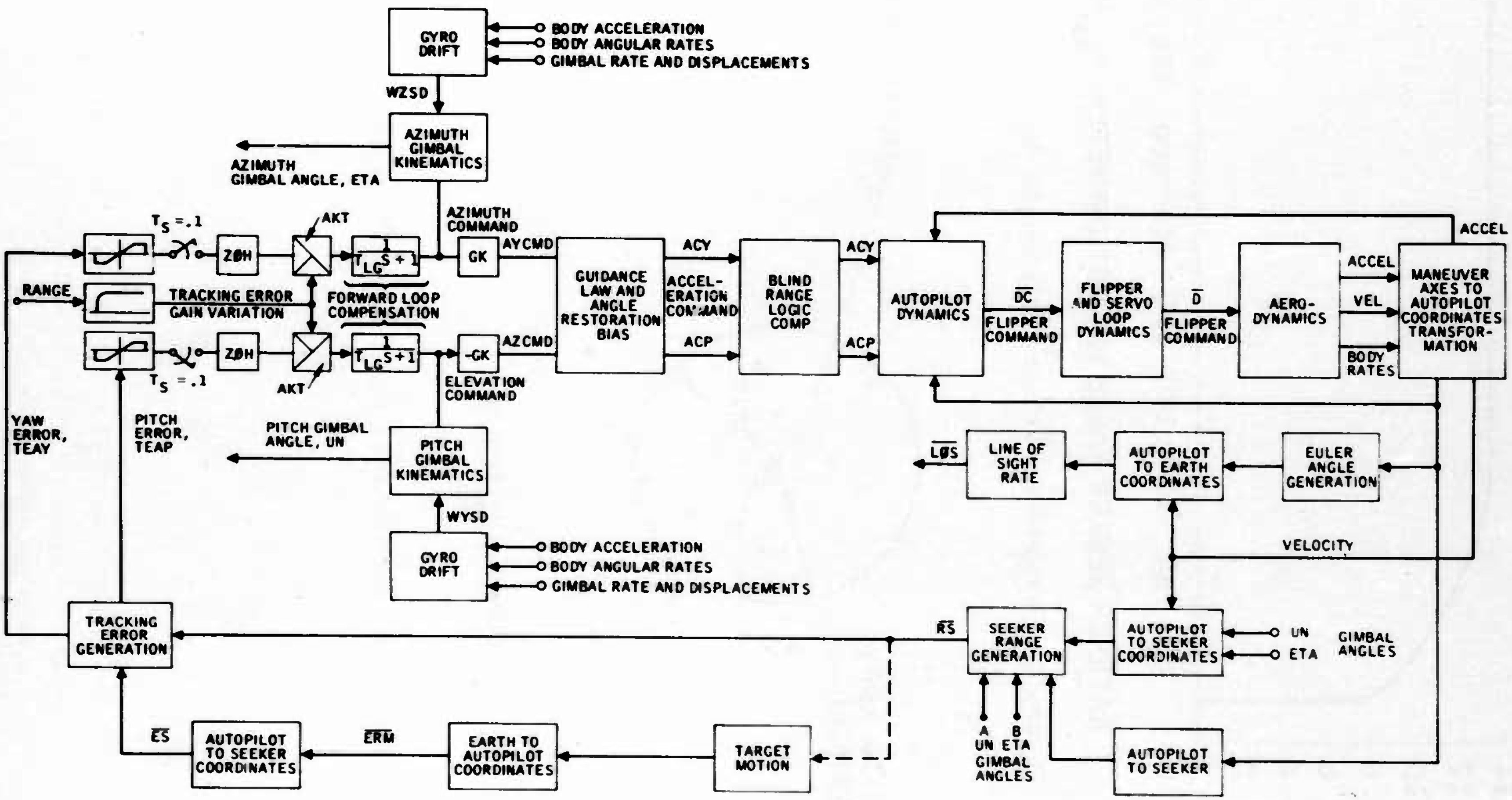



Figure 1. Close Air Support Missile Six-Degree-Of-Freedom Simulation, Simplified Block Diagram

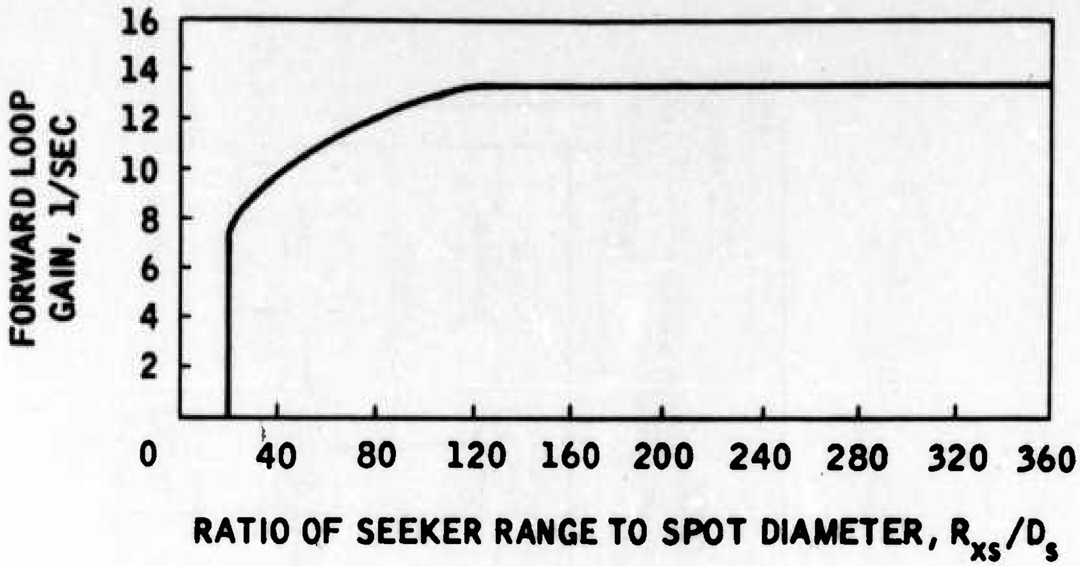



Figure 2. Forward Loop Gain Variation Model

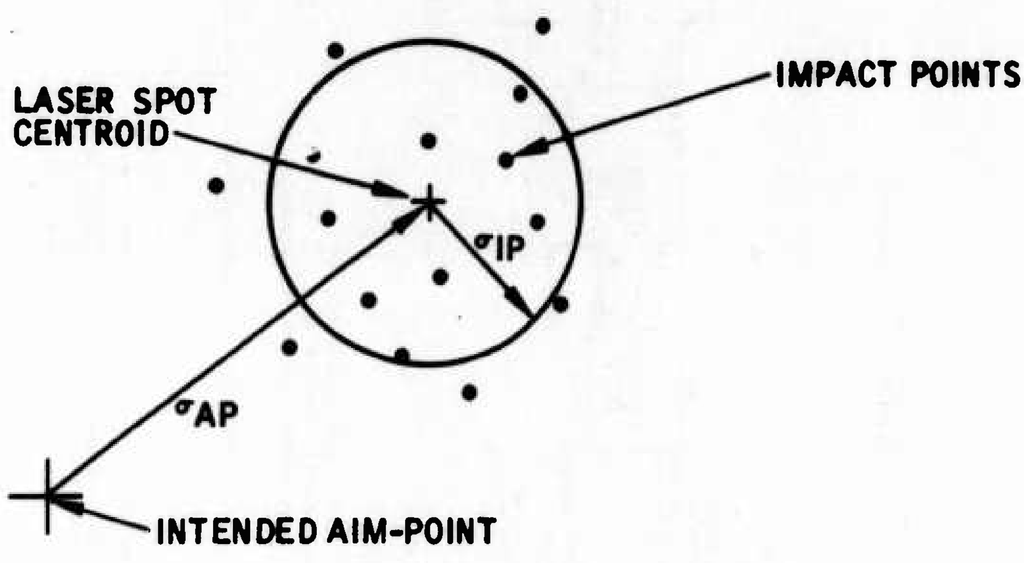



Figure 3. Designation and Impact Point Dispersions

# SECTION II

# SIX-DEGREE-OF-FREEDOM SYSTEM PERFORMANCE DIGITAL MODEL

## 2.1 OBJECTIVES, CONCEPTS, REQUIREMENTS, METHODS, AND TECHNIQUES

### 2.1.1 Objectives

(U) The simulation objectives are to provide a complete and highly detailed simulation of the entire missile system which can be used for final design verification and spot checks of parameter optimization results from simpler simulations.

### 2.1.2 Concepts

(U) This simulation represents the most complete all-analytical model of the entire CAS missile system. A bare minimum of simplifying assumptions are made in subsystem hardware representation. Seeker drift phenomena are included in their entirety. Autopilot transfer functions are not approximated in any way. In short, this simulation is the best overall missile system dynamics model.

### 2.1.3 Requirements

(U) The operation of this program requires the use of an object deck, a data deck, and a library of special functions which constitutes a special computational package called SADSAM. All these inputs must be compatible for use with the GE635 Computer.

### 2.1.4 Methods and Techniques

(U) As previously mentioned, the simulation was developed using a programming system called SADSAM. This system is used with the FORTRAN IV compiler language and provides an extensive library of functions and operations which lend themselves well to handling the bookkeeping and computational problems of many engineering systems. Typical of the available functions or operations are integrations, differentiations, linear transfer functions of almost any order, orthogonal transformations, Euler angle computations, and a variety of non-linear operations such as limiting and backlash. The computational speed of a simulation developed around this system is much greater than one programmed in a more conventional manner.

## 2.2 DESCRIPTION OF WEAPON SYSTEM EQUIPMENT USED

### 2.2.1 Introduction

(U) The dynamic performance of the CAS guided missile has been simulated by a digital computer model. The purpose of this section is to describe this model.

(U) The digital computer simulation model is organized in modules and written in FORTRAN IV compiler language. The CAS simulation model makes use of SADSAM III, a programming technique which is specifically intended for dynamic simulations and which achieves both high dynamic accuracy and high speed operation. In addition, it provides preprogrammed subroutines typical of those used in missile simulations, as an aid to the analyst.

(U) The CAS simulation program is organized into four principal modules: (1) universal seeker, (2) autopilot, (3) control surface, and (4) aerodynamics modules. In addition, there are six other minor or supporting modules incorporated into the simulation. These are (1) initial conditions computer, (2) aimpoint wander (target motion), (3) angle restoration bias (guidance law), (4) blind range filter, (5) track, and (6) gyro. Each of the eight subroutines and the main program are discussed in the paragraphs that follow. Also, there is a description of how the model is used as well as a description of the SADSAM programming technique.

(U) Paragraph 2.2.2 provides a brief overall description of the CAS model.

### 2.2.2 Background

(U) The dynamic model of the CAS missile describes the motion of the missile in three dimensions and makes use of all six degrees of freedom: three positions, three velocities, three attitude angles, and three angular rates. The two vector equations (translation and rotation) applying Newton's Second Law to the rigid missile are rigorously applied, and the kinetic and kinematic behavior of the gimballed seeker is also described in great detail. All significant contributors to seeker drift are represented as well as all significant aerodynamic forces and moments.

(U) The program is arranged in four basic modules describing the seeker, the autopilot, the control surfaces, and the aerodynamics. The modular representation is used because (1) it permits the model to be programmed and checked out more easily; (2) it permits changes and substitutions to be made more easily; and (3) it provides a good correspondence with the actual hardware elements of the missile, so that subsystem specialists can participate in the performance evaluation process in a more direct fashion.

(U) The simulation is exercised by the operator inputing values of all the system constants and setting the initial conditions of the system to the desired values. Due to the extensiveness of the simulation, numerous inputs are required for a complete initial conditions set. These are provided as the output of the initial conditions subroutine in a form convenient for use in the simulation. The input required for this subroutine establishes the missile configuration at the time of launch and is fully described in paragraph 2.3.3. The simulation begins with all dynamic elements at their steady state conditions. The printout interval and maximum simulation time is also input for each run, and at the option of the operator, any or all the system variables may be printed in any sequence.

## 2.3 DESCRIPTION OF DATA USED IN SIMULATION

### 2.3.1 Coordinate Systems

(U) Four different coordinate systems are in use in the simulation: (1) earth, (2) missile body (control surface), (3) autopilot, and (4) seeker. Earth coordinates are simply fixed in inertial space with the missile located at the origin. The Z-axis is vertical downward, and the X-axis is aligned with the ground projection of the original line-of-sight vector. The missile body axes are fixed in the missile with the X-axis aligned longitudinally and the Y- and Z-axes aligned with the control surfaces. Since the control surface orientation is nominally at 45-degree angles with the horizontal and vertical, these axes are also rotated in this manner. The autopilot axes are also fixed in the missile body with the X-axis oriented longitudinally but with the Y- and Z-axes rotated 45 degrees from the missile body axes. The seeker coordinates are fixed to the seeker head and are aligned with the autopilot axes when the seeker gimbal angles are set to zero.

(U) A fifth set of coordinate axes is used in the aerodynamics calculations. These are the maneuver axes which are obtained by rotating the missile body axes about the X-axis through the aerodynamic roll angle. In addition, a sixth coordinate set is also used for miss distance calculations. This miss distance coordinate set may be obtained from the inertial coordinate set simply by a rotation about the Y-axis which aligns the X-axis with the initial line of sight. When miss distance is measured in this coordinate set, it is taken as the missile-to-target distance at the point where the x-component of range reduces to zero.

(U) The coordinate sets described above are listed for convenience in Table I. Figure 4 depicts the Euler angle relationships among the various sets by means of piograms (or resolver chains). With the exception of earth-fixed coordinates, these coordinate systems are also shown graphically in Figure 5.

### 2.3.2 Main Program

(U) Within the simulation, each subroutine or module deals, for the most part, with only a single set of coordinates. Transformations between these coordinates are therefore performed largely in the main or call

TABLE I. SIMULATION COORDINATE SYSTEMS

| Coordinate System | Description |
|---|---|
| Inertial (Earth) | Fixed in earth with the origin at initial missile location. X-axis is horizontal and aligned with ground projection of initial line of sight and positive in direction toward target. Z-axis is vertical and positive downward, Y-axis horizontal and positive in the sense to complete a right-handed system. |
| Miss Distance | Fixed in earth with X-axis aligned with original line of sight and positive in direction toward target. This coordinate set is obtained by a rotation of the earth coordinates about the Y-axis. |
| Autopilot | Fixed in the missile body with Y- and Z-axes at 45 degrees to the planes of the control surfaces, and with the X-axis in the longitudinal axis of the missile, positive in the direction of flight. This set is related to the inertial system by three Euler rotations in the following sequence:<br>Earth-Yaw-Roll-Pitch-Autopilot<br>The positive sense of these rotations is the same as the positive sense of the axes about which the rotations take place. |
| Missile | Fixed in the missile body with the Y- and Z-axes in the planes of the control surfaces. This coordinate set is obtained by a rotation of the autopilot axes through 45 degrees about the positive X-axis. |
| Seeker | Fixed to the seeker head with the X-axis aligned with the boresight. This set is related to the autopilot system by two gimbal rotations in the following sequence:<br>Autopilot-Elevation-Azimuth-Seeker<br>The gimbal rotations, elevation, and azimuth are taken about the nominal autopilot Y- and Z-axes, respectively, with the positive sense of rotation being the same as that of the axis about which it takes place. |

TABLE I. SIMULATION COORDINATE SYSTEMS (CONCLUDED)

| Coordinate System | Description |
|---|---|
| Maneuver | This coordinate system is related to the missile coordinate system but is not fixed in the missile body. The X-axis is aligned with that of the missile set, but the Z-axis is selected so that the missile velocity vector lies in the XZ-plane. The direction of the lateral component of missile velocity fixes the positive direction of the Z-plane. The angle through which the missile axes must be rotated about the negative X-axis to coincide with the maneuver axes is called the aerodynamic roll angle, $\phi_a$. When no lateral component of velocity exists, $\phi_a$ is taken to be 45 degrees. |
| NOTE: All systems are in right-handed rectangular cartesian coordinates. | |

program which serves primarily to direct signal flows among the four functional subroutines and to permit the input and output of data.

(U) A FORTRAN listing of the call program appears in Table II. The flow chart and block diagram for this program appear in Figures 6 and 7, respectively. Table III is a glossary of the terms used in this program. This includes the dimensions and coordinate systems referred to as well as the subscripted variable or constant number used to identify the term.

### 2.3.3 Initial Condition Subroutine (Setic)

(U) The primary purpose of this subroutine is to accept the data which specifies the missile conditions at the time of launch and to convert this data into initial conditions useable by the simulation. Since the values of certain system parameters are also subject to change over a series of simulated trajectories, this subroutine also provides for a common area of data input shared by these parameters and the initial conditions. This common area is the T-array provided by the SADSAM system. The inputs and outputs of this subroutine are shown in Tables IV and V, respectively. The physical relationships between the various input quantities are indicated in Figures 8 through 10. The definition of the output quantities is the same as that of Table III.

# TABLE II. MAIN PROGRAM FORTRAN LISTING

```
$      FORTRAN DECK
CCALL        MAVERICK SIMULATION                                        CALL0010
      COMMON /SSAM/ IEND,ND,TNEXT,VMIN,STPMX,S12345,SUM222
     1,CETA,SETA,CNU,SNU,IMAX,NZ,INV(50),TITLE(250),DEL10,RTITLE(9)
     2,IFGEN,IMFGEN,MFGEN2,IFG2N
      COMMON /SSAM1/ READ,DELT,AUTOT,TIME
      COMMON /SSAM2/ V   (250),T   (250),C   (250)
      COMMON /TRAKER/ COUNT,TR,N1
     1          ,GFFX,GFFY

      EQUIVALENCE                                                       CALL0080
     1 (V( 1),ALT   ),(V(2 ),DAC   ),(V( 3),DPC   ),(V( 4),DYC   ),     CALL0090
     2 (V( 5),DA    ),(V( 6),DP    ),(V( 7),DY    ),(V( 8),VXM   ),     CALL0100
     3 (V( 9),VYM   ),(V(10),VZM   ),(V(11),WX    ),(V(12),WY    ),     CALL0110
     4 (V(13),WZ    ),(V(14),AXM   ),(V(15),AYM   ),(V(16),AZM   ),     CALL0120
     5 (V(17),AZCMD ),(V(18),AYCMD ),(V(19),YAW   ),(V(20),ROLL  ),     CALL0130
     6 (V(21),PITCH ),(V(22),RXS   ),(V(23),RYS   ),(V(24),RZS   ),     CALL0140
     7 (V(25),TEAP  ),(V(26),TEAY  ),(V(27),SEGA  ),(V(28),SAGA  ),     CALL0150
     8 (V(29),RX    ),(V(30),RY    ),(V(31),EP1   ),(V(32),EP2   ),     CALL0160
     9 (V(33),ALPHA ),(V(34),ALPHAP),(V(35),ALPHAY),(V(36),VXE   )      CALL0170
      EQUIVALENCE                                                       CALL0180
     1 (V(37),VYE   ),(V(38),VZE   ),(V(39),Q     ),(V(40),VM    ),     CALL0190
     2 (V(41),AM    ),(V(42),ACP   ),(V(43),ACY   ),(V(44),DWX   ),     CALL0200
     3 (V(45),DWY   ),(V(46),DWZ   ),(V(47),DDAC  ),(V(48),DDPC  ),     CALL0210
     5 (V(49),DDYC  ),(V(50),TSMISS),(V(51),YSMISS),(V(52),ZSMISS),     CALL0220
     6 (V(53),WXP   ),(V(54),WYP   ),(V(55),WZP   ),(V(56),DWXP  ),     CALL0230
     7 (V(57),DWYP  ),(V(58),DWZP  ),(V(59),AXP   ),(V(60),AYP   ),     CALL0240
     8 (V(61),AZP   ),(V(62),VXP   ),(V(63),VYP   ),(V(64),VZP   )      CALL0250
     X,(V(67),RXM   ),(V(68),RYM   ),(V(69),RZM   )                     CALL0260
      EQUIVALENCE                                                       CALL0270
     1(V(66),TOTMIS),(V(70),EMJ    ),(V(71),EMK    ),                   CALL0280
     1 (V(72),AXE   ),(V(73),AYE   ),(V(74),AZE   ),(V(75),ABRH  ),     CALL0290
     2 (V(76),ABRV  ),(V(77),HORBT ),(V(78),VERTBT),(V(79),SIGMAF),     CALL0300
     3 (V(80),XISURF),(V(81),XLOS  ),(V(82),YLOS  ),(V(83),ZLOS  ),     CALL0310
     4 (V(85),DE    ),(V(86),DEXS)
      EQUIVALENCE
     1 (V(110),WMD  ),(V(118),TEYD ),(V(119),TEPD ),(V(120),UMD  ),
     2 (V(121),ETAD ),(V(122),WXD  ),(V(123),WYD  ),(V(124),WZD  ),
     3 (V(125),XLOSD),(V(126),YLOSD),(V(127),ZLOSD),(V(128),ANT  ),
     4 (V(129),YAWD ),(V(130),ROLLD),(V(131),PITD ),(V(132),DED  ),
     5 (V(133),DEXSD), (V(134),DB), (V(135),DE1),(V(137),FLAG)
      EQUIVALENCE
     1(C(109),XK1),    (C(110),XK2),     (C(111),PK1),     (C(112),TAU1),
     2(C(113),TAU2),   (C(114),TB),      (C(115),TC)
      EQUIVALENCE
     1(C(116),W3S),      (C(117),K2T),        (C(118),DUMP),(C(156),PC),
     2 (C(159),DEV(1))
C                                                                       CALL0320
C                                                                       CALL0330
C     THIS MODEL HAS THE MISSILE FLYING AT A 45 DEG. ROLL ANGLE         CALL0340
      SUM222=0.0                                                        CALL0350
      READ=1.0                                                          CALL0360
    6 COUNT=0.0                                                         CALL0370
      CALL LOAD                                                         CALL0380
      TBX=TB                                                            CALL0381
      TCX=TC                                                            CALL0382
      DEL11=TC-TB
      C12=-1.
      B17=0.
```

# TABLE II. MAIN PROGRAM FORTRAN LISTING (CONTINUED)

```
      C1=0.
      B1=0.

C     C(1)   THROUGH C(12)  ARE RESERVED FOR THE MAIN PROGRAM             CALL0390
C     C(13)  THROUGH C(42)  ARE RESERVED FOR THE SEEKER SUBROUTINE        CALL0400
C     C(43)  THROUGH C(72)  ARE RESERVED FOR THE PILO   SUBROUTINE        CALL0410
C     C(73)  THROUGH C(96)  ARE RESERVED FOR THE AERO   SUBROUTINE        CALL0420
C     C(97) THROUGH C(102) ARE RESERVED FOR THE FLIPPER SUBROUTINE        CALL0430
C     C(109)   THROUGH C(120)  ARE RESERVED FOR THE SEEKER SUBROUTINE
C     C(103) - C(108) ARE RESERVED FOR AIM POINT WANDER ROUTINE           CALL0440
C     C(198)-C(250) RESERVED FOR BMAX DATA
C     C(1)=AUTOPILOT G BIAS                                               CALL0450
C     C(2)=AUTOPILOT ACTIVATION DELAY IN SECONDS                          CALL0460
C     C(3)= BLIND RANGE PITCH                                             CALL0470
C     C(4) = C(4)*D/DT(AYCMD) =GATE ANGLE ERROR                           CALL0480
C     C(5) = BLIND RANGE FILTER TIME CONSTANT. SET TO 0.0 TO EXCLUDE BRFCALL0490
C     C(6) = ANGLE RESTORATION GAIN                                       CALL0500
C     C(7) = REFERENCE RESTORATION ANGLE                                  CALL0510
C     C(8) = TIME CONSTANT ANGLE RESTORATION FILTER                       CALL0520
C     C(9) = BLIND RANGE YAW                                              CALL0530
C     V(118)TO V(133) SPECIAL PRINT OUT VARIABLES
      CALL SETIC                                                          CALL0540
      RTOD=57.2957795
  110 I=1
   20 COUNT=COUNT+1.0                                                     CALL0550
      IF (COUNT.GT.2.)   GO TO 304
      N1=1
      TR=0.
      C1Z=C1
      GO TO 330
  304 T1=TIME+DELT
      C1=AINT(T1/TC)
      B1=AINT((T1+DELT1)/TC)
      IF (C1.EQ.C1Z)  GO TO 310
  305 N1=1
      TR=TC
      C1Z=C1
      GO TO 330
  310 IF (B1.EQ.B1Z)  GO TO 320
      N1=0
      TR=TB
      B1Z=B1
      GO TO 330
  320 N1=-1
      TR=AMOD(T1,TC)
  330 CONTINUE
C     ANGLE RESTORATION BIAS                                              CALL0560
      IF (TIME.GT.C(2)) CALL ARB                                          CALL0570
C                                                                         CALL0580
C     BLIND RANGE FILTER                                                  CALL0590
      IF(FLAG.EQ.1.)CALLBRF
      IF (C(5).GT.0.0) CALL BRF                                           CALL0610
C                                                                         CALL0600
   50 CALL MAERO                                                          CALL0620
  120 I=2
```

TABLE II. MAIN PROGRAM FORTRAN LISTING (CONCLUDED)

```
      CALL MPILOT                                                       CALL0630
  130 I=3
      CALL MFLIP                                                        CALL0640
  140 I=4
      CALL R45R(WX,WY,WZ,WXP,WYP,WZP)                                   CALL0650
      CALL EULANG(WXP,WYP,WZP,YAW,ROLL,PITCH)                           CALL0660
      CALL R45R(DWX,DWY,DWZ,DWXP,DWYP,DWZP)                             CALL0670
      CALL R45R(AXM,AYM,AZM,AXP,AYP,AZP)                                CALL0680
      CALL R45R(VXM,VYM,VZM,VXP,VYP,VZP)                                CALL0690
      CALL EULTRN(1,1,VXE,VYE,VZE,VXP,VYP,VZP,YAW,ROLL,PITCH)           CALL0700
      CALL MSEEK                                                        CALL0710
  150 I=5
      CALL SEKTR (0,-1,RXM,RYM,RZM,RXS,RYS,RZS,SEGA,SAGA)               CALL0720
  160 I=6
      CALL EULTRN (0,1,RX,RY,ALT,RXM,RYM,RZM,YAW,ROLL,PITCH)            CALL0730
  170 I=7
      CALL EULTRN(0,1,AXE,AYE,AZE,AXP,AYP,AZP,YAW,ROLL,PITCH)           CALL0740
      RSQ=V(29)**2+V(30)**2+V(1)**2                                     CALL0770
      XLOS=(V(30)*V(38)-V(1)*V(37))/RSQ                                 CALL0780
      YLOS=(V(1)*V(36)-V(29)*V(38))/RSQ                                 CALL0790
      ZLOS=(V(29)*V(37)-V(30)*V(36))/RSQ                                CALL0800
      XLOSD=XLOS*RTOD
      YLOSD=YLOS*RTOD
      ZLOSD=ZLOS*RTOD
      WXD=WX*RTOD
      WYD=WY*RTOD
      WZD=WZ*RTOD
      TEYD=TEAY*RTOD
      TEPD= TEAP*RTOD
      UND=SEGA*RTOD
      ETAD=SAGA*RTOD
      YAWD=YAW*RTOD
      ROLLD=ROLL*RTOD
      PITD=PITCH*RTOD
      DED=DE*RTOD
      DEXSD=DEXS*RTOD
      ANT=SQRT((AYM)**2+(AZM)**2)
      V(65)=S12345                                                      CALL0810
C     IF (TIME.LT.TBX)  GO TO 60                                        CALL082X
C     TBX=TBX+TC                                                        CALL0821
C  60 CALL TTEST(TBX)                                                   CALL0822
      IF (TIME.LT.TCX)  GO TO 70
      TCX=TCX+TC
   70 CALL TTEST(TCX)
      CALL PRINTS(-V(22))
      CALL BMAX
      IF (IEND) 20,1000,1000                                            CALL0840
 1000 WRITE (6,1001) COUNT                                              CALL0850
 1001 FORMAT (1H1,28H TOTAL NUMBER OF ITERATIONS=,1PE15.7)              CALL0860
      GO TO 6                                                           CALL0920
      END                                                               CALL0930
```

(U) A listing of this subroutine is shown in Table VI, and a flow chart appears in Figure 11.

(U) Several points must be clarified regarding two of the inputs to this subroutine. The effective tracker time constant T(13) is used to establish initial tracking error angles only; it is not used thereafter. The assumption is made that, at time of launch, the tracking loop has achieved steady state, so that tracking error is proportional to the product of line-of-sight rate and tracker time constant. This initial error may be eliminated by setting this input to zero. In this case, the performance of the simulation would be otherwise unchanged.

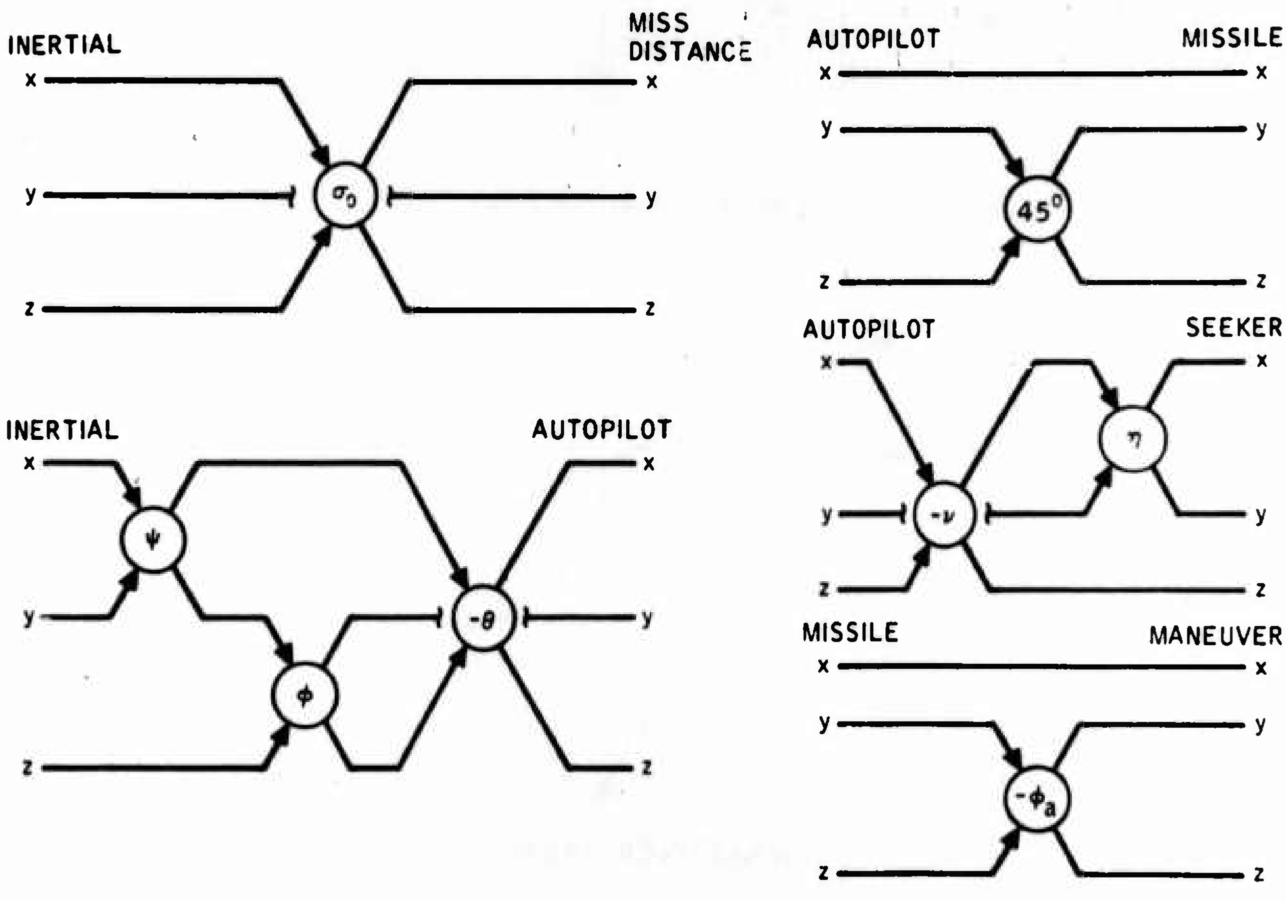



Figure 4. Piograms Showing Euler Angle Relationships Between Coordinate Sets

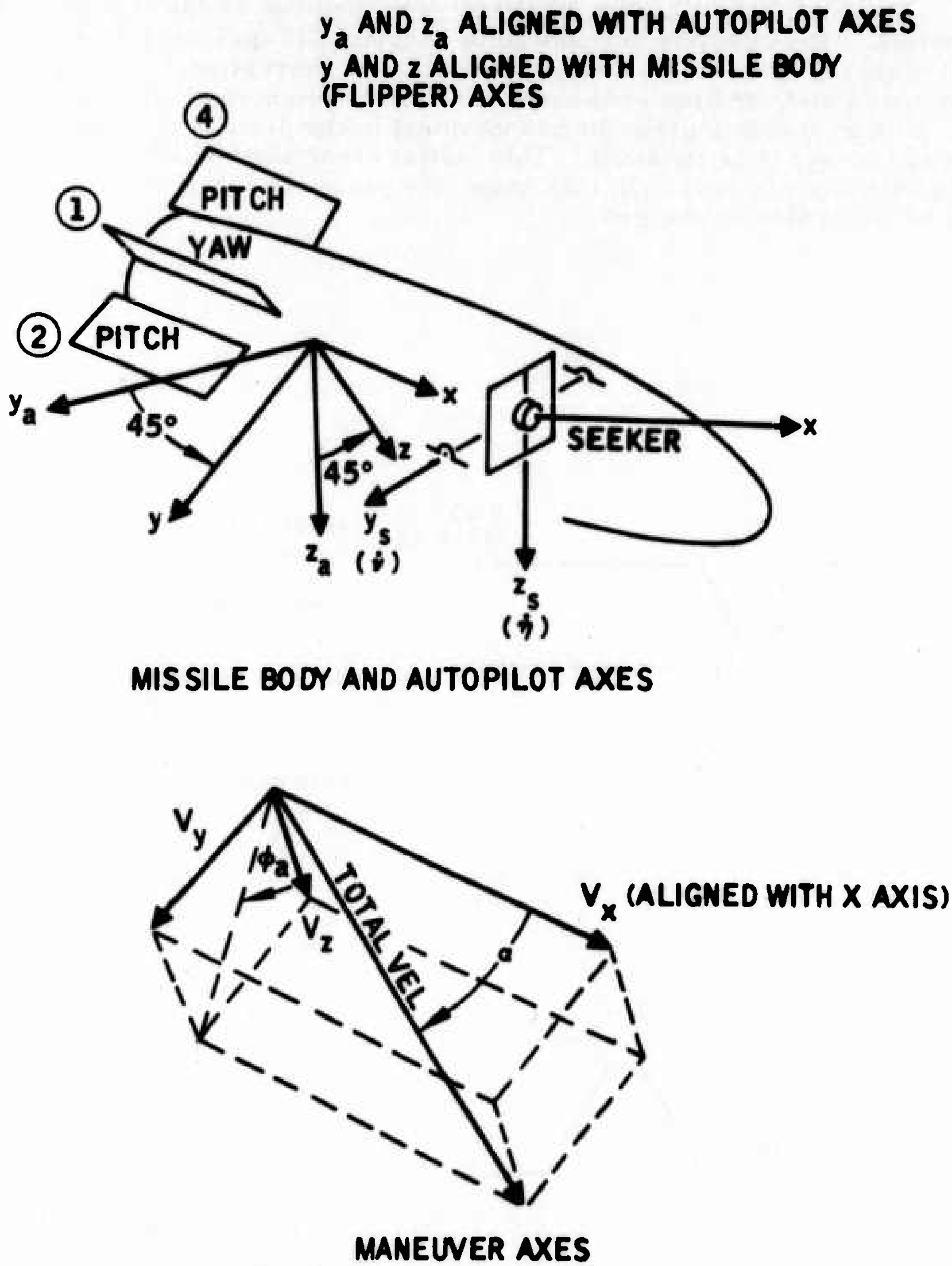



Figure 5. Simulation Coordinate System

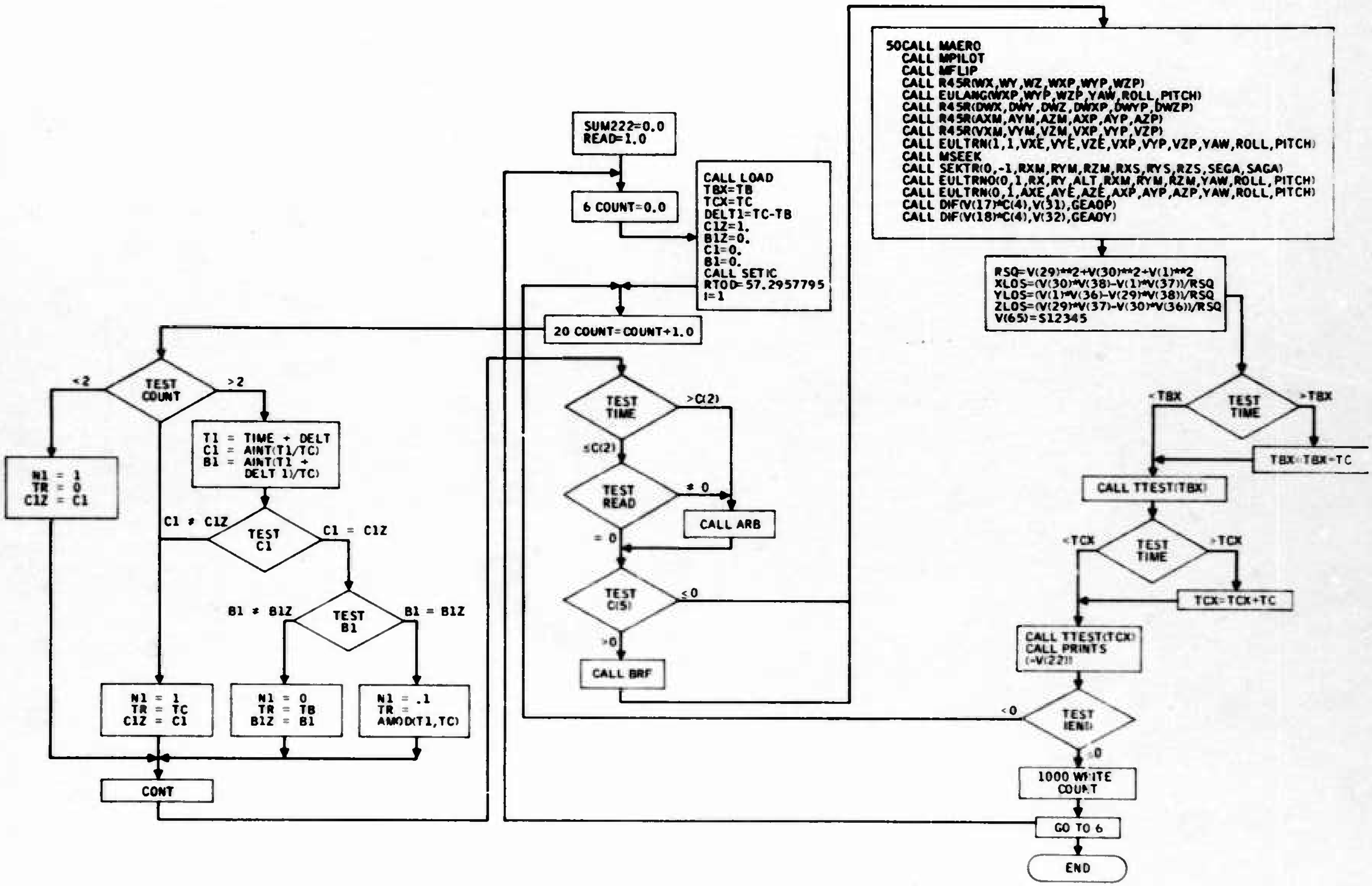
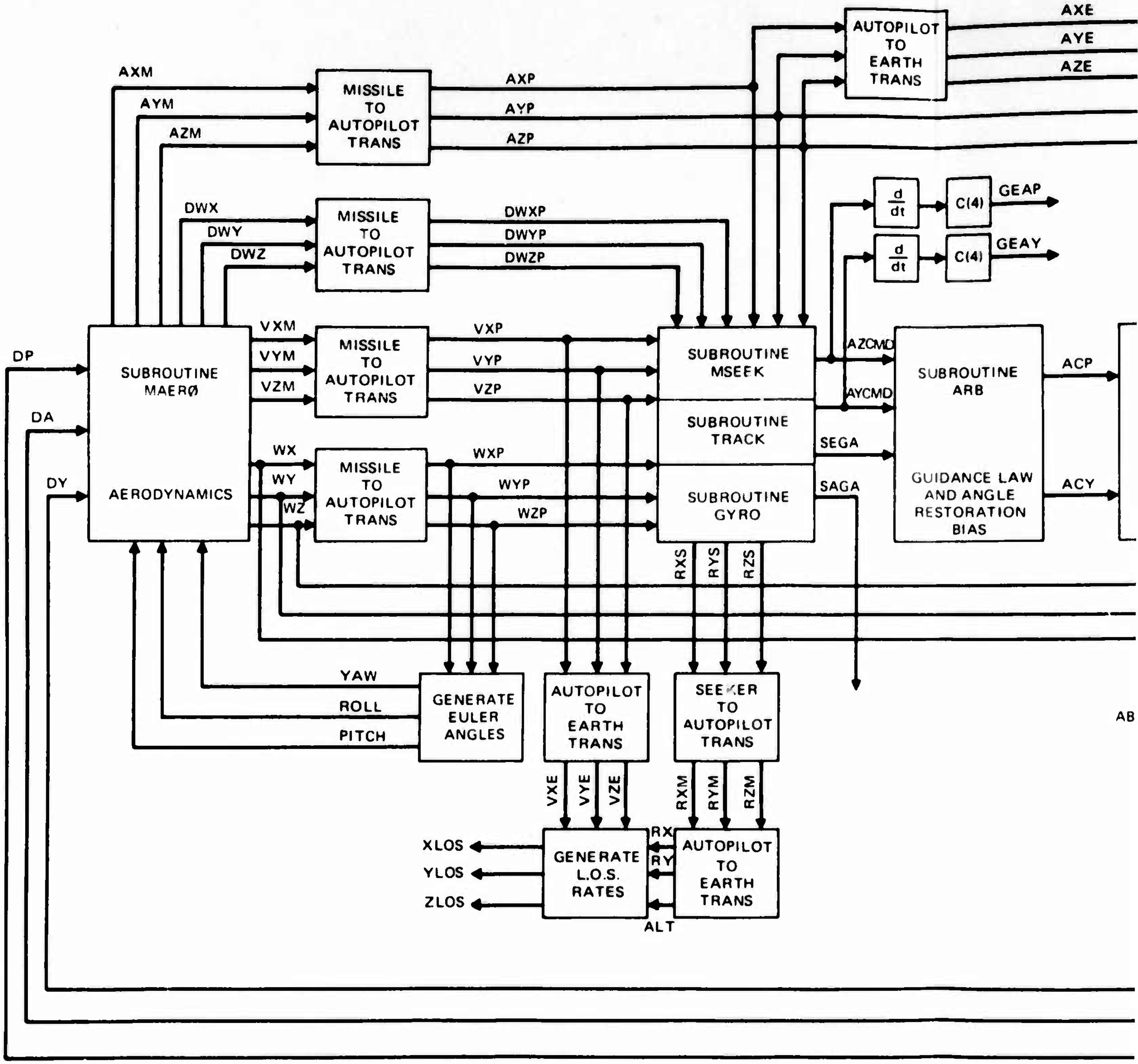



