AD NUMBER

AD-B004 003

NEW LIMITATION CHANGE

TO

DISTRIBUTION STATEMENT - A

Approved for public release; distribution is unlimited

LIMITATION CODE: 1

FROM

DISTRIBUTION STATEMENT - B

Distribution authorized to U.S. Gov't. agencies only.

LIMITATION CODE: 3

AUTHORITY

ASD; MAR 31, 1975

Report ASD/XR-TR-75-4

ANALYTICAL METHODOLOGY FOR EVALUATION OF PAYOFFS FOR INFRARED COUNTERMEASURES AND SUPPRESSION (EPICS)

I. A. Ratkovic
A. Leslie
Y. Nishimoto
Z. Neumark
R. A. Gosler

March 1975

Final Report

[redacted]bution limited to U.S. Government [redacted]
and [redacted] for this document must [redacted]
[redacted] Air Force Base, Ohio 45433.

Prepared for

AERONAUTICAL SYSTEMS DIVISION
Wright-Patterson Air Force Base, Ohio 45433

NOTICES

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

This technical report has been reviewed and is approved for publication.

JAMES H. HALL
Colonel, USAF
Deputy for Development Planning

REPRODUCTION QUALITY NOTICE

This document is the best quality available. The copy furnished to DTIC contained pages that may have the following quality problems:

- Pages smaller or larger than normal.
- Pages with background color or light colored printing.
- Pages with small type or poor printing; and or
- Pages with continuous tone material or color photographs.

Due to various output media available these conditions may or may not cause poor legibility in the microfiche or hardcopy output you receive.

☐ If this block is checked, the copy furnished to DTIC contained pages with color printing, that when reproduced in Black and White, may change detail of the original copy.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER ASD/XR-TR-75-4	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Analytical Methodology for Evaluation of Payoffs for Infrared Countermeasures and Suppression (EPICS)		5. TYPE OF REPORT & PERIOD COVERED Final Report
		6. PERFORMING ORG. REPORT NUMBER P74-479 HAC Ref D1757
7. AUTHOR(s) J. A. Ratkovic, A. Leslie, K. Nishimoto, Z. Neumark, R. A. Gesler		8. CONTRACT OR GRANT NUMBER(s) F33615-74-C-4076
9. PERFORMING ORGANIZATION NAME AND ADDRESS Hughes Aircraft Company Electro-Optical and Data Systems Group Electro-Optical Division, Culver City, CA 90230		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Aeronautical Systems Division Wright-Patterson Air Force Base, Ohio		12. REPORT DATE March 1975
		13. NUMBER OF PAGES 132
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Distribution limited to U.S. Government agencies only; test and evaluation; March 1975. Other requests for this document must be referred to Aeronautical Systems Division W-P, AFB, Ohio

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

18. SUPPLEMENTARY NOTES

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

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

An analytical methodology for assessing the impact of infrared suppression techniques on aircraft survivability in an IR missile threat environment has been developed. The methodology, designated as EPICS (Evaluation of Payoffs for Infrared Countermeasures and Suppression) consists of two digital computer programs: (1) ASDIR II, and (2) M/T/CM. The ASDIR II program generates aircraft IR signatures. The M/T/CM, a five-degrees-of-freedom program, computes the probability of aircraft

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

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20. Abstract (Continued)

survival based on a simulated missile-target engagement, including the dynamics and principal characteristics of the aircraft and the missile. The program can also simulate the deployment of decoys such as flares or pyrophorics.

The utility of the program lies in that it can provide guidelines during aircraft configuration studies, assess effects of design changes on aircraft survivability, and permits tradeoff studies to be made between various CMs such as suppression, shielding and flare deployment.

The programs are operational on the CDC 6600 digital computer at Wright-Patterson Air Force Base, Ohio.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS (Continued)

LIST OF TABLES

LIST OF TABLES (Continued)

INTRODUCTION

This document constitutes the final report on Analytical Methodology for Evaluation of Payoffs for Infrared Countermeasures and Suppression (EPICS), Contract No. F33615-75-C-4076. The prime objective of this study was to develop a methodology or analytical tool for rapidly and efficiently assessing the impact of infrared suppression techniques on aircraft survivability. Specifically, the intended purpose of the methodology is to provide a capability to analytically predict the effectiveness of aircraft design changes (primarily those related to infrared signature) on the probability of aircraft survival in a specified infrared threat environment.

The development of an analytical tool that meets the above objectives was achieved. This tool consists of two complementary digital computer programs: (1) an infrared target signature model (ASDIR II) and (2) a missile/target/countermeasures (M/T/CM)* model. A third program SPKINT (a subroutine) provides the interface between the two models. All three programs are fully operational on the CDC 6600 computer system. This total methodology system is designated as EPICS.

The first program, ASDIR II, was primarily developed by the Air Force. Hughes modified it and made it operational.** It is documented under a separate cover.*** The inputs to ASDIR II are engine specification data (gas dynamics or measured plume data to determine the engine exhaust plume radiation) and engine hot metal parts in terms of temperature and radiating

---

*The baseline for the M/T/CM program was developed by Hughes under "IRCM Simulation Study", Contract No. F33615-C-74-1680 for AFAL.

**This task constituted a deviation from the Statement of Work. Originally, a Hughes'-developed target signature program, IRSIG, was to be used. However, on Program Monitors instructions, ASDIR II was used instead.

***Stone, Charles W., Capt., USAF and Tate, Stanley, ASDIR II (Vol. I, II, and III), Deputy for Development Planning, Aeronautical Systems Division, Wright-Patterson Air Force Base, Ohio, No. ASD/XR-TR-75-1, January 1975.

area as a function of aspect. Similarly, the contribution to the total IR signature due to skin aerodynamic heating is an input in terms of temperature and radiating area for as many as twenty skin surfaces. Finally, the viewing geometry — target and observer altitudes, aspect angle, and slant range is an input to the ASDIR II program.

The outputs of this code are in the form of polar data for the source spectral radiant intensity, J_λ, integrated over the missile response band and the apparent spectral radiant intensity, $J_\lambda \tau_\lambda$, also integrated over the missile band, but at the point of a remote observer. These data then serve as input to the M/T/CM program.

The second major element of EPICS is the M/T/CM. This program is described in detail in this report. The program is a generic five-degrees-of-freedom dynamic simulation of the total missile/target encounter in a countermeasures environment. The prime countermeasures technique that can be evaluated using this program are IR signature reduction through suppression or shielding, and active decoys such as flares or pyrophorics. The output of the program is a probability of target survival (P_S) under a varied set of launch conditions and for various IR missile threats. The P_S is defined by

$$P_S = \frac{\text{number of misses}}{\text{total number of launch cases}}$$

As indicated in a preceding footnote, the baseline subroutines for the M/T /CM program were developed by Hughes under an earlier Air Force study program, however, in the present study contract this baseline program was considerably expanded and improved. In addition, the program was modified to accept inputs from the ASDIR II program with the aid of a subroutine called SPKINT. The total program was compiled on the CDC 6600 computer system (it was originally written for the SIGMA 5 Computer). The major changes to M/T/CM include:

Modularization of the program

Addition of flare control options (function of range and time-to-go)

Addition of superelevation angle subroutine for ground-to-air missiles

Addition of launch mode selection option in azimuth and elevation

Reduction of program execution time

Sample runs using all three programs were also made; see Section 10.

In this program, a specific target is represented by its physical characteristics, its dynamic parameters, and its infrared signature. The physical characteristics include the size of the aircraft, its wing span and the engine location. The dynamic parameters include initial velocity, accelerations, and maneuvers. The infrared target signature is represented by the effective apparent radiant intensity, $J\tau_{\Delta\lambda}$, as a function of range and aspect angle, and is calculated by ASDIR II. Interpolation on range and aspect angle between $(J\tau)_{\Delta\lambda}$ data points provide the appropriate values during the simulated flight which in turn are used to calculate the effective irradiance at the missile seeker.

Threat missiles are represented by a number of parameters divided into six categories: seeker, signal processing, guidance, aerodynamics, propulsion, and physical characteristics. Currently, 12 missiles, 25 aircraft, and 4 flare types have been defined and are part of the simulation file.

The M/T/CM program has been validated by comparing simulated engagements of captive missiles being decoyed by flares, with flight test results (using same flight conditions and the same missiles) conducted by the Naval Weapons Center, China Lake, California.

As mentioned above, the interface between the ASDIR II signature program and the M/T/CM program is provided by an auxiliary spectral integration subroutine, called SPKINT. This routine integrates the apparent spectral radiant intensity values, $J_\lambda \tau_\lambda$, over any specified spectral interval to obtain effective radiant intensity $(J\tau)_{\Delta\lambda}$. In general, the integration is performed for spectral intervals corresponding to the spectral bandpasses of the 12 missiles on file. The SPKINT subroutine is described in Section 9.

In summary, the EPICS methodology provides a tool to assess the impact that aircraft design has on the aircraft survivability in an infrared missile threat environment, evaluating design changes and conducting tradeoff studies during preliminary design and determining aircraft survivability in a flare countermeasures environment.

Figure 1 shows a flow diagram for the simulation program executive operation that calls all subroutines. The program is broken into eight major areas with each area being subsequently discussed in Sections 1 through 8 of this report.

Section 1 deals with the creation of files on which the necessary constants for the missile, target, and flare are stored. Long lists of input data can be eliminated, by creating files for each missile, target, and flare to be evaluated and the required constants can be specified by simply referring to a file name.

In Section 2, the launch geometry variables, the flare deployment strategy, the aircraft maneuver option, and all other program options are set. In this section, all program variables are initialized.

Section 3 of the program updates the position, velocity, and acceleration of the missile, target, and flare(s) with respect to inertial coordinates.

In Section 4, relative ranges, range rates, angles, and angular rates are determined between the missile and the target as well as the missile and flare(s).

The irradiance at the missile dome from the target and the flare(s) is computed in Section 5.

In Section 6, the aimpoint location is determined based on the irradiance levels of the sources in its field of view (FOV) and the type of signal processing in the missile. This aimpoint location is then fed back into the missile dynamics (Section 3) through missile guidance.

When the simulation program goes into an abort mode, the point of closest approach of the missile to the target is determined. The details are given in Section 7.

Section 8 describes how the probability of hit is determined based on the closest approach distance, aircraft dimensions, type, and lethality of the missile warhead.

The geometry for the missile and target encounter is shown in Figure 2. This simulation is based on a two-plane geometry, because a missile essentially processes its target position and rate information and provides guidance commands in two separate planes -- horizontal and vertical. Coupling between the two planes is accomplished by the range and velocity variables.