Figure 6. Call Program Flow Chart


AXM
AYM
AZM
MISSILE TO AUTOPILOT TRANS
AXP
AYP
AZP
AUTOPILOT TO EARTH TRANS
AXE
AYE
AZE
DWX
DWY
DWZ
MISSILE TO AUTOPILOT TRANS
DWXP
DWYP
DWZP
d/dt
C(4)
GEAP
GEAY
DP
DA
DY
SUBROUTINE MAERØ
AERODYNAMICS
VXM
VYM
VZM
MISSILE TO AUTOPILOT TRANS
VXP
VYP
VZP
WX
WY
WZ
MISSILE TO AUTOPILOT TRANS
WXP
WYP
WZP
SUBROUTINE MSEEK
SUBROUTINE TRACK
SUBROUTINE GYRO
AZCMD
AYCMD
SEGA
SAGA
SUBROUTINE ARB
GUIDANCE LAW AND ANGLE RESTORATION BIAS
ACP
ACY
RXS
RYS
RZS
YAW
ROLL
PITCH
GENERATE EULER ANGLES
AUTOPILOT TO EARTH TRANS
SEEKER TO AUTOPILOT TRANS
VXE
VYE
VZE
RXM
RYM
RZM
XLOS
YLOS
ZLOS
GENERATE L.O.S. RATES
RX
RY
ALT
AUTOPILOT TO EARTH TRANS
AB

2

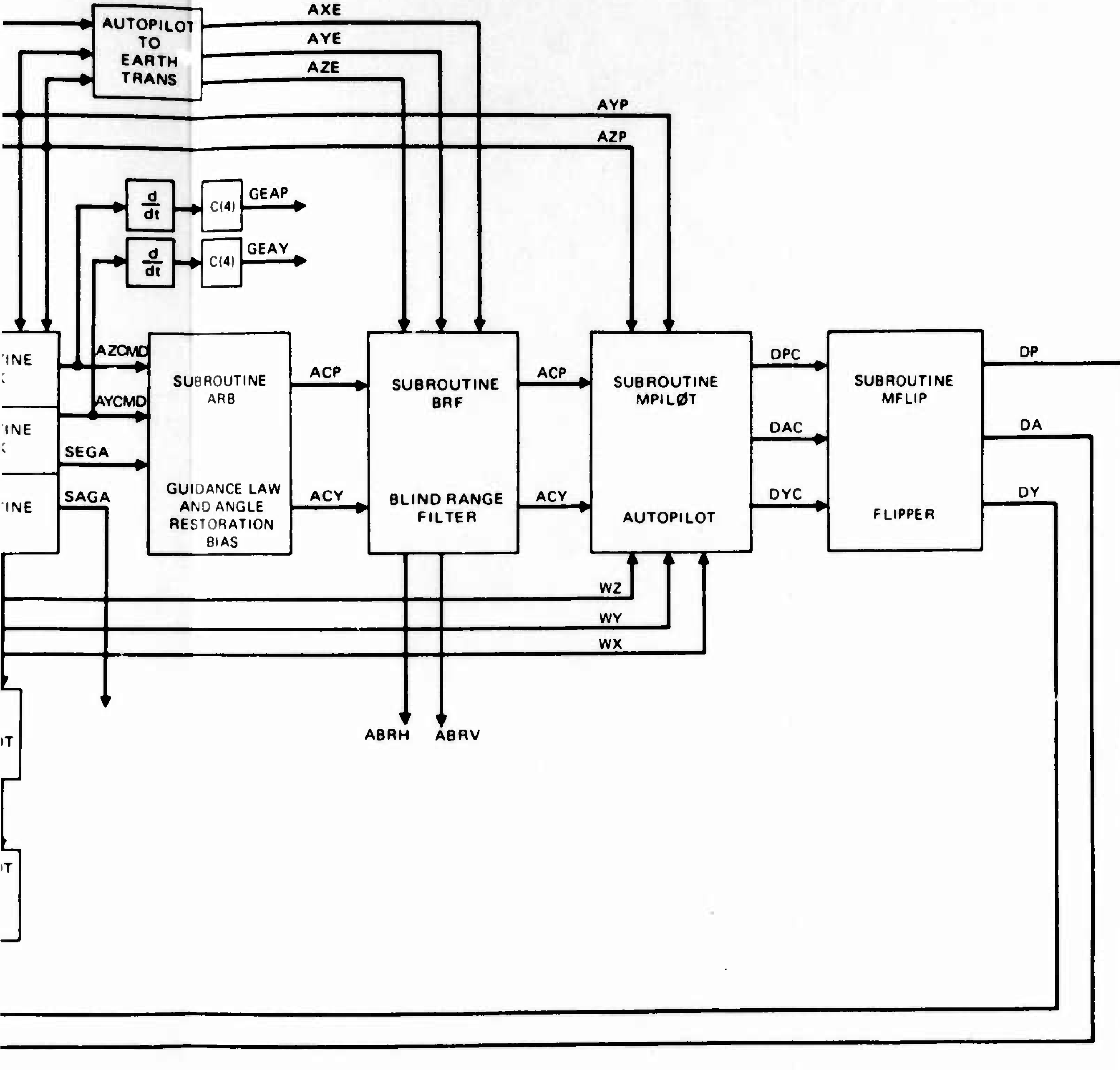



Figure 7. Call Program Flow Diagram


(The reverse of this page is blank.)

TABLE III. CALL PROGRAM GLOSSARY, V ARRAY

| Name | Quantity | Units | Coordinate System |
|---|---|---|---|
| V(1) | hm, Missile altitude above ground | ft | Inertial |
| V(2) | $\delta_{ac}$, Aileron deflection command | deg | Missile |
| V(3) | $\delta_{pc}$, Pitch deflection command | deg | Missile |
| V(4) | $\delta_{yc}$, Yaw deflection command | deg | Missile |
| V(5) | $\delta_a$, Aileron deflection | deg | Missile |
| V(6) | $\delta_p$, Pitch deflection | deg | Missile |
| V(7) | $\delta_y$, Yaw deflection | deg | Missile |
| V(8) | $V_x$, Missile velocity X-axis | ft/sec | Missile |
| V(9) | $V_y$, Missile velocity Y-axis | ft/sec | Missile |
| V(10) | $V_z$, Missile velocity Z-axis | ft/sec | Missile |
| V(11) | $\omega_x$, Angular velocity | rad/sec | Missile |
| V(12) | $\omega_y$, Angular velocity | rad/sec | Missile |
| V(13) | $\omega_z$, Angular velocity | rad/sec | Missile |
| V(14) | $A_x$, Propulsion and aerodynamic acceleration | g | Missile |
| V(15) | $A_y$, Propulsion and aerodynamic acceleration | g | Missile |
| V(16) | $A_z$, Propulsion and aerodynamic acceleration | g | Missile |
| V(17) | $A_{zc}$, Elevation maneuver command | g | Autopilot |
| V(18) | $A_{yc}$, Azimuth maneuver command | g | Autopilot |
| V(19) | $\psi$, Euler yaw angle | rad | |
| V(20) | $\phi$, Euler roll angle | rad | |
| V(21) | $\theta$, Euler pitch angle | rad | |
| V(22) | $R_x$, Seeker boresight range | ft | Seeker |
| V(23) | $R_y$, Seeker lateral range | ft | Seeker |
| V(24) | $R_z$, Seeker normal range | ft | Seeker |
| V(25) | $\epsilon_z$, Tracking error angle, pitch | rad | Seeker |
| V(26) | $\epsilon_y$, Tracking error angle, yaw | rad | Seeker |
| V(27) | $\nu$, Seeker elevation gimbal angle | rad | |
| V(28) | $\eta$, Seeker azimuth gimbal angle | rad | |

TABLE III. CALL PROGRAM GLOSSARY, V ARRAY (CONTINUED)

| Name | Quantity | Units | Coordinate System |
|---|---|---|---|
| V(29) | $R_i$, Horizontal longitudinal range component | ft | Inertial |
| V(30) | $R_j$, Horizontal lateral range component | ft | Inertial |
| V(31) | $\epsilon_{gz}$, Gate error angle, pitch | rad | Seeker |
| V(32) | $\epsilon_{gy}$, Gate error angle, yaw | rad | Seeker |
| V(33) | $\alpha$, Total miss angle of attack | deg | Missile |
| V(34) | $\alpha_p$, Missile pitch angle of attack | deg | Missile |
| V(35) | $\alpha_y$, Missile yaw angle of attack | deg | Missile |
| V(36) | $V_i$, Horizontal longitudinal velocity component | ft/sec | Inertial |
| V(37) | $V_j$, Horizontal lateral velocity component | ft/sec | Inertial |
| V(38) | $V_k$, Vertical velocity component | ft/sec | Inertial |
| V(39) | q, Dynamic pressure | $lb/ft^2$ | |
| V(40) | Total missile velocity | ft/sec | |
| V(41) | Missile Mach number | | |
| V(42) | $a_{cp}$, Acceleration command pitch | g | Autopilot |
| V(43) | $a_{cy}$, Acceleration command yaw | g | Autopilot |
| V(44) | $\dot{\omega}_x$ Scalar components of missile angular acceleration in missile axes | $rad/sec^2$ | Missile |
| V(45) | $\dot{\omega}_y$ | | |
| V(46) | $\dot{\omega}_z$ | | |
| V(47) | $\dot{\delta}_{ac}$, Aileron command rate | deg/sec | Missile |
| V(48) | $\dot{\delta}_{pc}$, Elevator command rate | deg/sec | Missile |
| V(49) | $\dot{\delta}_{yc}$, Rudder command rate | deg/sec | Missile |
| V(50) | Closest approach at end of flight | ft | |
| V(51) | Range component in Y seeker axis | ft | Seeker |
| V(52) | Range component in Z seeker axis | ft | Seeker |
| V(53) | $\omega'_x$ Missile body rates in autopilot axes | rad/sec | Autopilot |
| V(54) | $\omega'_y$ | | |
| V(55) | $\omega'_z$ | | |

TABLE III. CALL PROGRAM GLOSSARY, V ARRAY (CONTINUED)

| Name | Quantity | Units | Coordinate System |
|---|---|---|---|
| V(56) | $\omega'_x$ Scalar components of missile angular acceleration in autopilot axes | rad/sec$^2$ | Autopilot |
| V(57) | $\omega'_y$ | | |
| V(58) | $\omega'_z$ | | |
| V(59) | $A'_x$ Propulsive and aerodynamic acceleration components in autopilot axes | g | Autopilot |
| V(60) | $A'_y$ | | |
| V(61) | $A'_z$ | | |
| V(62) | $V'_x$ Missile velocity components in autopilot axes | ft/sec | Autopilot |
| V(63) | $V'_y$ | | |
| V(64) | $V'_z$ | | |
| V(65) | Special test variable - used as system diagnostic | | |
| V(66) | Total miss distance | ft | Miss Distance |
| V(67) | x component of range | ft | Autopilot |
| V(68) | y component of range | ft | Autopilot |
| V(69) | z component of range | ft | Autopilot |
| V(70) | y component of miss | ft | Miss Distance |
| V(71) | z component of miss | ft | Miss Distance |
| V(72) | x component of acceleration | g | Inertial |
| V(73) | y component of acceleration | g | Inertial |
| V(74) | z component of acceleration | g | Inertial |
| V(75) | y component of acceleration at blind range | g | Miss Distance |
| V(76) | z component of acceleration at blind range | g | Miss Distance |
| V(77) | Blind time in yaw channel | sec | |
| V(78) | Blind time in pitch channel | sec | |
| V(79) | Final line of sight angle (vertical) | rad | Inertial |
| V(80) | Final heading angle (horizontal) | rad | Inertial |
| V(81) | x component, LOS rate | rad/sec | Inertial |
| V(82) | y component, LOS rate | rad/sec | Inertial |
| V(83) | z component, LOS rate | rad/sec | Inertial |
| V(84) | $\Lambda$, Guidance gain | | |

TABLE III. CALL PROGRAM GLOSSARY, V ARRAY (CONTINUED)

| Name | Quantity | | Units | Coordinate System |
|---|---|---|---|---|
| V(85) | DE | Total yaw precession rate | | |
| V(86) | DEXS | Total pitch precession rate | | |
| V(87) | E | Yaw gyro inertial angle | | |
| V(88) | C1 | Yaw look angle (indicated) | | |
| V(90) | G1 | Forcing function cross-coupled equation 1 | | |
| V(91) | DG1 | Derivative forcing function cross-coupled equation 1 | | |
| V(92) | G2 | Forcing function cross-coupled equation 2 | | |
| V(93) | DG2 | Derivative forcing function cross-coupled equation 2 | | |
| V(94) | G1N | Integral forcing function cross-coupled equation 1 | | |
| V(95) | G2N | Integral forcing function cross-coupled equation 2 | | |
| V(96) | FFE | Forcing function yaw axis | | |
| V(97) | DFE | Derivative forcing function yaw axis | | |
| V(98) | FEXS | Forcing function pitch axis | | |
| V(99) | DFEXS | Derivative forcing function pitch axis | | |
| V(100) | | | | |
| V(101) | | | | |
| V(102) | | | | |
| V(103) | | | | |
| V(104) | | | | |
| V(105) | NOT USED | | | |
| V(106) | | | | |
| V(107) | | | | |
| V(108) | | | | |
| V(109) | | | | |
| V(110) | | | | |

TABLE III. CALL PROGRAM GLOSSARY, V ARRAY (CONCLUDED)

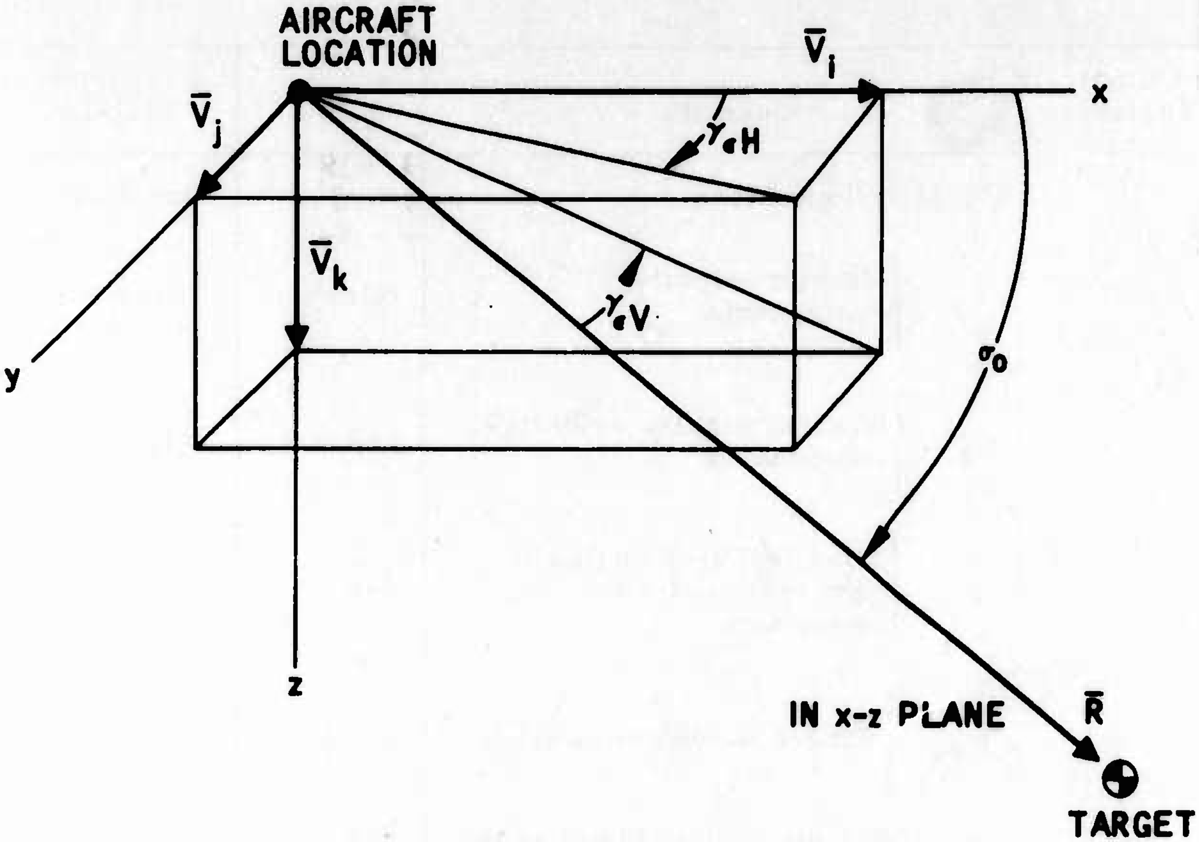
| Name | Quantity | Units | Coordinate System |
|---|---|---|---|
| V(111) | Sum1 - Tracker sampler bias | sec | |
| V(112) | TEAYD Tracker error yaw · RKAMG | deg | |
| V(113) | TEAPD Tracker error pitch · RKAMG | deg | |
| V(114) | TEAYS - Tracker ZØH output Signal, Yaw | deg | |
| V(115) | TEAPS - Tracker ZØH output Signal, Pitch | deg/sec | |
| V(116) | VSYP - Tracker output signal Pitch | deg/sec | |
| V(117) | VSPP - Tracker output signal yaw | deg/sec | |
| V(118) | TEYD - Tracking error - yaw | deg | |
| V(119) | TEPD - Tracking error - pitch | deg | |
| V(120) | UND Seeker elevation | deg | |
| V(121) | ETAD Seeker azimuth | deg | |
| V(122) | WXD | | |
| V(123) | WYD Missile angular velocity | deg | |
| V(124) | WZD | | |
| V(125) | XLOSD | | |
| V(126) | YLOSD LOS Rate, Inertial | deg/sec | |
| V(127) | ZLOSD | | |
| V(128) | ANT (New) | | |
| V(129) | $\psi$ Yaw D | | |
| V(130) | $\phi$ Roll D Error Angle | deg | |
| V(131) | $\theta$ Pitch | | |
| V(132) | DED Total precession rate, yaw | deg/sec | |
| V(133) | DEXSD Total precession rate, pitch | | |

TABLE IV. INPUT TO SUBROUTINE SETIC

| Input Location | Quantity | Units |
|---|---|---|
| T(1) | $\overline{R}$, Total range to target | ft |
| T(2) | $V_o$, Launch velocity | ft/sec |
| T(3) | $\sigma_o$, Line of sight angle | deg |
| T(4) | $\gamma_{EV}$, Heading error, vertical | deg |
| T(5) | $\gamma_{EH}$, Heading error, horizontal | deg |
| T(6) | $\phi'_a$, Aircraft roll angle | deg |
| T(7) | $A_{aL}$, Aircraft normal acceleration | g |
| T(8) | $A_{aY}$, Aircraft lateral acceleration | g |
| T(9) | $\alpha_{po}$, Aircraft angle of attack, trim | deg |
| T(10) | $\frac{\partial \alpha}{\partial A}$, Angle of attack, gain | deg/g |
| T(11) | $\phi_{\ell}$, Missile mounting angle, roll | deg |
| T(12) | $\theta_{\ell}$, Missile mounting angle, pitch | deg |
| T(13) | $\tau_a$, Effective tracker time constant used to calculate initial tracking error angle | sec |
| T(14) | $\omega_x$ } Missile body rates in autopilot axes | rad/sec |
| T(15) | $\omega_y$ } Missile body rates in autopilot axes | rad/sec |
| T(16) | $\omega_z$ } Missile body rates in autopilot axes | rad/sec |
| T(17) | $R_{BH}$, Blind range, horizontal | ft |
| T(18) | $R_{BV}$, Blind range, vertical | ft |
| T(19) | Steering bias, pitch | g |
| T(20) | Steering bias, yaw | g |
| T(21) | Roll rate bias | rad/sec |

TABLE V. SUBROUTINE SETIC OUTPUT

| Output Variable | Quantity | | Units | Coordinate System |
|---|---|---|---|---|
| V(1) | h, Missile altitude | | ft | Inertial |
| V(8) | $V_x$ | Missile velocity components | ft/sec | Missile |
| V(9) | $V_y$ | | | |
| V(10) | $V_z$ | | | |
| V(11) | $\omega_x$ | Missile angular velocity components | rad/sec | Missile |
| V(12) | $\omega_y$ | | | |
| V(13) | $\omega_z$ | | | |
| V(19) | $\psi$ | Missile Euler angles in yaw, roll and pitch sequences | rad | |
| V(20) | $\phi$ | | | |
| V(21) | $\theta$ | | | |
| V(22) | $R_{xs}$ | Target range components | ft | Seeker |
| V(23) | $R_{ys}$ | | | |
| V(24) | $R_{zs}$ | | | |
| V(27) | $\nu$, Seeker elevation gimbal angle | | rad | |
| V(28) | $\eta$, Seeker azimuth gimbal angle | | rad | |
| V(29) | $R_i$, Horizontal range component to target | | ft | Inertial |
| V(67) | $R_{xm}$ | Target range components | ft | Autopilot |
| V(68) | $R_{ym}$ | | | |
| V(69) | $R_{zm}$ | | | |

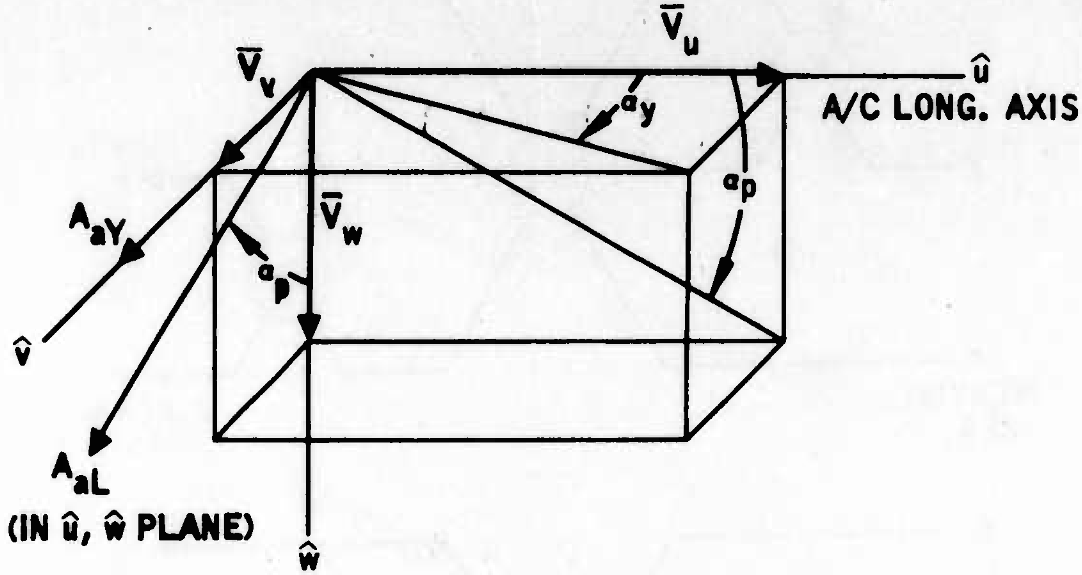



x, y, z    INERTIAL COORDINATE AXES

$\bar{V}_i$, $\bar{V}_j$, $\bar{V}_k$    AIRCRAFT VELOCITY COMPONENTS

THE INERTIAL AXES ARE SELECTED SO THAT THE INITIAL RANGE VECTOR IS CONTAINED IN THE x-z PLANE

Figure 8. Launch Geometry in Inertial Coordinates




$\hat{u}, \hat{v}, \hat{w}$ AIRCRAFT COORDINATE AXES

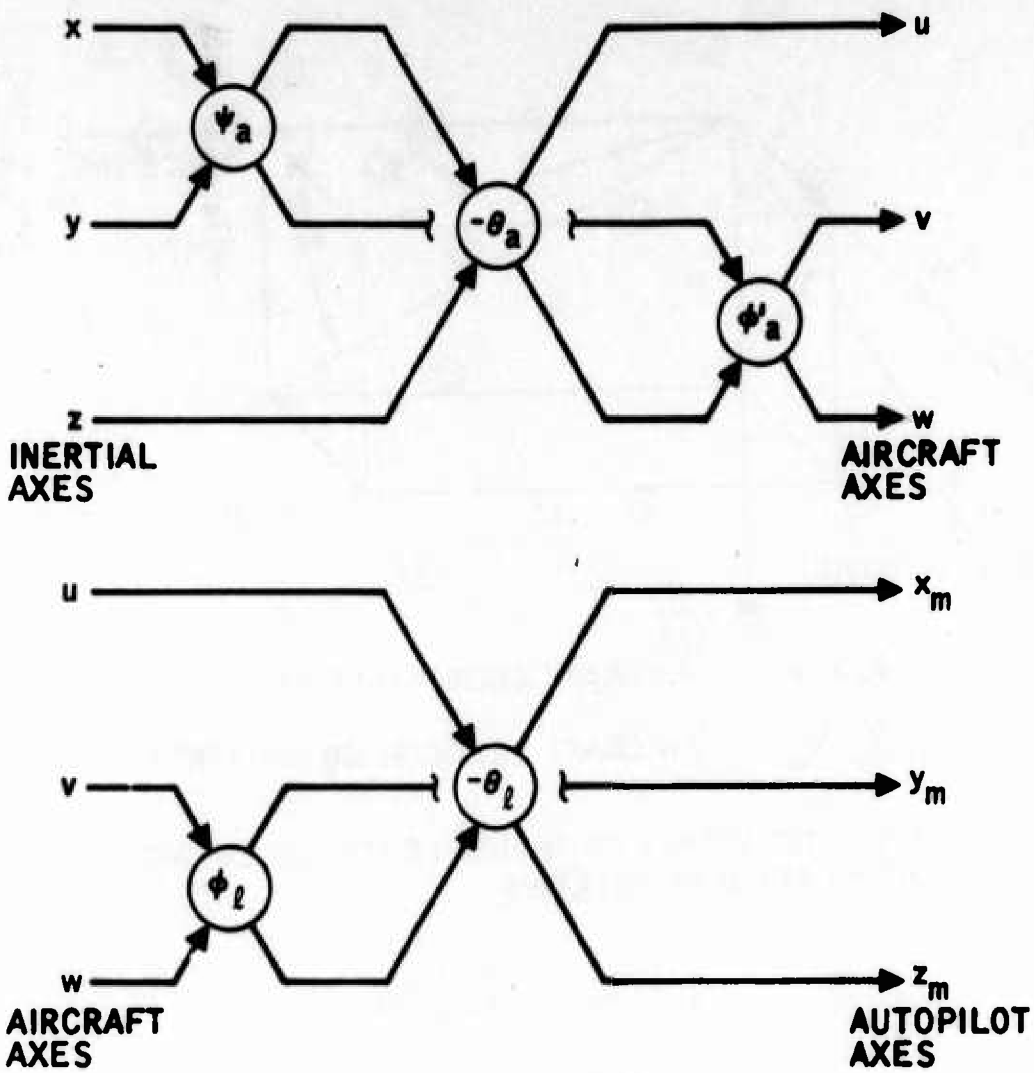
$\bar{V}_u, \bar{V}_v, \bar{V}_w$ AIRCRAFT VELOCITY COMPONENTS

ANGLE OF ATTACK COMPONENTS ARE CALCULATED WITHIN SETIC AS FOLLOWS:

$$\alpha_p = \alpha_{po} + \left(\frac{\partial \alpha}{\partial A}\right) A_{aL}$$

$$\alpha_y = \left(\frac{\partial \alpha}{\partial A}\right) A_{aY}$$

Figure 9. Launch Geometry in Aircraft Coordinates




$\psi_a$ AND $\theta_a$ ARE COMPUTED WITHIN SETIC.
ALL OTHER ANGLES ARE PROVIDED AS INPUTS.

Figure 10. Euler Angle Relations Between Coordinate Axis Sets

# TABLE VI. SUBROUTINE SETIC FORTRAN LISTING

```
$       FORTRAN DECK
CSETIC         SET INITIAL CONDITIONS  I                                SETI0010
      SUBROUTINE SETIC                                                  SETI0020
      COMMON /SSAM2/ V  (250),T  (250),C  (250)
      EQUIVALENCE                                                       SETI0050
     1 (T( 1),RANGE ),(T( 2),VEL   ),(T( 3),SIGMA ),(T( 4),HEV   ),     SETI0060
     2 (T( 5),HEH   ),(T( 6),ACROLL),(T( 7),ACCEL ),(T( 8),ACCELY),     SETI0070
     3 (T( 9),ALPO  ),(T(10),DALDA ),(T(11),PHIL  ),(T(12),THETAL),     SETI0080
     4 (T(13),TAUA  ),(T(14),WX    ),(T(15),WY    ),(T(16),WZ    )      SETI0090
      DATA RTOD/57.29578/                                               SETI0100
      C(9)=T(17)                                                        SETI0110
      C(3)=T(18)                                                        SETI0120
      GAMV=(SIGMA-HEV)/RTOD                                             SETI0130
      TANGV=TAN(GAMV)                                                   SETI0140
      HEHRAD=HEH/RTOD                                                   SETI0150
      TANH=TAN(HEHRAD)                                                  SETI0160
C     VELOCITY COMPONENTS IN EARTH AXES                                 SETI0170
      VI=VEL/SQRT(1.0+TANGV**2+TANH**2)                                 SETI0180
      VJ=VI*TANH                                                        SETI0190
      VK=VI*TANGV                                                       SETI0200
      ALPHAP=(ALPO+DALDA*ACCEL)/RTOD                                    SETI0210
      ALPHAY=DALDA*ACCELY/RTOD                                          SETI0220
      TANAP=TAN(ALPHAP)                                                 SETI0230
      TANAY=TAN(ALPHAY)                                                 SETI0240
C     VELOCITY COMPONENTS IN A/C AXES                                   SETI0250
      VU=VEL/SQRT(1.0+TANAP**2+TANAY**2)                                SETI0260
      VV=VU*TANAY                                                       SETI0270
      VW=VU*TANAP                                                       SETI0280
      APHI=ACROLL/RTOD                                                  SETI0290
      SPHI=SIN(APHI)                                                    SETI0300
      CPHI=COS(APHI)                                                    SETI0310
C     ESTABLISH A/C EULER ANGLES                                        SETI0320
      CON1=VV*SPHI+VW*CPHI                                              SETI0330
      CON2=SQRT(VU*VU+CON1*CON1)                                        SETI0340
      THETAA=ARSN(CON1/CON2)-ARSN(VK/CON2)                              SETI0350
      CON3=SQRT(VI*VI+VJ*VJ)                                            SETI0360
      PSIA=ARSN(VJ/CON3)-ARSN((VV*CPHI-VW*SPHI)/CON3)                   SETI0370
      SPSI=SIN(PSIA)                                                    SETI0380
      CPSI=COS(PSIA)                                                    SETI0390
      STHE=SIN(THETAA)                                                  SETI0400
      CTHE=COS(THETAA)                                                  SETI0410
      THETLR=THETAL/RTOD                                                SETI0420
      STHL=SIN(THETLR)                                                  SETI0430
      CTHL=COS(THETLR)                                                  SETI0440
      PHILR=PHIL/RTOD                                                   SETI0450
      SPHL=SIN(PHILR)                                                   SETI0460
      CPHL=COS(PHILR)                                                   SETI0470
      A13=-STHE*CTHL+CTHE*SPHI*SPHL*STHL-CTHE*CPHI*CPHL*STHL            SETI0480
      A23= CTHE*SPHI*CPHL+CTHE*CPHI*SPHL                                SETI0490
      A21= (CPSI*STHE*SPHI-SPSI*CPHI)*CPHL +(CPSI*STHE*CPHI+SPSI*SPHI)* SETI0500
     1     SPHL                                                         SETI0510
      CON4=SQRT(1.0-A23*A23)                                            SETI0520
C     ESTABLISH MISSILE EULER ANGLES                                    SETI0530
      PHI=ARSN(A23)                                                     SETI0540
      THETA=ARSN(-A13/CON4)                                             SETI0550
      PSIPR=ARSN(-A21/CON4)                                             SETI0560
      SSIG=SIN(SIGMA/RTOD)                                              SETI0570
      CSIG=COS(SIGMA/RTOD)                                              SETI0580
C     RANGE COMPONENTS IN EARTH AXES                                    SETI0590
      RI=RANGE*CSIG                                                     SETI0600
```

TABLE VI. SUBROUTINE SETIC FORTRAN LISTING (CONCLUDED)

```
      RK=RANGE*SSIG
C     ESTABLISH TRACKING ERROR VECTOR
      CON6=VK*CSIG-VI*SSIG
      EI=TAUA*CON6*SSIG
      EJ= TAUA*VJ
      EK=-TAUA*CON6*CSIG
      PI=RI-EI
      PK=RK-EK
C     ESTABLISH SEEKER GIMBAL ANGLES
      CALL EULTRN(1,-1,PI,EJ,PK,RX,RY,RZ,PSIPR,PHI,THETA)
C     VELOCITY COMPONENTS IN MISSILE AXES
      CALL EULTRN(-1,-1,VI,VJ,VK,VX,VY,VZ,PSIPR,PHI,THETA)
C     RANGE COMPONENTS IN MISSILE AXES
      CALL EULTRN(-1,-1,RI,0.0,RK,RXM,RYM,RZM,PSIPR,PHI,THETA)
C     VELOCITY COMPONENTS IN AUTOPILOT AXES
      CALL R45F(VX,VY,VZ,VXM,VYM,VZM)
      UN=ATAN2(-RZ,RX)
      ETA=ATAN2(RY,SQRT(RX*RX+RZ*RZ))
C     RANGE COMPONENTS IN SEEKER AXES
      CALL SEKTR(1,1,RXM,RYM,RZM,RXS,RYS,RZS,UN,ETA)
      V(8)=VXM
      V(9)=VYM
      V(10)=VZM
      V(19)=PSIPR
      V(20)=PHI
      V(21)=THETA
      V(27)=UN
      V(28)=ETA
      V(1)=RK
      V(29)=RI
      V(67)=RXM
      V(68)=RYM
      V(69)=RZM
      V(22)=RXS
      V(23)=RYS
      V(24)=RZS
      CALL R45F(WX,WY,WZ,V(11),V(12),V(13))
      RETURN
      END
```

(U) Also, the 45-degree rotation of the missile axes relative to the mounting hooks is not contained in the input T(11), $\phi_\ell$. This angle will nominally be approximately zero or ±90 degrees depending upon whether the missile is mounted on the bottom or sides of the pylon.