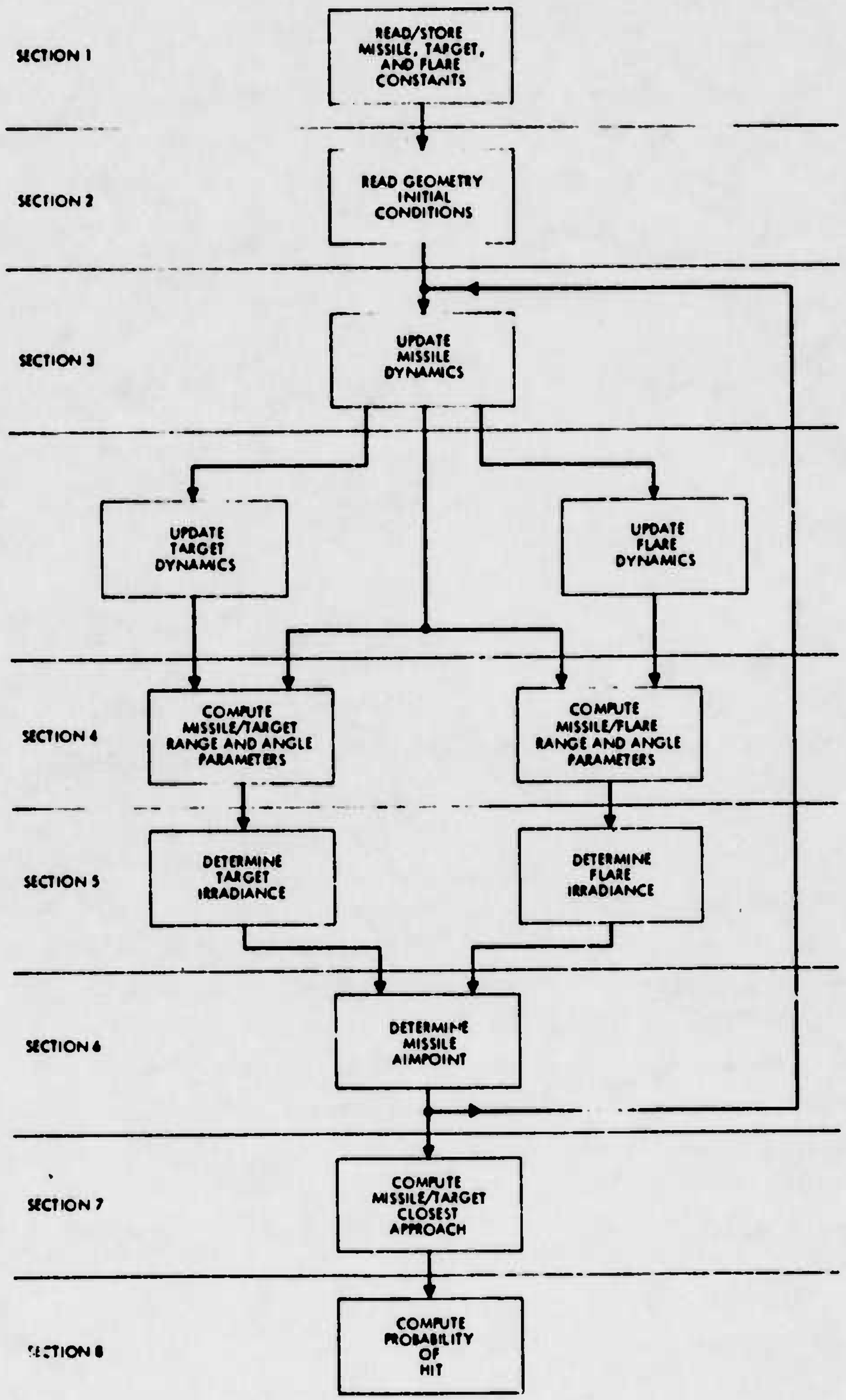


Figure 1. Simulation program executive operation flow diagram

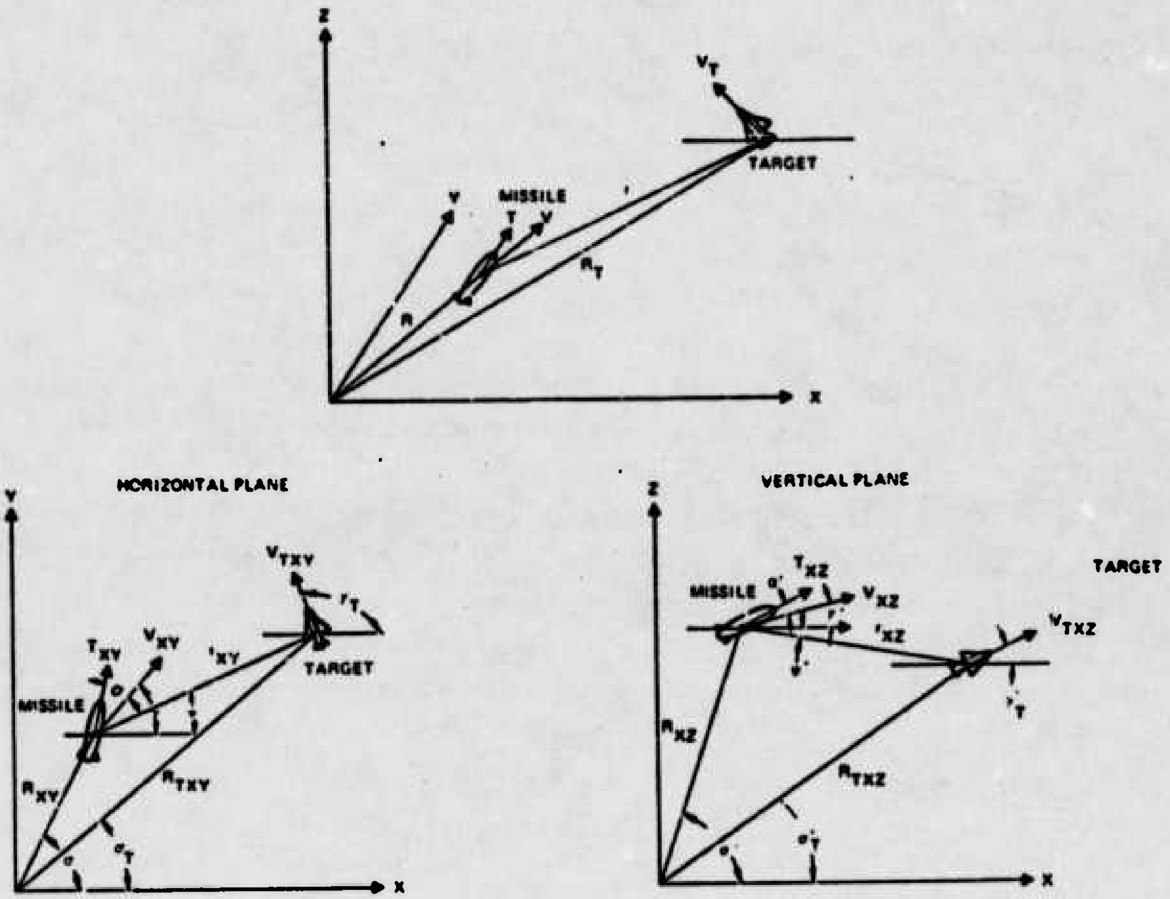


Figure 2. Missile and target geometry

The coordinate system in which missile, target, and flare positions are calculated is shown in Figure 2. This system has as its crigin a point on the ground directly below the launch point of the missile. The Z axis is parallel to gravity and positive up. Therefore the initial position of the missile is given by (O, O, Z_A), where Z_A is the launch altitude. The X axis is perpendicular to gravity and oriented such that the initial position of the target is in the X-Z plane. The Y-axis completes the orthognal, right-handed coordinate system. The initial position of the target is given by (X_T, O, Z_T) with X_T being the horizontal range between the missile and target at launch, and Z_T being the target altitude. From this definition, the line-of-sight (LOS) is in the X-Z plane from the missile to target at launch. The target aspect relative to the missile is set by the target velocity vector.

The state vectors listed in Table 1 represent the X-, Y-, and Z-components of position, velocity, and acceleration of the missile and target. The state variables are divided into two sets: one set for the horizontal plane and one set for the vertical plane.

Figure 3 shows the geometry that exists between the missile and an arbitrary flare (the K^{th} flare). Again, as in the case of the missile and target geometry, the problem is divided into two planes, horizontal and vertical.

The state vector for the K^{th} flare is listed in Table 2. As in the case of the missile and target state vector system, the first five components represent the position, velocity, and acceleration of the missile; while the next four components represent the position and velocity of the K^{th} flare. It is assumed that the flare has no thrusting device, and therefore no thrust acceleration terms are possible. As before, the state variables are divided into the sets -- one for each plane. The executive routine listing for the simulation program is presented in Table 3. All subroutines in the simulation program are called by this routine.

Table 1. Missile and target state variable definitions

Horizontal Plane	Vertical Plane
X(1) = X - Position of Missile	XP(1) = X - Position of Missile
X(2) = X - Velocity of Missile	XP(2) = X - Velocity of Missile
X(3) = Y - Position of Missile	XP(3) = Z - Position of Missile
X(4) = Y - Velocity of Missile	XP(4) = Z - Velocity of Missile
X(5) = Normal Acceleration (XY) of Missile	XP(5) = Normal Acceleration (XZ) of Missile
X(6) = X - Position of Target	XP(6) = X - Position of Target
X(7) = X - Velocity of Target	XP(7) = X - Velocity of Target
X(8) = Y - Position of Target	XP(8) = Z - Position of Target
X(9) = Y - Velocity of Target	XP(9) = Z - Velocity of Target
X(10) = Normal Acceleration (XY) of Target	XP(10) = Normal Acceleration (XZ) of Target

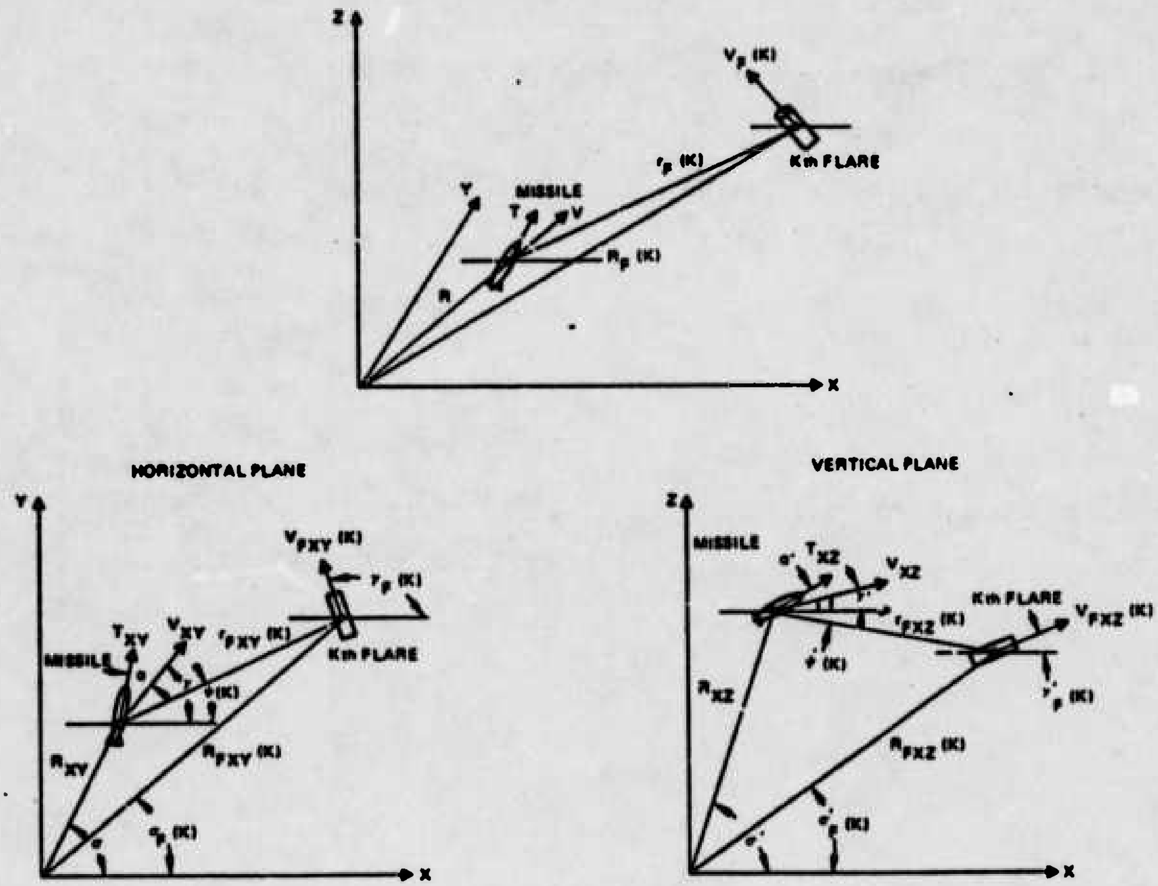


Figure 3. Missile and Kth flare geometry

Table 2. Missile and Kth flare state variable definitions

Horizontal Plane	Vertical Plane
XF(1,K) = X - Position of Missile	XFP(1,K) = X - Position of Missile
XF(2,K) = X - Velocity of Missile	XFP(2,K) = X - Velocity of Missile
XF(3,K) = Y - Position of Missile	XFP(3,K) = Z - Position of Missile
XF(4,K) = Y - Velocity of Missile	XFP(4,K) = Z - Velocity of Missile
XF(5,K) = Normal Acceleration (XY) of Missile	XFP(5,K) = Normal Acceleration (XZ) of Missile
XF(6,K) = X - Position of Kth Flare	XFP(6,K) = X - Position of Kth Flare
XF(7,K) = X - Velocity of Kth Flare	XFP(7,K) = X - Velocity of Kth Flare
XF(8,K) = Y - Position of Kth Flare	XFP(8,K) = Z - Position of Kth Flare
XF(9,K) = Y - Velocity of Kth Flare	XFP(9,K) = Z - Velocity of Kth Flare

Table 3. Simulation program executive routine

```
PROGRAM EPICS     74/74   OPT=1                          FTN 4.2+P383        05/31/75  13.30.19.

      PROGRAM EPICS(INPUT,OUTPUT,TAPE12,TAPE7,TAPE9,TAPE13,TAPE11,TAPE14 EPICS     2
     1              ,TAPE5=INPUT,TAPE6=OUTPUT)                           EPICS     3
                                                                         EPICS     4
      REAL JF,JIN                                                        EPICS     5
      DIMENSION HF(20),MTS(2),VF(20),RTLF(9,20),REL(12),IJDIR(8)         EPICS     6
      DIMENSION SF(20),WF(20),JF(20),VF(20)                              EPICS     7
      DIMENSION RF(20),IDF(20),TTGF(20),XFJ(10,20)                       EPICS     8
      COMMON /BLK1/ST(10),TL(10),SIN(10),HD(10),VMC(8),CDD(8),CNT(8),RMA EPICS     9
     *(10),ALPH(10,8),LL,SA,ALIM,FOV,AS,TMAX,DL,TS,TR,AG,WM,TGU,RNK,GKO, EPICS    10
     *IRNOM,MISTYPE(2),PLURS,NMIN,TDS(8)                                 EPICS    11
      COMMON /BLK2/FMS,AM,VMAX,X3,Y3,Z3,Z3,IAS,TGTALT,VELT,PANG(30),RINT EPICS    12
     *(30)                                                               EPICS    13
      COMMON /BLK3/IFTYPE(2),MK,DF,VL,TOURN,WFD,CDF,TIRAVN(5),REFI,MG,VF EPICS    14
     *LARE,TTN(103),JIN(147)                                             EPICS    15
      COMMON /BLK4/RMS(19),TAUT(19),TAUP(19)                             EPICS    16
      COMMON SSM1,SSM2,SSM3,SSM4,RIKS,IDSPLAY,MSRL,IFP                   EPICS    17
      COMMON PI,RMD,M1,VP,DF.T,VOLUV,TIN,V.                              EPICS    18
      COMMON DELV,AM,RKT,SEP,TUP,TLOA                                    EPICS    19
      COMMON NFLARE,IFDIR(20),MLS,ATIS,AV,FMACH,S..S                     EPICS    20
      COMMON IST(20,3)                                                   EPICS    21
      COMMON HH(20),LP,M                                                 EPICS    22
      COMMON TF(20),FSF(20),FFRA(20),SIMM,SIPM,SIDM,SIDPM,T,SI,SIP       EPICS    23
      COMMON VXF(20),VYF(20),VZF(20),Z(16),ZP(16),X(10),XP(10)           EPICS    24
      COMMON TF(3,20),TPP(S,20),XF(10,20),XFF(10,20),VT                  EPICS    25
      COMMON X20,X40,XP40,XT0,X30,XP30,THRUST                            EPICS    26
      COMMON ALPHE,ALPHAP,TX,TY,TZ,CX,CY,CZ,AX,AY,AZ                     MAR19     1
      NAMELIST/PARAM/SSM1,SSM4,IRNOM,WFD,SA,4.T4,FOV,AS,TMAX,            MAR19     2
     *GKO,RI,TS,TR,AS,M,IOPT,MG,TM,E,XP,L.,CDF,RINT,PANG,                MAR19     3
     *RMA,NMIN,VMC,CDD,CNT,TOURN,TGU,IRNO,XSDIR,NFLARE,G..,              MAR19     4
     *RKT,AM,RIKS,TLOW,TUP,AV,VDISP,MLS,IFC,TIN,TPLOT,REFI,              MAR19     5
     *IDSPLAY,ST,IDPLOT,RF,DDF,TTGF,SJPFLD,TATMED,.MSP,                  MAR19     6
     *LMSA,CLD,TL,SIN,HD,MTC,RUN,ALPHE,SS42,SSM6,FMG,                    MAR19     7
     *TGTALT,VELT,XP,YM,VS,PS                                            EPICS    37
      DATA(RJDIR(J),J=1,8)/90.,270.,0.,180.,45.,135.,225.,315./          MAR20     1
      REWIND 7                                                           MAR2.     2
      REWIND 11                                                          MAR2.     3
      REWIND 12                                                          MAR20     4
      REWIND 14                                                          MAR19     8
      CALL FLYCNST(RUN,IDPLOT,TOPT)                                      EPICS    43
      READ(5,17) NFLARE                                                  MAR20     5
      IF(NFLARE.EQ.0) GO TO 7                                            MAR19     9
   17 FORMAT(I1)                                                         EPICS    43
      READ(5,19) (IFDIR(J),J=1,NFLARE)                                   MAR19    10
   19 FORMAT(20I1)                                                       EPICS    45
    5 READ(5,15) (MTS(J),J=1,2)                                          EPICS    46
   15 FORMAT(2A6)                                                        MAR19    11
      CALL MSLCNST(MTS,MTP,M.S,G.)                                       MAR19    12
      CALL TGTCNST(IRNOM,TM)                                             EPICS    49
      CALL RVSTAU(TGTALT,IRNOM)                                          MAR19    13
   11 READ(5,PARAM)                                                      EPICS    51
      IF(IFMD.NE.0) GO TO 2                                              EPICS    52
      IF(MIP.GT.0) GO TO 3                                               EPICS    55
      RUN=RUN+1.                                                         EPICS    54
      CALL INITIAL(RM,T3,MK,DF,VELT,TGTALT,IAS,IRNOM,WFD,CDF,IFTYPE,MIST EPICS    55
     *YPE,NFLARE,MS,TL,TOURN,VM,VMAX,RJDIR,T.,RUN,AM,M)                  EPICS    56
      CALL LEAD(SL,X,XP,RKT,A4,UL,DEL.V,G,TS,CLD,ALPHE,LMSF,LMS1,SUPELD,T EPICS    57
     *MTHRD,PI)                                                          EPICS    58
      CALL MISPAGE(T(2),RUN,MISTYPE,SS4,..)
```

COPY AVAILABLE TO DDC DOES NOT PERMIT FULLY LEGIBLE PRODUCTION

(Table 3, continued)

```
PROGRAM EPICS       74/74   OPT=1                                  FTN 4.6+P478       05/01/79  13.5..19.

      GO TO 6                                                            EPICS     59
    6 T=T+DELT                                                           EPICS     60
      IF(T.LT.TMAX) GO TO 15                                             EPICS     61
      WRITE(6,14)                                                        EPICS     62
   14 FORMAT(/,2X,29HMISSILE LIFESPAN EXCEEDED,/)                        EPICS     63
      GO TO 17                                                           EPICS     64
   15 M1=M+1                                                             EPICS     65
      CALL GRAVITY (SIG,SIGP,SI,SIP,SIGM,SIGPM,SIGW,SIGN,WLTM,ALPHA,ALPH MAR19     16
     +AP,Z(3),ZP(3),DELT,LP,FOV,SS,TAU,T)                                EPICS     67
      IF(L2.EQ.2) GO TO 18                                               EPICS     68
      CALL THRUST(TL,SIG,W0,Z,ZP,ALPHA,ALPHAP,THRUST,M,T,DELT,TX,TY,TZ,G EPICS     69
     +)                                                                  EPICS     70
      CALL COMMAND(X(3),XP(3),PT,G,W,DL,VMM,RHO,VMS,CNA,ALOM,FMACH,ALPHA EPICS     71
     +,ALPHAP,AS,SCG,CNT,SIGW,SIGM,SIGN,SIGPM,THRUST,Z(3),ZP(3),DELT,TS,M EPICS    72
     +IOS,AX,AY,AZ,TQ,T,DEL)                                             EPICS     73
      CALL CHARG(FMACH,VMP,CDO,RHO,PI,S,DL,Z,ZP,ALPHA,ALPHAP,VMS,M,CX,CY EPICS     74
     +,CZ)                                                               EPICS     75
      X2=TX+AX*CX                                                        EPICS     76
      X4=TY+AY*CY                                                        EPICS     77
      X6=TZ+AZ*CZ-G                                                      EPICS     78
      X(2)=X(2)+X2*DELT                                                  EPICS     79
      X(4)=X(4)+X4*DELT                                                  EPICS     80
      XP(6)=XP(6)+X6*DELT                                                EPICS     81
      X(1)=X(1)+X(2)*DELT                                                EPICS     82
      X(3)=X(3)+X(4)*DELT                                                EPICS     83
      XP(5)=XP(5)+XP(6)*DELT                                             EPICS     84
      IF(XP(5).GT.0.) GO TO 20                                           EPICS     85
      WRITE(6,21) T                                                      EPICS     86
   21 FORMAT(2X,27HMISSILE HIT GROUND AT TIME ,F8.4)                     EPICS     87
      GO TO 17                                                           EPICS     88
   20 IF(MPLANE.EQ.1) GO TO 9                                            EPICS     89
      IF(IPOLDT.EQ.1) GO TO 12                                           EPICS     90
      IF(TP.GE.T) GO TO 9                                                EPICS     91
   12 CALL FLOTPL  (MM,M,TST,T,TF,FTS,TTSF,SSMP,XF,XFP,X,XP,TAJPN,VYF,V MAR19     15
     +YF,VZF,YFP,XFD,VFDO,DEL,RF,RPF,IPOLDT,M,PI,T)                      EPICS     93
      IF(M.EQ.3) GO TO 9                                                 EPICS     94
      CALL FLOTM  (T,DELT,TF,TAJPN,SF,SFP,TF,WFD,WG,WF,M,RMP,CDF,G,VF,X MAR19     16
     +,YP,XFD,VF,XFP,NM,MK)                                              EPICS     95
      CALL MPLRTPL(M,XF,YFP,XFD,VF,AF,DELT)                              MAR19     17
      CALL MPLANG(XF,M,ZF,PI)                                            MAR19     18
      CALL MPLANG(VFP,M,YFP,PI)                                          MAR19     19
      CALL FLPIR   (T,TF,DELF,M,VF,WF,AF,IOPT,TTN,JIN,FISSND,ISNOM,FSF,P MAR19     20
     +CPI,TAUPN,VTRDM,RMS,TAJP)                                          EPICS    101
      M1=1PP.*PI                                                         EPICS    102
      IF(M1.LT.M2) GO TO 6                                               EPICS    103
    9 CONTINUE                                                           EPICS    104
      DO 7 K=1,M                                                         EPICS    105
      GO TO (13,13,7), MM(K)                                             EPICS    106
   13 IF(XFM(9,K).GT.0.) GO TO 16                                        EPICS    107
      JF(K)=0.                                                           EPICS    108
      MF(K)=0.                                                           EPICS    109
      GO TO 7                                                            EPICS    110
   16 IF(SSMP.EQ.0.) GO TO 7                                             EPICS    111
      ZF1=ZF(1,K)*M1                                                     EPICS    112
      ZF2=ZFP(1,K)*M1                                                    EPICS    113
      ZF3=ZF(2,K)*M1                                                     EPICS    114
      ZF4=ZFP(2,K)*M1                                                    EPICS    115
```

(Table 3, concluded)

```
PROGRAM EPICS     74/74   OPT=1                    FTN 4.2+P380       05/31/79  13.33.19.

        WRITE(6,4920) N                                                  EPICS    116
    492 FORMAT(5X,7HNUMBER ,I4,1X,21HP-ARC CHARACTERISTICS)              EPICS    117
        WRITE(6,6790) XP(5,K),XP(6,K),XP(9,K)                            EPICS    118
        WRITE(6,6790) XP(7,K),XP(8,K),XP(9,K)                            EPICS    119
        WRITE(6,5760) RELP(1,K),RELP(2,K),RELP(3,K)                      EPICS    120
        WRITE(6,6790) RELP(7,K),RELP(9,K),VP(K)                          EPICS    121
    579 FORMAT(2X,5HXREL=,F8.2,2X,5HXDOT=,F8.2,1X,3HVP=,F9.2)            EPICS    122
    576 FORMAT(5X,5HXREL=,F10.2,2X,5HYREL=,F9.2,2X,5HZREL=,F8.2)         EPICS    123
    575 FORMAT(2X,6HVXREL=,F9.6,2X,6HVYREL=,F9.6,2X,6HVZREL=,F9.6)       EPICS    124
    576 FORMAT(5X,3HXP=,F12.3,2X,3HYP=,F12.6,2X,3HZP=,F12.6)             EPICS    125
        WRITE(6,5930) RP(K),HP(K)                                        EPICS    126
    593 FORMAT(5X,3HRP=,F12.5,2X,7HHEIGHT=,F12.5)                        EPICS    127
        WRITE(6,6920) RP(K),HP(K)                                        EPICS    128
    692 FORMAT(2X,13HINTENSITY=,F12.5,2X,11HIRRADIANCE=,F12.6)           EPICS    129
        WRITE(6,9060) ZF1,ZF2,ZF3,ZF4                                    EPICS    130
    906 FORMAT(2X,7HSIGNAL=,F12.5,2X,7HSIGNAL=,F12.5,2X,10HSIGNALTOT=,F12. EPICS    131
       +6,2X,10HSIGNALTOT=,F12.5)                                        EPICS    132
      7 CONTINUE                                                         EPICS    133
      5 CALL TGTPM   (T,TM,DELT,N,G,FM,VM,VT,X(5),X(6),X(8),X(7),X(9),X  MAR20      6
       +P(3),X7),X9),X(9),Z(P))                                          EPICS    135
      6 VMM=SQRT(X(2)*X(2)+X(4)*X(4)+XP(6)*XP(6))                        EPICS    136
        CALL MTRVEL  (X,XP,X2,X6,X8),X7,X9,X5),T,VMM,VMD,VT,VTD,REL,     MAR20      7
       +L2,TTG,RMISS,MISS)                                               EPICS    138
        IF(L2.EQ.2) GO TO 10                                             EPICS    139
        CALL MTSTANS  (X,ALPHA,X6,X7,X9,X7),Z,PI)                        MAR20      8
        CALL MTSTANS  (XP,ALPHAP,X6,X3,X9,X7),ZP,PI)                     MAR20      9
        CALL TGTIR   (X(7),X(8),X(9),VT,REL(1),REL(2),REL(3),REL(10),PI,R MAR20     10
       +INF,PANG,RAD,TOUT,AI,TINT)                                       EPICS    143
        CALL MSLAMPT (ZP,ZPP,Z,TP,RELP,REL,MT,ML,MP,M,MP,KT,LT,SI,SIP,SID, MAR20     11
       +SIDP,PR,RADOT,FOV,L2,MAIN,LL,T)                                  EPICS    145
        IF(L2.EQ.2) GO TO 10                                             EPICS    146
        IF(M1.LT.M2) GO TO 9                                             EPICS    147
        M1=0                                                             MAR20     12
        IF(SSNS.NE.0.) GO TO 22                                          EPICS    149
        GO TO 23                                                         EPICS    150
     22 MM1=M-1                                                          MAR19     21
        WRITE(9) DJM,MM1,XP(3),XP(8),(XP(8,MM),MM=1,MM1),X(3),X(8),      MAR19     22
       +(XP(8,MM),MM=1,MM1),X(1),X(5),(XP(6,MM),MM=1,MM1),SI,Z(3),(ZP(1,MM MAR19     23
       +),MM=1,MM1),STP,ZP(3),(ZP(1,MM),MM=1,MM1),(IGT(MM,M1),MM=1,MM1), MAR19     24
       +M1=1,2),(JP(MM),MM=1,MM1),(RELP(7,MM),MM=1,MM1),(RELP(8,MM),MM=1,M MAR19     25
       +M1),MT,REL(10),REL(11),T                                         MAR19     26
     23 IF(T.GT.0.) GO TO 32                                             EPICS    169
        IF(SSNS.GT.0.) GO TO 32                                          EPICS    170
        CALL PAGE2(IAG,VELT,TGTALT,ITYPE,Z,ZP,SI,STP,SID,SIDP,REL,X,XP,XR EPICS    171
       +D,X6D,XP6D,X7D,X33,XP9D)                                         EPICS    172
     32 CALL OUTPUT(X,XP,Z,ZP,ALPHA,ALPHAP,R6,RADOT,SIGM,SIPM,SID,SIDPM,X EPICS    173
       +ZD,X6D,XP6D,VT,VMM,FMACH,REL,RMISS,TS,THRUST,N,MT,T,SSN2,IX,TY,TZ EPICS    174
       +,BX,BY,BZ,CX,CY,CZ)                                              EPICS    175
        GO TO (9,1), L2                                                  EPICS    176
      1 IF(NTP.GT.0) GO TO 3                                             EPICS    177
        GO TO 11                                                         EPICS    178
     10 TMISS=RMISS                                                      EPICS    179
        CALL CLOSAPP  (REL,TTS,DELT,RMISS,CP,T,TMAX,XM,YM,ZM,TTGT)       MAR20     13
        CALL PHCOMP(REL,ZIP,PHC,YM,ZM,XM,23,PH)                          EPICS    181
        CALL FINPAGE(XM,YM,ZM,REL,RMISS,TMISS,TTGT,TTG,PH)               EPICS    182
        IF(SSNS.EQ.0.) GO TO 33                                          EPICS    183
        WRITE(13) RMISS,PH,TMISS,REL(13)                                 EPICS    184
        ENDFILE 13                                                       EPICS    185
     30 IF(SSNS.NE.0.) GO TO 29                                          EPICS    186
        IF(IOSPLOT.EQ.3) GO TO 11                                        EPICS    187
        ENDFILE 9                                                        EPICS    188
        WRITE(8,28) RMISS,T,PH                                           EPICS    189
     28 FORMAT(3E12.6)                                                   EPICS    190
     29 ENDFILE 8                                                        EPICS    191
        GO TO 11                                                         EPICS    192
      2 IF(SSNS.EQ.0.) GO TO 31                                          EPICS    193
        ENDFILE 13                                                       EPICS    194
        REWIND 13                                                        EPICS    195
     31 STOP                                                             EPICS    196
        END                                                              EPICS    197
```