### 2.3.4 Universal Seeker Subroutine

(U) Input to the Universal Seeker subroutine are the components of the missile linear velocity and acceleration components as well as the angular velocity and acceleration components, all in autopilot axes. This subroutine performs the coordinate transformations of these quantities to seeker axes and performs the appropriate integrations to yield seeker range components from which tracking error angles in the pitch and yaw planes are computed.

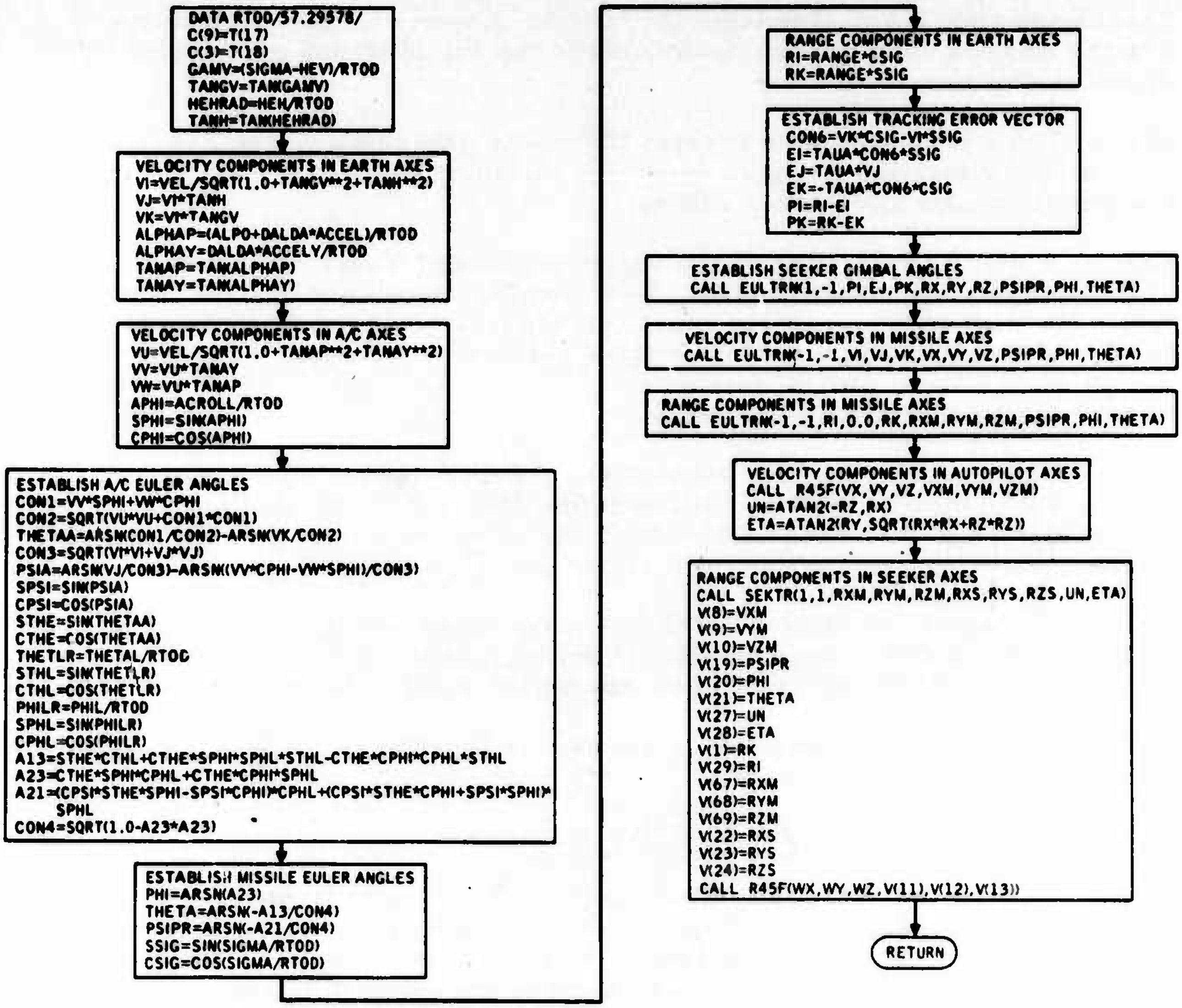



Figure 11. Subroutine Setic Flow Chart

(U) The Seeker subroutine has been modified for the CAS system simulation to accommodate two other subroutines which simulate tracker and gyro dynamics. The respective subroutines are Track and Gyro; their addition facilitates the replacement of different trackers and gyro models into the simulation.

(U) The Track subroutine accepts error signals from the Universal Seeker subroutine and simulates the tracker dynamics. Its output signals are the missile acceleration commands to the autopilot and precession rate signal to the gyro torquer.

(U) The Gyro subroutine accepts the inputs generated by the Track subroutine and simulates the gyro dynamics including drift. Its outputs are the gimbal angles and gimbal rates.

(U) A FORTRAN listing of the Universal Seeker Track and Gyro subroutines are shown in Tables VII, VIII, and IX, respectively. The subroutine block diagram and flow charts appear in Figures 12, 13, 14 and 15, respectively. Tables X, XI, and XII contain a glossary of terms.

### 2.3.5 Aimpoint Wander Subroutine

(U) The Aimpoint Wander subroutine, GWAND, is called out within the Seeker subroutine and either simulates the apparent target motion caused by the wandering of the seeke: airpoint or the actual motion of an evasive target. In both cases, the subroutine input is the boresight range $R_{xs}$.

(U) The equations implemented in each of these options are shown in Tables XIII and XIV, respectively. If neither option is desired, setting the parameter C(103) to zero will cause the entire subroutine to be bypassed.

(U) A FORTRAN listing of this subroutine appears in Table XV and its flow chart is shown in Figure 16.

### 2.3.6 Angle Restoration Bias (ARB) Subroutine

(U) The Angle Restoration Bias subroutine serves to implement the guidance law incorporated in the CAS missile. It operates on the acceleration commands from the Seeker subroutine to provide steering commands for the Autopilot subroutine. A FORTRAN listing of the subroutine appears in Table XVI. A block diagram and flow chart of the subroutine appear in Figures 17 and 18, respectively.

### 2.3.7 Blind Range Filter (BRF) Subroutine

(U) This subroutine simply provides the filtering for commands to the autopilot subroutine when blind range is reached. The FORTRAN listing for this subroutine appears in Table XVII. The subroutine block diagram and flow chart appear in Figures 19 and 20.

# TABLE VII. UNIVERSAL SEEKER FORTRAN LISTING

```
$       FORTRAN LSTOU,DECK
CUSEK        UNIVERSAL SEEKER
      SUBROUTINE MSEEK                                                        SEEK0020
      COMMON /SSAM1/ READ,DELT,AUTOT,TIME
      COMMON /SSAM2/ V  (250),T  (250),C  (250)
      COMMON /TRAKER/ COUNT,TR,N1,OFFX,OFFY
      COMMON /TRAKZ / TEAPO,TEAYO,DTEAP,DTEAY,WYSC,WZSC
      COMMON/GYR/VSY,VSP,WXS,WYS,WZS
      EQUIVALENCE                                                             SEEK0070
     1 (V(22),RXS   ),(V(23),RYS   ),(V(24),RZS   ),(V(25),TEAP  ),          SEEK0080
     2 (V(26),TEAY  ),(V(17),AZCMD ),(V(18),AYCMD ),(V(27),UN    ),          SEEK0090
     3 (V(28),ETA   ),     (V(31),EP1),      (V(32),EP2)
      EQUIVALENCE                                                             SEEK0110
     1 (V(53),WX    ),(V(54),WY    ),(V(55),WZ    ),(V(56),DWX   ),          SEEK0120
     2 (V(57),DWY   ),(V(58),DWZ   ),(V(59),AX    ),(V(60),AY    ),          SEEK0130
     3 (V(61),AZ    ),(V(62),VX    ),(V(63),VY    ),(V(64),VZ    )           SEEK0140
      EQUIVALENCE
     0(V(104),VS2),      (V(105),VS1),       (V(110),WND)
      EQUIVALENCE                                                             SEEK0150
     1(V(66),TOTMIS),(V(70),EMJ   ),(V(71),EMK    )                          SEEK0160
     2,(V(50),TSMISS),(V(51),YMISS ),(V(52),ZMISS )                          SEEK0170
      EQUIVALENCE
     1(C(13),TARNGE),(C(14),GK   ),(C(15),TAUAP),(C(16),OMEGAL),
     2(C(17),C1 ),(C(18),C2 ),(C(19),C3 ),(C(20),C4 ),(C(21),C5 ),           SEEK0200
     3(C(22),C6 ),(C(23),C7 ),(C(24),C8 ),(C(25),C9 ),(C(26),C10),           SEEK0220
     4(C(27),C11),(C(28),C12),(C(29),C13),(C(30),C14),(C(31),C15),           SEEK0210
     5(C(32),C16),(C(33),C17),(C(34),C18),(C(35),C19),(C(36),C20),           SEEK0230
     6(C(37),C21),(C(38),C22),(C(39),C23),(C(40),AK1),(C(41),TG    )         SEEK0240
      EQUIVALENCE
     1(C(109),XK1),     (C(110),XK2),      (C(111),PK1),      (C(112),TAU1),
     2(C(113),TAU2),    (C(114),TB),       (C(115),TC)
      DATA  RTOD/57.2957795/
      NAMELIST/NAMS/RXS,ERXS,EMJ,EMK,TOTMIS,TIME,DELT
      IF (READ.EQ.0.0) GO TO 50                                               SEEK0250
C     C(13) THROUGH C(42) ARE RESERVED FOR THIS SUBROUTINE                    SEEK0260
C     C(42) IS DRIFT CONTROL. SET TO +1.0 TO INCLUDE DRIFT                    SEEK0270
      TEAP=-EZS/RXS                                                            SEEK028
      TEAY=RYS/RXS                                                            SEEK0290
      WZSD=0.0                                                                SEEK0310
      WYSD=0.0                                                                SEEK0320
      SINSIG=SIN(T(3)/RTOD)                                                   SEEK0330
      COSSIG=COS(T(3)/RTOD)                                                   SEEK0340
      IF (C(103).EQ.0.0) CALL GWAND(RXS,ERY,ERZ)                              SEEK0360
C                                                                             SEEK0370
C     TRACKING ERROR ANGLES                                                   SEEK0380
   50 CONTINUE                                                                SEEK0390
      IF (C(103).EQ.0.0) GO TO 51                                             SEEK0400
      CALL GWAND (RXS,ERY,ERZ)                                                SEEK0410
      CALL COLTRN(0,-1,0.0,ERY,ERZ,ERXM,ERYM,ERZM,YAW,ROLL,PITCH)             SEEK0420
      CALL SEKTR (0,1,ERXM,ERYM,ERZM,EXS,EYS,EZS,UN,ETA)                      SEEK0430
      ESUM=RXS+EXS                                                            SEEK0440
      YMISS=RYS+EYS                                                           SEEK0450
      ZMISS=RZS+EZS                                                           SEEK0460
      GO TO 52                                                                SEEK0470
   51 ESUM=RXS                                                                SEEK0480
      YMISS=RYS                                                               SEEK0490
      ZMISS=RZS                                                               SEEK0500
   52 TEAP=-ZMISS/ESUM                                                         SEEK051
      TEAY=YMISS/ESUM                                                         SEEK0520
      IF (ABS(TEAP).GE.TARNGE) GO TO 62                                       SEEK0540
```

## TABLE VII. UNIVERSAL SEEKER FORTRAN LISTING (CONCLUDED)

```
      IF (ABS(TEAY).GE.TARNGE) GO TO 62                                 SEEK0550
      CALL DIF(TEAP,DTEAP,TEAPO)                                        SEEK0560
      CALL DIF(TEAY,DTEAY,TEAYO)                                        SEEK0570
      IF (DTEAP) 80,82,81                                               SEEK 571
   80 CALL SPTEST(TEAP,DTEAP,-TARNGE)                                   SEEK 572
      GO TO 82                                                          SEEK 573
   81 CALL SPTEST(TEAP,DTEAP, TARNGE)                                   SEEK 574
   82 IF (DTEAY) 83,85,84                                               SEEK 575
   83 CALL SPTEST(TEAY,DTEAY,-TARNGE)                                   SEEK 576
      GO TO 85                                                          SEEK 577
   84 CALL SPTEST(TEAY,DTEAY, TARNGE)                                   SEEK 578
   85 CONTINUE                                                          SEEK 579
      GO TO 65                                                          SEEK0580
   62 TEAP=0.0                                                          SEEK0590
      DTEAP=0.0                                                         SEEK0600
      TEAPO=0.0                                                         SEEK0610
      TEAY=0.0                                                          SEEK0620
      DTEAY=0.0                                                         SEEK0630
      TEAYO=0.0                                                         SEEK0640
C                                                                       SEEK0650
C     CONTROL COMMAND                                                   SEEK0660
   65 CONTINUE
      CALL TRACK
      CALL GYRO
      CALL SEKTR(1,1,-VX,-VY,-VZ,VRXS,VRYS,VRZS,UN,ETA)                 SEEK1200
      CALL VECTV(VRXS,VRYS,VRZS,RXS,RYS,RZS,WXS,WYS,WZS,DRXS,DRYS,DRZS) SEEK1220
      CALL SPTEST(RXS,DRXS,C(3))                                        SEEK1222
      CALL SPTEST(RXS,DRXS,C(9))                                        SEEK1224
      CALL SPTEST(RXS,DRXS,10.)                                         SEEK1226
      IF ((V(1).GT.0.0).AND.(ESUM.GT.10.0)) GO TO 140                   SEEK1260
      DTO=  -(ESUM/DRXS)                                                SEEK1240
      IEND=1.0                                                          SEEK1280
      YMISS=YMISS+DTO*DRYS                                              SEEK1290
      ZMISS=ZMISS+DTO*DRZS                                              SEEK1300
      V(50)=SQRT(V(51)**2+V(52)**2)                                     SEEK1310
      V(77)=C(9)/VRXS                                                   SEEK1320
      V(78)=C(3)/VRXS                                                   SEEK1330
      V(79)=ATAN2(V(1),V(29))                                           SEEK1340
      V(80)=ATAN2(V( 30),V(29))                                         SEEK1350
  140 CALL INTER(DRXS,DRX1,DRX2,RXS0,RXS)                               SEEK1360
      CALL INTER(DRYS,DRY1,DRY2,RYS0,RYS)                               SEEK1370
      CALL INTER(DRZS,DRZ1,DRZ2,RZS0,RZS)                               SEEK1380
      RI=V(29)*COSSIG+V(1)*SINSIG                                       SEEK1390
      RJ=V(30)                                                          SEEK1400
      RK=-V(29)*SINSIG+V(1)*COSSIG                                      SEEK1410
      VI=V(36)*COSSIG+V(38)*SINSIG                                      SEEK1420
      VJ=V(37)                                                          SEEK1430
      VK=-V(36)*SINSIG+V(38)*COSSIG                                     SEEK1440
      EMJ=RJ-RI*VJ/VI                                                   SEEK1450
      EMK=RK-RI*VK/VI                                                   SEEK1460
      TOTMIS=SQRT(EMJ**2+EMK**2)                                        SEEK1470
      IF(RXS.LE.15.)      WRITE(6,NAMS)
  120 RETURN                                                            SEEK1480
      END                                                               SEEK1490
```

# TABLE VIII. TRACK ROUTINE FORTRAN LISTING

```
C       A2 NAVII TRACKER A-6  WITH LAG ONLY
      SUBROUTINE TRACK
      COMMON /SSAM1/ READ,DELT,AUTOT,TIME
      COMMON /SSAM2/ V  (250),T  (250),C  (250)
      COMMON /SSAM/ IEND,ND,TNEXT,VMIN,STPMX,S12345,SUM222
     1,CETA,SETA,CNU,SNU,TMAX,NZ,LNV(50),TITLE(250),DELTO,RTITLE(9)
     2,IFGEN,IMFGEN,MFGEN2,IFG2N
      COMMON /GYR/VSY,VSP,WXS,WYS,WZS
      COMMON /TRAKZ / TEAPO,TEAYO,DTEAP,DTEAY,WYSC,WZSC
      EQUIVALENCE
     1(V(17),AZCMD),    (V(18),AYCMD),    (V(25),TEAP),    (V(26),TEAY),
     2(V(22),RXS),   (V(112),TEAYD), (V(113),TEAPD), (V(116),VSYP),
     3(V(117),VSPP)
      EQUIVALENCE
     1(C(142),SK),     (C(143),AKT),    (C(144),TS),     (C(145),OMEGLD),
     2(C(146),GKK),    (C(147),BIAS),   (C(148),TLDP),   (C(149),TLGP),
     3(C(150),TLDY),   (C(151),TLGY), (C(152),SPOT)
      NAMELIST
     A/NAMB/
     BTEAY,TEAYD,TEAP,TEAPD,TEAYS,TEAPS,SUM1,VSYP,VSPP,AYCMD,AZCMD,VSY,
     CVSP,TIME,GLVSYO,GLVSPO,TLDP,TLGP,TLDY,TLGY,TEAPK,TEAYK,RKAMG,SPXS
      DATA  RTOD/57.2957795/
      SPXS=RXS/SPOT
      CALL FGEN1(RDUMG,SPXS,RKAMG,-1)
      IF(READ.NE.0.0)  GO TO 10
      GO TO 14
   10 SUM1=BIAS
      GO TO 16
   14 IF(TIME.LT.SUM1) GO TO 20
   16 TEAPK=TEAP*RKAMG
      TEAYK=TEAY*RKAMG
      TEAYD=RTOD*TEAYK
      TEAPD=RTOD*TEAPK
      CALL FGEN1(TDUMY,TEAYD,TEAYS,-1)
      CALL FGEN1(TDUMP,TEAPD,TEAPS,-1)
      CONT=1.
      SUM1=SUM1+TS
   20 CALL TTEST(SUM1)
      VSYP=AKT*TEAYS
      VSPP=AKT*TEAPS
      CALL DIF(VSYP,DVSYP,VDUMY)
      CALL DIF(VSPP,DVSPP,VDUMP)
      CALL LAG(VSPP,DVSPP,GLVSPO,GLVSP,DLGVSP,TLGP,GDUMP)
      CALL LAG(VSYP,DVSYP,GLVSYO,GLVSY,DLGVSY,TLGY,GDUMY)
      AYCMD=GKK*GLVSY
      AZCMD=-GKK*GLVSP
      CALL LIMIT(GLVSY,DLGVSY,OMEGLD,-OMEGLD)
      CALL LIMIT(GLVSP,DLGVSP,OMEGLD,-OMEGLD)
      VSY=GLVSY*SK
      VSP=GLVSP*SK
      IF(CONT.EQ.1.)  WRITE(6,NAMB)
      CONT=0.
      RETURN
      END
```

## TABLE IX. GYRO SUBROUTINE FORTRAN LISTING

```
CGYRO       GYRO SIDE RAIL        GNUT OPTION
      SUBROUTINE GYRO
      DIMENSION  F01B(4),F12E(4),F04C(3)
      COMMON /SSAM1/ READ,DELT,AUTOT,TIME
      COMMON /SSAM2/ V  (250),T  (250),C  (250)
      COMMON /TRAKER/ COUNT,TH,N1
     1       ,GFFX,GFFY
      COMMON
     1/GYR/VSY,VSP,CE,WYS,WZS
      EQUIVALENCE
     1(V(27),UN),        (V(28),ETA)
      EQUIVALENCE
     1(V(53),WA),        (V(54),WY),        (V(55),WZ),        (V(56),DWA),
     2(V(57),DWY),       (V(58),DWZ),       (V(59),GA),        (V(60),AY),
     3(V(61),AZ)
      EQUIVALENCE
     1(V(85),DE),        (V(86),DEXS),      (V(87),E),         (V(88),EXS),
     2(V(89),G1),        (V(90),G1),        (V(91),DG1),       (V(92),G2),
     3(V(93),DG2),       (V(94),G1N),       (V(95),G2N),       (V(96),FFE),
     4(V(97),DFE),       (V(98),FEXS),      (V(99),DFEXS)
      EQUIVALENCE
     1(C(116),W3S),       (C(117),K2T),      (C(118),DUMP),    (C(119),RAIL)
     2,(C(136),GNUT),     (C(137),DFR),      (C(138),DST),     (C(139),DSU),
     3(C(140),DAN),       (C(141),DDU), (C(153),CF1),  (C(154),CF2),
     4(C(155),CF3)
      REAL  I1R,I1RXE,I2E,I2S,     I2EXS,I3S,I3T,I4C,I4D,I4DXC,I1E
      REAL  MC,MP,LR,K01B,K02,K12F,K04C,MPX,K3E,K3S,K2E,K2S,K2EXS,K1E,
     1K1BXE,M3,M23,M123,K2T
      REAL  KRR,KRT,KGT,KGR,M23S,M123S
      NAMELIST
     A/NAMZZ/
     BUEF,E,FHS,FFXS,ET,EXS,EXSS,EXST,GA2,GBP,GEP,HUXCD,KGR,KGT,
     CK12T,KRR,KRT,M123S,M23S,SE,W13,XEWM,XGEXS,XGSE,XKG,XKR,
     DXKRM,XTE,XU
     E/NAMA/
     1A1,A2,AGB,AGE,B,B1,B2,C1,CA,CB,CD,CE,CF,CH,COSB,COSC,COSD,COSE,
     2COUNT,CX1,CX2,CX3,     D1N,D2N,DB,DBX,DC,DDB,DDE1,DDXC,DE,DE1,
     3DEEXS,DEX,DEXS,DFE,DFEXS,DG1,DG2,DX,DWA,DWBXE,DWDXC,E,F1,EXS,
     4FB,FC,FDB,FDC,FDE,FE,FFE,FEXS,G1,G1N,G1Z,G2,G2N,G2Z,GH,GA2,GBL,
     5GBXE,GC,GE,GEL,GFXS,GS,HDDXC,MC,MP,RE1,RE2,RX1,RX2,SASF,SEC2,SECB,
     6SECC,SECD,SECE,SINB,SINC,SIND,SINE,STAN,     TAN2,TANC,TANE,
     7TANY,TE,TEA,TEDU,TEF,TEP,TES,TEU,TEXS,TEXSA,TEXSF,TEXSP,TEXSS,
     8TEXSU,TXDU,W1,W12,W2,WA,WBXE,WD,WDXC,WE,WEBXE,WN,WT,    XFM,
     9XMK,XMP,XIM,XIMC,   ES,XS
     A,TAND,TIME,DELT,CX4C
C
C     MAVERICK GYRO ESTIMATED AND CALCULATED PARAMETERS  (7-15-69)
C
      DATA
     1I3S,I3T,I2E,I2S,I2EXS/4.27E 4,2.04E 4,2.36E 4,2.98E 4,2.70E 4/,
     2I1H,I1F,I1BXE/1.46E 4,1.53E 4,2.94E 4/,
     3I4C,I4D,I4DXC/1.75E 3,1.94E 3,2.88E 3/,
     4M3,M23,M123/1.12E 3,2.53E 3,2.95E 3/,
```

TABLE IX. GYRO SUBROUTINE FORTRAN LISTING (CONTINUED)

```
     5K3F,K3S,K1F,K1RXF/0.42E-7,2.10E-7,0.52E-7,1.00E-7/,
     6K2F,K2S,K2FXS/0.30E-7,0.70E-7,0.75E-7/,
     7RCA,RPL,GCA,GPL/10.,4540.,20.,4540./,
     8F01H/33.0,7.3,1.9,0.22/,
     9F12E/22.5,2.1,0.5,0.16/,
     AF04C/11.0,1.3,0.4/,
     BY,K02,K01B,K12E,K04C/1.2,150.,0.,0.,0./,
     CV01B,V12E,V04C/4.0,1.5,2.5/,
     DUS,UBXE,UE,UEXS,UD,UDXC,MPX/1.00,1.00,1.00,1.00,0.50,0.50,2.0/,
     ELR,OL,OT,DUB,THET/0.215,12.0,21.0372,0.50,0./,
     FG,R/980.,57.2957795/
      IF (READ.EQ.0.)  GO TO 3
C
CCCCC INITIAL COMPUTATIONS
C
      M23S=M23**2
      M123S=M123**2
      EXSS=I2FXS-I2S
      MDXCD=I4DXC-I4D
      ET=I2E+I3T
      BEF=I1BXE-I1F
      EEXS=I2E-I2EXS
      SE=I2S-I2E
      EXST=I3T+I2EXS
      EBS=I1B+I2S
      WI3=W3S*I3S
      GHP=GPL/M123
      GEF=GPL/M23
      KRR=9.E-7*RCA/RPL+9.E-10*RCA
      KRT=1.4E-3/(RPL*RCA)+1.9E-7/SQRT(RCA)
      KGT=1.4E-3/(GPL*GCA)+1.9E-7/SQRT(GCA)
      KGR=9.E-7*GCA/GPL+9.E-10*GCA
      XKR=K3E-K3S+KRR-KRT
      XKG=K2F+KGT-KGR/2.
      XKRM=XKR*M3**2
      XGSE=XKRM+M23S*(XKG-K2S)
      XGEXS=M23S*(XKG-K2EXS)
      XTF=XKRM+M23S*(K2EXS-K2S)
      XERM=M123S*(K1E-K1BXE)
      GA2=GA**2
      XU=DUB*W3S**2/980.
      XU=GNUT*XU
      WRITE (6,NAMZ)
      IF(RAIL.EQ.1.0) GO TO 25
      B=UN
      E1=ETA
      GO TO 26
 25   CONTINUE
      B=ETA
      E1=UN
 26   CONTINUE
      EXS=B
      E=E1
      ES=E
      XS=EXS
      KOUNT=0
      KDUMP=DUMP+.01
C
CCCCC MISSILE FRAME RATES AND ACCELERATIONS
C
```

TABLE IX. GYRO SUBROUTINE FORTRAN LISTING (CONTINUED)

```
    3 CONTINUE
      KOUNT=KOUNT+1
      GB2=GB**2
      AGB=ABS(GB)
      IF (AGB.LE.GBP)  GO TO 210
      GBL=AGB-GBP
      GO TO 220
  210 GBL=0.
  220 CONTINUE
      FB=F01B(1)+F01B(2)*GBL+F01B(3)*SQRT(GA2+GC**2)+F01B(4)*GBL**2
      FB=FB*CF1
      FC=F04C(1)*CF2+(F04C(2)*ABS(GC)+F04C(3)*SQRT(GA2+GB2))*CF3
C
C     MISSILE RAIL POSITION
      IF(RAIL.EQ.1.0) GO TO 40
C            BOTTOM RAIL
      WB=WY
      WC=WZ
      DWB=DWY
      DWC=DWZ
      GB=AY
      GC=AZ
      B=UN
      E1=ETA
      VE=VSY
      VEXS=VSP
      GO TO 42
   40 CONTINUE
C            SIDE RAIL
      WB=WZ
      WC=WY
      DWB=DWZ
      DWC=DWY
      GB=AZ
      GC=AY
      B=ETA
      E1=UN
      VE=VSP
      VEXS=VSY
   42 CONTINUE
      DEEXS=DE*DEXS
      SINB=SIN(B)
      COSB=COS(B)
      SINE=SIN(E1)
      COSE=COS(E1)
      SECE=1./COSE
      TANE=TAN(E1)
      TAN2=TANE**2
      SEC2=SECE**2
      STAN=SECE*TANE
      SINC=COSB*SINE
      C1=ARSN(SINC)
      COSC=COS(C1)
      SECC=1./COSC
      TANC=TAN(C1)
      COSD=COSB*COSE/COSC
      SECD=1./COSD
      SIND=SINB/COSC
      TAND=SIND/COSD
      WE=WA*SINB+WC*COSB
```

# TABLE IX. GYRO SUBROUTINE FORTRAN LISTING (CONTINUED)

```
      WBXE=WA*COSB-WC*SINB
      WD=-WA*SINC+WB*COSC
      WDXC=WA*COSC+WB*SINC
      DWBXE=DWA*COSB-DWC*SINB-WB*WE
      DWDXC=DWA*COSC+DWB*SINC+WC*WD
      WEBXE=WE*WBXE
      DDXC=WD*WDXC
      HDDXC=HDXCD*DDXC
C
CCCCC TE COMPUTATIONS
C
      MC=-SIND*TANE*(1.+LR*SINC)
      MP=COSD*     (COSB-LR*SINB**2*SINE)/COSB
      XIM=I4C*MP
      CE=SECE*WBXE+TANE*DEXS
      CF=2.*SIND*SECC+SINB*TAN2
      CX2=SINC*SIND**2*DE+CF*DEXS.
      CA=FBS*SEC2*TANE
      CX1=TAND*DWDXC-SECD**2*DDXC-TANC*SECC*DEXS**2
      CD=SECE*(I1B*(TANE*DWBXE-SEC2*WEBXE)+BEE*WEBXE)
      CH=I2S*TANE*(SECE*DWBXE-STAN*WEBXE)
      CX3=TANC*SECC*DEXS+CF*DE
      CB=TAND*DWDXC-SECD**2*DDXC-SINC*(SIND*DE)**2
      HE1=XIM*CH+HDXCD*MP*DDXC
      HE2=EXSS*CF*DEXS-WI3*DEXS+XIM*CX1+HDXCD*MP*DDXC
C
CCCCC TEXS COMPUTATIONS
C
      XIMC=I4C*MC
      RX1=CD+CH+SE   *SECE*WBXE*DE+WI3*DE+XIMC*CB+HDXCD*MC*DDXC
      RX2=CD+CH                        +XIMC*CX1+HDXCD*DDXC*MC
C
CCCCC PRECESSION TORQUE
C
      TEXSP=(VE*SECE+VEXS*MC)*K2T
      TEP=-VEXS*K2T*MP
C
CCCCC UNBALANCE TORQUE
C
      GE=GA*SINB+GC*COSB
      GBXE=GA*COSB-GC*SINB
      GS=GA*COSB*COSE+GB*SINE-GC*SINB*COSE
      SBSE=SINB*SINE
      SECB=SINE*COSB
      GEXS=-GA*SECB+GB*COSE+GC*SBSE
      UU=UDXC*(GA*SINC-GB*COSC)+UD*(GA*COSC+GB*SINC)
      XMP=MPX*(1.-COSB+LR*SINB*SBSE)
      TEXSU=(US+UBXE*SECE-UEXS*TANE)*GE-UE*SECE*GBXE+UU*MC
     1-XMP*GE+MPX*(1.+LR*SECB)*SINB*GS
      TEU=-US*GEXS+UEXS*GS+UU*MP-XMP*GEXS
C
CCCCC FRICTION TORQUE
C
      DB=-WB+WA*COSB*TANE-WC*SINB*TANE+DEXS*SECE
      DE1=-WC*COSB-WA*SINB+DE
      DC=COSD*DE1-SIND*SINE*DB
      FDB=SIGN(1.,DB)
      IF(DB.EQ.0.) FDB=0.
      FDC=SIGN(1.,DC)
      IF(DC.EQ.0.) FDC=0.
```

## TABLE IX. GYRO SUBROUTINE FORTRAN LISTING (CONTINUED)

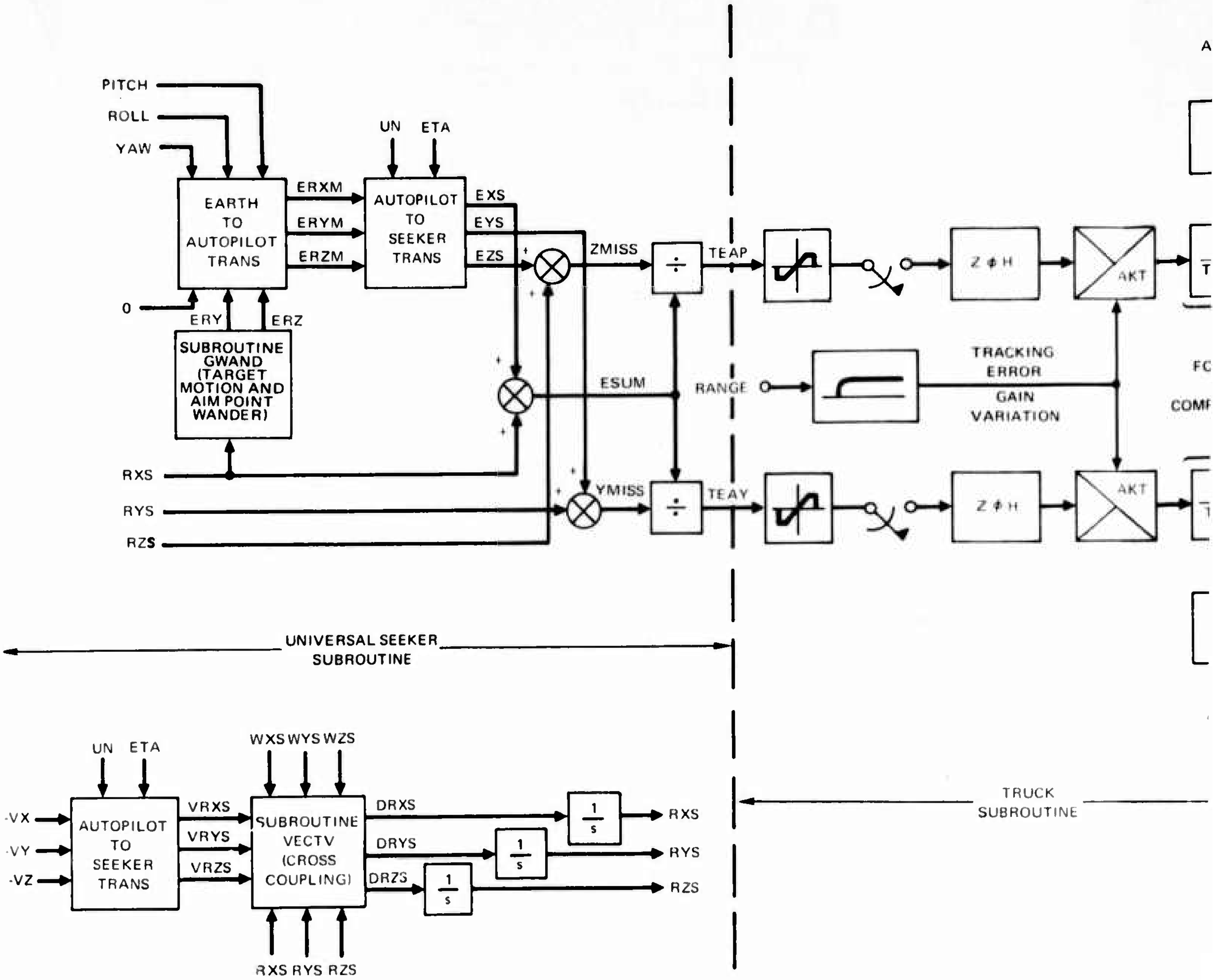
```
      FDE=SIGN(1.,DE1)
      IF(DE1.EQ.0.) FDE=0.
      AGE=ABS(GE)
      IF (AGE.LE.GEP) GO TO 230
      GEL=AGE-GEP
      GO TO 240
  230 GEL=0.
  240 CONTINUE
      FE1=F12E(1)*CF2
      FE2=(F12E(2)*GEL+F12E(3)*SQRT(GB2+GBXE**2)+F12E(4)*GEL**2)*CF3
      FE=FE1+FE2
      XFM=FC*FDC+V04C*DC
      TEXSF=-(FB   *FDB+V01B*DB)*SECE-XFM*MC
      TEF=-(FE   *FDE+V12E*DE1)-XFM*MP
C
CCCCC SPRING TORQUE
C
      TANY=TAN(Y*E1)
      XMK=K04C*COSB*TANY
      TEXSS=-(K01B*SECE+K02)*TAN(Y*B)-XMK*MC
      TES=-(K12E+K02*COSB)*TANY-XMK*MP
C
CCCCC ANISOELASTICITY TORQUES
C
      TEXSA=GS*GE*XGSI-GE*GEXS*TANE*XGEXS*GBXE*GE*SECE*XEBM
      TEA=-GEXS*GS*XIE
C
CCCCC DYNAMIC UNBALANCE TORQUE
C
      WT=W3S*TIME+THETZR
      TXDU=XU*SIN(WT)
      TEDU=XU*COS(WT)
C
CCCCC FINAL COMPUTATIONS
C
      TE=G*(TEP+DFR*TEF+DST*TES+DSU*TEU+DAN*TEA+DDU*TEDU)
      TEXS=G*(TEXSP+DFR*TEXSF+DST*TEXSS+DSU*TEXSU+DAN*TEXSA+DDU*TXDU)
      IF(GNUT.EQ.1.0) GO TO 250
C
CCCC  GYRO WITHOUT NUTATION
      CX4C=I4C*DWDXC*SIND/COSD-I4C*DGXC*SECD*SECD*HDXCD*DEXC
      TXGI=(CD*CH)+CX4C*MC
      TEGI=CX4C*MP
      TEXS=TEXS+TXGI
      TE=TE+TEGI
      DE=TEXS/W13
      DEXS=-TE/W13
      CALL INTER(DE,DEM1,DEM2,EU0,E)
      CALL INTER(DEXS,DXM1,DXM2,FX0,EXS)
      CALL INTER(DB,DBM1,DBM2,BU0,B)
      CALL INTER(DE1,DE1M1,DE1M2,E10,E1)
      GO TO 260
 250  CONTINUE
C
CCCC  GYRO WITH NUTATION
      A1=ET+XIM*COSD
      B1=-XIM*SIND*TANE
      A2=XIMC*COSD
      B2=EXST-XIMC*SIND*TANE+I1B*SEC2+I2S*TAN2
      DX=A1*B2-A2*B1
```

# TABLE IX. GYRO SUBROUTINE FORTRAN LISTING (CONCLUDED)

```
      W1=(-A2*(EXSS*CE-W13-XIM*CX3)
     1*W3*(CA*DE*EEXS*TANE*DE*SE*TANE*DE-XIMC*CX3))/DX
      W2=(A1*(CA*DEXS*SE*CE*W13-XIMC*CX2*EEXS*DEXS*TANE)*A2*XIM*CX2)/DX
      G1=(Q2*(TE-RE1)-Q1*(TEXS-RX1))/DX
      G2=(A1*(TEXS-RX2)-A2*(TE-RE2))/DX
      CALL DIF(G1,DG1,DUMA)
      CALL DIF(G2,DG2,DUMB)
      CALL GRATE(1,G1,DG1,G1Z,G1N,D1N,DUMX1)
      CALL GRATE(1,G2,DG2,G2Z,G2N,D2N,DUMX2)
      CALL DIF(DB,DDB,DUMR)
      CALL DIF(DE1,DDE1,DUMU)
      CALL GRATE(1,DB,DDB,AZ,B,DAX,DUM7)
      CALL GRATE(1,DE1,DDE1,E1Z,E1,DEX,DUM8)
      W12=W1*W2
      WN=SQRT(W12)
      FFE=(G1+W1*G2Z)/W12+ES
      CALL DIF(FFE,DFE,DUMC)
      FEXS=(G2-W2*G1Z)/W12+XS
      CALL DIF(FEXS,DFEXS,DUMD)
      CALL LDSEC(FFE,DFE,EZ,E,DE,0.,1.,WN,0.,DUMX3,DUMX4)
      CALL LDSEC(FEXS,DFEXS,EXSZ,FXS,DEXS,0.,1.,WN,0.,DUMX5,DUMX6)
  260 CONTINUE
      IF(PAIL.EQ.1.0) GO TO 270
      WYS=DEXS
      WZS=DE
      UN=B
      ETA=E1
      GO TO 280
  270 CONTINUE
      WYS=DE
      WZS=DEXS
      UN=E1
      ETA=B
  280 CONTINUE
      KX=MOD(KOUNT,KDUMP)
      IF ((KX.EQ.0 ) .OR.(COUNT.LE.5.))  WRITE(6,NAMA)
      RETURN
      END
```



(The reverse of this page is blank.)