1. FILE CREATION

Information used to create missile, target and flare file data has been collected at Hughes over a long period. Pertinent sources are listed in the references at the end of this section.

MISSILE FILE

The missile information is divided into seven categories: seeker, signal processing, guidance unit, aerodynamics, motor, physical characteristics, and other characteristics. Table 1-1 is the missile file subroutine. See Table 1-2 for sample file data.

Seeker

The seeker is defined as:

1. Seeker look angle (deg), variable name SA. This is the angle about the longitudinal axis of the missile in any plane that the gyro is free to move.
2. Max gyro rate (deg/sec.), variable name WLIM.
3. FOV (deg), variable name FOV.
4. Gyro time lag (sec.), variable name TGU.

Signal Processing

The signal processing includes:

1. The detector bandwidth of the missile, variable name IBNDM. The bandwidth value is currently deleted from the missile page printout.
2. Aimpoint type is selected on the basis of geometric centroid, irradiance centroid, and maximum irradiance.

The present available missiles are divided into two categories: the con-scan missile which is a maximum irradiance tracker, and the spin-scan missile which is an irradiance centroid tracker. Geometric centroid does not apply to the current systems

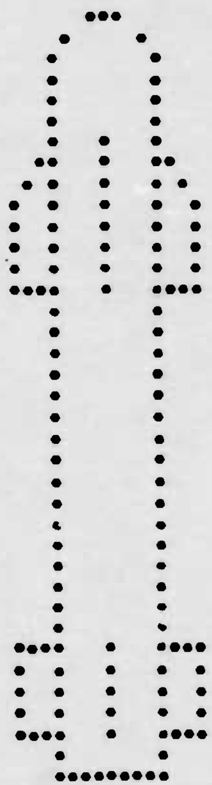
Table 1-1. Missile file subroutine

```
SUBROUTINE MSLCNST    74/74    OPT=1                          FTN 4.2+P380        05/01/75  13.30.35.

          SUBROUTINE MSLCNST(NTS,NTC,N.P,G.)                                          MAR19       29
          DIMENSION NTS(2)                                                            MSLCONST     3
          COMMON /BLK1/ST(10),TL(10),SIM(10),WD(10),VMC(8),CDO(8),CNS(8),CNA          MSLCONST     4
         +(10),ALPH(10,8),LL,SA,WLIM,FOV,AS,TMAX,D1,TS,T9,AG,WN,TGU,RMK,CKO,          MSLCONST     5
         +IRNOM,MISTYPE(2),BLURC,HMIN,ZZZ(9)                                          MSLCONST     6
          NLS=3                                                                       MSLCONST     7
          GL=0.                                                                       MSLCONST     8
          NTS=0                                                                       MSLCONST     9
          NPTS=10                                                                     MSLCONST    10
          RAD=3.141592/180.                                                           MSLCONST    11
        6 READ(7) (ST(LA),LA=1,NPTS)                                                  MSLCONST    12
          READ(7) (TL(LA),LA=1,NPTS)                                                  MSLCONST    13
          READ(7) (SIM(LA),LA=1,NPTS)                                                 MSLCONST    14
          READ(7) (WD(LA),LA=1,NPTS)                                                  MSLCONST    15
          READ(7) (VMC(LA),LA=1,8)                                                    MSLCONST    16
          READ(7) (CDO(LA),LA=1,8)                                                    MSLCONST    17
          READ(7) (CNS(LA),LA=1,8)                                                    MSLCONST    18
          READ(7) (CNT(LA),LA=1,8)                                                    MSLCONST    19
          READ(7) (CNA(LA),LA=1,10)                                                   MSLCONST    20
          READ(7) ((ALPH(M1,M2),M1=1,10),M2=1,8)                                      MSLCONST    21
          READ(7) LL,SA,WLIM,FOV,AS,TMAX,D1,TS,T9,AG,WN                               MSLCONST    22
          TS=.1                                                                       MSLCONST    23
          SA=SA*RAD                                                                   MSLCONST    24
          WLIM=WLIM*RAD                                                               MSLCONST    25
          FOV=FOV*RAD                                                                 MSLCONST    26
          AS=AS*32.2                                                                  MSLCONST    27
          READ(7) TGU,RMK,CKO,IRNOM                                                   MSLCONST    28
          READ(7) (MISTYPE(LA),LA=1,2),BLURC,HMIN                                     MSLCONST    29
          IF(MISTYPE(1).NE.NTS(1)) GO TO 6                                            MSLCONST    30
          IF(MISTYPE(2).NE.NTS(2)) GO TO 6                                            MSLCONST    31
          REWIND 7                                                                    MSLCONST    32
          RETURN                                                                      MSLCONST    33
          END                                                                         MSLCONST    34
```

Table 1-2. Missile constants file

RUN 1

MISSILE CONSTANTS



MISSILE 1

SEEKER

SEEKER LOOK ANGLE	(DEG)	84.
MAX GYRO RATE	(DEG/SEC)	15.0
FIELD OF VIEW	(DEG)	1.9
GYRO TIME LAG	(SEC)	.06

SIGNAL PROCESSING

DETECTOR BANDWIDTH (MICRONS)
AIMPOINT TYPE 1 IRRADIANCE CENTROID

GUIDANCE UNIT

NAVIGATION CONSTANT		4.0
NO GUIDANCE PERIOD	(SEC)	.2
G - LIMIT	(G'S)	9.
MISSILE TIME CONST	(SEC)	.10

AERODYNAMICS

MACH NUMBER	.00	.22	.44	.70	.90	1.12	1.45	1.75
C(C)	.45	.47	.49	.52	.79	1.22	1.21	1.16
C(DB)	.07	.09	.11	.14	.22	.40	.25	.16
C(NMAX)	.00	2.85	2.90	3.30	3.52	3.98	3.44	3.46

C(N)

MACH NO	.0	.2	.3	.6	1.0	1.4	1.8	2.2	2.8	3.5
.00	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
.22	.0	1.0	2.0	3.9	6.0	8.3	10.7	13.0	15.9	16.0
.44	.0	.9	1.8	3.6	5.5	7.7	9.9	12.8	15.7	16.0
.70	.0	.9	1.7	3.4	5.0	6.8	8.8	11.1	13.8	16.0
.90	.0	.8	1.7	3.3	4.8	6.3	8.2	10.4	13.0	15.7
1.12	.0	.7	1.4	2.9	4.5	6.0	7.6	9.8	11.9	14.3
1.45	.0	1.0	1.9	3.8	5.6	7.6	9.6	11.4	13.9	16.0
1.75	.0	1.0	2.0	4.0	6.0	8.0	10.0	12.0	14.0	16.0

MOTOR

		1	2	3	4
THRUST LEVEL	(LB)	0	250	0	0
SWITCHOVER TIME	(SEC)	.2	5.6	5.6	5.6
SPECIFIC IMPULSE	(1/SEC)	195.	195.	195.	195.
MOTOR WEIGHT DROP	(LB)	.0	.0	.0	.0

PHYSICAL CHARACTERISTICS			OTHER CHARACTERISTICS		
DIAMETER	(FT)	2.7/12	LIFESPAN	(SEC)	7.
WEIGHT	(LB)	20.3	LAUNCH VELOCITY	(FT/SEC)	95.
MAX FLIPPER DEFLECTION	(DEG)	17.0	MAX FLIPPER TRAVEL	(DEG)	0.

Guidance Unit

The guidance unit is composed of:

1. The navigation constant, variable name GKO
2. No guidance period (sec), variable name TB
3. G-limit (g's), variable name AS
4. Missile time constant (sec), variable name TS.

Aerodynamics

A sampling of eight values has been adopted for the parameters below to cover the broad range values in order of convenience in handling data information for table lookups. (Exceptions will be noted.) The following variables are functions of Mach number table, variable name VMC:

1. Chord force, variable name CDO.
2. Base drag coefficient, variable name CDB.
3. Maximum normal force coefficient, variable name CNT.
4. Angle of attack (deg) (80 samples), variable name ALPH. This parameter is a function of both mach number and normal force, variable name CNA.

Motor

The motor parameters are depicted as tables of 10 samples related to time. These are:

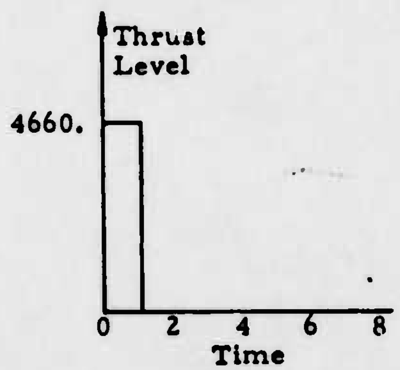
1. Time (sec.), variable name ST
2. Thrust (lb.), variable name TL
3. Specific impulse (1/sec.), variable name SIM
4. Motor weight drop (lb.), variable name WID.

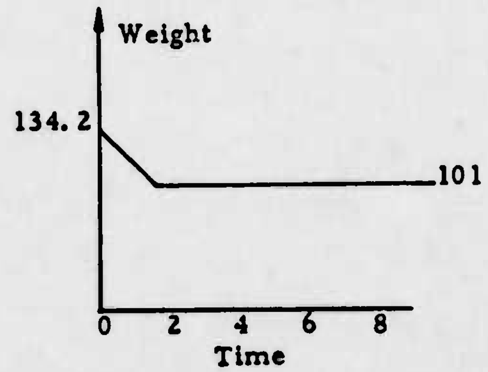
See Table 1-3.

Table 1-3. Motor characteristics

Example:

Time	0.	1.	1.395	1.405	2.	4.	5.	6.	7.	8.
Thrust	4660.	4660.	4660.	0.	0.	0.	0.	0.	0.	0.
Specific Impulse	210.	210.	210.	210.	210.	210.	210.	210.	210.	210.
Motor Weight Drop	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.