PITCH
ROLL
YAW
EARTH TO AUTOPILOT TRANS
ERXM
ERYM
ERZM
UN
ETA
AUTOPILOT TO SEEKER TRANS
EXS
EYS
EZS
0
ERY
ERZ
SUBROUTINE GWAND (TARGET MOTION AND AIM POINT WANDER)
ZMISS
TEAP
ESUM
RANGE
TRACKING ERROR
GAIN VARIATION
YMISS
TEAY
AKT
AKT
RXS
RYS
RZS
UNIVERSAL SEEKER SUBROUTINE
UN
ETA
WXS WYS WZS
-VX
-VY
-VZ
AUTOPILOT TO SEEKER TRANS
VRXS
VRYS
VRZS
SUBROUTINE VECTV (CROSS COUPLING)
DRXS
DRYS
DRZS
1/s
RXS
RYS
RZS
RXS RYS RZS
TRUCK SUBROUTINE

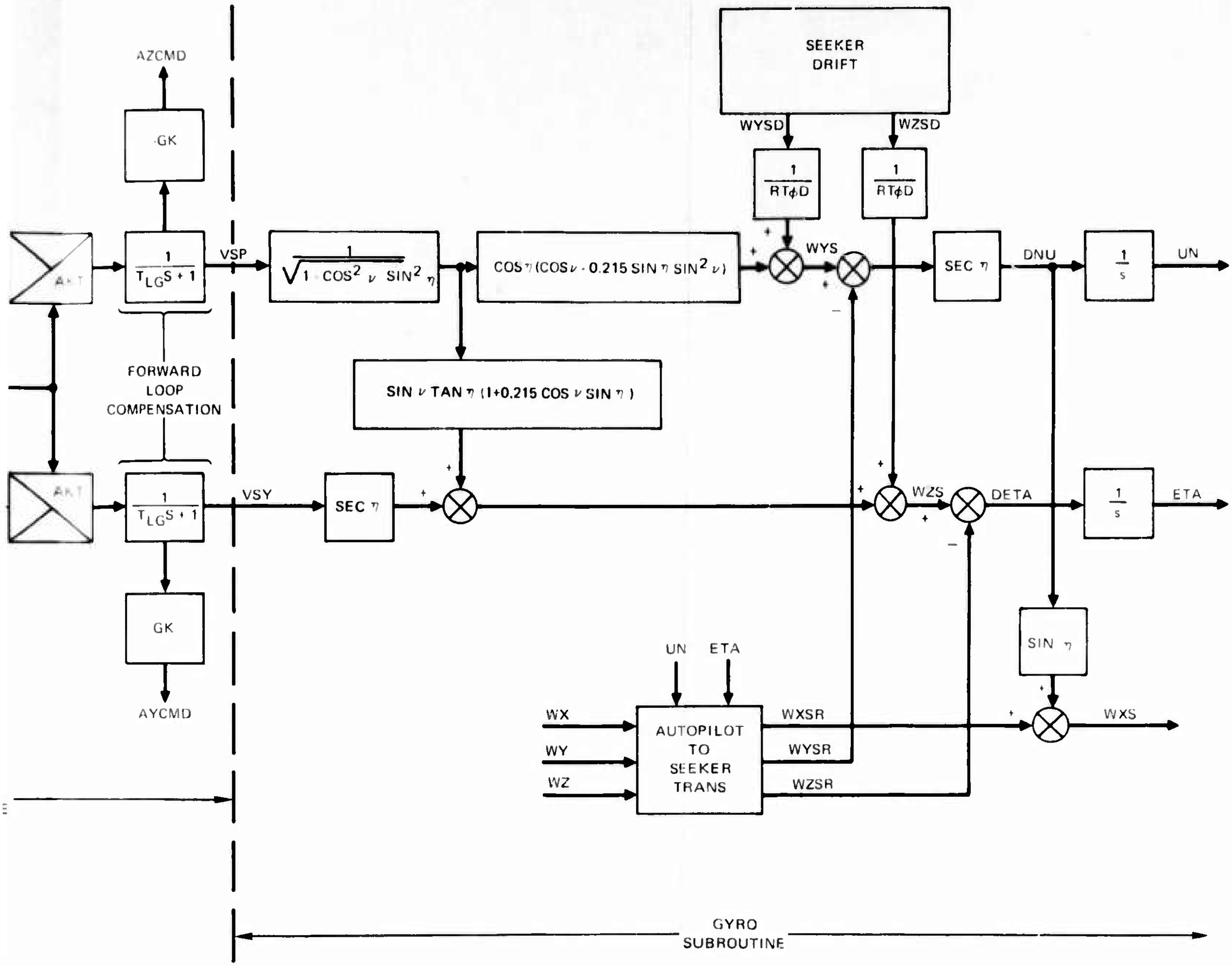



Figure 12. Universal Seeker/ Gyro Subroutine Block Diagram



(The reverse of this page is blank.)

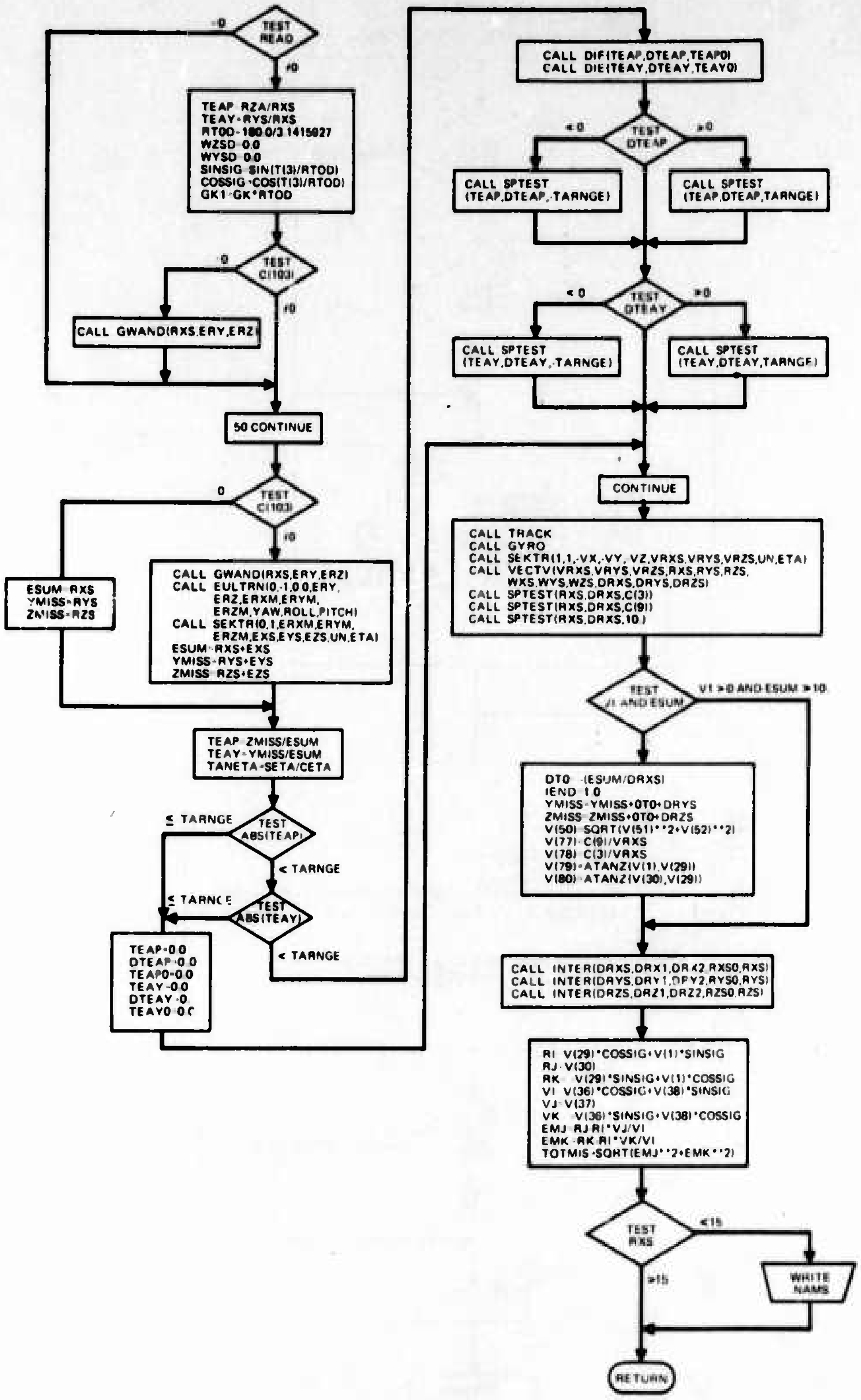



Figure 13. Seeker Subroutine Flow Chart

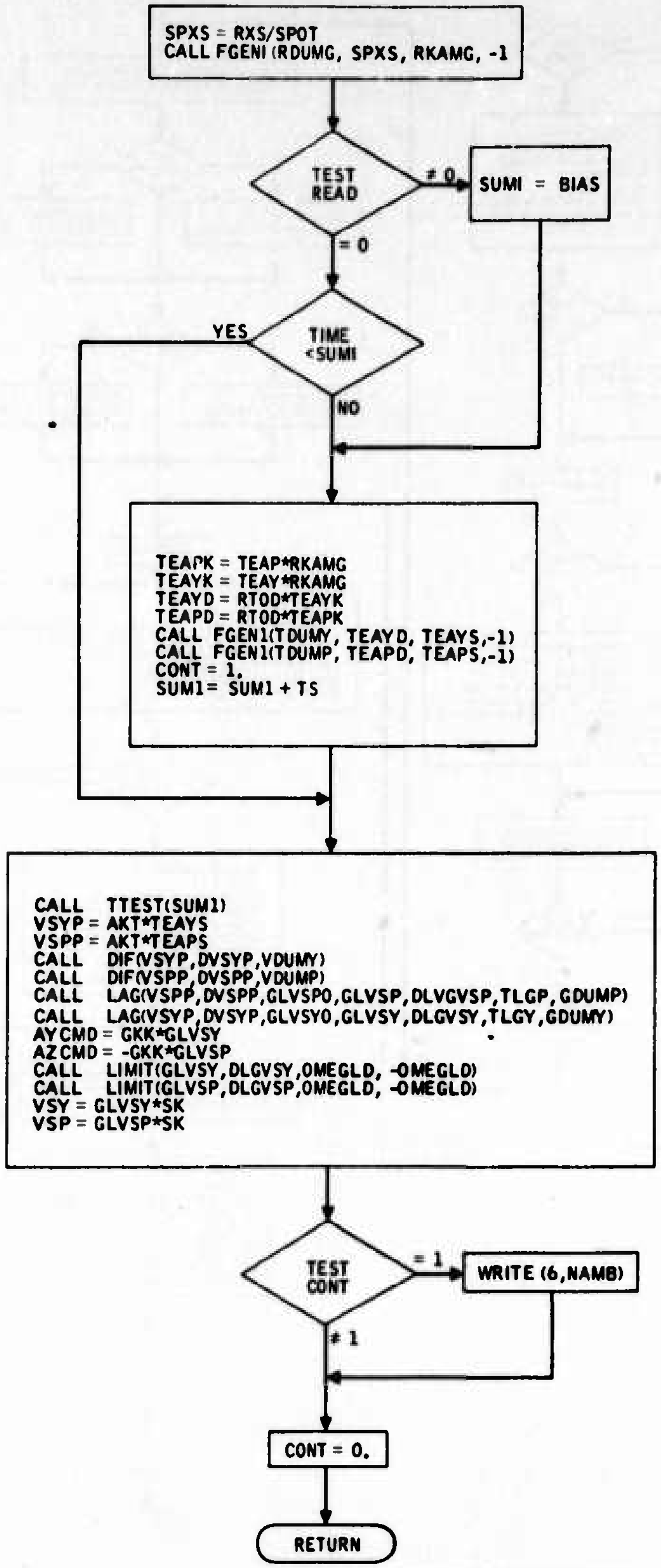
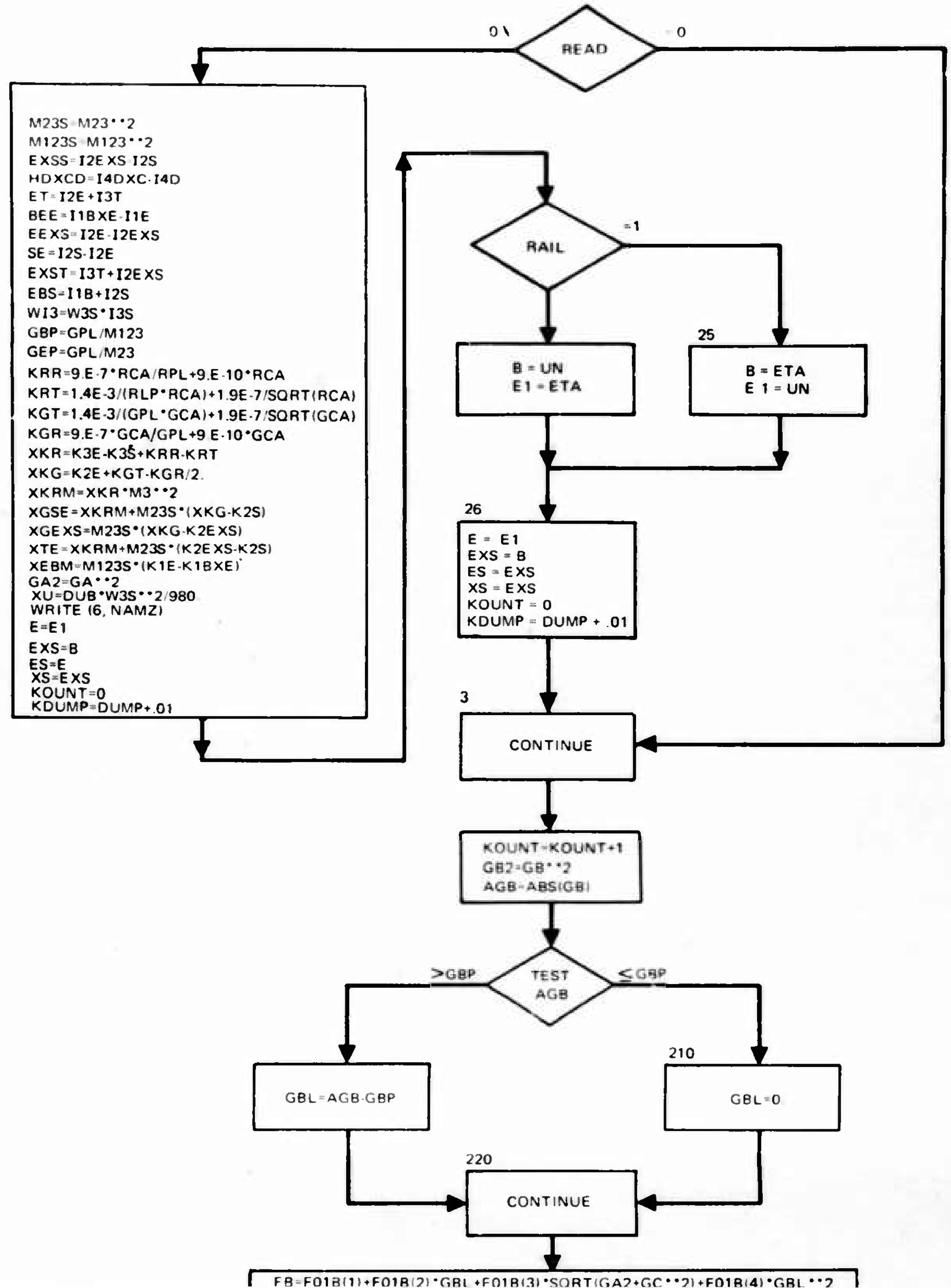



Figure 14. Universal Subroutine TRACK Flow Chart


READ
≠0
=0
M23S=M23**2
M123S=M123**2
EXSS=I2EXS·I2S
HDXCD=I4DXC·I4D
ET=I2E+I3T
BEE=I1BXE·I1E
EEXS=I2E·I2EXS
SE=I2S·I2E
EXST=I3T+I2EXS
EBS=I1B+I2S
WI3=W3S*I3S
GBP=GPL/M123
GEP=GPL/M23
KRR=9.E-7*RCA/RPL+9.E-10*RCA
KRT=1.4E-3/(RLP*RCA)+1.9E-7/SQRT(RCA)
KGT=1.4E-3/(GPL*GCA)+1.9E-7/SQRT(GCA)
KGR=9.E-7*GCA/GPL+9.E-10*GCA
XKR=K3E-K3S+KRR-KRT
XKG=K2E+KGT-KGR/2.
XKRM=XKR*M3**2
XGSE=XKRM+M23S*(XKG-K2S)
XGEXS=M23S*(XKG-K2EXS)
XTE=XKRM+M23S*(K2EXS-K2S)
XEBM=M123S*(K1E-K1BXE)
GA2=GA**2
XU=DUB*W3S**2/980
WRITE (6, NAMZ)
E=E1
EXS=B
ES=E
XS=EXS
KOUNT=0
KDUMP=DUMP+.01
RAIL
=1
B = UN
E1 = ETA
25
B = ETA
E 1 = UN
26
E = E1
EXS = B
ES = EXS
XS = EXS
KOUNT = 0
KDUMP = DUMP + .01
3
CONTINUE
KOUNT=KOUNT+1
GB2=GB**2
AGB=ABS(GB)
TEST
AGB
>GBP
≤GBP
GBL=AGB-GBP
210
GBL=0
220
CONTINUE
FB=F01B(1)+F01B(2)*GBL+F01B(3)*SQRT(GA2+GC**2)+F01B(4)*GBL**2

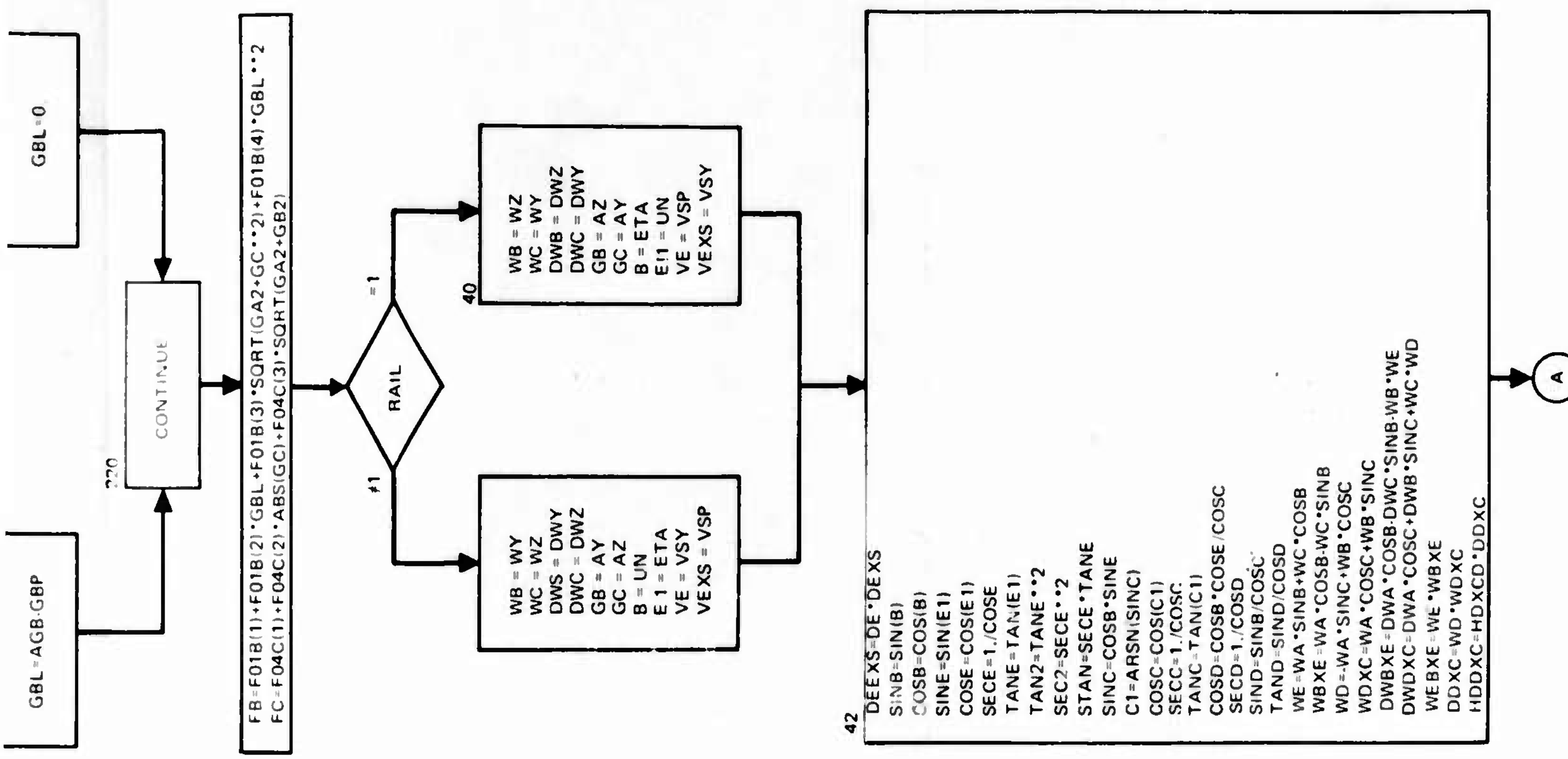
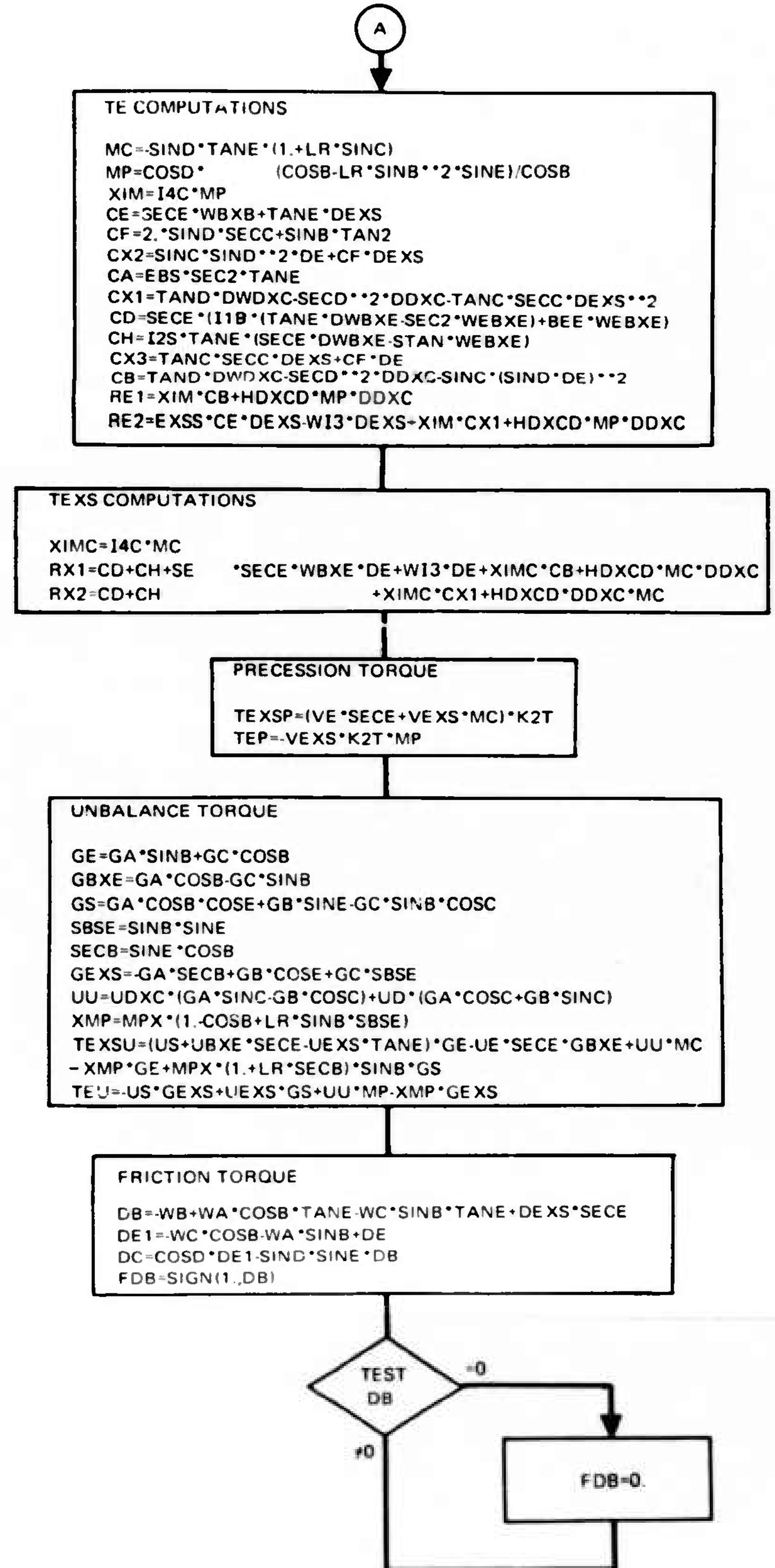



Figure 15. Gyro Subroutine Flow Chart (CAS 6 DOF Simulation)



(The reverse of this page is blank.)


A
TE COMPUTATIONS
MC=-SIND*TANE*(1.+LR*SINC)
MP=COSD* (COSB-LR*SINB**2*SINE)/COSB
XIM=I4C*MP
CE=SECE*WBXB+TANE*DEXS
CF=2.*SIND*SECC+SINB*TAN2
CX2=SINC*SIND**2*DE+CF*DEXS
CA=EBS*SEC2*TANE
CX1=TAND*DWDXC-SECD**2*DDXC-TANC*SECC*DEXS**2
CD=SECE*(I1B*(TANE*DWBXE-SEC2*WEBXE)+BEE*WEBXE)
CH=I2S*TANE*(SECE*DWBXE-STAN*WEBXE)
CX3=TANC*SECC*DEXS+CF*DE
CB=TAND*DWDXC-SECD**2*DDXC-SINC*(SIND*DE)**2
RE1=XIM*CB+HDXCD*MP*DDXC
RE2=EXSS*CE*DEXS-WI3*DEXS+XIM*CX1+HDXCD*MP*DDXC
TEXS COMPUTATIONS
XIMC=I4C*MC
RX1=CD+CH+SE *SECE*WBXE*DE+WI3*DE+XIMC*CB+HDXCD*MC*DDXC
RX2=CD+CH +XIMC*CX1+HDXCD*DDXC*MC
PRECESSION TORQUE
TEXSP=(VE*SECE+VEXS*MC)*K2T
TEP=-VEXS*K2T*MP
UNBALANCE TORQUE
GE=GA*SINB+GC*COSB
GBXE=GA*COSB-GC*SINB
GS=GA*COSB*COSE+GB*SINE-GC*SINB*COSC
SBSE=SINB*SINE
SECB=SINE*COSB
GEXS=-GA*SECB+GB*COSE+GC*SBSE
UU=UDXC*(GA*SINC-GB*COSC)+UD*(GA*COSC+GB*SINC)
XMP=MPX*(1.-COSB+LR*SINB*SBSE)
TEXSU=(US+UBXE*SECE-UEXS*TANE)*GE-UE*SECE*GBXE+UU*MC
-XMP*GE+MPX*(1.+LR*SECB)*SINB*GS
TEU=-US*GEXS+UEXS*GS+UU*MP-XMP*GEXS
FRICTION TORQUE
DB=-WB+WA*COSB*TANE-WC*SINB*TANE+DEXS*SECE
DE1=-WC*COSB-WA*SINB+DE
DC=COSD*DE1-SIND*SINE*DB
FDB=SIGN(1.,DB)
TEST DB
=0
≠0
FDB=0.

2

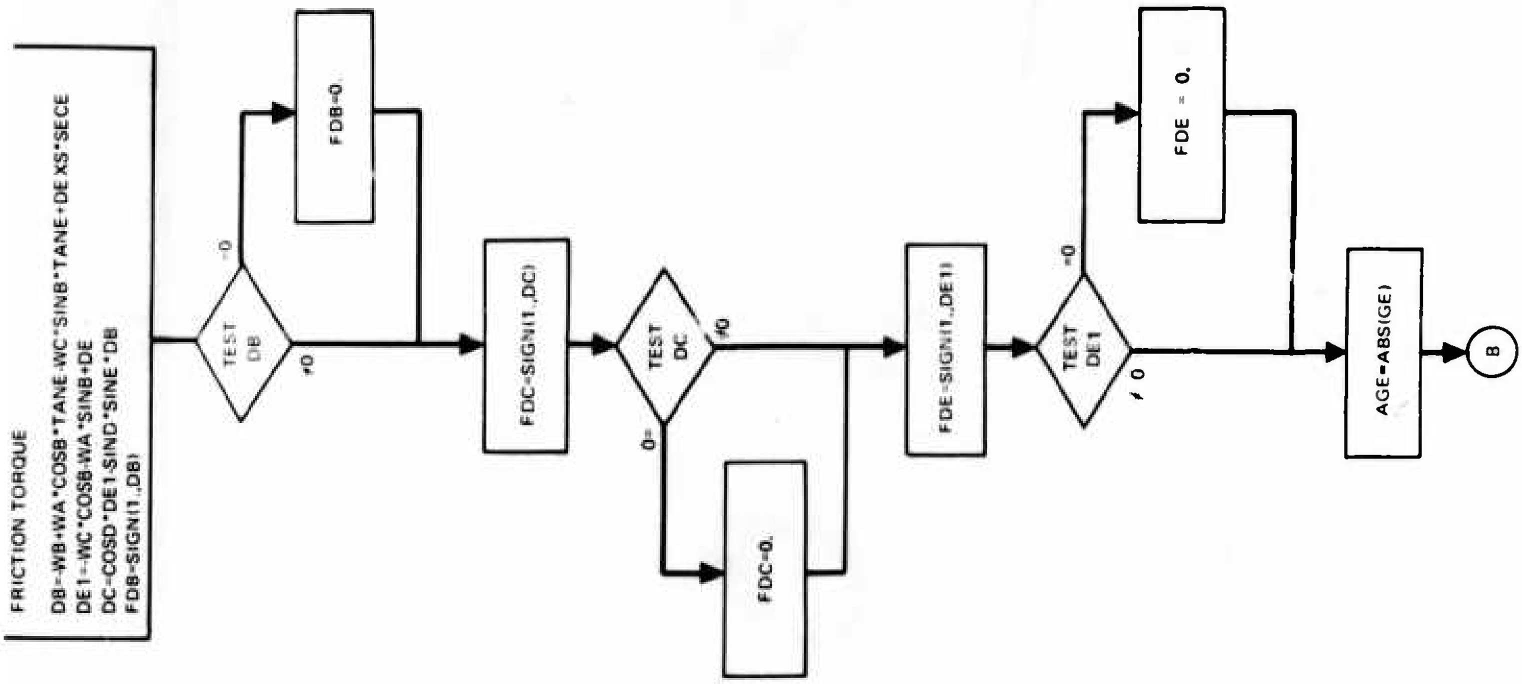
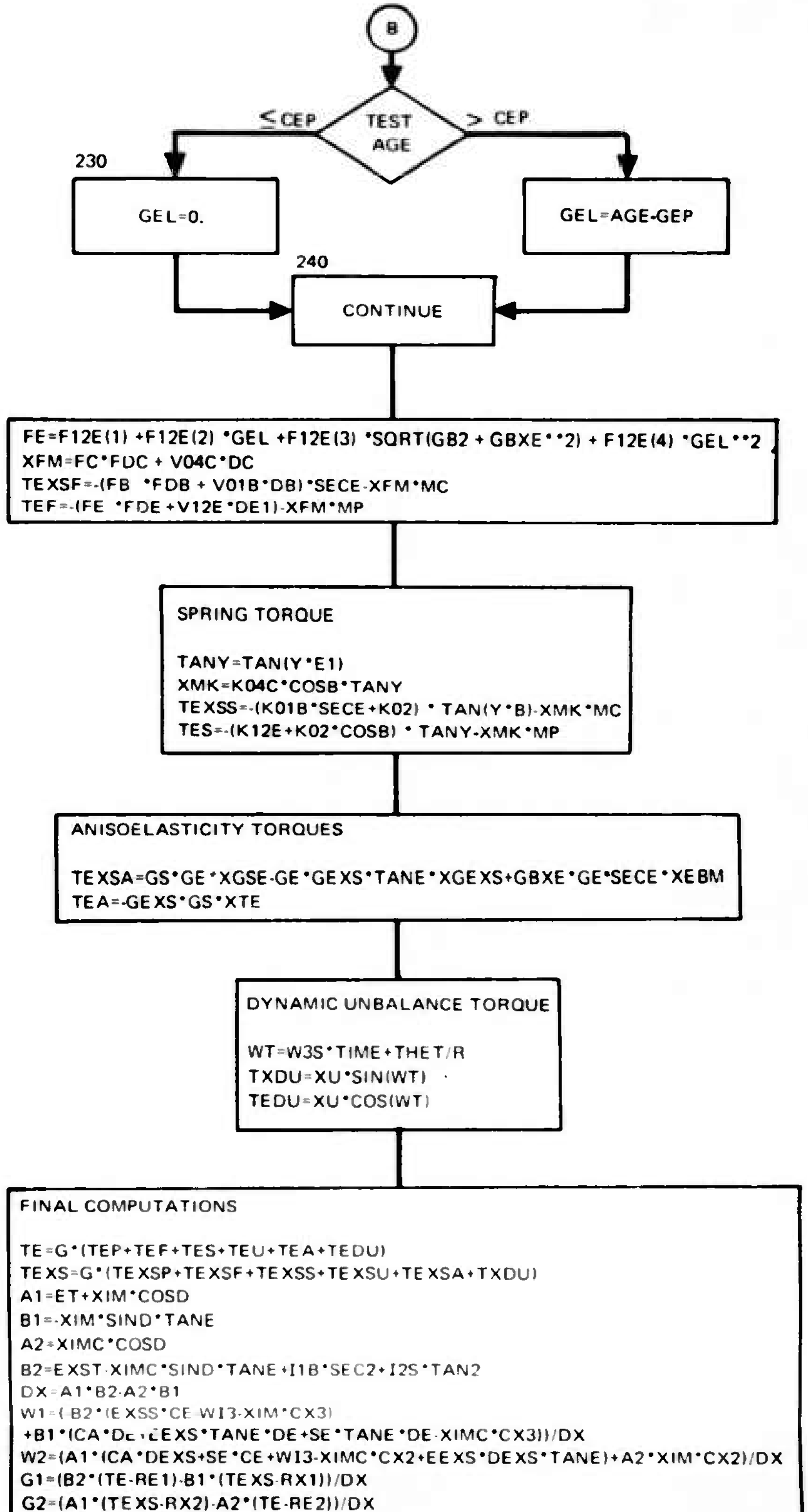



Figure 15. Gyro Subroutine Flow Chart (CAS 6 DOF Simulation) (Continued)



(The reverse of this page is blank.)



B
TEST AGE
≤ CEP
> CEP
230
GEL=0.
GEL=AGE-GEP
240
CONTINUE
FE=F12E(1) +F12E(2) *GEL +F12E(3) *SQRT(GB2 + GBXE**2) + F12E(4) *GEL**2
XFM=FC*FDC + V04C*DC
TEXSF=-(FB *FDB + V01B*DB)*SECE-XFM*MC
TEF=-(FE *FDE+V12E*DE1)-XFM*MP
SPRING TORQUE
TANY=TAN(Y*E1)
XMK=K04C*COSB*TANY
TEXSS=-(K01B*SECE+K02) * TAN(Y*B)-XMK*MC
TES=-(K12E+K02*COSB) * TANY-XMK*MP
ANISOELASTICITY TORQUES
TEXSA=GS*GE*XGSE-GE*GEXS*TANE*XGEXS+GBXE*GE*SECE*XEBM
TEA=-GEXS*GS*XTE
DYNAMIC UNBALANCE TORQUE
WT=W3S*TIME+THET/R
TXDU=XU*SIN(WT)
TEDU=XU*COS(WT)
FINAL COMPUTATIONS
TE=G*(TEP+TEF+TES+TEU+TEA+TEDU)
TEXS=G*(TEXSP+TEXSF+TEXSS+TEXSU+TEXSA+TXDU)
A1=ET+XIM*COSD
B1=-XIM*SIND*TANE
A2=XIMC*COSD
B2=EXST-XIMC*SIND*TANE+I1B*SEC2+I2S*TAN2
DX=A1*B2-A2*B1
W1=(-B2*(EXSS*CE-WI3-XIM*CX3)
+B1*(CA*DE+EEXS*TANE*DE+SE*TANE*DE-XIMC*CX3))/DX
W2=(A1*(CA*DEXS+SE*CE+WI3-XIMC*CX2+EEXS*DEXS*TANE)+A2*XIM*CX2)/DX
G1=(B2*(TE-RE1)-B1*(TEXS-RX1))/DX
G2=(A1*(TEXS-RX2)-A2*(TE-RE2))/DX

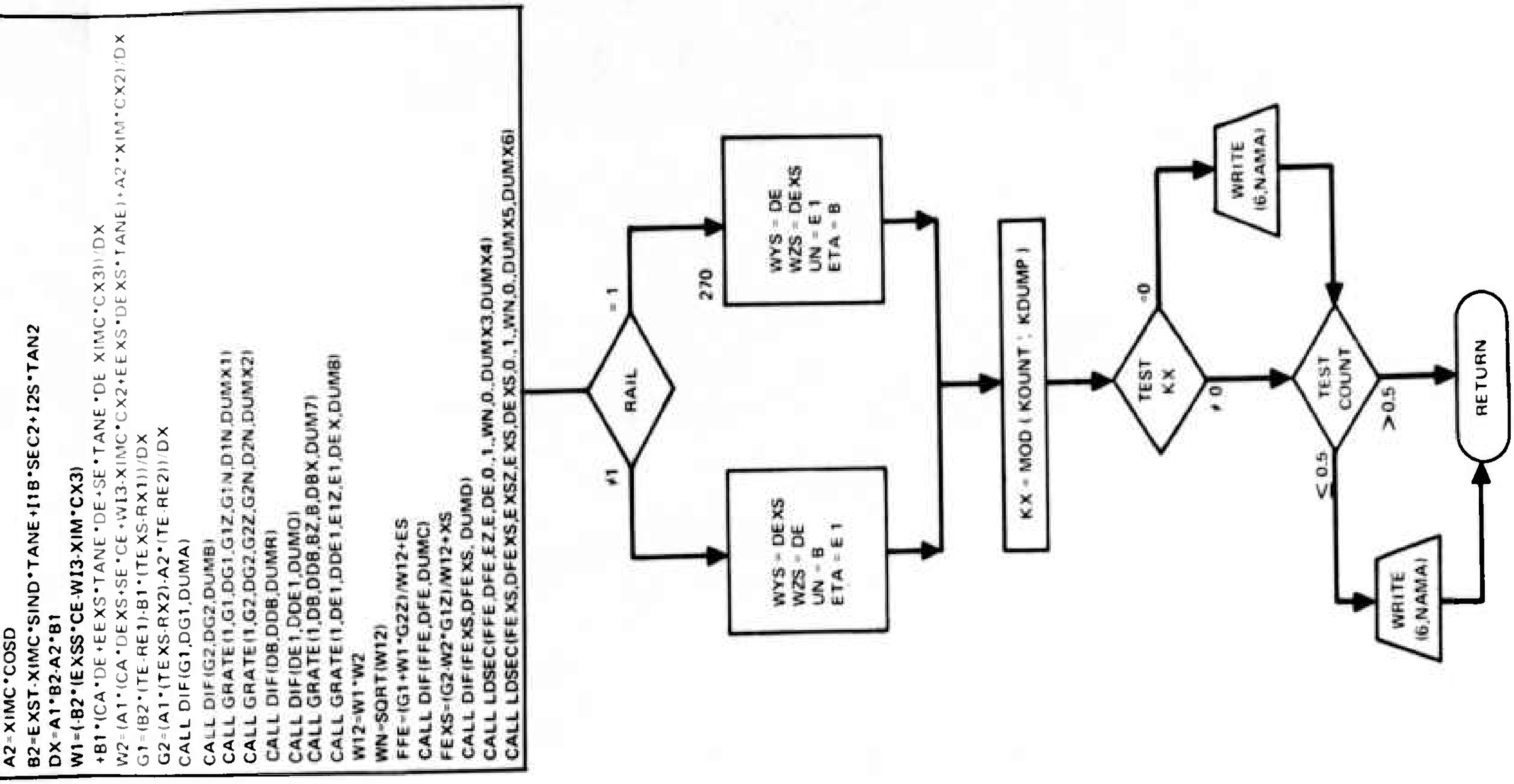



Figure 15. Gyro Subroutine
Flow Chart (CAS 6 DOF
Simulation) (Continued)


(The reverse of this page is blank.)