Equations

1. Thrust = TL
2. Weight (new) = Weight (old) - Thrust (DELT/SIM) - WD(t)

DELT = Integration Step Size

Physical Characteristics

Physical characteristics of a missile are:

1. Diameter (in.), variable name DI
2. Weight (lb.), variable name W
3. Max. flipper deflection (deg.), variable name AG.

Other Characteristics

Other characteristics are:

1. Lifespan (sec.), variable name TMAX
2. Launch velocity (ft./sec.), variable name X(Z)
3. Max. flipper travel (deg.), variable name FMAX.

Information not shown but required

Information not shown but required is:

1. Missile kill radius used in part to determine probability of hit of missile, variable name RMK
2. Missile name, variable name MISTYPE
3. Blur circle, variable name BLURC
4. Minimum detectable irradiance, variable name HMIN.

FLARE FILE

The flare information has two categories: physical characteristics, and other characteristics. The flare file subroutine is presented as Table 1-4.

Table 1-4. Flare file subroutine

```
                                                          FTN 4.2+P383)     05/01/79  13.30.32.
SUBROUTINE FLRCNST    74/74   OPT=1
                                                                            MAR19        27
        SUBROUTINE FLRCNST(RUN,IOPLOT,IOPT)                                 FLRCONST      3
        REAL JIN                                                            FLRCONST      4
        COMMON /BLK3/IFTYPE(2),MK,DF,X.,TBURN,NFO,CDF,FIBAND(9),REFI,WG,VF  FLRCONST      5
       *LARE,TTN(100),JIN(100)                                              FLRCONST      6
        COMMON DUM(5),IDUM(2),IFC                                           FLRCONST      7
        DO 1 J=1,5                                                          FLRCONST      8
        IF(J.GT.2) GO TO 2                                                  FLRCONST      9
        IDUM(J)=0                                                           FLRCONST     10
      2 DUM(J)=0.                                                           FLRCONST     11
      1 CONTINUE                                                            FLRCONST     12
        RUN=0.                                                              FLRCONST     13
        IFC=1                                                               FLRCONST     14
        IOPLOT=1                                                            FLRCONST     15
        IOPT=2                                                              FLRCONST     16
     12 READ(12) (IFTYPE(LA),LA=1,2)                                        FLRCONST     17
        READ(12) MK,DF,XL,TBURN,NFD,CDF,(FIBAND(JJ),JJ=1,9),REFI,WG,VFLARE  FLRCONST     18
        READ(12) NPTS,(TTN(LA),JIN(LA),LA=1,NPTS)                           FLRCONST     19
        REWIND 12                                                           FLRCONST     20
        RETURN                                                              FLRCONST     21
        END
```

Note that the program can handle pyrophorics if they are modelled as special flares, e.g. short rise time, short burn time, high peak intensity and high drag coefficient. However, the program may require some development to more fully take account of the burning and aerodynamic characteristics of this type of countermeasure.

Physical Characteristics

Physical characteristics are:

1. Diameter (in.), variable name DF
2. Length (in.), variable name XL
3. Weight (lb.), variable name WFO
4. Grain Weight (lb.), variable name WG.

Other Characteristics

Other characteristics are:

1. Burn time (sec.), variable name TBURN
2. Reference intensity (watts/sterad), variable name REF
3. Spectral band constants, variable name FIBAND
4. Drag coefficient, variable name CDF
5. Dispenser ejection velocity (ft/sec), variable name VFLARE
6. Flare type (name), variable name IFTYPE
7. Type of surface area, variable name MK.

TARGET FILE

Target information has three categories: physical characteristics, initial condition, and other characteristics. Table 1-5 gives the target file subroutine.

Table 1-5. Target file subroutine

```
SUBROUTINE TGTCNST    74/74    OPT=1                    FTN 4.2+P443        05/01/75  13.33.37.

            SUBROUTINE TGTCNST(IANDM,TM)                                               MAR19       29
            COMMON /BLK2/FMS,AM,VMAX,XR,XM,YS,ZS,IAC,TGTALT,VELT,PANG(70),RINT         TGTCONST     3
           +(37)                                                                       TGTCONST     4
            TM=0.                                                                      TGTCONST     5
            READ(11) FMS,AM,VMAX,XR,XM,YS,ZS                                           TGTCONST     6
            DO 10 IANDT=1,5                                                            TGTCONST     7
            READ(11) IAC,TGTALT,VELT                                                   TGTCONST     8
         20 READ(11) NPTS,(PANG(J),J=1,NPTS)                                           TGTCONST     9
            READ(11) (RINT(J),J=1,NPTS)                                                TGTCONST    10
         21 DO 11 J=1,NPTS                                                             TGTCONST    11
            PANG(2*NPTS-J+1)=360.-PANG(J)                                              TGTCONST    12
            RINT(2*NPTS-J+1)=RINT(J)                                                   TGTCONST    13
         11 CONTINUE                                                                   TGTCONST    14
            IF(IANDM.EQ.IANDT) GO TO 12                                                TGTCONST    15
         10 CONTINUE                                                                   TGTCONST    16
            STOP                                                                       TGTCONST    17
         12 REWIND 11                                                                  TGTCONST    18
            RETURN                                                                     TGTCONST    19
            END                                                                        TGTCONST    20
```

Physical Characteristics

Physical characteristics are:

1. Longitudinal distance from tailpipe to tip of tail (ft), variable name XB
2. Longitudinal distance from tailpipe to nose (ft.), variable name XN
3. Wingspan (ft.), variable name ZS.

Initial Condition

Initial condition is:

1. Maximum aircraft turn (g's), variable name FMG
2. Maximum aircraft forward acceleration (g's), variable name AM
3. Maximum aircraft speed (mach), variable name VM
4. Target altitude (ft.), variable name XP(8)
5. Target velocity (ft/sec.), variable name X(7).

Other Characteristics

Intensity as a function of polar angle are tables

1. Polar angle (deg.), variable name PANG
2. Intensity (kw/sterad), variable name RINT
3. Target type (name), variable name IAC.

ATMOSPHERE FILE

The last file to be discussed is the atmospheric spectral transmittance tables. These tables are a function of range, altitude, temperature, and band region for target and flare.

1. Range (ft.), variable name RNG
2. Atmospheric spectral transmittance of target, variable name TAUT
3. Atmospheric spectral transmittance of flare, variable name TAUF.

Table 1-6 gives the atmospher file subroutine.

Table 1-6. Atmosphere file subroutine

```
SUBROUTINE RVSTAU      74/74   OPT=1                    FTN 4.2+P390        03/01/79  13.38.39.

          SUBROUTINE RVSTAU(TGTALT,IMNOM)
          COMMON /BLK6/RNG(19),TAUT(19),TAUF(19)
          IF(TGTALT.LT.1000.) GO TO 13
          IF(TGTALT.LT.21000.) GO TO 16
          IALTA=3
          GO TO 15
       13 IALTA=1
          GO TO 15
       16 IALTA=2
       15 DO 18 LA=1,3
          DO 17 LM=1,5
          READ(16) IALTQ,IMNOD
          READ(16) (RNG(LC),TAUT(LC),TAUF(LC),LC=1,19)
          IF(IALTQ.EQ.IALTA.AND.IMNOM.EQ.IMNOD) GO TO 19
       17 CONTINUE
       18 CONTINUE
          WRITE(6,10)
       10 FORMAT(1X,26HRANG-ALT. DO NOT COMPARE)
          STOP
       19 REWIND 16
          RETURN
          END
```

DATA SOURCES

(1) J.A. Ratkovic, K. Nishimoto "IRCM Simulation Study (U)" 4 Quarterly Progress Report prepared by Hughes Aircraft Company, Culver City, Calif. for Air Force Avionics Laboratory, Wright-Patterson Air Force Base, Ohio, July 1973

(2) IRCM Simulation Study (U), Feb. 1972, Volume II Hughes Aircraft Co., Culver City, CA, SDN G-5812

(3) Fighter-Launched Missiles (Trends), Eurasian Communist Countries (U), Dec. 1971, Defense Intelligence Agency, Report No. T65-09-26B, SDN J-58432

(4) DAWN (Develop Attack Warning Needs) (U), Final Report, Nov. 1972, General Research Corps., Santa Barbara, CA, SDN G-61305

(5) Foreign Material Exploitation Report, Interim Report, Project Graduation Level (U), July 1972, Missile Intelligence Agency, U.S. Army Missile Command, Redstone Arsenal, Alabama, SDN G-61357

(6) Shoulder Fired Surface to Air Missile System Comparison Summary (U), June 1972, Micom Foreign Intelligence Office, U.S. Army Missile Command, Redstone Arsenal, Alabama, SDN G-61358

(7) AIM-9D Simulation Parameters, Gene Younkin, Technical Note 4055-2-68, U.S. Naval Weapons Center, China Lake, Calif., Dec. 1967

(8) Hughes AIM-4D Aerodynamic Data, SRS-585, Revised 1 January 1965

(9) Assessment of Aerodynamic Studies of Foreign Tactical Missiles, Leroy Spearman and Charlie Jackson, Jr., NASA Langley Research Center, Hampton, Va., Feb. 1971

(10) ANAB Missile Wing Evaluation, FTD-CW-09-4-70, Feb. 1970.

Report on ASH Air-to-Air Missile Exploitation, Ministry of Defense, TM 88169, Dec. 1969.

(11) NAVWEPS Report TN 4063-233, AIM-9L Wind Tunnel Test Report (U), Oct. 1972

(12) PMS 12AD44-1/2174, Final Stability and Control Report for the AIM-54A Missile (U), 27 April 1973

(13) Radiometric Data and Mission Profile, Dept. of the Air Force, Headquarters Aeronautical Systems Division, Wright-Patterson AFB, Dec. 1972, SDN G-61377

(14) B-52 Infrared Radiation Patterns, Beoing Co., Wichita Division, July 1968, SDN J-59090

2. PROGRAM INITIALIZATION

This portion of the simulation program initializes all program constants, launch geometry variables, and determines the missile level angle computations. See Figure 2-1. Section 2.1 describes the initialization procedures, and Section 2.2 the lead angle computations.

2.1 PROGRAM INITIALIZATION

In the missile, target, flare simulation program, basic parameters have to be defined, and the program involved in doing this is initial. The three categories to be discussed will be definitions of constants, initialization, and launch geometry. A block diagram of these computations is shown in Figure 2-2. See Table 2-1 for program listing.

<u>Definitions of Constants</u>

The following constants are defined in the simulation program:

1. Gravity, (ft/sec.) variable name G, 32.2

2. Pi, variable name PI, 3.141592

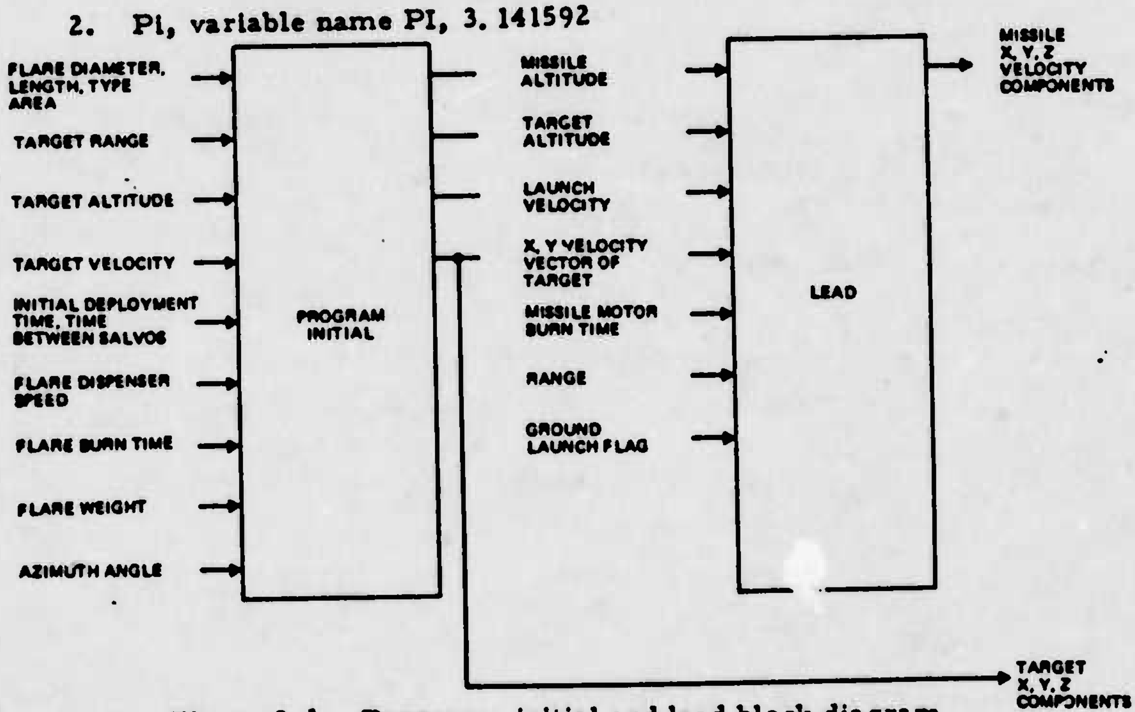
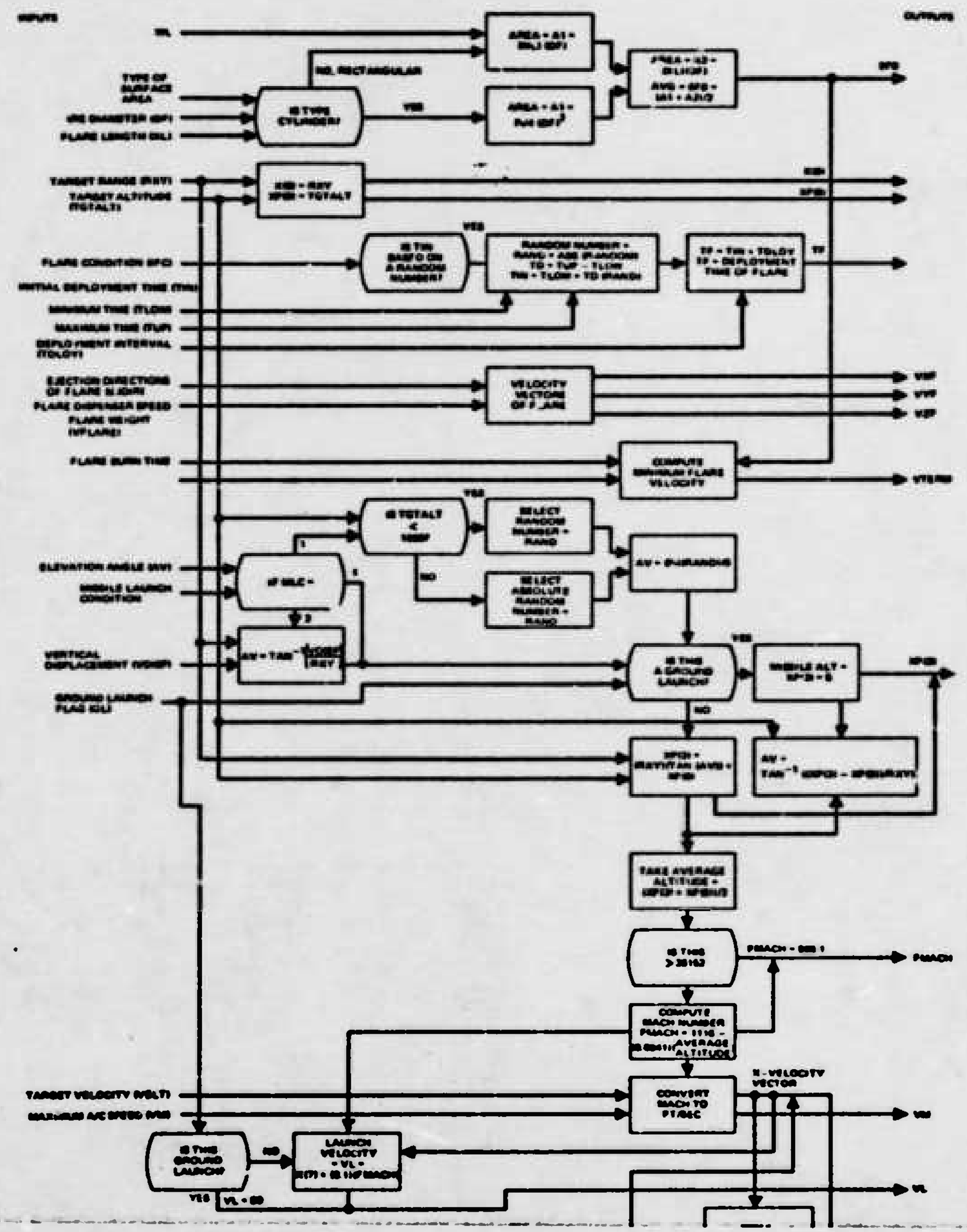


Figure 2-1. Programs initial and lead block diagram



INPUTS
OUTPUTS
TYPE OF SURFACE AREA
FLARE LENGTH
IS TYPE CYLINDER?
NO, RECTANGULAR
YES
TARGET RANGE
TARGET ALTITUDE (TOTALT)
FLARE CONDITION
INITIAL DEPLOYMENT TIME
MINIMUM TIME (TLOW)
MAXIMUM TIME (TUP)
DEPLOYMENT INTERVAL (TDLOY)
EJECTION DIRECTIONS OF FLARE
FLARE DISPENSER SPEED
FLARE WEIGHT (WFLARE)
FLARE BURN TIME
VELOCITY VECTORS OF FLARE
COMPUTE MINIMUM FLARE VELOCITY
VXF
VYF
VZF
VTERM
ELEVATION ANGLE
MISSILE LAUNCH CONDITION
VERTICAL DISPLACEMENT
GROUND LAUNCH FLAG
SELECT RANDOM NUMBER
SELECT ABSOLUTE RANDOM NUMBER
IS THIS A GROUND LAUNCH?
TAKE AVERAGE ALTITUDE
COMPUTE MACH NUMBER
PMACH
TARGET VELOCITY
MAXIMUM A/C SPEED
CONVERT MACH TO FT/SEC
X-VELOCITY VECTOR
IS THIS GROUND LAUNCH?
LAUNCH VELOCITY
VL

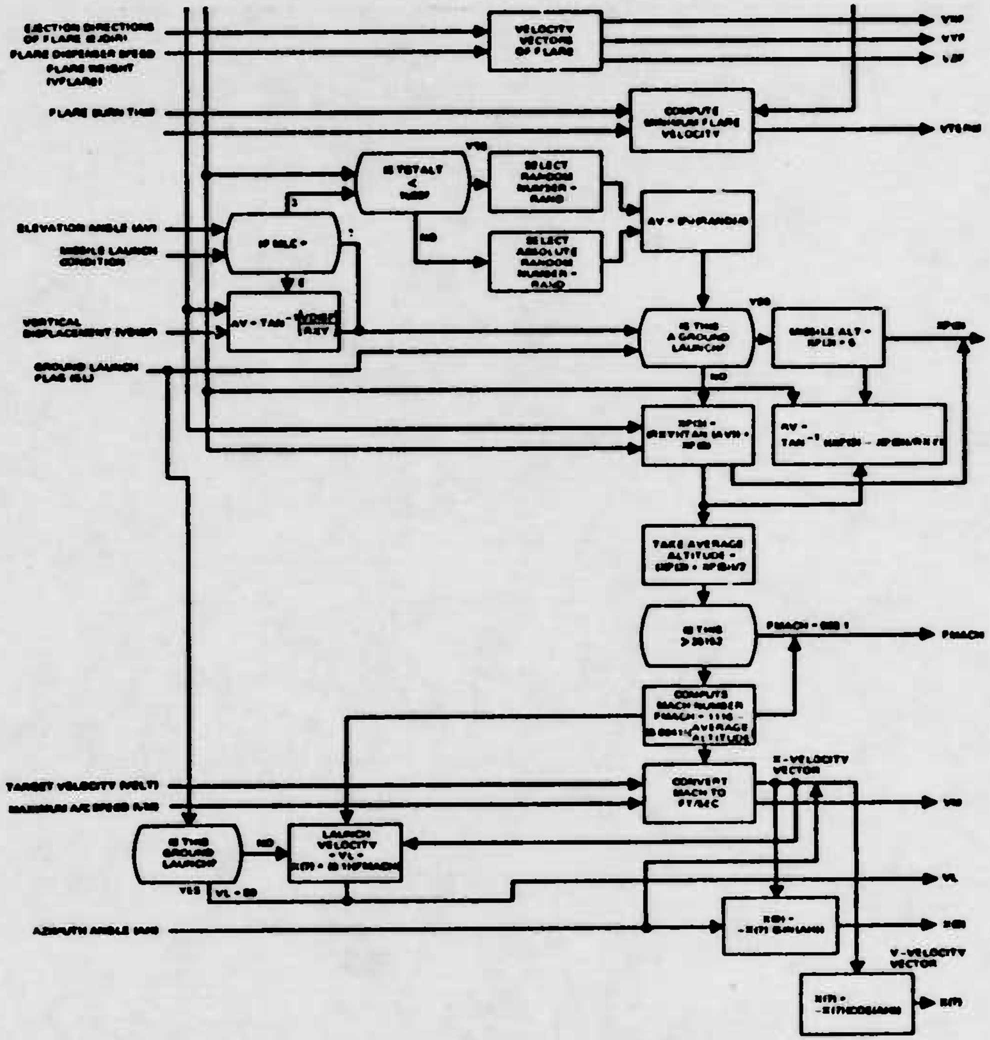


Figure 2-2. Program initial block diagram.

Table 2-1. Initial subroutine

```
SUBROUTINE INITIAL     74/74   OPT=1                        FTN 4.2+P398      05/01/75  13.38.41.

      SUBROUTINE INITIAL(D1,TS,MK,DF,VELT,TGTALT,IAC,ISWON,NFO,COF,IFTYP
     +E,MISTYP,VFLARE,W3,XL,TRJON,VV,ANAC,EJDIR,TL,RJM,MM,NI)
      COMMON SSW1,SSW2,SSW3,SSW4,DIAS,IDSPLAY,MSG1,IFC
      COMMON PI,RHO,W1,W2,DELT,TOLDV,TIN,WL
      COMMON DFLV,AM,RXV,SFB,YUP,TLOW
      COMMON NFLARE,IFDIR(20),WLD,ANISP,AV,FMACH,SL,G
      COMMON IGY(20,3)
      COMMON MM(23),LP,N
      COMMON TF(23),FSF(23),VTERM(23),SIWW,SIPW,SIDW,SIDPW,T,SI,SIP
      COMMON VXF(23),VYF(23),VZF(23),Z(10),ZP(10),X(10),XP(10)
      COMMON ZF(3,23),ZFP(3,23),XF(10,23),XFP(10,23),VT
      COMMON X20,X40,XP40,X70,X90,XP90,THRUST
      COMMON ALPHA,ALPHAP,TX,TY,TZ,DX,DY,DZ,AX,AV,AZ
      DIMENSION MISTYPE(2),IFTYPE(2),EJDIR(9)
      DIMENSION IX(20),IFX(20),TL(10)
      WRITE(6,2)
    2 FORMAT(1H1)
      G=32.2
      PI=3.141592
      N=0
      DF=.0064716
      WL=2.
      THRUST=TL(1)
      DO 1 J=1,716
    1 TF(J)=0.
      DO 3 J=1,60
      IF(J.GT.23) GO TO 3
      MM(J)=1
    3 IGY(J,1)=0
      RHO2=.002469
      DELT=.01
      CTR=.2/DELT
      W2=CTR
      W1=W2
      GO TO (6,7,8,8), MK
    6 A1=16./144.*DF*XL
      A2=A1
      A3=DF*DF/144.
      DFL=(A1/4.+A2)/2.
      GO TO 40
    7 A1=PI*(DF/12.)*(DF/12.)*.25
      GO TO 9
    8 A1=XL*DF*(1./12.)*(1./12.)
    9 A2=XL*DF*(1./12.)*(1./12.)
   40 SFB=(A1+A2)/2.
      X(5)=RXV
      X(7)=VELT
      X7=VELT
      XP(6)=TGTALT
      DFLV=1999.
      AM=AM*PI/180.
      GO TO (10,11), IFC
   11 RAND=ABS(RANF(1))
      TD=TUP-TLOW
      TIN=TLOW+TD*RAND
   10 IF(SSW4.EQ.1.) GO TO 12
      IF(X(7).GT..0) GO TO 13
```