## TABLE X. SEEKER (MSEEK) SUBROUTINE

| Name | Quality | Units | Coordinate System |
|---|---|---|---|
| V(17) AZCMD | $A_{zc}$, Elevation maneuver command | g's | Autopilot |
| V(18) AYCMD | $A_{yc}$, Azimuth maneuver command | g's | Autopilot |
| V(22) RXS | $R_x$, Seeker boresight range | ft | Seeker |
| V(23) RYX | $R_y$, Seeker lateral range | ft | Seeker |
| V(24) RZS | $R_z$, Seeker normal range | ft | Seeker |
| V(25) TEAP | $\epsilon_p$, Tracking error angle, pitch | rad | Seeker |
| V(26) TEAY | $\epsilon_y$, Tracking error angle, yaw | rad | Seeker |
| V(27) UN | $\nu$, Seeker elevation gimbal angle | rad | |
| V(28) ETA | $\eta$, Seeker azimuth gimbal angle | rad | |
| V(53) WX<br>V(54) WY<br>V(55) WZ | $\omega_x'$, $\omega_y'$, $\omega_z'$ Missile body rates in autopilot axes | rad/sec | Autopilot |
| V(56) DWX<br>V(57) DWY<br>V(58) DWZ | $\dot{\omega}_x'$, $\dot{\omega}_y'$, $\dot{\omega}_z'$ Scalar components of missile angular acceleration in auto-pilot axes | $rad/sec^2$ | Autopilot |
| V(59) AX<br>V(60) AY<br>V(61) AZ | $A_x'$, $A_y'$, $A_z'$ Propulsive and aero-dynamic acceleration components autopilot axes | g's | Autopilot |
| V(62) VX<br>V(63) VY<br>V(64) VZ | $V_x'$, $V_y'$, $V_z'$ Missile velocity components in auto-pilot axes | ft/sec | Autopilot |
| V(66) TOTMISS | Total Miss Distance | ft | Miss Distance |
| V(70) EMJ | Y Component of Miss | ft | Miss Distance |
| V(71) EMK | Z Components of Miss | ft | Miss Distance |

TABLE X. SEEKER (MSEEK) SUBROUTINE (CONTINUED)

| Name | Quantity | Units |
|---|---|---|
| C(13) TARNGE | $\frac{\epsilon_{max}}{2}$, Half the seeker field of view | rad |
| C(14) GK | $K_g$, Guidance gain | g's/deg/sec |
| C(15) TAUAP | $\tau'_A$, Tracker time constant | sec |
| C(16) OMEGAL | $\Omega_{CL}$, Precession rate limit | rad/sec |
| C(17) C1 | Seeker drift term | rad/sec |
| C(18) C2 | Seeker drift term | rad/sec |
| C(19) C3 | Seeker drift term | rad/sec/g |
| C(20) C4 | Seeker drift term | |
| C(21) C5 | Seeker drift term | 1/sec |
| C(22) C6 | Seeker drift term | 1/sec |
| C(23) C7 | Seeker drift term | sec |
| C(24) C8 | Seeker drift term | sec |
| C(25) C9 | Seeker frift term | rad/sec/g |
| C(26) C10 | Seeker drift term | rad/sec/g |
| C(27) C11 | Seeker drift term | 1/sec/g |
| C(28) C12 | Seeker drift term | $rad/sec/g^2$ |
| C(29) C13 | Seeker drift term | $rad/sec/g^2$ |
| C(30) C14 | Seeker drift term | $rad/sec/g^2$ |
| C(31) C15 | Seeker drift term | rad/sec |
| C(32) C16 | Seeker drift term | rad/sec |
| C(33) C17 | Seeker drift term | rad/sec/g |
| C(34) C18 | Seeker drift term | |
| C(35) C19 | Seeker drift term | 1/sec |
| C(36) C20 | Seeker drift term | sec |
| C(37) C21 | Seeker drift term | rad/sec/g |
| C(38) C22 | Seeker drift term | rad/sec/g |
| C(39) C23 | Seeker drift term | $rad/sec/g^2$ |
| C(40) AK1 | $K_1$, Tracking loop velocity gain | 1/sec |

TABLE X. SEEKER (MSEEK) SUBROUTINE (CONCLUDED)

| Name | Quantity | Units |
| --- | --- | --- |
| C(41) TG | Gimbal preload | g |
| C(42) | Drift control, set to 1.0 to include drift | |
| C(109) | Not used | |
| C(110) | Not used | |
| C(111) | Not used | |
| C(112) | Not used | |
| C(113) | Not used | |
| C(114) | Not used | |
| C(115) | Not used | |

TABLE XI. TRACKER GLOSSARY OF TERMS

| Name | Quantity | Units | Coordinate System |
|---|---|---|---|
| | V Array | | |
| V(17) | $A_{zc}$, Elevation maneuver command | g | Autopilot |
| V(18) | $A_{yc}$, Azimuth maneuver command | g | Autopilot |
| V(22) | $R_x$, Seeker boresight range | ft | Seeker |
| V(25) | $\epsilon_z$, Tracking error angle, pitch | rad | Seeker |
| V(26) | $\epsilon_y$, Tracking error angle, yaw | rad | Seeker |
| V(112) | TEAYD Tracker error yaw RKAMG | deg | |
| V(113) | TEAPD Tracker error pitch RKAMG | deg | |
| V(116) | VSYP - Tracker output signal Pitch | deg/sec | |
| V(117) | VSPP - Tracker output signal Yaw | deg/sec | |
| | C Array | | |
| C(142) | SK Torquer gain coefficient | V/deg/sec | |
| C(143) | AKT - Tracker gain constant | 1sec | |
| C(144) | TS - Sampling period | sec | |
| C(145) | OMEGLD - Precession rate limit | deg/sec | |
| C(146) | GKK - Guidance gain | g/deg/sec | |
| C(147) | BIAS - Sampling rate offset bias | sec | |
| C(148) | TLDP - Tracker filter lead time constant pitch | sec | |
| C(149) | TLGP - Tracker filter lag time constant pitch | sec | |
| C(150) | TLOY - Tracker filter lead time constant-Yaw | sec | |
| C(151) | TLGY - Tracker filter lag time constant-Yaw | sec | |
| C(152) | SPOT - Tracker spot size | ft | |

## TABLE XII. GYRO GLOSSARY OF TERMS

| Name | Quantity | | Units | Coordinate System |
|---|---|---|---|---|
| | **V Array** | | | |
| V(27) | $\nu$, Seeker elevation gimbal angle | | rad | |
| V(28) | $\eta$, Seeker azimuth gimbal angle | | rad | |
| V(53) | $\omega'_x$ | Missile body rates in autopilot axes | rad/sec | Autopilot |
| V(54) | $\omega'_y$ | | | |
| V(55) | $\omega'_z$ | | | |
| V(56) | $\dot{\omega}'_x$ | Scalar components of missile angular acceleration in autopilot axes | $rad/sec^2$ | Autopilot |
| V(57) | $\dot{\omega}'_y$ | | | |
| V(58) | $\dot{\omega}'_z$ | | | |
| V(59) | $A'_x$ | Propulsive and aerodynamic acceleration components in autopilot axes | g | Autopilot |
| V(60) | $A'_y$ | | | |
| V(61) | $A'_z$ | | | |
| V(85) | DE | Total yaw precession rate | | |
| V(86) | DEXS | Total pitch precession rate | | |
| V(87) | E | Yaw gyro inertial angle | | |
| V(88) | C1 | Yaw look angle (indicated) | | |
| V(90) | G1 | Forcing function cross-coupled equation 1 | | |
| V(91) | DG1 | Derivative forcing function cross-coupled equation 1 | | |
| V(92) | G2 | Forcing function cross-coupled equation 2 | | |
| V(93) | DG2 | Derivative forcing function cross-coupled equation 2 | | |
| V(94) | G1N | Integral forcing function cross-coupled equation 1 | | |
| V(95) | G2N | Integral forcing function cross-coupled equation 2 | | |
| V(96) | FFE | Forcing function yaw axis | | |
| V(97) | DFE | Derivative forcing function yaw axis | | |

TABLE XII. GYRO GLOSSARY OF TERMS (CONCLUDED)

| Name | Quantity | Units | Coordinate System |
|---|---|---|---|
| | V Array (Continued) | | |
| V(98) | FEXS Forcing function pitch axis | | |
| V(99) | DFEXS Derivative forcing function pitch axis | | |
| | C Array | | |
| C(116) | 3S - Gryo motor speed | rad/sec | |
| C(117) | K2T - Precession torque coefficient | gcm/V | |
| C(118) | Dump program control logic | B = 0 | |
| C(119) | - Rail control logic | S = 1.0 | |
| C(136) | GNUT - Program logic control - W/O-0, W = 1.0 | | |
| C(137) | DFR Coulomb friction drift factor | Dim | |
| C(138) | DST Spring torque drift factor | Dim | |
| C(139) | DSU Unbalance drift factor | Dim | |
| C(140) | DAN Anisoelastic drift factor | Dim | |
| C(141) | DDU Dynamic unbalance factor | Dim | |
| C(153) | CF1 Friction factor coefficient | D | |
| C(154) | CF2 Friction factor coefficient | D | |
| C(155) | CF2 Friction factor coefficient | D | |

TABLE XIII. SUBROUTINE GWAND USED TO SIMULATE AIMPOINT WANDER

(This option is exercised when C(106) ≠ 0)

For

$$C(103) \le RXS \le 8.35*\ C(103),$$

$$ERY = \frac{H*RXS}{C(103)*\ C(105)*\ C(106)}$$

$$ERZ = \frac{-Z*RXS}{C(103)*\ C(105)*\ C(106)}$$

Apparent target motion is y and z earth axes

Otherwise

$$ERY = ERZ = 0.$$

Where

$$H = f_1(a)$$

$$Z = f_2(a)$$

functions $f_1$ and $f_2$ are described by function generators 1A and 1B

and

$$a = C(104)*\left[-0.563 + \sqrt{2.45 - 2.42\left(1 - \frac{C(103)}{RXS}\right)}\right]$$

or

$$a = C(104),$$ whichever is smaller

TABLE XIV. SUBROUTINE GWAND USED TO SIMULATE TARGET MOTION

(This option is exercised when C(106) = 0)

$y_T$ is target displacement in the positive earth fixed y direction.

$$y_T = V_f \left[t - \tau (1 - e^{-t/\tau})\right]$$

$$\dot{y}_T = V_f (1 - e^{-t/\tau})$$

$$\ddot{y}_T = a\, e^{-t/\tau}$$

Where

$a$ = C(103)* 32.2, initial target acceleration

$V_f$ = C(104), final target velocity

$\tau$ = $V_f/a$

$t$ = time measured from the point when boresight range equals C(105)

## TABLE XV. SUBROUTINE GWAND FORTRAN LISTING

```
$      FORTRAN DECK
GWAND           AIM POINT WANDER                                        WAND0010
      SUBROUTINE GWAND(RXS,ERY,ERZ)                                     WAND0020
      COMMON /SSAM1/ READ,DELT,AUTOT,TIME
      COMMON /SSAM2/ V  (250),T  (250),C  (250)
      EQUIVALENCE (C(103),RF),(C(104),A),(C(105),PLOTK), (C(106),PHOTOK)WAND0050
C IF C(103)IS SET TO 0, THIS SUBROUTINE WILL BE BYPASSED                 WAND0060
C IF C(106)IS NON 0,AIM POINT WANDER WILL BE SIMULATED                   WAND0070
C IF C(106)IS SET TO 0, TARGET MOTION WILL BE SIMULATED WHERE            WAND0080
C     1) C(103)=INITAL TARGET ACCEL. IN GS                               WAND0090
C     2) C(104)=FINAL TARGET VELOCITY IN FPS                             WAND0100
C     3) C(105)=SEEKER RANGE AT START OF TARGET MOTION                   WAND0110
C TARGET MOTION OBEYS THE FOLLOWING EQUATIONS                            WAND0120
C     DDY= A*EXP(-T/TAU)                                                 WAND0130
C     DY= T*A*(1-EXP(-T/TAU))                                            WAND0140
C     Y= T*A*(T-TAU*(1-EXP(-T/TAU)))                                     WAND0150
      IF (READ.EQ.0.0)GO TO 50                                          WAND0160
      GX=PLOTK*PHOTOK*RF                                                WAND0170
      ERY=0.0                                                           WAND0180
      ERZ=0.0                                                           WAND0190
      SMA=0.0                                                           WAND0200
      CALL FGEN1(IA,SMA,H,-1)                                           WAND021
      CALL FGEN1(IB,SMA,Z,-1)                                           WAND022
      IF (C(103).EQ.0.0) GO TO 100                                      WAND0230
      RTST=8.35*RF                                                      WAND0240
      AC=32.2*C(103)                                                    WAND0250
      TAC=C(104)                                                        WAND0260
      TAU=TAC/AC                                                        WAND0270
      GO TO 100                                                         WAND0280
   50 IF (C(103).EQ.0.0) GO TO 100                                      WAND0290
      IF (C(106).EQ.0.0) GO TO 200                                      WAND0300
      IF (RXS.GT.RTST) GO TO 100                                        WAND0310
      IF (RXS.LT.RF)   GO TO 100                                        WAND0320
      RHO=(1.0-RF/RXS)                                                  WAND0330
      SMA=A*(-.563+SQRT(2.45-2.42*RHO))                                 WAND0340
      IF (SMA.GT.A) SMA=A                                               WAND0350
      CALL FGEN1(IA,SMA,H,-1)                                           WAND036
      CALL FGEN1(IB,SMA,Z,-1)                                           WAND037
      ERY=H*RXS/GX                                                      WAND0380
      ERZ=-Z*RXS/GX                                                     WAND0390
  100 RETURN                                                            WAND0400
  200 IF (RXS.LT.C(105)) GO TO 250                                      WAND0410
      TSTART=TIME                                                       WAND0420
      CALL DIF (RXS,DRXS,RXSO)                                          WAND0430
      CALL SPTEST(-RXS,-DRXS,-C(105))                                   WAND0440
      GO TO 100                                                         WAND0450
  250 TTT=TIME-TSTART                                                   WAND0460
      ERY=TAC*(TTT-TAU*(1.0-EXP(-TTT/TAU)))                             WAND0470
      GO TO 100                                                         WAND0480
      END                                                               WAND0490
```

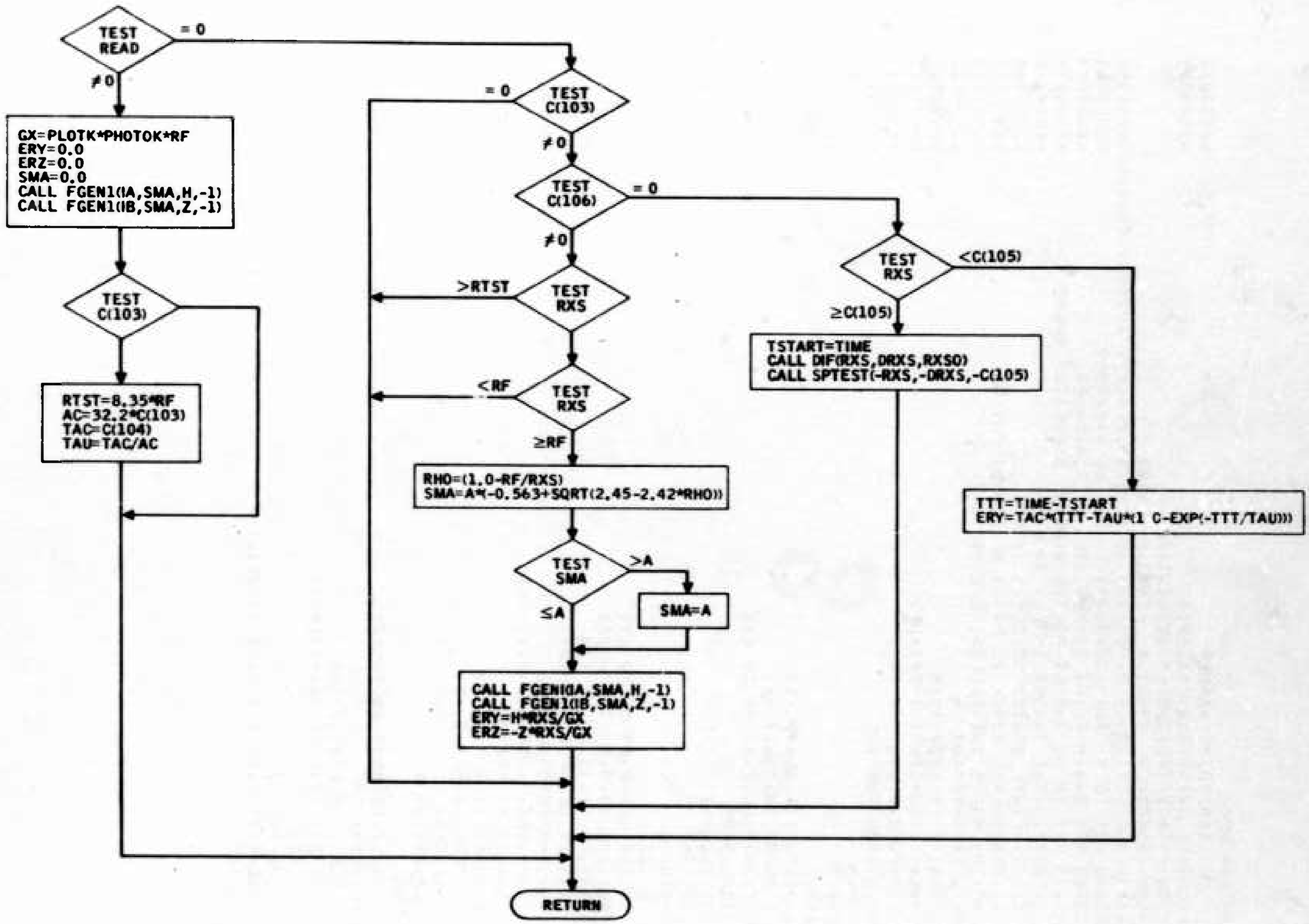



Figure 16. Subroutine GWAND Flow Chart

## TABLE XVI. SUBROUTINE ARB FORTRAN LISTING

```
$      FORTRAN DECK
CARB        GUIDANCE LAW MODIFICATION
      SUBROUTINE ARB
      COMMON /SSAM2/ V  (250),T  (250),C  (250)
      COMMON /SSAM1/ READ,DELT,AUTOT,TIME
      EQUIVALENCE
     1 (V( 1),ALT   ),(V(2 ),DAC   ),(V( 3),DPC   ),(V( 4),DYC   ),
     2 (V( 5),DA    ),(V( 6),DP    ),(V( 7),DY    ),(V( 8),VXM   ),
     3 (V( 9),VYM   ),(V(10),VZM   ),(V(11),WX    ),(V(12),WY    ),
     4 (V(13),WZ    ),(V(14),AXM   ),(V(15),AYM   ),(V(16),AZM   ),
     5 (V(17),AZCMD ),(V(18),AYCMD ),(V(19),YAW   ),(V(20),ROLL  ),
     6 (V(21),PITCH ),(V(22),RXS   ),(V(23),RYS   ),(V(24),RZS   ),
     7 (V(25),TEAP  ),(V(26),TEAY  ),(V(27),SEGA  ),(V(28),SAGA  ),
     8 (V(29),RX    ),(V(30),RY    ),(V(31),GEAP  ),(V(32),GEAY  ),
     9 (V(33),ALPHA ),(V(34),ALPHAP),(V(35),ALPHAY),(V(36),VXE   )
      EQUIVALENCE
     1 (V(37),VYE   ),(V(38),VZE   ),(V(39),Q     ),(V(40),VM    ),
     2 (V(41),AM    ),(V(42),ACP   ),(V(43),ACY   ),(V(44),DWX   ),
     3 (V(45),DWY   ),(V(46),DWZ   ),(V(47),DDAC  ),(V(48),DDPC  ),
     5 (V(49),DDYC  ),(V(50),XSMISS),(V(51),YSMISS),(V(52),ZSMISS),
     6 (V(53),WXP   ),(V(54),WYP   ),(V(55),WZP   ),(V(56),DWXP  ),
     7 (V(57),DWYP  ),(V(58),DWZP  ),(V(59),AXP   ),(V(60),AYP   ),
     8 (V(61),AZP   ),(V(62),VXP   ),(V(63),VYP   ),(V(64),VZP   )
      EQUIVALENCE
     1 (C(109),PSIP0),(C(110),BP),(C(111),AKSIQP)
      IF(READ.EQ.0.) GO TO 10
      PSIPN=0.
   10 ERROR=SEGA*57.295/A*C(7)
      IF (ERROR.GT.0.0) ERROR=0.0
      CALL DIL  (ERROR,DE,DEI)
      CALL LAG (DEI,DE,BIASE,BIASEN,DBIAS,C(8),DUMDUM)
      DPSIP=AZCMD/C(14)
      CALL  INTER(DPSIP,DUM1,DUM2,PSIP,PSIPN)
      PSIDFP=PSIP-PSIP0
      IF(PSIDFP.LT.0.) PSIDFP=0.
      ACP=AZCMD-BP+AKSIQP*PSIDFP
      ACY=AYCMD+T(20)
   40 ACP=ACP-BIASE*C(6)+  T(19)
      IF (READ.EQ.0.) GO TO 50
      ACP=0.
      ACY=0.
   50 RETURN
      END
```

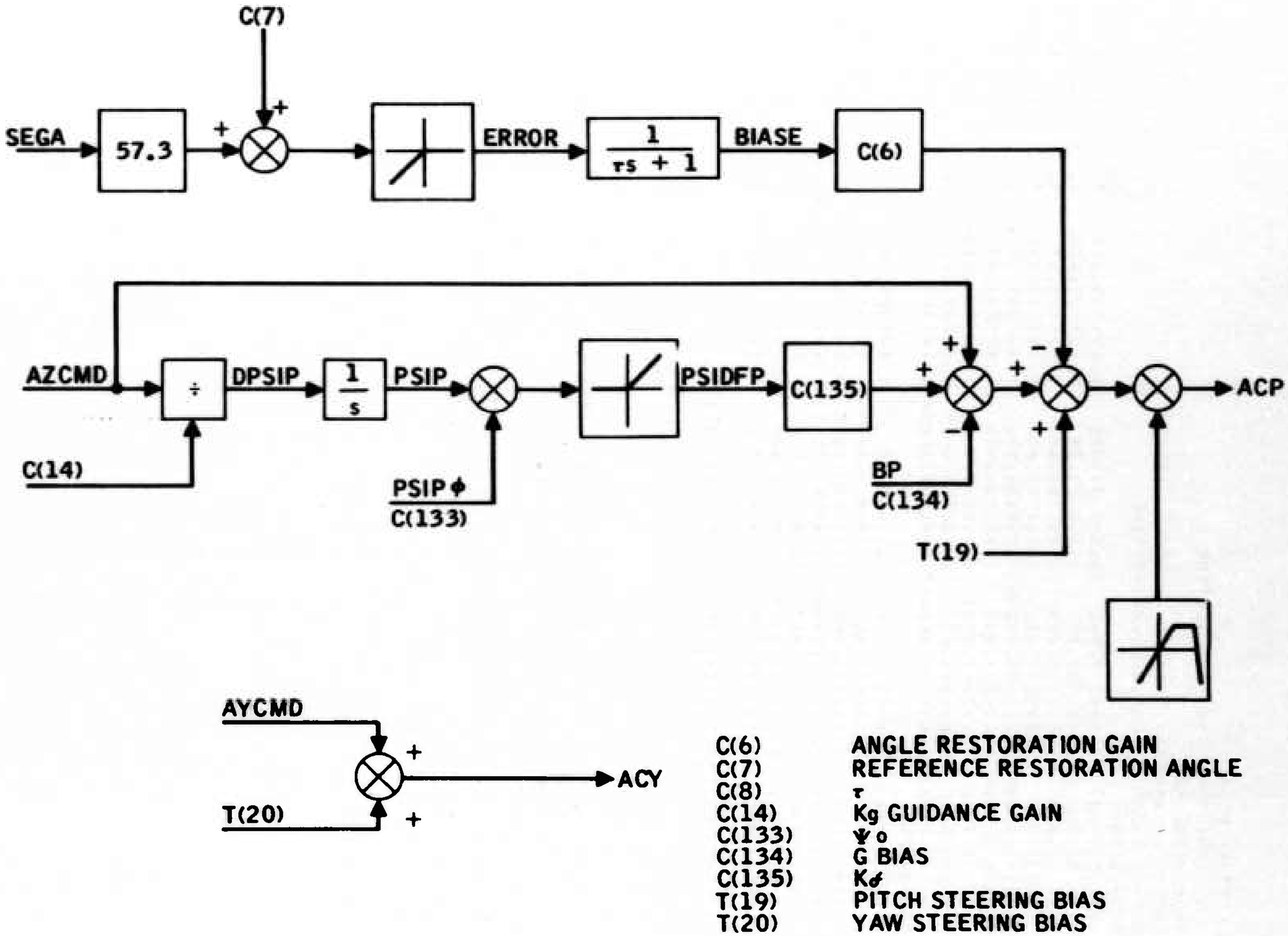



Figure 17. Subroutine ARB Block Diagram

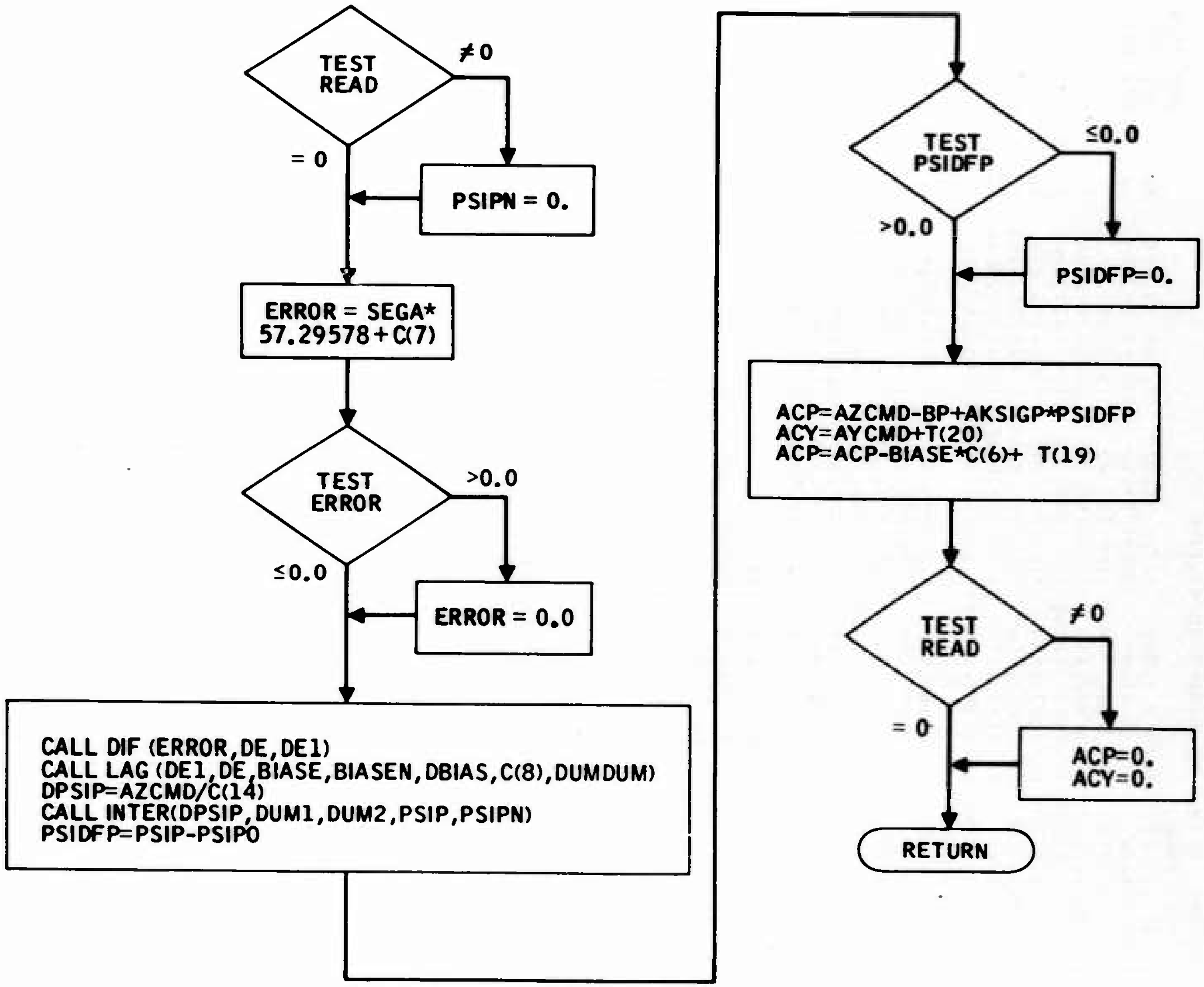



Figure 18. Subroutine ARB Flow Chart

# TABLE XVII. SUBROUTINE BRF FORTRAN LISTING

```
$      FORTRAN DECK
CBRF        BLIND RANGE FILTER                                          BRF 0010
      SUBROUTINE BRF                                                    BRF 0020
      COMMON /SSAM2/ V  (250),T  (250),C  (250)
      COMMON /SSAM1/ READ,DELT,AUTOT,TIME
      EQUIVALENCE                                                       BRF 0070
     1 (V( 1),ALT   ),(V(2 ),DAC   ),(V( 3),DPC   ),(V( 4),DYC   ),    BRF 0080
     2 (V( 5),DA    ),(V( 6),DP    ),(V( 7),DY    ),(V( 8),VXM   ),    BRF 0090
     3 (V( 9),VYM   ),(V(10),VZM   ),(V(11),WX    ),(V(12),WY    ),    BRF 0100
     4 (V(13),WZ    ),(V(14),AXM   ),(V(15),AYM   ),(V(16),AZM   ),    BRF 0110
     5 (V(17),AZCMD ),(V(18),AYCMD ),(V(19),YAW   ),(V(20),ROLL  ),    BRF 0120
     6 (V(21),PITCH ),(V(22),RXS   ),(V(23),RYS   ),(V(24),RZS   ),    BRF 0130
     7 (V(25),TEAP  ),(V(26),TEAY  ),(V(27),SEGA  ),(V(28),SAGA  ),    BRF 0140
     8 (V(29),RX    ),(V(30),RY    ),(V(31),GEAP  ),(V(32),GEAY  ).    BRF 0150
     9 (V(33),ALPHA ),(V(34),ALPHAP),(V(35),ALPHAY),(V(36),VXE   )     BRF 0160
      EQUIVALENCE                                                       BRF 0170
     1 (V(37),VYE   ),(V(38),VZE   ),(V(39),Q     ),(V(40),VM    ),    BRF 0180
     2 (V(41),AM    ),(V(42),ACP   ),(V(43),ACY   ),(V(44),DWX   ),    BRF 0190
     3 (V(45),DWY   ),(V(46),DWZ   ),(V(47),DDAC  ),(V(48),DDPC  ),    BRF 0200
     5 (V(49),DDYC  ),(V(50),TSMISS),(V(51),YSMISS),(V(52),ZSMISS),    BRF 0210
     6 (V(53),WXP   ),(V(54),WYP   ),(V(55),WZP   ),(V(56),DWXP  ),    BRF 0220
     7 (V(57),DWYP  ),(V(58),DWZP  ),(V(59),AXP   ),(V(60),AYP   ),    BRF 0230
     8 (V(61),AZP   ),(V(62),VXP   ),(V(63),VYP   ),(V(64),VZP   )     BRF 0240
      IF (READ.EQ.0.0) GO TO 5                                          BRF 0250
      COSSIG=COS(T  (3)*3.1415927/180.0)
      SINSIG=SIN(T  (3)*3.1415927/180.0)
    5 IF (RXS.LT.C(9)) GO TO 10                                         BRF 0280
      CALL DIF(V(43),DSIGY,DUM2)                                        BRF 0290
      CALL LAG(V(43),DSIGY,XXY0,XXYN,DXXY,C(5),DUM6)                    BRF 0300
      V(75)=V(73)                                                       BRF 0310
      GO TO 20                                                          BRF 0320
   10 ACY=XXYN                                                          BRF 0330
   20 IF (RXS.LT.C(3)) GO TO 40                                         BRF 0340
      CALL DIF(V(42),DSIGP,DUM1)                                        BRF 0350
      CALL LAG(V(42),DSIGP.XXP0,XXPN,DXXP,C(5),DUM5)                    BRF 0360
      V(76)=-V(72)*SINSIG+V(74)*COSSIG                                  BRF 0370
      GO TO 50                                                          BRF 0380
   40 ACP=XXPN                                                          BRF 0390
   50  RETURN                                                           BRF 0400
      END                                                               BRF 0410
```

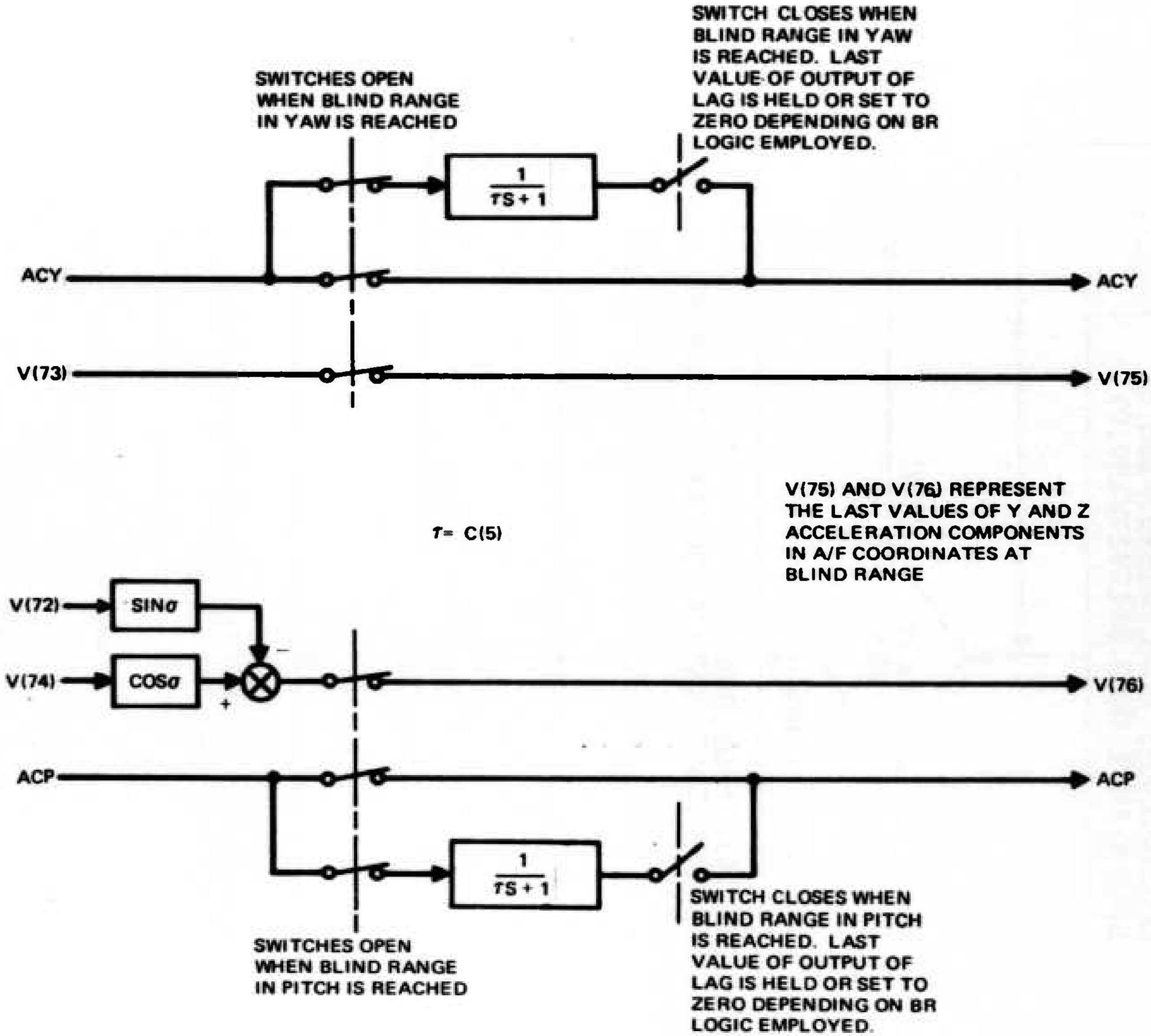



Figure 19. Subroutine BRF Block Diagram

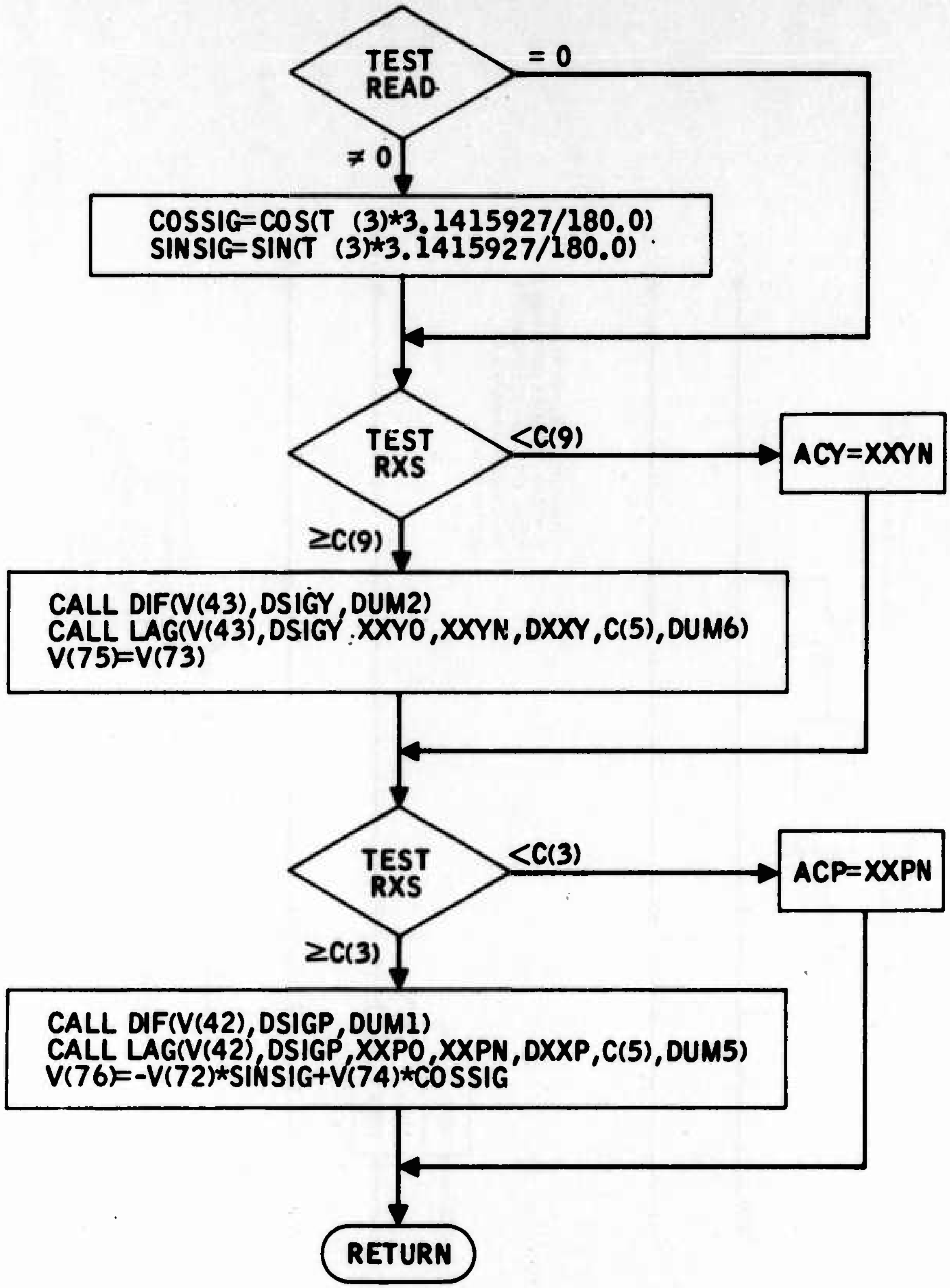



Figure 20. Subroutine BRF Flow Chart

### 2.3.8 Autopilot Subroutine (MPILOT)

(U) This subroutine simulates the behavior of the missile autopilot. It accepts inputs from the ARB and BRF subroutines as well as linear and angular acceleration components of the missile. Its outputs are the command control deflections in yaw, pitch and roll. A FORTRAN listing of this subroutine appears in Table XVIII. The corresponding flow chart and block diagram appear in Figure 21 and 22, respectively. A glossary of terms used in this subroutine appears in Table XIX.