Table 2-1 (Continued)

```
SUBROUTINE INITIAL     74/74   OPT=1                        FTN 4.2+P534,     09/31/75  13.33.41.

                                                                   INITIALT   59
      IPS=NHIL                                                     INITIALT   60
      GO TO 14                                                     INITIALT   61
   13 IPS=NHB/8                                                    INITIALT   62
   14 WRITE(13) RUN,NFLARE,(IFDIR(KJ),KJ=1,NFLARE),TAC,XP(4),AM,RXY,IPS,  INITIALT   63
     +X(7)                                                         INITIALT   64
   12 GO TO (15,15,17), NLC                                        INITIALT   65
   15 AV=AV*PI/180.                                                INITIALT   66
      GO TO 18                                                     INITIALT   67
   17 IF(XP(3).LT.1000.) GO TO 19                                  MAR19      32
      RAND=RANF(1)                                                 INITIALT   69
      GO TO 20                                                     MAR19      33
   19 RAND=ABS(RANF(1))                                            INITIALT   71
   20 AV=PI*RAND/4.                                                INITIALT   72
      GO TO 18                                                     INITIALT   73
   16 AV=ATAN2(VDIST,RXY)                                          INITIALT   74
   18 IF(GL.GT.0.) GO TO 21                                        INITIALT   75
      XP(3)=RXY*TAN(AV)+XP(4)                                      INITIALT   76
      GO TO 22                                                     INITIALT   77
   21 XP(3)=0.                                                     INITIALT   78
      AV=ATAN2((XP(3)-XP(4)),RXY)                                  INITIALT   79
   22 AVALT=(XP(3)+XP(4))/2.                                       INITIALT   80
      IF(AVALT.GT.36152.) GO TO 23                                 INITIALT   81
      FMACH=1115.-.00361*AVALT                                     INITIALT   82
      GO TO 24                                                     INITIALT   83
   23 FMACH=958.1                                                  INITIALT   84
   24 X(7)=X(7)*FMACH                                              INITIALT   85
      X(3)=-X(7)*SIN(AM)                                           INITIALT   86
      IF(GL.EQ.0.) GO TO 4                                         INITIALT   87
      VL=0.                                                        INITIALT   88
      GO TO 5                                                      INITIALT   89
    4 VL=X(7)+.1*FMACH                                             INITIALT   90
    5 X(7)=-X(7)*COS(AM)                                           INITIALT   91
      VM=VMAX*FMACH                                                INITIALT   92
      RHO=RHOZ*EXP(-C7*(XP(8)))                                    INITIALT   93
      CONST=.5*RHO*CDPS                                            INITIALT   94
      NFCNT=0                                                      INITIALT   95
      IFCNT=1                                                      INITIALT   96
      ICOMP=IFDIR(1)                                               INITIALT   97
      DO 25 J=1,NFLARE                                             INITIALT   98
      IF(ICOMP.EQ.IFDIR(J+1)) GO TO 95                             INITIALT   99
      ICOMP=IFDIR(J+1)                                             INITIALT  100
      NFCNT=NFCNT+1                                                INITIALT  101
      IX(NFCNT)=IFCNT                                              INITIALT  102
      IFCNT=1                                                      INITIALT  103
      IFX(NFCNT)=IFDIR(J)                                          INITIALT  104
      GO TO 25                                                     INITIALT  105
   95 IFCNT=IFCNT+1                                                INITIALT  106
   25 CONTINUE                                                     INITIALT  107
      IF(NFCNT.NE.0) GO TO 27                                      INITIALT  108
      NFCNT=1                                                      INITIALT  109
      IX(NFCNT)=IFCNT                                              INITIALT  110
   27 TMULT=0.                                                     INITIALT  111
      SS=0.                                                        INITIALT  112
      DO 28 K=1,30,NFCNT                                           INITIALT  113
      DO 29 KJ=1,NFCNT                                             INITIALT  114
      IF(K+KJ-1.GT.30) GO TO 30                                    INITIALT  115
      ANG=FJDIR(IFX(KJ))*PI/180.
```

Table 2-1 (Continued)

```
SUBROUTINE INITIAL    74/74    OPT=1                              FTN 4.2+P390 -    05/01/75  13.30.41.

          VHF(K+KJ-1)=VFLARE*COS(ANG)                                        INITIALT  116
          VIF(K+KJ-1)=VFLARE*SIN(ANG)                                        INITIALT  117
          TF(K+KJ-1)=TIME+SS*TMULT                                           INITIALT  118
          PSF(K+KJ-1)=IX(KJ)                                                 INITIALT  119
       29 CONTINUE                                                           INITIALT  120
          SS=SS+1                                                            INITIALT  121
          TMULT=TDLAY                                                        INITIALT  122
       28 CONTINUE                                                           INITIALT  123
       33 IF(IOSPLAY.EQ.0) GO TO 31                                          INITIALT  124
          IBLNK=64                                                           INITIALT  125
          IF(X7.GT..0) GO TO 32                                              INITIALT  126
          IPS=6HNIL                                                          INITIALT  127
          GO TO 33                                                           INITIALT  128
       32 IPS=6HA/B                                                          INITIALT  129
       33 WRITE(9,39) IBLNK,IAC,IBLNK,DL,X0(3),X7,X0(9),IPS,(IFOTR(K),K=1,2) INITIALT  130
         +,IX(1),(DLOT,TIN,RET,AH,AV                                         INITIALT  131
       39 FORMAT(1A6,6E12.6,A6,3I1,5I1I.3)                                   INITIALT  132
       31 IF(SSW6.EQ.0.) GO TO 34                                            INITIALT  133
          WRITE(13) AV,TOLOT,TIN,(IFTYPE(J),J=1,I),(MISTYPE(J),J=1,I)        INITIALT  134
       34 VH=SQRT(X(7)*X(7)+X(9)*X(9)+VT*(1)*VT*(1))                         INITIALT  135
          VV=VIF(1)                                                          INITIALT  136
          VTRM=1331387.                                                      INITIALT  137
          V=SQRT(VH*VH+VV*VV)                                                INITIALT  138
          TT=-DELT                                                           MAR19      34
          NT=230./DELT+1                                                     MAR19      35
          DO 35 ITM=1,NT                                                     MAR19      36
          TT=TT+DELT                                                         MAR19      37
          A=1.-TT/TBURN                                                      INITIALT  140
          IF(NK.NE.6) GO TO 61                                               INITIALT  141
          AREA=AFL*A*A                                                       INITIALT  142
          WE=WFO*A*A                                                         INITIALT  143
          GO TO 62                                                           INITIALT  144
       61 AREA=SFO*A*A                                                       INITIALT  145
          WE=WFO-WG*(1.-AREA/SFO)                                            INITIALT  146
       62 VHD=-CONST*AREA/WE*V*VH                                            INITIALT  147
          VVD=-CONST*AREA/WE*V*VV-G                                          INITIALT  148
          VH=VH+VHD*DELT                                                     INITIALT  149
          VV=VV+VVD*DELT                                                     INITIALT  150
          V=SQRT(VH*VH+VV*VV)                                                INITIALT  151
          IF(V.GT.VTRM) GO TO 36                                             INITIALT  152
          VTRM=V                                                             INITIALT  153
       35 CONTINUE                                                           INITIALT  154
          WRITE(6,37)                                                        INITIALT  155
       37 FORMAT(1X,24HMIN VELOCITY NOT REACHED)                             INITIALT  156
          STOP                                                               INITIALT  157
       36 DO 38 J=1,20                                                       INITIALT  158
       38 VTERM(J)=VTRM                                                      INITIALT  159
          RETURN                                                             INITIALT  160
          END                                                                INITIALT  161
```

3. Air density at sea level, variable name RHOZ (slugs/ft^3) and coefficient of exponential variation with altitude, variable name CZ (ft^{-1})

4. Atmospheric density as a function of altitude, variable name RHO

5. Speed of sound as a function of altitude, variable name FMACH

6. Surface area, variable name SFB (ft^2).

Atmospheric density is given as:

RHO = RHOZ * EXP (-CZ * ALTITUDE)
RHOZ = STD air density (sea level)

where,

ALTITUDE = XP(8) = TARGET ALTITUDE

Speed of sound is a function of altitude. Altitude of missile and target are averaged and checked whether,

IF

Altitude > 36152, FMACH = 968.1

IF

Altitude ≤ 36152, FMACH = 1116.0 - 0.0041 * Altitude

<u>Initialization</u>

Pertinent missile, target and flare parameters, including position, velocity, and acceleration are initialized. Certain parameters previously defined by data file input are discussed in Section 1. They are:

1. Target range (ft.), variable name X(6)
2. Target velocity (ft/sec.), variable name X(7)
3. Target altitude (ft.), variable name XP(8)
4. Missile altitude (ft.), variable name XP(3).

Other required initial conditions are:

1. Elevation angle as a function of a random number (radians), variable name AV
2. Flare time between salvos (sec.), variable name TF
3. Flare dispenser ejection velocity components (ft./sec.), variable name VXF, VYF, VZF
4. Minimum flare velocity (ft./sec.), variable name VTERM.

Minimum flare velocity is precomputed here to be used later as a check on radiant intensity. This velocity is based on flare burn time, surface area, weight, and drag.

Launch Geometry

Options available for missile launch conditions are:

1. Input azimuth angle (AH) and horizontal range (RXY)(in which the program selects the elevation angle (AV) on a random basis.
2. Input azimuth angle, horizontal range, and elevation angle.
3. Input azimuth angle, horizontal range, and vertical displacement (Vdisp).

2.2 LEAD DETERMINATION

This subroutine computes initial velocity components for the missile at launch. Several alternate launch modes are available. Different launch modes may be used in the elevation and azimuth planes. See Figure 2-3 for a block diagram description.

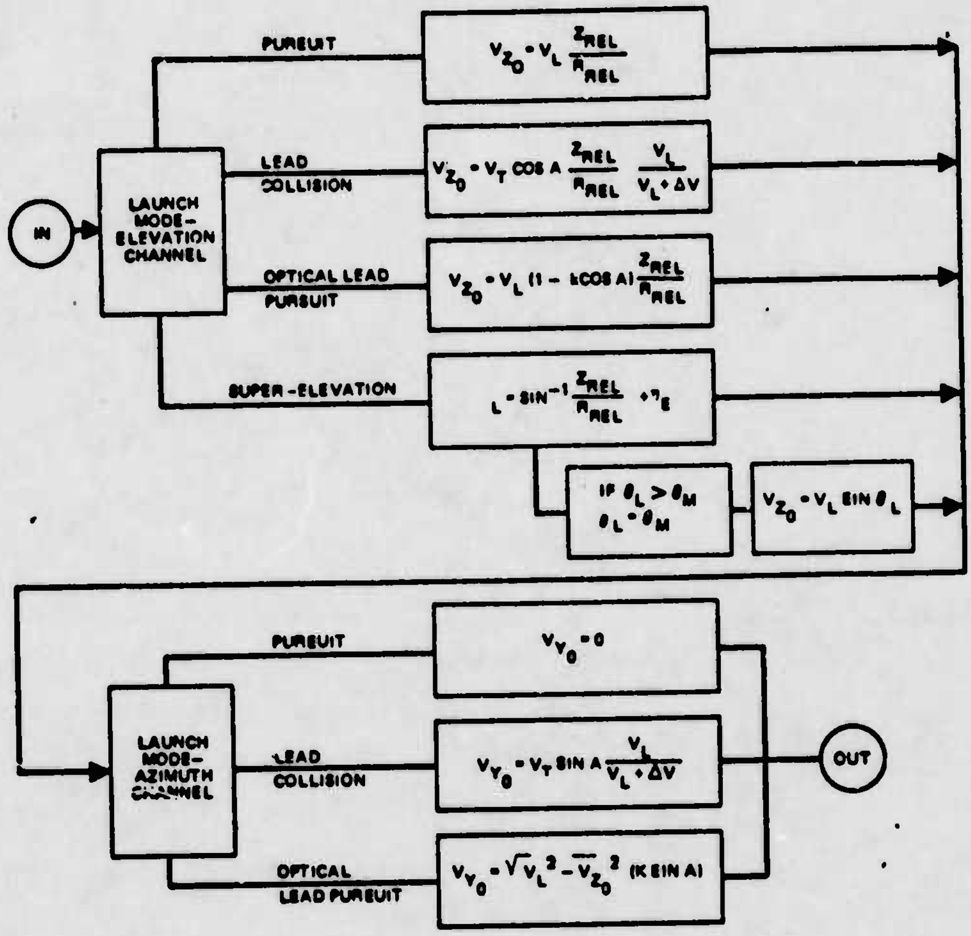


Figure 2-3. Lead subroutine block diagram

The following laws are available

1. Lead collision with maximum lead limit.
2. Pursuit
3. Visual lead pursuit with maximum lead limit.
4. Pursuit with super-elevation angle (elevation plane only).

Lead collision launch is based on attempting to put the missile on a collision course for a missile velocity of

$$V_L + \Delta V,$$

where V_L is launch velocity, and ΔV is an incremental velocity.

In pursuit launch, the missile is launched on a line directly toward the target. This mods is used when the launcher or the missile does not have capability for lead launch.

In visual lead pursuit, a lead angle is estimated by the person launching the missile. The value of lead is generally restricted to a small angle in this mode. The lead angle η_L is computed from

$$\text{Sin } \eta_L = K \sin A_T,$$

where A_T is target aspect angle, and K is an empirical constant ~ 0.5.

Often the missile is launched in a trajectory above the target. The angle above the angle of launch is called the super-elevation angle. The superelevation angle is input as a constant, with a limit on the total elevation angle of launch. A program listing is contained in Table 2-2.