### 2.3.9 Flipper Subroutine

(U) The Flipper subroutine simply accepts the outputs of the Autopilot subroutine and processes them to obtain the actual control surface deflections in the three control axes. The listing, block diagram, and flow chart are shown in Table XX and Figures 23 and 24, respectively. A glossary of terms appears in Table XXI.

### 2.3.10 Aerodynamic Subroutine

(U) The Aerodynamic (Aero) subroutine is the most complex of the entire program. The complete set of aerodynamic equations for forces and moments on the missile (as available from wind tunnel tests or actual flight tests) is programmed. Table XXII contains the subroutine listing, and Figure 25 is the block diagram of this portion of the simulation. A flow chart is shown in Figure 26, and a glossary of terms is contained in Table XXIII.

(U) The aerodynamic data is readily available only in maneuver axes; therefore, the force and moment coefficients are generated in these axes. Because of the complexity of the final equations of motion, the aerodynamic coefficients are generated in a series of steps which are labeled intermediate expressions, secondary expressions, and primary expressions; these steps are shown in sections (A), (B), and (C), respectively, of Table XXIV. It is desirable to integrate the equations of motion in missile coordinates; so the coefficient generates in maneuver axes are transformed to missile axes, as shown by the equations in Table XXV. The final six equations of motion in missile axes are shown in Table XXVI.

(U) This subroutine contains numerous parameters which are input as function generators and are used in the generation of the aerodynamic coefficients. This is achieved through curve fitting techniques applied to raw aerodynamic data. A separate list of the parameters contained in the function generators is supplied in Table XXVII.

(U) Among the options available in this subroutine are the selection of different values of missile mass and moments of inertia during several stages of thrusting. Three points in time are chosen (corresponding to the thrust interval) and corresponding values of mass and moments of inertia are also selected. During the portion of the simulation contained within this initial interval, the mass and moments of inertia are varied linearly between the selected values.

# TABLE XVIII. AUTOPILOT SUBROUTINE FORTRAN LISTING

```
$      FORTRAN DECK
CPILC         MAVERICK AUTOPILOT                                          PIL00010
      SUBROUTINE MPILOT                                                   PIL00020
      COMMON /SSAM1/ READ,DELT,AUTOT,TIME
      COMMON /SSAM2/ V  (250),T  (250),C  (250)
      EQUIVALENCE                                                         PIL00050
     1 (V(12),WY    ),(V(13),WZ    ),(V(11),WX    ),(V(20),PHI   ),       PIL00060
     2 (V(42),ACP   ),(V(14),AX    ),(V(43),ACY   ),(V(2 ),DAC   ),       PIL00070
     3 (V(3 ),DPC   ),(V(4 ),DYC   )                                      PIL00080
      EQUIVALENCE                                                         PIL00090
     1 (V(60),AY    ),(V(61),AZ    ),(V(47),DDAC  ),(V(48),DDPC  ),       PIL00100
     2 (V(44),DWX   ),(V(45),DWY   ),(V(46),DWZ   ),(V(49),DDYC  )        PIL00110
      EQUIVALENCE                                                         PIL00120
     1(C(43),GSW   ),(C(44), AK   ),(C(45), BJ   ),(C(46),PHIJ  ),        PIL00130
     2(C(47),ACCLIM),(C(48),DIFLIM),(C(49),TYALD ),(C(50),TYALG ),        PIL00140
     3(C(51),TYPLD ),(C(52),TYPLG ),(C(53),DEADT ),(C(54),GROLIM),        PIL00150
     4(C(55),TRCLG ),(C(56),RSW   ),(C(57),TAUACC),(C(58),TAURG ),        PIL00160
     5(C(59),DPHIJ ),(C(60),DALT  ),(C(61),TR1   ),(C(62),TR2   ),        PIL00170
     6(C(63),DAL1  ),(C(64),DAL2  ),(C(65),WGL   )                        PIL00180
C     C(43) THROUGH C(72) RESERVED FOR THIS SUBROUTINE                    PIL00190
      IF (READ.EQ.0.0) GO TO 50                                           PIL00200
      CON=180.0/3.1415927                                                 PIL00210
      RK=CON*BJ                                                           PIL00220
      PHIK=CON*PHIJ                                                       PIL00230
      DPHIK=DPHIJ*CON                                                     PIL00240
      YIN1=0.0                                                            PIL00250
      PIN1=0.0                                                            PIL00260
      PHIN=PHI                                                            PIL00270
      PHIO=PHI                                                            PIL00280
      DPHI=0.0                                                            PIL00290
      ISW=0                                                               PIL00300
      DAL=DAL1                                                            PIL00310
      PHIO=T(6)/(TRCLG*CON)                                               PIL00320
      PHIN=T(6)/(TRCLG*CON)                                               PIL00330
      YIN30=0.                                                            PIL00340
      WXL=0.0                                                             PIL00350
      WXLO=0.0                                                            PIL00360
      THETAX=0.0                                                          PIL00370
      THTAXN=0.0                                                          PIL00380
      DAC1=0.0                                                            PIL00390
      AZN=0.0                                                             PIL00400
      AZ1=0.0                                                             PIL00410
      PIN2=0.0                                                            PIL00420
      PIN2O=0.0                                                           PIL00430
      WYL=0.0                                                             PIL00440
      WYLO=0.0                                                            PIL00450
      PIN3=0.0                                                            PIL00460
      PIN3O=0.0                                                           PIL00470
      AYN=0.0                                                             PIL00480
      AY1=0.0                                                             PIL00490
      YIN2=0.0                                                            PIL00500
      YIN2O=0.0                                                           PIL00510
      WZL=0.0                                                             PIL00520
      WZLO=0.0                                                            PIL00530
      YIN3=0.0                                                            PIL00540
      AZO=0.                                                              PIL00550
      DUPIN=0.                                                            PIL00560
      AYO=0.                                                              PIL00570
      DUYIN=0.                                                            PIL00580
      YIN3O=0.                                                            PIL00590
```

## TABLE XVIII. AUTOPILOT SUBROUTINE FORTRAN LISTING (CONCLUDED)

```
      DM1=0.                                                            PIL00600
      DM2=0.                                                            PIL00610
      DM3=0.                                                            PIL00620
      DM4=0.                                                            PIL00630
      DM5=0.                                                            PIL00640
      DM6=0.                                                            PIL00650
      DM7=0.                                                            PIL00660
      DM8=0.                                                            PIL00670
      DM9=0.                                                            PIL00680
      DM10=0.                                                           PIL00690
      DM11=0.                                                           PIL00700
      DM12=0.                                                           PIL00710
      GO TO 65                                                          PIL00720
C                                                                       PIL00730
C     PITCH CONTROL                                                     PIL00740
   50 IF(ABS(WX).GE.RSW) ISW=1                                          PIL00750
      IF((AX.GT.GSW).OR.(TIME.LT.DEADT)) GO TO 65                       PIL00760
   60 CALL DIF(AZ,DAZZ,AZO)                                             PIL00770
      CALL LAG(AZ,DAZZ,AZ1,AZN,DAZ,TAUACC,DM1)                          PIL00780
      CALL DIF(ACP,DACP,ACPO)                                           PILO 785
      CALL LIMIT(ACP,DACP,ACCLIM,-ACCLIM)                               PIL00790
      PIN1=AK*(ACP-AZ1)                                                 PIL00800
      CALL DIF (PIN1,DPIN,DDPIN)                                        PIL00810
      CALL LDLAG(PIN1,DPIN ,PIN20,PIN2,DPIN2,TYALD,TYALG,DM2)           PIL00820
      CALL LIMIT(PIN2,DPIN2,DIFLIM,-DIFLIM)                             PIL00830
      CALL LAG(WY,DWY,WYLO,WYL,DWYL,TAURG,DM3)                          PIL00840
      CALL LIMIT(WYL,DWYL,WGL,-WGL)                                     PIL00850
      CALL LDLAG(WYLO*BK,DWYL*BK,PIN30,PIN3,DPIN3,TYDLD,TYDLG,DM4)      PIL00860
C                                                                       PIL00870
C     YAW CONTROL                                                       PIL00880
      CALL DIF(AY,DAYY,AYO)                                             PIL00890
      CALL LAG(AY,DAYY,AY1,AYN,DAY,TAUACC,DM5)                          PIL00900
      CALL DIF(ACY,DACY,ACYO)                                           PILO 905
      CALL LIMIT(ACY,DACY,ACCLIM,-ACCLIM)                               PIL00910
      YIN1=AK*(AY1-ACY)                                                 PIL00920
      CALL DIF(YIN1,DYIN,DDYIN)                                         PIL00930
      CALL LDLAG(YIN1,DYIN ,YIN20,YIN2,DYIN2,TYALD,TYALG,DM6)           PIL00940
      CALL LIMIT(YIN2,DYIN2,DIFLIM,-DIFLIM)                             PIL00950
      CALL LAG(WZ,DWZ,WZLO,WZL,DWZL,TAURG,DM7)                          PIL00960
      CALL LIMIT(WZL,DWZL,WGL,-WGL)                                     PIL00970
      CALL LDLAG(WZLO*BK,DWZL*BK,YIN30,YIN3,DYIN3,TYDLD,TYDLG,DM8)      PIL00980
      DPC=PIN20+PIN30+YIN20                                             PIL00990
      DDPC=DPIN2+DPIN3+DYIN2                                            PIL01000
      DYC=YIN20+YIN30-PIN20                                             PIL01010
      DDYC=DYIN2+DYIN3-DPIN2                                            PIL01020
C                                                                       PIL01030
C     ROLL CONTROL                                                      PIL01040
   65 IF((TIME.LT.DEADT).AND.(ISW.EQ.0)) GO TO 100                      PIL01050
   70 CALL LAG(WX,DWX,WXLO,WXL,DWXL,TAURG,DM9)                          PIL01060
      CALL LIMIT(WXL,DWXL,GROLIM,-GROLIM)                               PIL01070
      CALL LAG( 0.0, 0.0,PHIO,PHIN,DPHI,TRCLG,DM10)                     PIL01080
      WX1=WXLO+PHIO+T(21)                                               PIL01090
      DWX1=DWXL+DPHI                                                    PIL01100
   90 CALL GRATE(1,WX1,DWX1,THETAX,TETAXN,DTAX,DM11)                    PIL01110
      RIN1=DPHIK*WX1+PHIK*THETAX                                        PIL01120
      DRIN1=DPHIK*DWX1+PHIK*DTAX                                        PIL01130
      CALL LDLAG(RIN1,DRIN1,DAC,DAC1,DDAC,TR1,TR2,DM12)                 PIL01140
      IF (TIME.GT.DALT) DAL=DAL2                                        PIL01150
      CALL LIMIT(DAC1,DDAC,DAL,-DAL)                                    PIL01160
  100 CALL DIF(AX,DAXX,DAXXO)                                           PIL01170
      CALL SPTEST(-AX,-DAXX,-GSW)                                       PIL01180
      CALL TTEST(DEADT)                                                 PIL01190
      CALL TTEST(DALT)                                                  PIL01200
      RETURN                                                            PIL01210
      END                                                               PIL01220
```

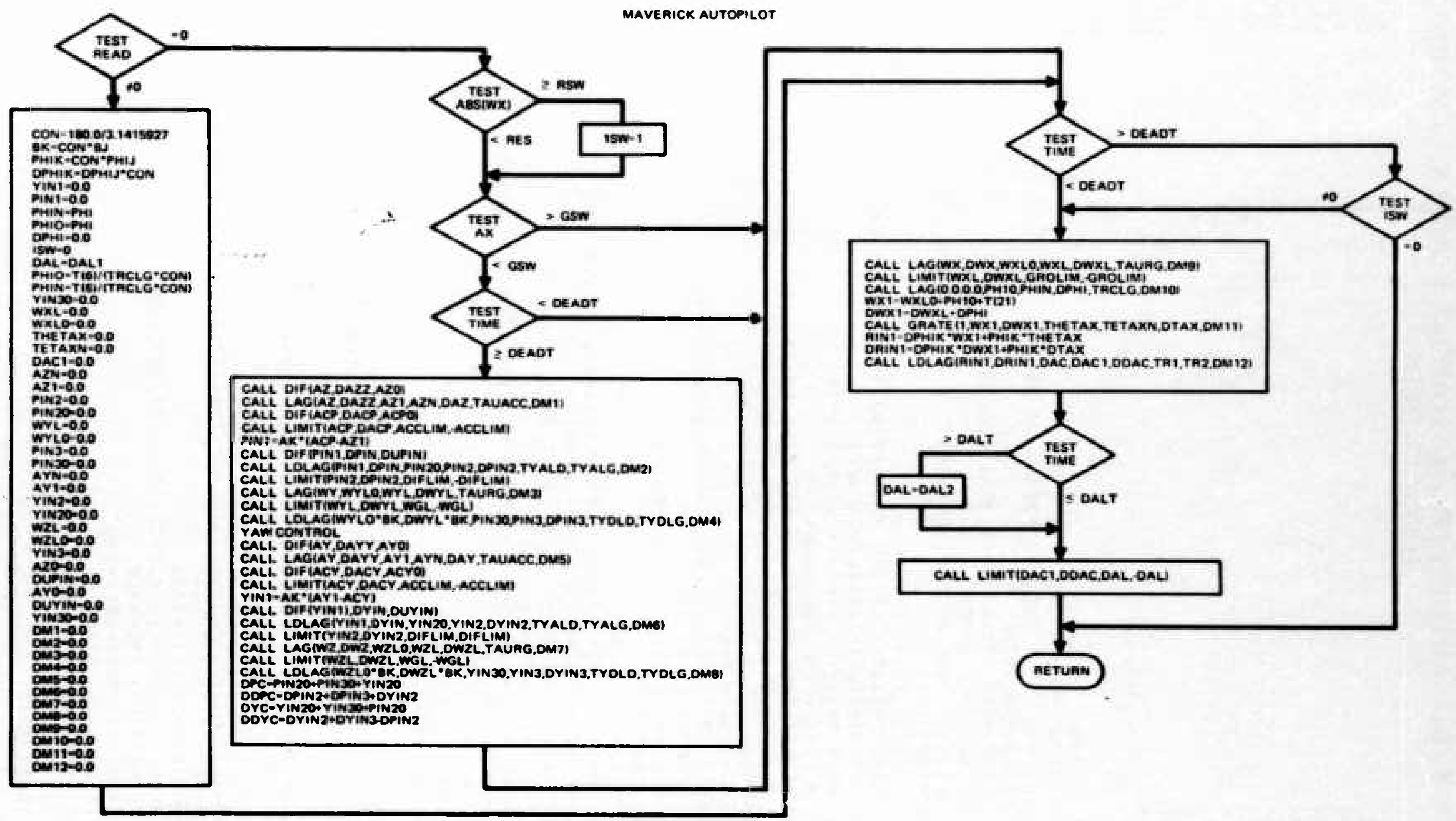



Figure 21. Autopilot Subroutine Flow Chart

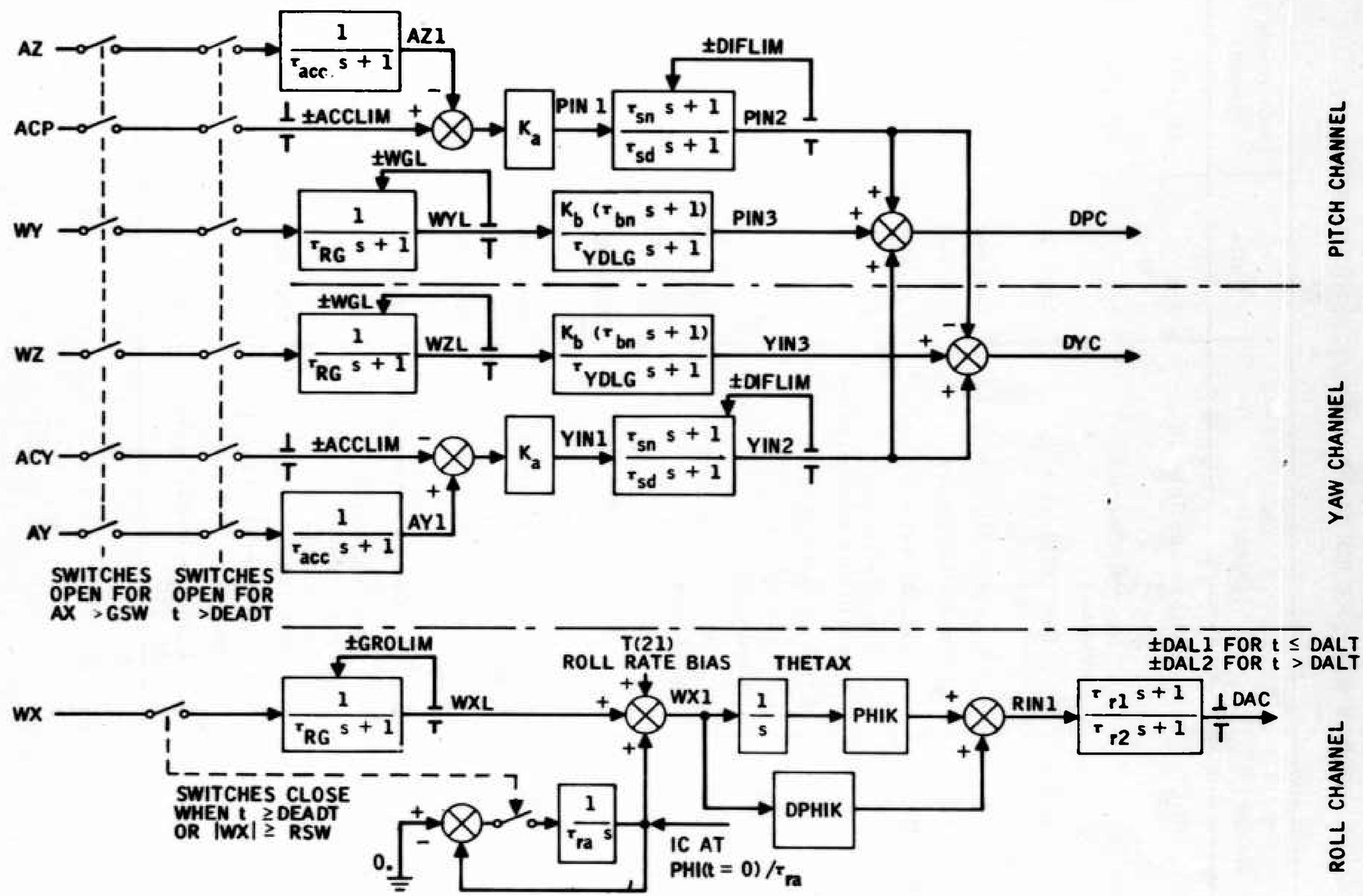



Figure 22. Autopilot Subroutine Block Diagram

TABLE XIX. AUTOPILOT (MPILOT) SUBROUTINE

| Name | Quantity | Units | Coordinate System |
|---|---|---|---|
| V(2) DAC | $\delta_{ac}$, Aileron deflection command | deg | Missile |
| V(3) DPC | $\delta_{pc}$, Pitch deflection command | deg | Missile |
| V(4) DYC | $\delta_{yc}$, Yaw deflection command | deg | Missile |
| V(11) WX | $\omega_x$, Angular velocity | rad/sec | Missile |
| V(12) WY | $\omega_y$, Angular velocity | rad/sec | Missile |
| V(13) WZ | $\omega_z$, Angular velocity | rad/sec | Missile |
| V(14) AX | $A_x$, Propulsive and aerodynamic acceleration | g's | Missile |
| V(20) PHI | $\phi$, Euler roll angle | rad | |
| V(42) ACP | $A_{cp}$, Acceleration command pitch | g's | Autopilot |
| V(43) ACY | $A_{cy}$, Acceleration command yaw | g's | Autopilot |
| V(44) DWX | $\dot{\omega}_x$, Scalar, components of missile angular acceleration in autopilot axes | rad/sec$^2$ | Missile |
| V(45) DWY | $\dot{\omega}_y$, Scalar, components of missile angular acceleration in autopilot axes | rad/sec$^2$ | Missile |
| V(46) DWZ | $\dot{\omega}_z$, Scalar, components of missile angular acceleration in autopilot axes | rad/sec$^2$ | Missile |
| V(47) DDAC | $\dot{\delta}_{ac}$, Aileron command rate | deg/sec | Missile |
| V(48) DDPC | $\dot{\delta}_{pc}$, Elevator command rate | deg/sec | Missile |
| V(49) DDYC | $\dot{\delta}_{yc}$, Rudder command rate | deg/sec | Missile |
| V(60) AY | $a_y$ missile lateral acceleration components | g's | Missile |
| V(61) AZ | $a_z$ missile lateral acceleration components | g's | Missile |
| T(21) | Roll rate bias | rad/sec | Missile |
| C(43) GSW | Autopilot lateral channel activation switch level | g's | |
| C(44) AK | $K_a$, Lateral channel gain | deg/g | |

TABLE XIX. AUTOPILOT (MPILOT) SUBROUTINE (CONCLUDED)

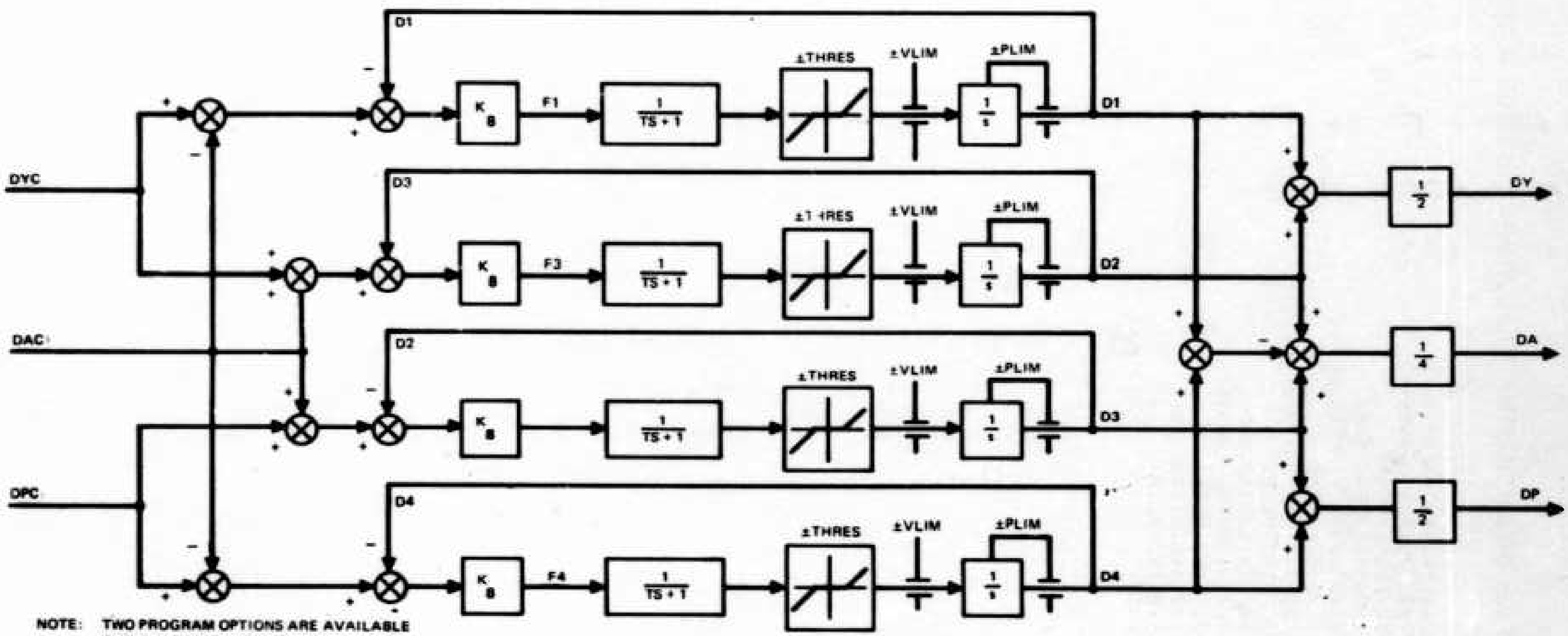
| Name | Quantity | Units |
|---|---|---|
| C(45) BJ | $K_b$, Damping gain | deg/deg/sec |
| BK | $K_b$, Damping gain | deg/rad/sec |
| C(46) PHIJ | $\phi_k$, Roll channel gain | deg/deg |
| PHIK | $\phi_k$, Roll channel gain | deg/rad |
| C(47) ACCLIM | Acceleration limit, lateral channels | g's |
| C(48) DIFLIM | Command limit, lateral channels | deg |
| C(49) TYALD | $\tau_{sn}$, Lead time constant | sec |
| C(50) TYALG | $\tau_{sd}$, Lag time constant | sec |
| C(51) TYDLD | $\tau_{bn}$, Lead time constant | sec |
| C(52) TYDLG | $\tau_{YDLG}$, Lag time constant | sec |
| C(53) DEADT | Autopilot activation delay | sec |
| C(54) GROLIM | Roll rate signal limit | rad/sec |
| C(55) TRCLG | $\tau_{ra}$ | sec |
| C(56) RSW | Roll rate switch level | rad/sec |
| C(57) TAUACC | $\tau_{acc}$, Lateral channel time constant | sec |
| C(58) TAURG | $\tau_{RG}$, Lateral channel time constant | sec |
| C(59) DPHIJ | $\dot{\phi}_K$, Roll rate gain | deg/deg/sec |
| DPHIK | $\dot{\phi}_K$, Roll rate gain | deg/rad/sec |
| C(60) DALT | Roll channel limit change time | sec |
| C(61) TR1 | $\tau_{r1}$, Lead time constant | sec |
| C(62) TR2 | $\tau_{r2}$, Lag time constant | sec |
| C(63) DAL1 | Roll command limit | deg |
| C(64) DAL2 | Roll command limit | deg |
| C(65) WGL | Lateral channel rate limit | rad/sec |

# TABLE XX. FLIPPER SUBROUTINE FORTRAN LISTING

```
$      FORTRAN DECK
CFLIP       FLIPPER WITH THRESHOLD PROVISION                            FLIP0010
      SUBROUTINE MFLIP                                                  FLIP0020
      COMMON /SSAM1/ READ,DELT,AUTOT,TIME
      COMMON /SSAM2/ V  (250),T  (250),C  (250)
      EQUIVALENCE                                                       FLIP0050
     1 (V( 2),DAC   ),(V(47),DDAC  ),(V( 3),DPC   ),(V(48),DDPC  ),     FLIP0060
     2 (V( 4),DYC   ),(V(49),DDYC  ),(V( 5),DA    ),(V( 6),DP    ),     FLIP0070
     3 (V( 7),DY    )                                                   FLIP0080
      EQUIVALENCE                                                       FLIP0090
     1 (C(97),GAIN  ),(C(98),TAU   ),(C(99),VLIM  ),(C(100),PLIM ),     FLIP0100
     2 (C(101),THRES)                                                   FLIP0110
      IF (READ.EQ.0.0) GO TO 10                                         FLIP0120
      D1=0.0                                                            FLIP0130
      D2=0.0                                                            FLIP0140
      D3=0.0                                                            FLIP0150
      D4=0.0                                                            FLIP0160
      DD1=0.0                                                           FLIP0170
      DD2=0.0                                                           FLIP0180
      DD3=0.0                                                           FLIP0190
      DD4=0.0                                                           FLIP0200
      TA1=1.0/GAIN                                                      FLIP0210
   10 IF (THRES.EQ.0.0) GO TO 15                                        FLIP0220
      F1=(DYC-DAC-D1)*GAIN                                              FLIP0230
      DF1=(DDYC-DDAC-DD1)*GAIN                                          FLIP0240
      F2=(DAC+DPC-D2)*GAIN                                              FLIP0250
      DF2=(DDAC+DDPC-DD2)*GAIN                                          FLIP0260
      F3=(DYC+DAC-D3)*GAIN                                              FLIP0270
      DF3=(DDYC+DDAC-DD3)*GAIN                                          FLIP0280
      F4=(DPC-DAC-D4)*GAIN                                              FLIP0290
      DF4=(DDPC-DDAC-DD4)*GAIN                                          FLIP0300
      GO TO 20                                                          FLIP0310
   15 F1=DYC-DAC                                                        FLIP0320
      DF1=DDYC-DDAC                                                     FLIP0330
      F2=DAC+DPC                                                        FLIP0340
      DF2=DDAC+DDPC                                                     FLIP0350
      F3=DYC+DAC                                                        FLIP0360
      DF3=DDYC+DDAC                                                     FLIP0370
      F4=DPC-DAC                                                        FLIP0380
      DF4=DDPC-DDAC                                                     FLIP0390
      CALL   LAG(F1,DF1,D10,D1,DD1,TA1,DU1)                             FLIP0400
      CALL  VLIMIT(D10,D1,DD1,VLIM,-VLIM)                               FLIP0410
      CALL  LIMIT(D1,DD1,PLIM,-PLIM)                                    FLIP0420
      CALL   LAG(F2,DF2,D20,D2,DD2,TA1,DU2)                             FLIP0430
      CALL  VLIMIT(D20,D2,DD2,VLIM,-VLIM)                               FLIP0440
      CALL  LIMIT(D2,DD2,PLIM,-PLIM)                                    FLIP0450
      CALL   LAG(F3,DF3,D30,D3,DD3,TA1,DU3)                             FLIP0460
      CALL  VLIMIT(D30,D3,DD3,VLIM,-VLIM)                               FLIP0470
      CALL  LIMIT(D3,DD3,PLIM,-PLIM)                                    FLIP0480
      CALL   LAG(F4,DF4,D40,D4,DD4,TA1,DU4)                             FLIP0490
      CALL  VLIMIT(D40,D4,DD4,VLIM,-VLIM)                               FLIP0500
      CALL  LIMIT(D4,DD4,PLIM,-PLIM)                                    FLIP0510
      GO TO 30                                                          FLIP0520
   20 CALL  LAG(F1,DF1,Z10,Z1,DZ,TAU,DU1)                               FLIP0530
      CALL  DBAND(Z10,DZ,ZZ1,THRES,-THRES)                              FLIP0540
      CALL  LIMIT(ZZ1,DZ1, VLIM,-VLIM)                                  FLIP0550
      CALL  GRATE(1,ZZ1,DZ1,D10,D1,DD1,DU2)                             FLIP0560
      CALL  LIMIT (D1,DD1,PLIM,-PLIM)                                   FLIP0570
      CALL  LAG(F2,DF2,Z20,Z2,DZ,TAU,DU3)                               FLIP0580
      CALL  DBAND(Z20,DZ,ZZ2,THRES,-THRES)                              FLIP0590
```

TABLE XX. FLIPPER SUBROUTINE FORTRAN LISTING (CONCLUDED)

```
      CALL LIMIT(ZZ2,DZ,VLIM,-VLIM)                                      FLIP0600
      CALL GRATE(1,ZZ2,DZ,D20,D2,DD2,DU4)                                FLIP0610
      CALL LIMIT (D2,DD2,PLIM,-PLIM)                                     FLIP0620
      CALL LAG(F3,DF3,Z30,Z3,DZ,TAU,DU5)                                 FLIP0630
      CALL DBAND(Z30,DZ,ZZ3,THRES,-THRES)                                FLIP0640
      CALL LIMIT(ZZ3,DZ,VLIM,-VLIM)                                      FLIP0650
      CALL GRATE(1,ZZ3,DZ,D30,D3,DD3,DU6)                                FLIP0660
      CALL LIMIT(D3,DD3,PLIM,-PLIM)                                      FLIP0670
      CALL LAG(F4,DF4,Z40,Z4,DZ,TAU,DU7)                                 FLIP0680
      CALL DBAND(Z40,DZ,ZZ4,THRES,-THRES)                                FLIP0690
      CALL LIMIT(ZZ4,DZ,VLIM,-VLIM)                                      FLIP0700
      CALL GRATE(1,ZZ4,DZ,D40,D4,DD4,DU8)                                FLIP0710
      CALL LIMIT(D4,DD4,PLIM,-PLIM)                                      FLIP0720
   30 DA=.25*(D2-D4-D1+D3)                                               FLIP0730
      DP=.5*(D2+D4)                                                      FLIP0740
      DY=.5*(D1+D3)                                                      FLIP0750
      RETURN                                                             FLIP0760
      END                                                                FLIP0770
```




NOTE: TWO PROGRAM OPTIONS ARE AVAILABLE WITH AND WITHOUT THRESHOLD. WHEN THRESHOLD IS NOT USED A DIFFERENT PROGRAMMING TECHNIUQE IS EMPLOYED IN THE INTEREST OF EFFICIENCY.

Figure 23. Flipper Subroutine Block Diagram

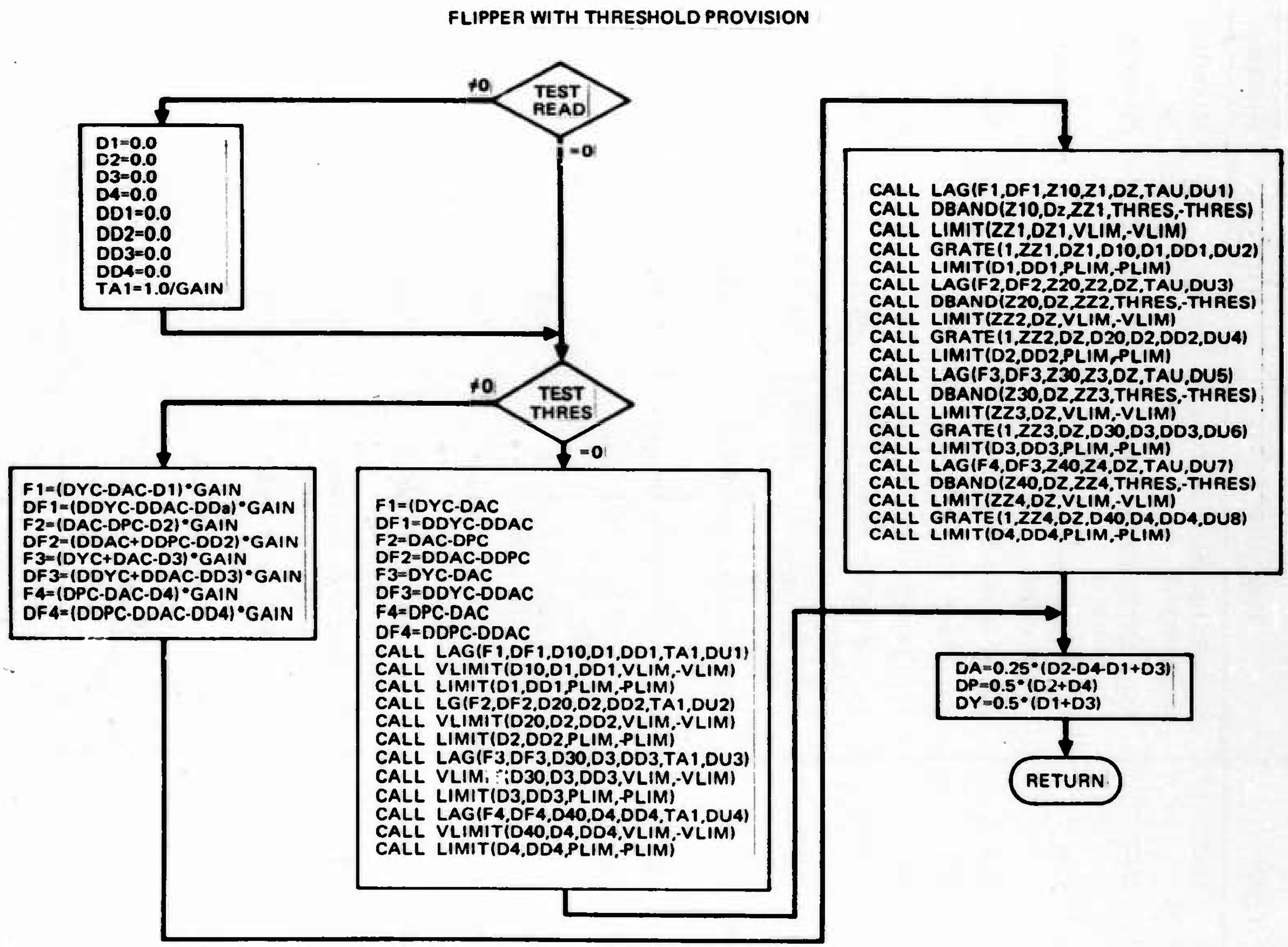



Figure 24. Flipper Subroutine Flow Chart

TABLE XXI. FLIPPER (MFLIP) SUBROUTINE

| Name | | Quantity | Units | Coordinate System |
|---|---|---|---|---|
| V(2) | DAC | $\delta_{ac}$, Aileron deflection command | deg | Missile |
| V(3) | DPC | $\delta_{pc}$, Pitch deflection command | deg | Missile |
| V(4) | DYC | $\delta_{yc}$, Yaw deflection command | deg | Missile |
| V(5) | DA | $\delta_a$, Aileron deflection | deg | Missile |
| V(6) | DP | $\delta_p$, Pitch deflection | deg | Missile |
| V(7) | DY | $\delta_y$, Yaw deflection | deg | Missile |
| V(47) | DDAC | $\dot{\delta}_{ac}$, Aileron command rate | deg/sec | Missile |
| V(48) | DDPC | $\dot{\delta}_{pc}$, Elevator command rate | deg/sec | Missile |
| V(49) | DDYC | $\dot{\delta}_{yc}$, Rudder command rate | deg/sec | Missile |
| C(97) | GAIN | $K_\delta$, Servo velocity gain | 1/sec | |
| C(98) | TAU | $\tau_\delta$, Control surface time constant | sec | |
| C(99) | VLIM | $\dot{\delta}_L$, Control surface velocity limit | deg/sec | |
| C(100) | PLIM | $\delta_L$, Control surface angle limit | deg | |
| C(101) | THRES | $\delta_{th}$, Control surface rate threshold | deg/sec | |
| | D1 | $\delta_1$, Control surface No. 1 deflection | deg | |
| | D2 | $\delta_2$, Control surface No. 2 deflection | deg | |
| | D3 | $\delta_3$, Control surface No. 3 deflection | deg | |
| | D4 | $\delta_4$, Control surface No. 4 deflection | deg | |

## TABLE XXII. AERO SUBROUTINE FORTRAN LISTING

```
$       FORTRAN DECK
CMAERO          MAVERICK AERO
      SUBROUTINE MAERO
      COMMON /SSAM1/ READ,DELT,AUTOT,TIM
      COMMON /SSAM2/ V  (250),T  (250),C  (250)
      EQUIVALENCE
     1 (V( 1),RELALT),(V( 5),DA    ),(V( 6),DP    ),(V( 7),DY    ),
     2 (V( 8),VXM   ),(V( 9),VYM   ),(V(10),VZM   ),(V(11),WX    ),
     3 (V(12),WY    ),(V(13),WZ    ),(V(14),AX    ),(V(15),AY    ),
     4 (V(16),AZ    ),(V(33),ALPHA ),(V(34),ALPHAP),(V(35),ALPHAY)
      EQUIVALENCE
     2 (V(45),DWY   ),(V(46),DWZ   ),(V(39),Q     ),(V(40),VM    ),
     3 (V(41),AM    ),(V(44),DWX   ),(V(84),CAPLAM)
      EQUIVALENCE
     1(C(73),S     ),(C(74),D     ),(C(75),FSL   ),(C(76),TBOOST),
     2(C(77),TSUST ),(C(78),ELTB  ),(C(79),XBAR  ),(C(80),CLP   ),
     3(C(81),AE    ),(C(82),AJX0  ),(C(83),AJX1  ),(C(84),AJXT  ),
     4(C(85),AJY0  ),(C(86),AJY1  ),(C(87),AJYT  ),(C(88),AMASS0),
     5(C(89),AMASS1),(C(90),AMASST)
     6,(C(91),TSEPAR),(C(92),TGTALT)
      DATA RD2/114.59156/
      DATA RTOD/57.29578/
      IF (READ.EQ.0.0) GO TO 5
      FLUB=1.414213/(32.2*57.29578)
      SD = S*D
      G=32.2
      DOV2=D/2.0
      RXB=RD2*(XBAR/D)**2
      AEOS=AE/S
      CONALF=2.0*XBAR**2/(3.0*D**2)
      DOLTB=D/ELTB
      DAJX1=(AJX0-AJX1)/TBOOST
      DAJY1=(AJY0-AJY1)/TBOOST
      DMASS=(AMASS0-AMASS1)/TBOOST
      TDIF=TSUST-TBOOST
      DAJX2=(AJX1-AJXT)/TDIF
      DAJY2=(AJY1-AJYT)/TDIF
      DMASS2=(AMASS1-AMASST)/TDIF
    5 VYZ2=VYM**2+VZM**2
      VYZ=SQRT(VYZ2)
      VSQ=VYZ2+VXM**2
      VM=SQRT(VSQ)
      TIME=TIM+TSEPAR
      ALT=RELALT+TGTALT
C
C  MACH NUMBER
      CALL FGEN1(I1,ALT,SVEL,-1)
      AM=VM/SVEL
C
C  AERODYNAMIC ROLL ANGLE, PHIA
      IF (VYZ.NE.0.0) GO TO 10
      CPHIA=.70711
      SPHIA=.70711
      GO TO 20
   10 CPHIA=VZM/VYZ
      SPHIA=VYM/VYZ
C
C  ANGLE OF ATTACK, ALPHA
   20 ALPHA=ATAN(VYZ/VXM)*RTOD
      ALPHA2=ALPHA**2
```

```
      ALPHA3=ALPHA2*ALPHA                                               AERO0590
      ALPHAP=ALPHA*CPHIA                                                AERO0600
      ALPHAY=ALPHA*SPHIA                                                AERO0610
      CALL DIF(ALPHAP,DALFP,DALFP1)                                     AERO0620
      CALL DIF(ALPHAY,DALFY,DALFY1)                                     AERO0630
C                                                                       AERO0640
C  FLIPPER DEFLECTIONS IN MANEUVER AXES                                 AERO0650
      DT=DP*CPHIA-DY*SPHIA                                              AERO0660
      DR=DY*CPHIA+DP*SPHIA                                              AERO0670
C                                                                       AERO0680
C  AUXILIARY FUNCTIONS                                                  AERO0690
      COS4PH=1.0-6.0*CPHIA**2*SPHIA**2                                  AERO0700
      SIN4PH=5.18*(ABS(CPHIA)-ABS(SPHIA))*CPHIA*SPHIA                   AERO0710
      DD2V=DOV2/VM                                                      AERO0720
C                                                                       AERO0730
C  MANEUVER AXES AERODYNAMIC COEFFICIENTS                               AERO0740
C     ******************************                                    AERO0750
C  INTERMEDIATE EXPRESSIONS                                             AERO0760
C  C SUB SMALL MO                                                       AERO0770
      N=0                                                               AERO0780
      CALL FGEN1(I2,AM,SM1,N)                                           AERO079
      CALL FGEN1(I3,AM,SM2,N)                                           AERO080
      CALL FGEN1(I4,AM,SM3,N)                                           AERO081
      CALL FGEN1(I5,AM,SM4,N)                                           AERO082
      CSMO=SM1*ALPHA+(SM2+SM3*COS4PH)*ALPHA2 + SM4*ALPHA3               AERO0830
C                                                                       AERO0840
C  DELTA C SUM SMALL M                                                  AERO0850
      CALL FGEN1(I6,AM,SM5,N)                                           AERO086
      CALL FGEN1(I7,AM,SM6,N,                                           AERO087
      CALL FGEN1(I8,AM,SM7,N)                                           AERO088
      DCSM= (SM5+(SM6 + SM7*COS4PH)*ALPHA2)*DT                          AERO0890
C                                                                       AERO0900.
C  C SUB SMALL NO                                                       AERO0910
      CALL FGEN1(I9,AM,SN1,N)                                           AERO092
      CALL FGEN1(I10,AM,SN2,N)                                          AERO093
      CSNO=(SN1*ALPHA2 + SN2*ALPHA3)*SIN4PH                             AERO0940
C                                                                       AERO0950
C  DELTA C SUB SMALL N                                                  AERO0960
      CALL FGEN1(I11,AM,SN3,N)                                          AERO097
      CALL FGEN1(I12,AM,SN4,N)                                          AERO098
      CALL FGEN1(I13,AM,SN5,N)                                          AERO099
      CALL FGEN1(I14,AM,SN6,N)                                          AERO100
      CALL FGEN1(I15,AM,SN7,N)                                          AERO101
      CALL FGEN1(I16,AM,SN8,N)                                          AERO1020
      DCSN=(SN3 + (SN4 + SN5*COS4PH)*ALPHA2)*DR                         AERO1030
     1 + ((SN6 + SN7*COS4PH)*ALPHA + SN8*ALPHA2*DT)*DA                  AERO1040
C                                                                       AERO1050
C  C SUB NO                                                             AERO1060
      CALL FGEN1(I17,AM,CN1,N)                                          AERO1070
      CALL FGEN1(I18,AM,CN2,N)                                          AERO1080
      CALL FGEN1(I19,AM,CN3,N)                                          AERO1090
      CALL FGEN1(I20,AM,CN4,N)                                          AERO1100
      CNO=CN1*ALPHA + (CN2+CN3*COS4PH)*ALPHA2 + CN4*ALPHA3              AERO1110
C                                                                       AERO1120
C  DELTA C SUB N                                                        AERO1130
      DCN=DOLTB*DCSM                                                    AERO1140
C                                                                       AERO1150
C  C SUB YO                                                             AERO1160
      CALL FGEN1(I21,AM,Y1,N)                                           AERO1170
      CALL FGEN1(I22,AM,Y2,N)                                           AERO1180
```

## TABLE XXII. AERO SUBROUTINE FORTRAN LISTING (CONTINUED)

```
      CYO=(Y1*ALPHA2 + Y2*ALPHA3)*SIN4PH                                AERO1190
C                                                                       AERO1200
C  DELTA C SUB Y                                                        AERO1210
      DCY=DULTB*DCSN                                                    AERO1220
C                                                                       AERO1230
C  C SUB CO                                                             AERO1240
      CALL FGEN1(123,AM,CDW,N)                                          AERO1250
      CALL FGEN1(124,AM,C2,N)                                           AERO126
      CALL FGEN1(125,AM,C3,N)                                           AERO127
      CALL FGEN1(127,AM,CDF1,N)                                         AERO128
      CALL FGEN1(128,AM,CDF2,N)                                         AERO129
      IF((TIME.GT.TSUST ).AND.(READ.EQ.0.0))GO TO 30                    AERO1300
      CALL FGEN1(126,AM,CDB,N)                                          AERO1310
      GO TO 40                                                          AERO1320
   30 CDB=0.0                                                           AERO1330
   40 CDF=ALT*(CDF1 + ALT*CDF2)                                         AERO1340
      C1=CDW + CDF + CDB*AEOS                                           AERO1350
      CCO=C1+C2*ALPHA+C3*ALPHA2                                         AERO1360
C                                                                       AERO1370
C  DELTA C SUB  C                                                       AERO1380
      CALL FGEN1(130,AM,C4,N)                                           AERO1390
      CALL FGEN1(131,AM,C5,N)                                           AERO1400
      DCC=C4*(ALPHAP*UP-ALPHAY*DY)+C5*DT**2                             AERO1410
C                                                                       AERO1420
C  C SUB LO                                                             AERO1430
      CALL FGEN1(132,AM,SL1,N)                                          AERO1440
      CALL FGEN1(133,AM,SL2,N)                                          AERO1450
      CALL FGEN1(134,AM,SL3,N)                                          AERO1460
      CALL FGEN1(135,AM,SL4,N)                                          AERO1470
      CALL FGEN1(136,AM,SL5,N)                                          AERO1480
      CALL FGEN1(137,AM,SL6,N)                                          AERO1490
      CALL FGEN1(138,AM,SL7,N)                                          AERO1500
      CALL FGEN1(139,AM,SL8,N)                                          AERO1510
      CALL FGEN1(140,AM,SL9,N)                                          AERO1520
      CALL FGEN1(140,AM,SL10,N)                                         AERO153
      CLO=(SL1 +SL2*ALPHA + SL3*ALPHA2 + SL4*ALPHA3)*ALPHA2*SIN4PH      AERO1540
C                                                                       AERO1550
C  DELTA C SUB L                                                        AERO1560
      DCL=(SL5+(SL6+SL7*COS4PH)*ALPHA )*DA+(SL8*ALPHA+(SL9+SL10*COS4PH) AERO1570
     1*ALPHA2)*DR                                                       AERO1580
      IF ((TIME.GT.TSUST).AND.(READ.EQ.0.0)) GO TO 50                   AERO1590
      IF (TIME.GT.TBOOST) GO TO 45                                      AERO1600
      AMASS=AMASS0-DMASS*TIME                                           AERO1610
      AJY=AJY0-DAJY1*TIME                                               AERO1620
      AJX=AJX0-DAJX1*TIME                                               AERO1630
      GO TO 46                                                          AERO1640
   45 TDIF=TIME-TBOOST                                                  AERO1650
      AMASS=AMASS1-DMASS2*TDIF                                          AERO1660
      AJY=AJY1-DAJY2*TDIF                                               AERO1670
      AJX=AJX1-DAJX2*TDIF                                               AERO1680
   46 AJXY=(AJY-AJX)/AJY                                                AERO1690
C  1.0 - DELTA L/LTB                                                    AERO1710
      CALL FGEN1(141,TIME,DLOD,-1)                                      AERO1700
      CALL FGEN1(142,TIME,OMDLOL,-1)                                    AERO1720
      CALL FGEN1(144,TIME,THRUST,-1)                                    AERO1730
      CALL FGEN1(145,ALT,PRES,-1)                                       AERO1740
      TOM=(THRUST +AE*(PSL-PRES))/AMASS                                 AERO1750
      GO TO 60                                                          AERO1760
   50 TOM=0.0                                                           AERO1770
   60 CALL FGEN1(143,ALT,RHO,-1)                                        AERO1780
```

TABLE XXII. AERO SUBROUTINE FORTRAN LISTING (CONTINUED)

```
C                                                                       AER01790
C  PRIMARY EXPRESSIONS                                                  AER01800
      CSUBM=CSMO-DLOD*CNO+OMDLOL*DCSM                                   AER01810
      CSLCN=CSNO+DLOD*CYO+OMDLOL*DCSN                                   AER01820
      CSUBN=CNO+DCN                                                     AER01830
      CSUBY=CYO-DCY                                                     AER01840
      CSUBC=CCO+DCC                                                     AER01850
      CSUBL=CLO + DCL                                                   AER01860
      CNOALP=CN1+ALPHA*(CN2+CN3*COS4PH+ALPHA*CN4)                       AER01870
      CMO=RXB*CNOALP                                                    AER01880
      CMDALF=DO2V*CONALF*CNOALP                                         AER01890
C                                                                       AER01900
C                                                                       AER01910
C  AERODYNAMIC COEFFICIENTS IN MISSILE AXES                             AER01920
      CSUBMB=CSUBM*CPHIA+CSLCN*SPHIA                                    AER01930
     1 + CMDALF*DALFP                                                   AER01940
      CSLCNB=CSLCN*CPHIA-CSUBM*SPHIA                                    AER01950
     1 - CMDALF * DALFY                                                 AER01960
      CSUBNB=CSUBN*CPHIA -CSUBY*SPHIA                                   AER01970
      CSUBYB=CSUBY*CPHIA +CSUBN*SPHIA                                   AER01980
C  DYNAMIC PRESSURE, Q                                                  AER01990
      Q=RHO*VSQ/2.0                                                     AER02000
C                                                                       AER02010
C  ACCELERATIONS                                                        AER02020
      CON1 = Q*S/AMASS                                                  AER02030
      CON2 = Q*SD/AJX                                                   AER02040
      CON3 = Q*SD/AJY                                                   AER02050
      COX=-DO2V*CLP                                                     AER02060
      COYZ=-DO2V*CMQ                                                    AER02070
      TAUX=1.0/(COX*CON2)                                               AER02080
      TAUZ =1.0/(COYZ*CON3)                                             AER02090
      TS=WX*AJXY*TAUZ                                                   AER02100
      AXM=CON1*CSUBC + TOM                                              AER02110
      AYM=CON1*CSUBYB                                                   AER02120
      AZM=CON1*CSUBNB                                                   AER02130
      CALL EULTRN(0,-1,0.0,0.0,0.0,GX,GY,GZ,YAW,ROLL,PITCH)             AER02140
      CALL R45F(GX,GY,GZ,GXM,GYM,GZM)                                   AER02150
C                                                                       AER02160
      TX=CSUBL/COX                                                      AER02170
      CALL DIF(TX,DTX,DX1)                                              AER02180
      TY=CSUBMB/COYZ+TS*WZ                                              AER02190
      CALL DIF(TY,DTY,DY1)                                              AER02200
      TZ=CSLCNB/COYZ-TS*WY                                              AER02210
      CALL DIF(TZ,DTZ,DZ1)                                              AER02220
      IF(READ .EQ. 0.0)GO TO 70                                         AER02230
      TX = WX                                                           AER02230
      TY = WY                                                           AER02240
      TZ = WZ                                                           AER02250
      DTX = 0.0                                                         AER02270
      DTY = 0.0                                                         AER02280
      DTZ = 0.0                                                         AER02290
   70 DVX=AXM+GXM+WZ*VYM-WY*VZM                                         AER02300
      DVY=AYM+GYM+WX*VZM-WZ*VXM                                         AER02310
      DVZ=AZM+GZM+WY*VXM-WX*VYM                                         AER02320
      CALL INTER(DVX,XX1,XX2,VXO,VXM)                                   AER02330
      CALL INTER(DVY,YY1,YY2,VYO,VYM)                                   AER02340
      CALL INTER(DVZ,ZZ1,ZZ2,VZO,VZM)                                   AER02350
      CALL LAG (TX,DTX,WXO,WX,DWX,TAUX,DUMX)                            AER02360
      CALL LAG (TY,DTY,WYO,WY,DWY,TAUZ,DUMY)                            AER02370
      CALL LAG (TZ,DTZ,WZO,WZ,DWZ,TAUZ,DUMZ)                            AER02380
```

TABLE XXII. AERO SUBROUTINE FORTRAN LISTING (CONCLUDED)

```
      AX=AXM/G
      AY=AYM/G
      AZ=AZM/G
      ZAP1=C(44)*C(14)*1.414213
      ZAP2=C(44)*V(40)*FLUR
      ZAP3=2.*AMASS/(RHO*V(40)*S)
      ZAP4=(CN1+(CN2+CN3*COS4PH)*ALPHA+CN4*ALPHA2)/(SM1+(SM2+SM3*COS4PH)
     1      *ALPHA+SM4*ALPHA2)-DOLTB
      ZAP5=SM5+(SM6+SM7*COS4PH)*ALPHA2
      CAPLAM = ZAP1/(ZAP2+C(45)-ZAP3/(ZAP4*ZAP5*57.295787))
      RETURN
      END
```



(The reverse of this page is blank.)

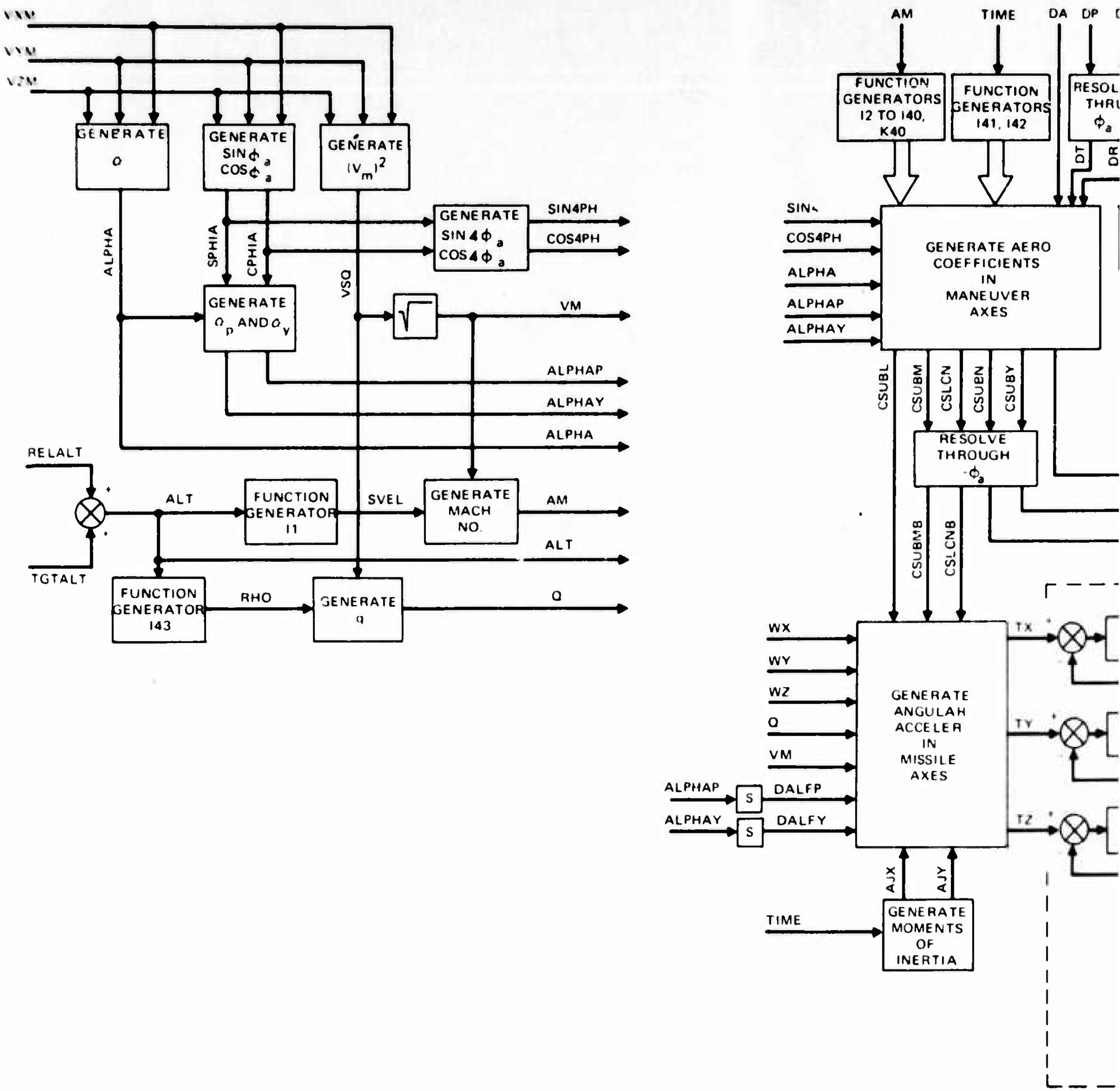



Figure 2
Bl

(The revers

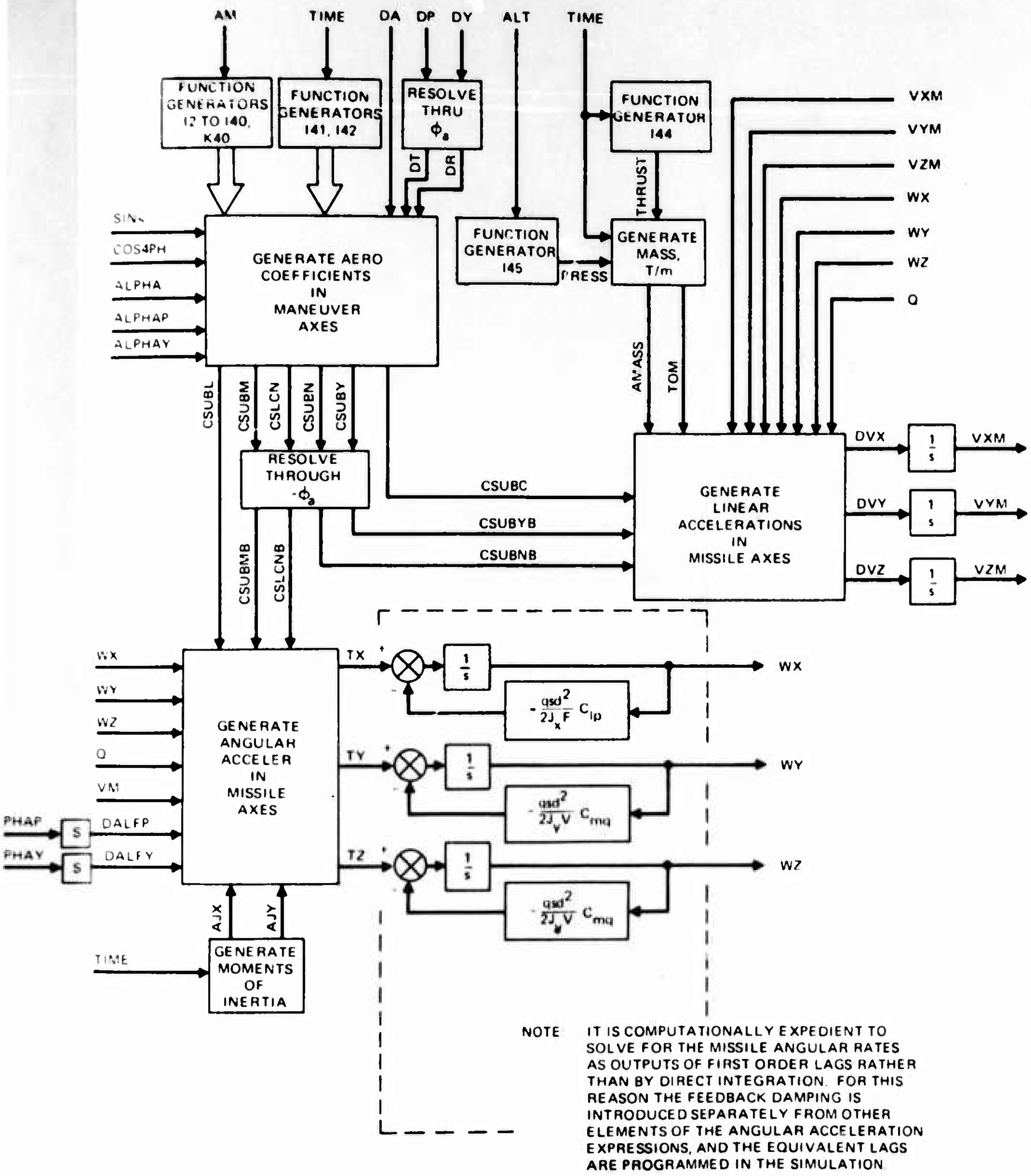
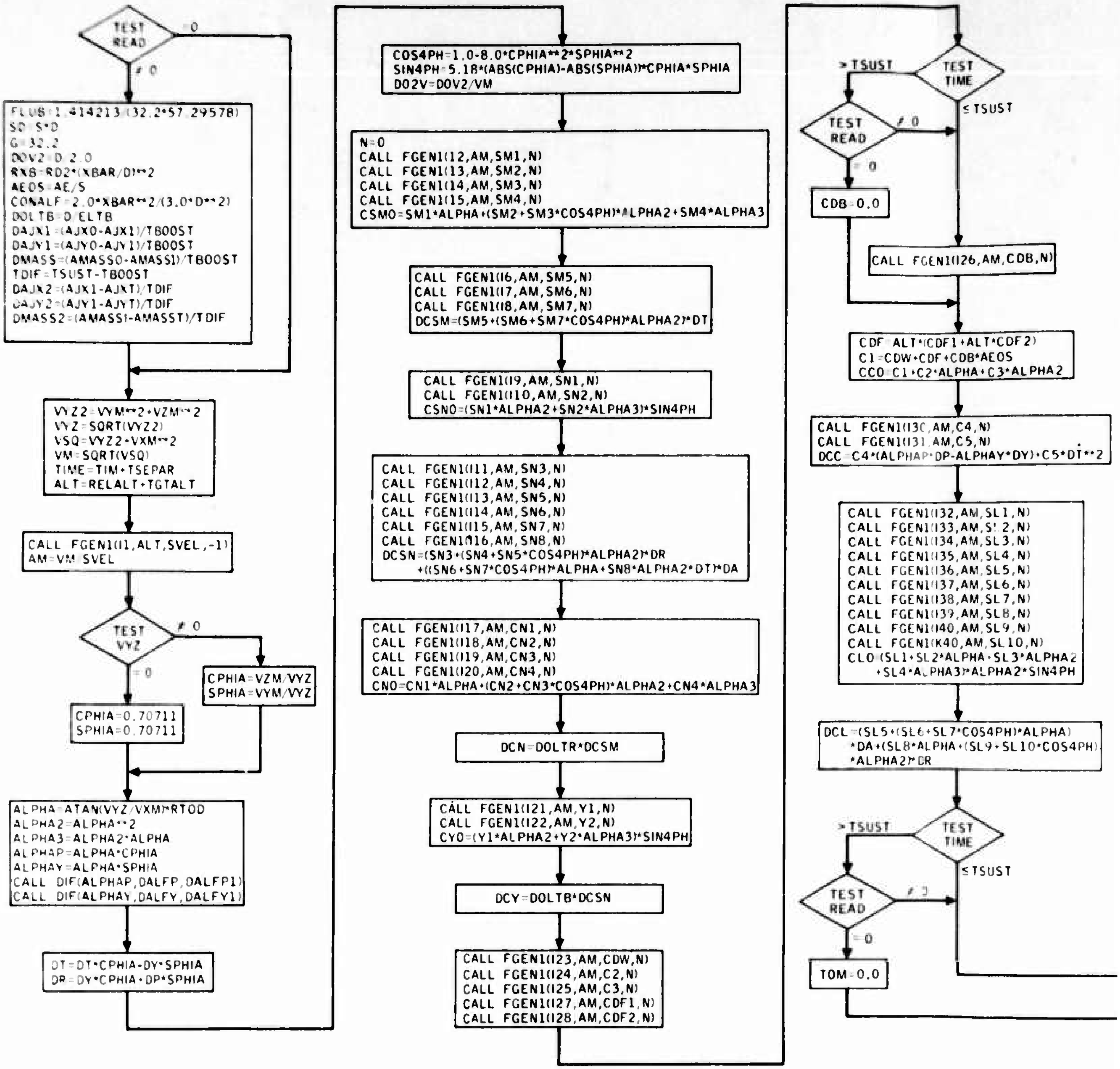



Figure 25. Aero Subroutine Block Diagram



(The reverse of this page is blank.)



TEST READ
=0
≠0
FLUS=1.414213/(32.2*57.29578)
SD=S*D
G=32.2
DOV2=D/2.0
RXB=RD2*(XBAR/D)**2
AEOS=AE/S
CONALF=2.0*XBAR**2/(3.0*D**2)
DOLTB=D/ELTB
DAJX1=(AJX0-AJX1)/TBOOST
DAJY1=(AJY0-AJY1)/TBOOST
DMASS=(AMASS0-AMASS1)/TBOOST
TDIF=TSUST-TBOOST
DAJX2=(AJX1-AJXT)/TDIF
DAJY2=(AJY1-AJYT)/TDIF
DMASS2=(AMASS1-AMASST)/TDIF
VYZ2=VYM**2+VZM**2
VYZ=SQRT(VYZ2)
VSQ=VYZ2+VXM**2
VM=SQRT(VSQ)
TIME=TIM+TSEPAR
ALT=RELALT+TGTALT
CALL FGEN1(I1,ALT,SVEL,-1)
AM=VM/SVEL
TEST VYZ
≠0
=0
CPHIA=VZM/VYZ
SPHIA=VYM/VYZ
CPHIA=0.70711
SPHIA=0.70711
ALPHA=ATAN(VYZ/VXM)*RTOD
ALPHA2=ALPHA**2
ALPHA3=ALPHA2*ALPHA
ALPHAP=ALPHA*CPHIA
ALPHAY=ALPHA*SPHIA
CALL DIF(ALPHAP,DALFP,DALFP1)
CALL DIF(ALPHAY,DALFY,DALFY1)
DT=DT*CPHIA-DY*SPHIA
DR=DY*CPHIA+DP*SPHIA
COS4PH=1.0-8.0*CPHIA**2*SPHIA**2
SIN4PH=5.18*(ABS(CPHIA)-ABS(SPHIA))*CPHIA*SPHIA
DO2V=DOV2/VM
N=0
CALL FGEN1(12,AM,SM1,N)
CALL FGEN1(13,AM,SM2,N)
CALL FGEN1(14,AM,SM3,N)
CALL FGEN1(15,AM,SM4,N)
CSMO=SM1*ALPHA+(SM2+SM3*COS4PH)*ALPHA2+SM4*ALPHA3
CALL FGEN1(I6,AM,SM5,N)
CALL FGEN1(I7,AM,SM6,N)
CALL FGEN1(I8,AM,SM7,N)
DCSM=(SM5+(SM6+SM7*COS4PH)*ALPHA2)*DT
CALL FGEN1(I9,AM,SN1,N)
CALL FGEN1(I10,AM,SN2,N)
CSNO=(SN1*ALPHA2+SN2*ALPHA3)*SIN4PH
CALL FGEN1(I11,AM,SN3,N)
CALL FGEN1(I12,AM,SN4,N)
CALL FGEN1(I13,AM,SN5,N)
CALL FGEN1(I14,AM,SN6,N)
CALL FGEN1(I15,AM,SN7,N)
CALL FGEN1(I16,AM,SN8,N)
DCSN=(SN3+(SN4+SN5*COS4PH)*ALPHA2)*DR
+((SN6+SN7*COS4PH)*ALPHA+SN8*ALPHA2*DT)*DA
CALL FGEN1(I17,AM,CN1,N)
CALL FGEN1(I18,AM,CN2,N)
CALL FGEN1(I19,AM,CN3,N)
CALL FGEN1(I20,AM,CN4,N)
CNO=CN1*ALPHA+(CN2+CN3*COS4PH)*ALPHA2+CN4*ALPHA3
DCN=DOLTR*DCSM
CALL FGEN1(I21,AM,Y1,N)
CALL FGEN1(I22,AM,Y2,N)
CYO=(Y1*ALPHA2+Y2*ALPHA3)*SIN4PH
DCY=DOLTB*DCSN
CALL FGEN1(I23,AM,CDW,N)
CALL FGEN1(I24,AM,C2,N)
CALL FGEN1(I25,AM,C3,N)
CALL FGEN1(I27,AM,CDF1,N)
CALL FGEN1(I28,AM,CDF2,N)
TEST TIME
>TSUST
≤TSUST
TEST READ
≠0
=0
CDB=0.0
CALL FGEN1(I26,AM,CDB,N)
CDF=ALT*(CDF1+ALT*CDF2)
C1=CDW+CDF+CDB*AEOS
CCO=C1+C2*ALPHA+C3*ALPHA2
CALL FGEN1(I30,AM,C4,N)
CALL FGEN1(I31,AM,C5,N)
DCC=C4*(ALPHAP*DP-ALPHAY*DY)+C5*DT**2
CALL FGEN1(I32,AM,SL1,N)
CALL FGEN1(I33,AM,SL2,N)
CALL FGEN1(I34,AM,SL3,N)
CALL FGEN1(I35,AM,SL4,N)
CALL FGEN1(I36,AM,SL5,N)
CALL FGEN1(I37,AM,SL6,N)
CALL FGEN1(I38,AM,SL7,N)
CALL FGEN1(I39,AM,SL8,N)
CALL FGEN1(I40,AM,SL9,N)
CALL FGEN1(K40,AM,SL10,N)
CLO=(SL1+SL2*ALPHA+SL3*ALPHA2
+SL4*ALPHA3)*ALPHA2*SIN4PH
DCL=(SL5+(SL6+SL7*COS4PH)*ALPHA)
*DA+(SL8*ALPHA+(SL9+SL10*COS4PH)
*ALPHA2)*DR
TEST TIME
>TSUST
≤TSUST
TEST READ
≠0
=0
TOM=0.0

2

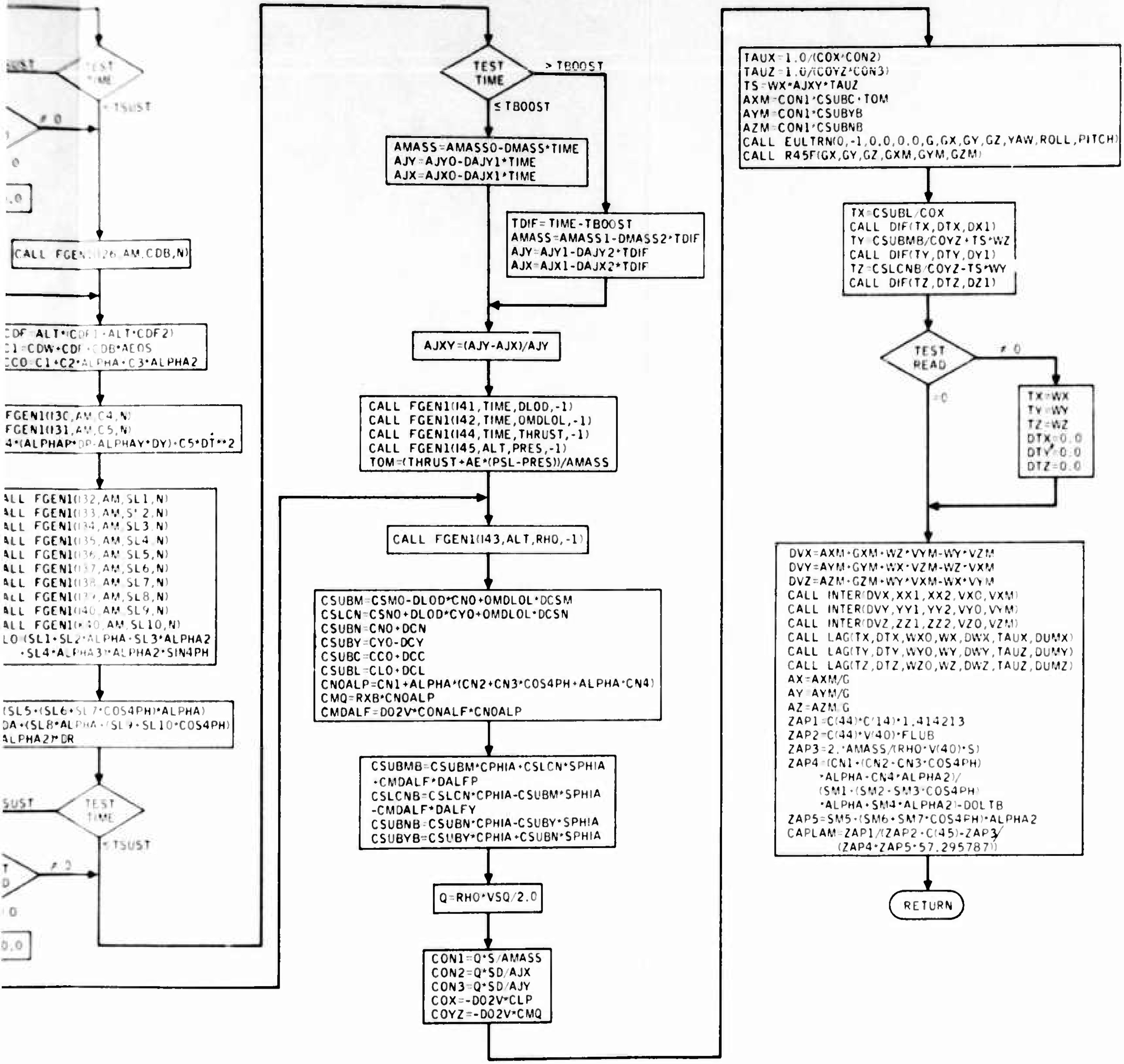



Figure 26. Aero Subroutine Flow Chart



(The reverse of this page is blank.)

## TABLE XXIII. AERODYNAMICS (MAERO) SUBROUTINE

| Name | | Quantity | Units | Coordinate System |
|---|---|---|---|---|
| V(1) | RELALT | $h_m$, Missile altitude above ground | ft | Inertial |
| V(5) | DA | $\delta_a$, Aileron deflection | deg | Missile |
| V(6) | DP | $\delta_p$, Pitch deflection | deg | Missile |
| V(7) | DY | $\delta_y$, Yaw deflection | deg | Missile |
| V(8) | VXM | $V_x$, Velocity X-axis missile | ft/sec | Missile |
| V(9) | VYM | $V_y$, Velocity Y-axis missile | ft/sec | Missile |
| V(10) | VZM | $V_z$, Velocity Z-axis missile | ft/sec | Missile |
| V(11) | WX | $\omega_x$ | | |
| V(12) | WY | $\omega_y$ } Components of angular velocity in missile axes | rad/sec | Missile |
| V(13) | WZ | $\omega_z$ | | |
| V(14) | AX | $A_x$ | | |
| V(15) | AY | $A_y$ } Propulsive and aerodynamic acceleration | g | Missile |
| V(16) | AZ | $A_z$ | | |
| V(33) | ALPHA | $\alpha$, Total missile angle of attack | deg | Missile |
| V(34) | ALPHAP | $\alpha_p$, Missile pitch angle of attack | deg | Missile |
| V(35) | ALPHAY | $\alpha_y$, Missile yaw angle of attack | deg | Missile |
| V(39) | Q | q, Dynamic pressure | $lb/ft^2$ | |
| V(40) | VM | Total missile velocity. | ft/sec | |
| V(41) | AM | Missile Mach number | | |
| V(44) | DWX | $\dot{\omega}_x$ | | |
| V(45) | DWY | $\dot{\omega}_y$ } Scalar components of missile angular acceleration in autopilot axes | $rad/sec^2$ | Missile |
| V(46) | DWZ | $\dot{\omega}_z$ | | |
| C(73) | S | S, Missile ref. area | $ft^2$ | |
| C(74) | D | d, Missile ref. diameter | ft | |
| C(75) | PSL | $P_{SL}$, Sea level pressure | $lb/ft^2$ | |
| C(76) | TBOOST | $t_b$, Booster burn time | sec | |
| C(77) | TSUST | $t_s$, Sustainer burn time | sec | |
| C(78) | ELTB | $l_{TB}$, Tail length (burnout) | ft | |

TABLE XXIII. AERODYNAMICS (MAERO) SUBROUTINE (CONCLUDED)

| Name | Quantity | Units | Coordinate System |
|---|---|---|---|
| C(79) XBAR | $\overline{X}$, cg to control surface trailing edge distance | ft | |
| C(80) CLP | $C_{l_p}$, Roll damping coefficient | 1/rad | |
| C(81) AE | $A_e$, Nozzle exit area | $ft^2$ | |
| C(82) AJXO | $J_{xo}$, Launch roll inertia | slug-$ft^2$ | |
| C(83) AJX1 | $J_{x1}$, End-of-boost roll inertia | slug-$ft^2$ | |
| C(84) AJXT | $J_{xt}$, End-of-sustain roll inertia | slug-$ft^2$ | |
| C(85) AJYO | $J_{yo}$, Launch lateral inertia | slug-$ft^2$ | |
| C(86) AJY1 | $J_{y1}$, End-of-boost lateral inertia | slug-$ft^2$ | |
| C(87) AJYT | $J_{yt}$, End-of-sustain lateral inertia | slug-$ft^2$ | |
| C(88) AMASSO | $M_o$, Launch mass | slugs | |
| C(89) AMASS1 | $M_1$, End-of-boost mass | slugs | |
| C(90) AMASST | $M_t$, End-of-sustain mass | slugs | |
| C(91) TSEPAR | Burn time prior to launch | sec | |
| C(92) TGALT | Target altitude | ft | Inertial |

## TABLE XXIV. AERODYNAMICS SUBROUTINE

| Intermediate Expressions (A) |
| --- |
| $C_1 = C_{D\omega} + (C_{DF1} + C_{DF2}\, h)\, h + \left\| C_{DB} \frac{A_e}{S} \right\|$ |
| $\delta_T = \delta_p \cos \phi_a - \delta_y \sin \phi_a$ |
| $\delta_R = \delta_y \cos \phi_a + \delta_p \sin \phi_a$ |
| $\cos 4\phi_a = \cos (4\phi_a) = 1\text{-}8 \sin^2 \phi_a \cos^2 \phi_a$ |
| $\sin 4\phi_a = 5.18 \left( \left\| \cos \phi_a \right\| - \left\| \sin \phi_a \right\| \right) \cos \phi_a \sin \phi_a$ |

| Secondary Expressions (B) |
| --- |
| $C_{mo} = m_1\alpha + (m_2 + m_3 \cos 4\phi_a)\, \alpha^2 + m_4\, \alpha^3$ |
| $C_{No} = N_1\alpha + (N_2 + N_3 \cos 4\phi_a)\, \alpha^2 + N_4\, \alpha^3$ |
| $\Delta C_m = (m_5 + (m_6 + m_7 \cos 4\phi_a)\, \alpha^2)\, \delta_T$ |
| $C_{no} = (n_1\, \alpha^2 + n_2\, \alpha^3) \sin 4\phi_a$ |
| $C_{yo} = (y_1\, \alpha^2 + y_2\, \alpha^3) \sin 4\phi_a$ |
| $\Delta C_n = (n_3 + (n_4 + n_5 \cos 4\phi_a)\, \alpha^2)\, \delta_R + ((n_6 + n_7 \cos 4\phi_a)\, \alpha + n_8\, \alpha^2\, \delta_T)\delta_a$ |
| $C_{co} = C_1 + C_2\alpha + C_3\, \alpha^2$ |
| $\Delta C_c = (C_4\alpha + C_5\, \delta_T)\, \delta_T$ |
| $C_{\ell o} = (\ell_1 + \ell_2\alpha + \ell_3\, \alpha^2 + \ell_4\, \alpha^3)\, \alpha^2 \sin 4\phi_a$ |
| $\Delta C_\ell = (\ell_5 + (\ell_6 + \ell_7 \cos 4\phi_a)\, \alpha)\, \delta_a + (\ell_8\alpha + (\ell_9 + \ell_{10} \cos 4\phi_a)\alpha^2)\, \delta_R$ |
| $\frac{d}{2V} C_{m\dot{\alpha}} = \frac{d}{2V} (N_1 + \alpha(N_2 + N_3 \cos 4\phi_a + \alpha N_4)) \left( \frac{2\overline{X}^2}{3d^2} \right)$ |

TABLE XXIV. AERODYNAMICS SUBROUTINE (CONCLUDED)

| Primary Expressions (C) |
|---|
| $C_m = C_{mo} - \frac{\Delta \ell}{d} C_{no} + \left(1 - \frac{\Delta \ell}{TB}\right) \Delta C_m$ |
| $C_n = C_{no} + \frac{\Delta \ell}{d} C_{yo} + \left(1 - \frac{\Delta \ell}{\ell_{TB}}\right) \Delta C_n$ |
| $C_N = C_{No} + \frac{d}{\ell_{TB}} \Delta C_m$ |
| $C_y = C_{yo} - \frac{d}{\ell_{TB}} \Delta C_n$ |
| $C_c = C_{co} + \Delta C_c$ |
| $C_\ell = C_{\ell o} + \Delta C_\ell$ |

TABLE XXV. AERODYNAMIC COEFFICIENTS IN MISSILE BODY AXES

$$C_{mb} = C_m \cos \phi_a + C_n \sin \phi_a$$

$$C_{nb} = C_n \cos \phi_a - C_m \sin \phi_a$$

$$C_{yb} = C_y \cos \phi_a + C_n \sin \phi_a$$

$$C_{Nb} = C_N \cos \phi_a - C_y \sin \phi_a$$

TABLE XXVI. EQUATIONS OF MOTION IN MISSILE BODY AXES

$$\dot{V}_x = \frac{qs}{m} C_c + g_x + \frac{T}{m} + \omega_z V_y - \omega_y V_z$$

$$\dot{V}_y = \frac{qs}{m} C_{yb} + g_y + \omega_x V_z - \omega_z V_x$$

$$\dot{V}_z = \frac{qs}{m} C_{Nb} + g_z + \omega_y V_x - \omega_x V_y$$

$$\dot{\omega}_x = \frac{qsd}{J_x} \left\{ C_{\ell} + \frac{d}{2V} C_{\ell p} \omega_x \right\} + \left( \frac{J_y - J_z}{J_x} \right) \omega_y \omega_z$$

$$\dot{\omega}_y = \frac{qsd}{J_y} \left\{ C_{mb} + \frac{d}{2V} \left( C_{mq} \omega_y + C_{m_{\dot{\alpha}}} \dot{\alpha}_y \right) \right\} + \left( \frac{J_z - J_x}{J_y} \right) \omega_z \omega_x$$

$$\dot{\omega}_z = \frac{qsd}{J_z} \left\{ C_{nb} + \frac{d}{2V} \left( C_{mq} \omega_z + C_{m_{\dot{\alpha}}} \dot{\alpha}_p \right) \right\} + \left( \frac{J_x - J_y}{J_z} \right) \omega_x \omega_y$$

$$J_y = J_z$$

TABLE XXVII. FUNCTION GENERATORS IN AERODYNAMIC SUBROUTINE

| FGEN No. | Input | Output | Symbol | Units |
|---|---|---|---|---|
| I 1 | ALT | SVEL | Sonic velocity | ft/sec |
| I 2 | AM | SM1 | $m_1$ | 1/deg |
| I 3 | AM | SM2 | $m_2$ | $1/deg^2$ |
| I 4 | AM | SM3 | $m_3$ | $1/deg^2$ |
| I 5 | AM | SM4 | $m_4$ | $1/deg^3$ |
| I 6 | AM | SM5 | $m_5$ | 1/deg |
| I 7 | AM | SM6 | $m_6$ | $1/deg^3$ |
| I 8 | AM | SM7 | $m_7$ | $1/deg^3$ |
| I 9 | AM | SN1 | $n_1$ | $1/deg^2$ |
| I 10 | AM | SN2 | $n_2$ | $1/deg^2$ |
| I 11 | AM | SN3 | $n_3$ | 1/deg |
| I 12 | AM | SN4 | $n_4$ | $1/deg^3$ |
| I 13 | AM | SN5 | $n_5$ | $1/deg^3$ |
| I 14 | AM | SN6 | $n_6$ | $1/deg^2$ |
| I 15 | AM | SN7 | $n_7$ | $1/deg^2$ |
| I 16 | AM | SN8 | $n_8$ | $1/deg^4$ |
| I 17 | AM | CN1 | $N_1$ | 1/deg |
| I 18 | AM | CN2 | $N_2$ | $1/deg^2$ |
| I 19 | AM | CN3 | $N_3$ | $1/deg^2$ |
| I 20 | AM | CN4 | $N_4$ | $1/deg^3$ |
| I 21 | AM | Y1 | $Y_1$ | $1/deg^2$ |
| I 22 | AM | Y2 | $Y_2$ | $1/deg^3$ |
| I 23 | AM | CDW | $C_{DW}$ | – |
| I 24 | AM | C2 | $C_2$ | 1/deg |
| I 25 | AM | C3 | $C_3$ | $1/deg^2$ |
| I 27 | AM | CDF1 | $C_{DF1}$ | 1/ft |
| I 28 | AM | CDF2 | $C_{DF2}$ | $1/ft^2$ |
| I 26 | AM | CDB | $C_{DB}$ | – |
| I 30 | AM | C4 | $C_4$ | $1/deg^2$ |
| I 31 | AM | C5 | $C_5$ | $1/deg^2$ |

## TABLE XXVII. FUNCTION GENERATORS IN AERODYNAMICS SUBROUTINE (CONCLUDED)

| FGEN No. | Input | Output | Symbol | Units |
|---|---|---|---|---|
| I 32 | AM | SL1 | $\ell_1$ | $1/deg^2$ |
| I 33 | AM | SL2 | $\ell_2$ | $1/deg^3$ |
| I 34 | AM | SL3 | $\ell_3$ | $1/deg^4$ |
| I 35 | AM | SL4 | $\ell_4$ | $1/deg^5$ |
| I 36 | AM | SL5 | $\ell_5$ | 1/deg |
| I 37 | AM | SL6 | $\ell_6$ | $1/deg^2$ |
| I 38 | AM | SL7 | $\ell_7$ | $1/deg^2$ |
| I 39 | AM | SL8 | $\ell_8$ | $1/deg^2$ |
| I 40 | AM | SL9 | $\ell_9$ | $1/deg^3$ |
| K 40 | AM | SL10 | $\ell_{10}$ | $1/deg^3$ |
| I 41 | TIME | DLØD | $\Delta\ell/d$ | — |
| I 42 | TIME | ØMDLØL | $1 - \Delta\ell/\ell_{TB}$ | — |
| I 44 | TIME | THRUST | Rocket motor thrust | lb |
| I 45 | ALT | PRES | Ambient pressure | $lb/ft^2$ |
| I 43 | ALT | RHØ | Air density | $slug/ft^3$ |

### 2.3.11 Program Glossary

(U) A master program glossary defining all the elements appearing in both the V and C arrays is contained in Tables XXVIII and XXIX, respectively.

## 2.4 TYPE OF SIMULATION FACILITIES TO BE USED

(U) The only equipment that is required is the GE-635 or any other digital computer capable of compiling FORTRAN IV source decks. No training equipment or mockups are required.

## 2.5 INSTRUMENTATION

(U) None is required.

## 2.6 DATA REDUCTION AND ANALYSIS TECHNIQUES

(U) No special data reduction techniques are required.

TABLE XXVIII. MASTER GLOSSARY, V ARRAY

| Name | Quantity | Units | Coordinate System |
|---|---|---|---|
| V(1) | hm, Missile altitude above ground | ft | Inertial |
| V(2) | $\delta_{ac}$, Aileron deflection command | deg | Missile |
| V(3) | $\delta_{pc}$, Pitch deflection command | deg | Missile |
| V(4) | $\delta_{yc}$, Yaw deflection command | deg | Missile |
| V(5) | $\delta_a$, Aileron deflection | deg | Missile |
| V(6) | $\delta_p$, Pitch deflection | deg | Missile |
| V(7) | $\delta_y$, Yaw deflection | deg | Missile |
| V(8) | $V_x$, Missile velocity X-axis | ft/sec | Missile |
| V(9) | $V_y$, Missile velocity Y-axis | ft/sec | Missile |
| V(10) | $V_z$, Missile velocity Z-axis | ft/sec | Missile |
| V(11) | $\omega_x$, Angular velocity | rad/sec | Missile |
| V(12) | $\omega_y$, Angular velocity | rad/sec | Missile |
| V(13) | $\omega_z$, Angular velocity | rad/sec | Missile |
| V(14) | $A_x$, Propulsion and aerodynamic acceleration | g | Missile |
| V(15) | $A_y$, Propulsion and aerodynamic acceleration | g | Missile |
| V(16) | $A_z$, Propulsion and aerodynamic acceleration | g | Missile |
| V(17) | $A_{zc}$, Elevation maneuver command | g | Autopilot |
| V(18) | $A_{yc}$, Azimuth maneuver command | g | Autopilot |
| V(19) | $\psi$, Euler yaw angle | rad | |
| V(20) | $\phi$, Euler roll angle | rad | |
| V(21) | $\theta$, Euler pitch angle | rad | |
| V(22) | $R_x$, Seeker boresight range | ft | Seeker |
| V(23) | $R_y$, Seeker lateral range | ft | Seeker |
| V(24) | $R_z$, Seeker normal range | ft | Seeker |
| V(25) | $\epsilon_z$, Tracking error angle, pitch | rad | Seeker |
| V(26) | $\epsilon_y$, Tracking error angle, yaw | rad | Seeker |
| V(27) | $\nu$, Seeker elevation gimble angle | rad | |
| V(28) | $\eta$, Seeker azimuth gimble angle | rad | |

TABLE XXVIII. MASTER GLOSSARY, V ARRAY (CONTINUED)

| Name | Quantity | Units | Coordinate System |
|---|---|---|---|
| V(29) | $R_i$, Horizontal longitudinal range component | ft | Inertial |
| V(30) | $R_j$, Horizontal lateral range component | ft | Inertial |
| V(31) | $\epsilon_{gz}$, Gate error angle, pitch | rad | Seeker |
| V(32) | $\epsilon_{gy}$, Gate error angle, yaw | rad | Seeker |
| V(33) | a, Total missile angle of attack | deg | Missile |
| V(34) | $a_p$, Missile pitch angle of attack | deg | Missile |
| V(35) | $a_y$, Missile yaw angle of attack | deg | Missile |
| V(36) | $V_i$, Horizontal longitudinal velocity component | ft/sec | Inertial |
| V(37) | $V_j$, Horizontal lateral velocity component | ft/sec | Inertial |
| V(38) | $V_k$, Vertical velocity component | ft/sec | Inertial |
| V(39) | q, Dynamic pressure | $lb/ft^2$ | |
| V(40) | Total missile velocity | ft/sec | |
| V(41) | Missile Mach number | | |
| V(42) | $a_{cp}$, Acceleration command pitch | g | Autopilot |
| V(43) | $a_{cy}$, Acceleration command yaw | g | Autopilot |
| V(44) | $\dot{\omega}_x$ } Scalar components of missile angular acceleration in missile axes | $rad/sec^2$ | Missile |
| V(45) | $\dot{\omega}_y$ | | |
| V(46) | $\dot{\omega}_z$ | | |
| V(47) | $\dot{\delta}_{ac}$, Aileron command rate | deg/sec | Missile |
| V(48) | $\dot{\delta}_{pc}$, Elevator command rate | deg/sec | Missile |
| V(49) | $\dot{\delta}_{yc}$, Rudder command rate | deg/sec | Missile |
| V(50) | Closest approach at end of flight | ft | |
| V(51) | Range component in Y seeker axis | ft | Seeker |
| V(52) | Range component in Z seeker axis | ft | Seeker |
| V(53) | $\omega'_x$ } Missile body rates in autopilot axes | rad/sec | Autopilot |
| V(54) | $\omega'_y$ | | |
| V(55) | $\omega'_z$ | | |

TABLE XXVIII. MASTER GLOSSARY, V ARRAY (CONTINUED)

| Name | Quantity | Units | Coordinate System |
|---|---|---|---|
| V(56) | $\dot{\omega}'_x$ } Scalar components of missile angular acceleration in autopilot axes | rad/sec$^2$ | Autopilot |
| V(57) | $\dot{\omega}'_y$ | | |
| V(58) | $\dot{\omega}'_z$ | | |
| V(59) | $A'_x$ } Propulsive and aerodynamic acceleration components in autopilot axes | g | Autopilot |
| V(60) | $A'_y$ | | |
| V(61) | $A'_z$ | | |
| V(62) | $V'_x$ } Missile velocity components in autopilot axes | ft/sec | Autopilot |
| V(63) | $V'_y$ | | |
| V(64) | $V'_z$ | | |
| V(65) | Special test variable - used as system diagnostic | | |
| V(66) | Total miss distance | ft | Miss Distance |
| V(67) | x component of range | ft | Autopilot |
| V(68) | y component of range | ft | Autopilot |
| V(69) | z component of range | ft | Autopilot |
| V(70) | y component of miss | ft | Miss Distance |
| V(71) | z component of miss | ft | Miss Distance |
| V(72) | x component of acceleration | g | Inertial |
| V(73) | y component of acceleration | g | Inertial |
| V(74) | z component of acceleration | g | Inertial |
| V(75) | y component of acceleration at blind range | g | Miss Distance |
| V(76) | z component of acceleration at blind range | g | Miss Distance |
| V(77) | Blind time in yaw channel | sec | |
| V(78) | Blind time in pitch channel | sec | |
| V(79) | Final line of sight angle (vertical) | rad | Inertial |
| V(80) | Final heading angle (horizontal) | rad | Inertial |
| V(81) | x component, LOS rate | rad/sec | Inertial |
| V(82) | y component, LOS rate | rad/sec | Inertial |
| V(83) | z component, LOS rate | rad/sec | Inertial |
| V(84) | $\Lambda$, Guidance gain | | |

TABLE XXVIII. MASTER GLOSSARY, V ARRAY (CONTINUED)

| Name | | Quantity | Units | Coordinate System |
|---|---|---|---|---|
| V(85) | DE | Total yaw precession rate | | |
| V(86) | DEXS | Total pitch precession rate | | |
| V(87) | E | Yaw gyro inertial angle | | |
| V(88) | C1 | Yaw look angle (indicated) | | |
| V(90) | G1 | Forcing function cross-coupled equation 1 | | |
| V(91) | DG1 | Derivative forcing function cross-coupled equation 1 | | |
| V(92) | G2 | Forcing function cross-coupled equation 2 | | |
| V(93) | DG2 | Derivative forcing function cross-coupled equation 2 | | |
| V(94) | D1N | Integral forcing function cross-coupled equation 1 | | |
| V(95) | G2N | Integral forcing function cross-coupled equation 2 | | |
| V(96) | FFE | Forcing function yaw axis | | |
| V(97) | DFE | Derivative forcing function yaw axis | | |
| V(98) | FEXS | Forcing function pitch axis | | |
| V(99) | DFEXS | Derivative forcing function pitch axis | | |
| V(100) – V(110) | | Not Used | | |

TABLE XXVIII. MASTER GLOSSARY, V ARRAY (CONCLUDED)

| Name | Quantity | Units | Coordinate System |
|---|---|---|---|
| V(111) | Sum 1 - Tracker sampler bias | sec | |
| V(112) | TEAYD Tracker error yaw · RKAMG | deg | |
| V(113) | TEAPD Tracker error pitch · RKAMG | deg | |
| V(114) | TEAYS - Tracker ZØH output signal, yaw | deg | |
| V(115) | TEAPS - Tracker ZØH output signal, pitch | deg | |
| V(116) | VSYP - Tracker output signal pitch | deg/sec | |
| V(117) | VSPP - Tracker output signal yaw | deg/sec | |
| V(118) | TEYD - Tracking error - yaw | deg | |
| V(119) | TEPD - Tracking error - pitch | deg | |
| V(120) | UND Seeker elevation | deg | |
| V(121) | ETAD Seeker azimuth | deg | |
| V(122) | WXD | | |
| V(123) | WYD Missile angular velocity | deg | |
| V(124) | WZD | | |
| V(125) | XLOSD | | |
| V(126) | YLOSD LOS Rate, inertial | deg/sec | |
| V(127) | ZLOSD | | |
| V(128) | ANT (New) | | |
| V(129) | $\psi$ Yaw D | | |
| V(130) | $\phi$ Roll D Error angle | deg | |
| V(131) | $\theta$ Pitch | | |
| V(132) | DED Total precession rate, yaw | | |
| V(133) | DEXSD Total precession rate, pitch | deg/sec | |

TABLE XXIX. MASTER GLOSSARY, C ARRAY*

| Name | Quantity | Units |
|---|---|---|
| C(1) | Not used | |
| C(2) | Autopilot activation time | sec |
| C(3) | Blind range, pitch | ft |
| C(4) | Gate error angle/$\frac{d}{dt}$ V (18) | $\frac{\text{mills}}{\text{g/sec}}$ |
| C(5) | Blind range filter time constant | sec |
| C(6) | $K_p$, Angle restoration gain | g/deg |
| C(7) | $\beta_o$, Restoration angle bias | deg |
| C(8) | $\tau_\beta$, Angle restoration filter time constant | sec |
| C(9) | Blind range, yaw | ft |
| C(10) | Not used | |
| C(11) | Not used | |
| C(12) | Not used | |
| C(13) | Half field of view, seeker | rad |
| C(14) | $K_g$, Guidance gain | g/deg/sec |
| C(15) | $\tau_a'$, Tracker time constant | sec |
| C(16) | Precession rate limit | rad/sec |
| C(17) | $C_1$ Seeker drift terms | rad/sec |
| C(18) | $C_2$ Seeker drift terms | rad/sec |
| C(19) | $C_3$ Seeker drift terms | rad/sec/g |
| C(20) | $C_4$ Seeker drift terms | |
| C(21) | $C_5$ Seeker drift terms | 1/sec |
| C(22) | $C_6$ Seeker drift terms | 1/sec |
| C(23) | $C_7$ Seeker drift terms | sec |
| C(24) | $C_8$ Seeker drift terms | sec |
| C(25) | $C_9$ Seeker drift terms | rad/sec/g |
| C(26) | $C_{10}$ Seeker drift terms | rad/sec/g |
| C(27) | $C_{11}$ Seeker drift terms | 1/sec/g |
| C(28) | $C_{12}$ Seeker drift terms | rad/sec/$g^2$ |
| C(29) | $C_{13}$ Seeker drift terms | rad/sec/$g^2$ |

*See Appendix V - Volume III - System Analysis Document for typical values of the C array.

TABLE XXIX. MASTER GLOSSARY, C ARRAY (CONTINUED)

| Name | Quantity | Units |
|---|---|---|
| C(30) | $C_{14}$ | rad/sec/g$^2$ |
| C(31) | $C_{15}$ | rad/sec |
| C(32) | $C_{16}$ | rad/sec |
| C(33) | $C_{17}$ | rad/sec/g |
| C(34) | $C_{18}$ Seeker drift terms | |
| C(35) | $C_{19}$ | 1/sec |
| C(36) | $C_{20}$ | sec |
| C(37) | $C_{21}$ | rad/sec/g |
| C(38) | $C_{22}$ | rad/sec/g |
| C(39) | $C_{23}$ | rad/sec/g$^2$ |
| C(40) | $K_1$, Tracking loop velocity gain | 1/sec |
| C(41) | TG, Gimbal [illegible]load | g |
| C(42) | Drift control (set to 1.0 to include drift) | |
| C(43) | Autopilot lateral channel activation switch level | g |
| C(44) | $K_a$, Autopilot lateral channel gain | deg/g |
| C(45) | $K_b$, Autopilot damping gain | deg/deg/sec |
| C(46) | $\phi_K$, Roll channel gain | deg/deg |
| C(47) | Acceleration limit, lateral channels | g |
| C(48) | Command limit, lateral channels | deg |
| C(49) | $\tau_{sn}$, Lead time constant | sec |
| C(50) | $\tau_{sd}$, Lag time constant | sec |
| C(51) | $\tau_{bn}$, Lead time constant | sec |
| C(52) | $\tau_{YDLG}$, Lag time constant | sec |
| C(53) | Roll channel deadtime | sec |
| C(54) | Roll rate signal limit | rad/sec |
| C(55) | $\tau_{ra}$, Time constant in autopilot | sec |
| C(56) | Roll rate switch level | rad/sec |
| C(57) | $\tau_{acc}$, Lateral channel time constant | sec |
| C(58) | $\tau_{RG}$, Lateral channel time constant | sec |

TABLE XXIX. MASTER GLOSSARY, C ARRAY (CONTINUED)

| Name | Quantity | Units |
|---|---|---|
| C(59) | $\dot{\phi}_K$, Roll rate gain | deg/deg/sec |
| C(60) | Roll channel limit change time | sec |
| C(61) | $\tau_{r1}$, Lead time constant | sec |
| C(62) | $\tau_{r2}$, Lag time constant | sec |
| C(63) | Roll command limit | deg |
| C(64) | Roll command limit | deg |
| C(65) | Lateral channel rate limit | rad/sec |
| C(66) | Not used | |
| C(67) | Not used | |
| C(68) | Not used | |
| C(69) | Not used | |
| C(70) | Not used | |
| C(71) | Not used | |
| C(72) | Not used | |
| C(73) | S, Reference area | $ft^2$ |
| C(74) | d, Reference diameter | ft |
| C(75) | Sea level pressure | $lb/ft^2$ |
| C(76) | Booster burn time | sec |
| C(77) | Sustainer burn time | sec |
| C(78) | $\ell_{TB}$, Tail length, burnout | ft |
| C(79) | $\overline{X}$, cg to flipper distance | ft |
| C(80) | $C_{\ell_p}$, Roll damping coefficient | 1/rad |
| C(81) | $A_e$, Nozzle exit area | $ft^2$ |
| C(82) | $J_{xo}$, Launch roll inertia | $slug\text{-}ft^2$ |
| C(83) | $J_{x1}$, End-of-boost roll inertia | $slug\text{-}ft^2$ |
| C(84) | $J_{xt}$, End-of-sustain roll inertia | $slug\text{-}ft^2$ |
| C(85) | $J_{yo}$, Launch lateral inertia | $slug\text{-}ft^2$ |
| C(86) | $J_{y1}$, End-of-boost lateral inertia | $slug\text{-}ft^2$ |
| C(87) | $J_{yt}$, End-of-sustain lateral inertia | $slug\text{-}ft^2$ |
| C(88) | $M_o$, Launch mass | slugs |

TABLE XXIX. MASTER GLOSSARY, C ARRAY (CONTINUED)

| Name | Quantity | Units |
|---|---|---|
| C(89) | $M_i$, End-of-boost mass | slugs |
| C(90) | $M_t$, End-of-sustain mass | slugs |
| C(91) | Burn time prior to launch | sec |
| C(92) | Target altitude | ft |
| C(93) | Not used | |
| C(94) | Not used | |
| C(95) | Not used | |
| C(96) | Not used | |
| C(97) | Control surface servo gain | 1/sec |
| C(98) | Control surface time constant | sec |
| C(99) | Control surface velocity limit | deg/sec |
| C(100) | Control surface angle limit | deg |
| C(101) | Control surface rate threshold | deg/sec |
| C(102) | Not used | |
| C(103) | Dual purpose input: Min. range of aimpoint wander when C(106) ≠ 0. Initial target accelera-tion when C(106) = 0. Subroutine bypassed when C(103) = 0. | ft<br>g |
| C(104) | Dual purpose input: A, used in aimpoint wander when C(106) ≠ 0. Final target velocity when C(106) = 0. | ft/sec |
| C(105) | Dual purpose input: PLØTK, used in aimpoint wander when C(106) ≠ 0. Boresight range at which target motion starts when C(106) = 0. | ft |
| C(106) | PHØTØK, used in aimpoint wander | |
| C(107) | Not used | |
| C(108) | Not used | |
| C(109) | LOS memory threshold in guidance law | deg |
| C(110) | G-Bias | g |
| C(111) | LOS memory gain in guidance law | g/deg |

TABLE XXIX. MASTER GLOSSARY, C ARRAY (CONTINUED)

| Name | Quantity | Units |
|---|---|---|
| C(112) | | |
| C(113) | | |
| C(114) | TB Sampling time (function of distribution to target (0.0800) | |
| C(115) | TC Sampling time control (0.1000) | |
| C(116) | 3S - Gyro rotor speed | rad/sec |
| C(117) | K2T - Precession torque coefficient | gcm/V |
| C(118) | Dump program control logic | B = 0 |
| C(119) | - Rail control logic | S = 1.0 |
| C(120) | | |
| C(121) | | |
| C(122) | | |
| C(123) | | |
| C(124) | | |
| C(125) | | |
| C(126) | Not used | |
| C(127) | | |
| C(128) | | |
| C(129) | | |
| C(130) | | |
| C(131) | | |
| C(132) | | |
| C(133) | PSIPØ LOS Memory threshold guidance law | deg |
| C(134) | BP - G-Bias | g |
| C(135) | AKSIGP LOS Memory gain in guidance law | g/deg |
| C(136) | GNUT - Program logic control - W/0-0, W = 1.0 | |
| C(137) | DFR Coulomb friction drift factor | |
| C(138) | DST Spring torque drift factor | |
| C(139) | DSU Unbalance drift factor | |
| C(140) | DAN Anisoelastic drift factor | |

TABLE XXIX. MASTER GLOSSARY, C ARRAY (CONCLUDED)

| Name | Quantity | Units |
|---|---|---|
| C(141) | DDU Dynamic unbalance factor | |
| C(142) | SK Torquer gain coefficient | V/deg/sec |
| C(143) | AKT - Tracker gain constant | 1 sec |
| C(144) | TS - Sampling period | sec |
| C(145) | OMEGLD - Precession rate limit | deg/sec |
| C(146) | GKK - Guidance gain | g/deg/sec |
| C(147) | BIAS - Sampling rate offset bias | sec |
| C(148) | TLDP - Tracker filter lead time constant - Pitch | sec |
| C(149) | TLGP - Tracker filter lag time constant - Pitch | sec |
| C(150) | TLOY - Tracker filter lead time constant - Yaw | sec |
| C(151) | TLGY - Tracker filter lag time constant - Yaw | sec |
| C(152) | SPOT - Tracker spot size | Ft |
| C(153) | CF1 Friction factor coefficient | |
| C(154) | CF2 Friction factor coefficient | |
| C(155) | CF3 Friction factor coefficient | |
| C(156) | | |
| C(157) | | |
| C(158) | Not used | |
| C(159) | | |
| C(160) | | |

## INITIAL DISTRIBUTION

| | |
|---|---|
| HQ USAF (RDPA) | 1 |
| HQ USAF (RDQRM) | 1 |
| HQ USAF (XOOW) | 2 |
| AFSC (DLP) | 1 |
| AFSC (DLTW) | 1 |
| AFSC (DLXP) | 1 |
| AFSC (SDWM) | 1 |
| TAC (DRPN) | 1 |
| AFAL (TE) | 1 |
| AFAL (NV) | 1 |
| ASD (SD) | 2 |
| ASD (ENYY) | 2 |
| FTD (PDJR) | 5 |
| AFFDL (FYS) | 1 |
| RADC (IRAP) | 1 |
| AUL (AUL/LSE-70-239) | 1 |
| SANDIA CORP | 1 |
| ARMY AIR MOBILITY DIV | 2 |
| DEP CH STAFF USMC CODE AAW-IC | 1 |
| CH NAV OPRS (OP-982E21) | 1 |
| OODR&E, OAD/COMBAT SUPPORT | 1 |
| WPNS SYS EVAL GP | 1 |
| ARPA (TIO) | 1 |
| NAV AIR SYS COMD, AIR-350B | 1 |
| DASA INFO & ANAL CTR GE CO-TEMPO | 1 |
| RAND CORP | 1 |
| NWC (CODE 456) | 2 |
| NWC (CODE 753) | 1 |
| REDSTONE SCI INFO CTR | 1 |
| USA ENGR R&D LABS | 1 |
| BATTELLE MEMORIAL INST | 1 |
| INST FOR DEF ANALYSES | 1 |
| DDC | 2 |
| DL | 1 |
| DLB | 1 |
| DLW | 1 |
| DLU | 1 |
| DLX | 1 |
| DLOSL | 2 |
| AFSC (DLW) | 1 |
| ASD (SD-65) | 1 |
| DLWS | 1 |
| PACAF (IGY) | 1 |
| ASD (ENYS) | 1 |



(The reverse of this page is blank)

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)*<br>Hughes Aircraft Company<br>Canoga Park, California | 2a. REPORT SECURITY CLASSIFICATION<br>UNCLASSIFIED<br>2b. GROUP |

3. REPORT TITLE

CLOSE AIR SUPPORT WEAPON ENGINEERING DESIGN STUDY VOLUME VI. MISSILE SIMULATION

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

Final Report - September 1970 to December 1970

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

Dr. W. S. Walker
G. E. Blackshaw
R. W. Knowles

| 6. REPORT DATE | 7a. TOTAL NO. OF PAGES | 7b. NO. OF REFS |
|---|---|---|
| January 1971 | 110 | |

| | |
|---|---|
| 8a. CONTRACT OR GRANT NO.<br>F08635-71-C-0048<br>b. PROJECT NO.<br>670B<br>c. Task No. 00<br>d. Work Unit No. 00 | 9a. ORIGINATOR'S REPORT NUMBER(S)<br><br>9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*<br>AFATL-TR-71-7, Volume VI |

10. DISTRIBUTION STATEMENT

Distribution limited to U. S. Government agencies only; this report documents test and evaluation; distribution limitation applied January 1971 . Other requests for this document must be referred to the Air Force Armament Laboratory (DLWS), Eglin Air Force Base, Florida 32542.

| 11. SUPPLEMENTARY NOTES | 12. SPONSORING MILITARY ACTIVITY |
|---|---|
| Available in DDC | Air Force Armament Laboratory<br>Air Force Systems Command<br>Eglin Air Force Base, Florida 32542 |

13. ABSTRACT

(U) The objective of the engineering design study of the close air support weapon (CASW) was to provide design considerations for the new close air support missile (CASM). The derivation of the missile was undertaken based on the modification of an existing missile. This study incorporates operational requirement and warhead effectiveness studies for various close air support targets leading to warhead and launch envelope recommendations. A thorough analysis of the system performance and terminal accuracy was conducted. Missile simulation models and a system description, including missile, launcher, avionics, and aerospace ground equipment (AGE) are provided. A cost analysis exercise was conducted for the design, development, test and evaluation (DDT&E) and production of the candidate approach. This report consists of six volumes: Management Summary, Operational Analysis and Warhead Effectiveness, System Analysis, System Design, Cost Analysis, and Missile Simulation.

UNCLASSIFIED
Security Classification

| 14. KEY WORDS | LINK A | | LINK B | | LINK C | |
|---|---|---|---|---|---|---|
| | ROLE | WT | ROLE | WT | ROLE | WT |
| Close Air Support Weapon | | | | | | |
| Missile Operational Requirements | | | | | | |
| Missile Warhead Effectiveness | | | | | | |
| Missile Launch Envelope | | | | | | |
| Missile System Performance | | | | | | |
| Missile Terminal Accuracy | | | | | | |
| Missile Simulation Model | | | | | | |

UNCLASSIFIED
Security Classification