Table 2-2. Lead subroutine

```
SUBROUTINE LEAD       74/74   OPT=1                      FTN 4.2+P380       09/01/75  13.30.49.

      SUBROUTINE LEAD(GL,X,XP,RXY,AM,VL,DELV,S,T0,SLO,ALOMX,LMSI,LMSA,SU
     *PELO,THTMXO,PI)
      DIMENSION X(10),XP(10)
      ALTO=XP(6)-XP(3)
      R=SQRT(RXY*RXY+ALTO*ALTO)
      GO TO (110,120,130,140,150), LMSI
  110 XP(6)=VL*ALTO/R
      GO TO 200
  120 A=S*TM*RXY/R-X(7)*ALTO/R
      VE=VL+DELV
      VER=SQRT(VE*VE-X(9)*X(9)-A*A)
      XP(6)=(VER*ALTO/R+A*RXY/R)*(VL/VE)
      GO TO 200
  130 SLOMX=SIN(ALOMX*PI/180.)
      SLO=CLO*COS(AM)*ALTO/R*PI/180.
      IF(ABS(SLO).LE.SLOMX) GO TO 189
      SLO=SIGN(SLOMX,SLO)
  189 THETAL=ASIN(ALTO/R)+ASIN(SLO)
      XP(6)=VL*SIN(THETAL)
      GO TO 200
  140 SUPELR=SUPELO*PI/180.
      THTMXR=THTMXO*PI/180.
      THETAL=ASIN(ALTO/R)
      IF(THETAL-THTMXR) 141,141,149
  141 THETAL=THETAL+SUPELR
      IF(THETAL-THTMXR) 149,149,142
  142 THETAL=THTMXR
  149 XP(6)=VL*SIN(THETAL)
      GO TO 200
  150 XP(6)=0.
  200 GO TO (210,220,230), LMSA
  210 X(6)=0.
      GO TO 300
  220 X(6)=X(9)*VL/(VL+DELV)
      GO TO 300
  230 SLOMX=SIN(ALOMX*PI/180.)
      SLO=-CLO*SIN(AM)*PI/180.
      IF(ABS(SLO).LE.SLOMX) GO TO 289
      SLO=SIGN(SLOMX,SLO)
  289 X(6)=SQRT(VL*VL-XP(6)*XP(6))*SLO
  300 X(7)=SQRT(VL*VL-X(6)*X(6)-XP(6)*XP(6))
      RETURN
      END
```

3. DYNAMICS

Figure 3-1 is a block diagram of the major components of the simulation program and shows the dynamics computations to be performed. Basically, the dynamics portion of the simulation program determines the X-, Y-, and Z-components of acceleration, velocity, and position for for the missile, target, and flare(s).

The forces controlling the missile trajectory are thrust, chord, commanded, and gravity. The missile velocity and position are computed by integrating the total acceleration.

The target is considered to fly nominally straight and level, i.e., no maneuver. However, the program does allow for three maneuver options: (1) turn in any direction, (2) straight acceleration, and (3) turn with acceleration.

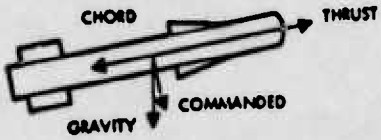
When the flare(s) is deployed, the flare deployment strategy controls how many are deployed, in what direction they are deployed, and how often they are deployed. The forces which govern the flare motion are drag and gravity.

3.1 MISSILE DYNAMICS

The missile equations of motion are governed by the four forces shown in Table 3-1. This table also lists the X-, Y-, and Z- components of each force. Figure 3-2 shows a block diagram of the computations performed in this portion of the program. Basically, the missile dynamics are updated as follows:

1. The aimpoint position is fed into the gyro subroutine from which gyro position and rate are output.
2. These gyro rates and positions are then fed into a subroutine which simulates the guidance unit of the missile and computes the acceleration components which are to be commanded by the missile.
3. The acceleration components due to thrust and chord forces are subsequently computed and resolved.

Table 3-1. Forces acting on the missile



FORCE	X COMPONENT	Y COMPONENT	Z COMPONENT
THRUST	$\text{THRUST} \cdot \dfrac{\cos(\gamma+\alpha)}{\left[1+\cos^2(\gamma+\alpha)\tan^2(\gamma'+\alpha')\right]^{1/2}}$	$\text{THRUST} \cdot \dfrac{\sin(\gamma+\alpha)}{\left[1+\cos^2(\gamma+\alpha)\tan^2(\gamma'+\alpha')\right]^{1/2}}$	$\dfrac{\text{THRUST} \cdot \cos(\gamma+\alpha)\cdot\tan(\gamma'+\alpha')}{\left[1+\cos^2(\gamma+\alpha)\cdot\tan^2(\gamma'+\alpha')\right]^{1/2}}$
CHORD	$-1/2\,\rho C_c\,\pi\,(D_M/2)^2 V^2 \cdot \dfrac{\cos(\gamma+\alpha)}{\left[1+\cos^2(\gamma+\alpha)\tan^2(\gamma'+\alpha')\right]^{1/2}}$	$-1/2\,\rho C_c\,\pi\,(D_M/2)^2 V^2 \cdot \dfrac{\sin(\gamma+\alpha)}{\left[1+\cos^2(\gamma+\alpha)\tan^2(\gamma'+\alpha')\right]^{1/2}}$	$-1/2\,\rho C_c\,\pi\,(D_M/2)^2 V^2 \cdot \dfrac{\cos(\gamma+\alpha)\tan(\gamma'+\alpha')}{\left[1+\cos^2(\gamma+\alpha)\tan^2(\gamma'+\alpha')\right]^{1/2}}$
COMMANDED	$-\dfrac{AW\dot{\psi}}{G\left[1+\cos^2\psi\cdot\tan^2\psi'\right]^{1/2}} \cdot \left[\dfrac{\sin(\gamma+\alpha)}{\cos(\gamma-\psi+\alpha)} + \dfrac{\sin(\gamma'+\alpha')}{\cos(\gamma'-\psi'+\alpha')}\right]$	$\dfrac{A\dot{\psi}W}{G\cos(\gamma-\psi-\alpha)} \cdot \cos(\gamma+\alpha) \cdot \dfrac{1}{\left[1+\cos^2\psi\cdot\tan^2(\psi')\right]^{1/2}}$	$\dfrac{A\dot{\psi}W}{G\cos(\gamma'-\psi'+\alpha')} \cdot \dfrac{\cos\psi/\cos\psi'}{\left[1+\cos^2\psi\cdot\tan^2\psi'\right]^{1/2}}$
GRAVITY	0	0	W

G = GRAVITATIONAL CONSTANT D_M = MISSILE DIAMETER ρ = ATMOSPHERIC DENSITY

A = NAVIGATION PARAMETER   W = MISSILE WEIGHT C_c = CHORD COEFFICIENT

4. The missile acceleration components due to thrust, chord, commanded and gravity forces are summed to obtain the X-, Y-, and Z-missile acceleration components.

5. These components are then integrated to obtain the missile velocity components.

6. Finally, these velocity components are subsequently integrated to obtain missile position.

The remainder of this Section describes the gyro position and rate computations, the thrust computations, the chord force computations, and the commanded acceleration computations in detail. Table 3-2 shows the portion of the main program which involves the missile dynamics computations.

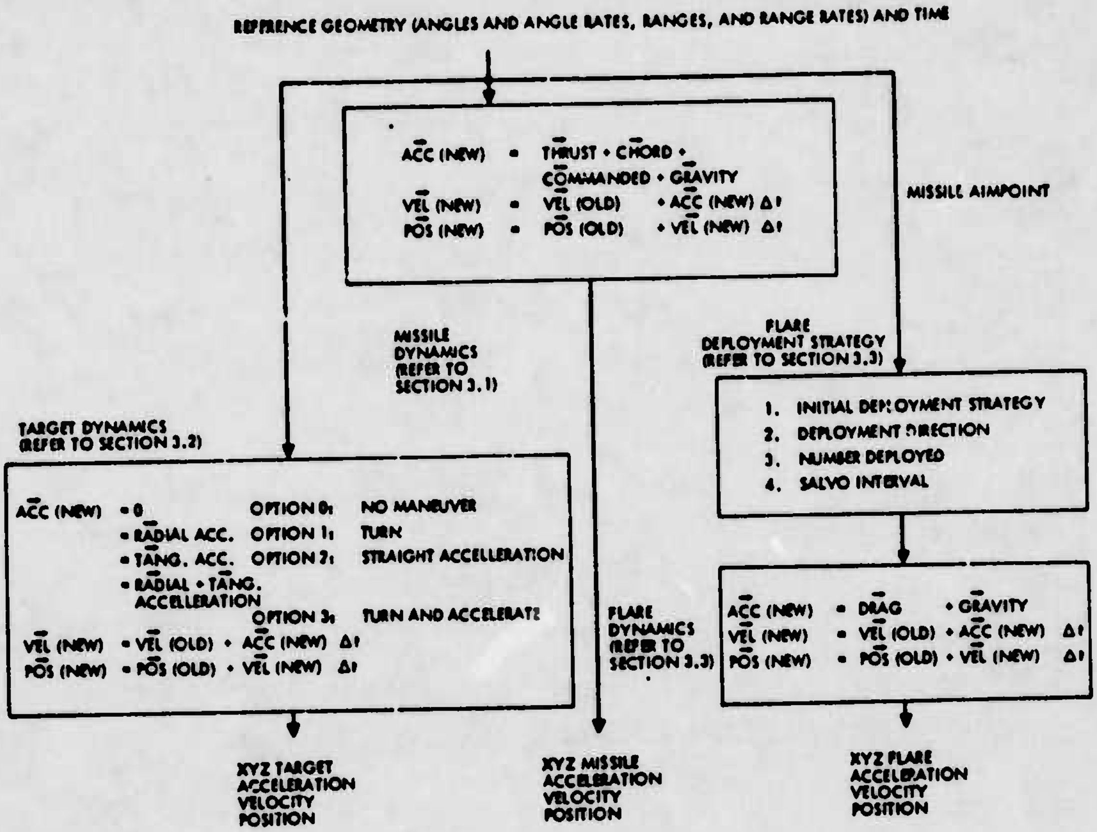
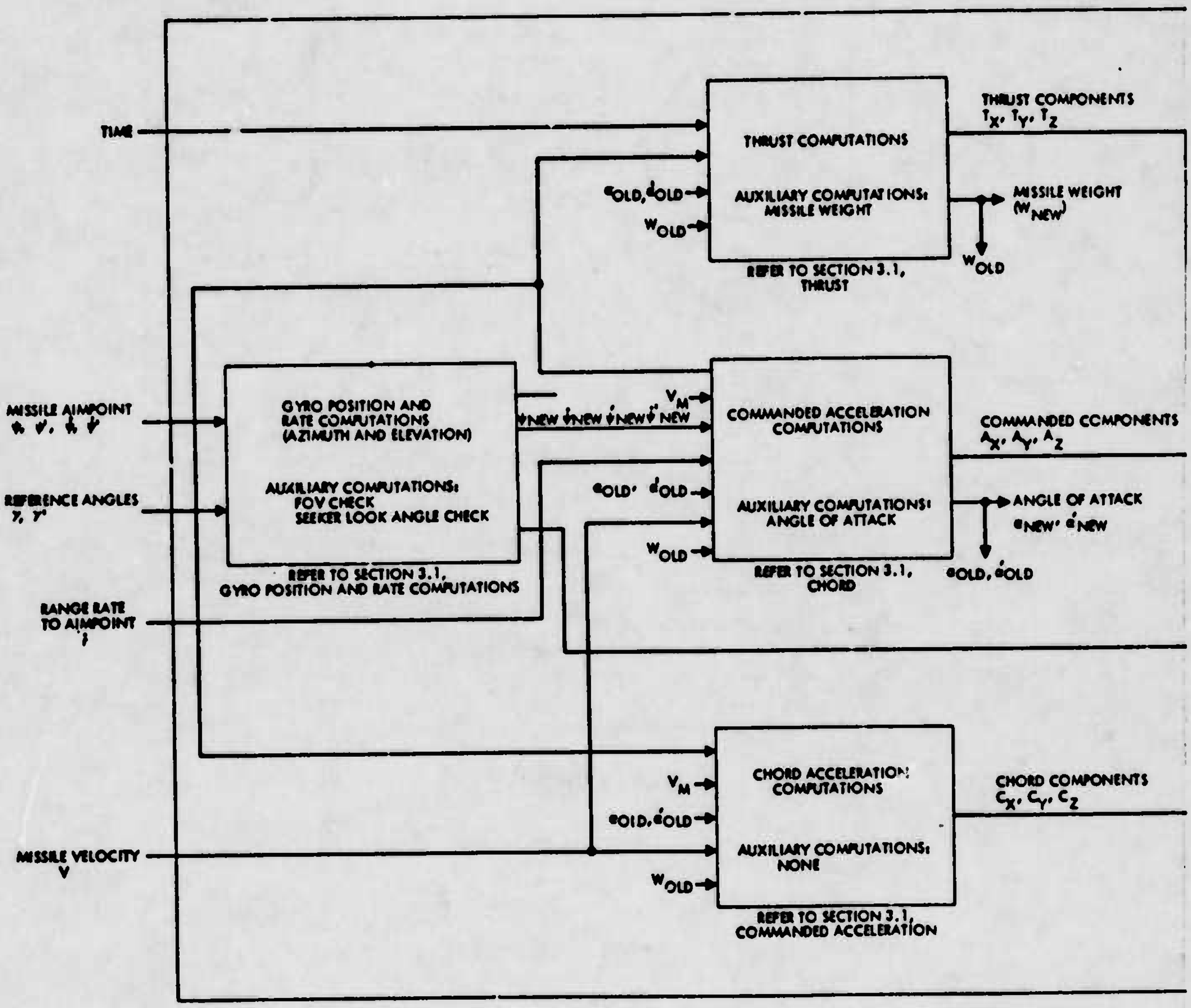
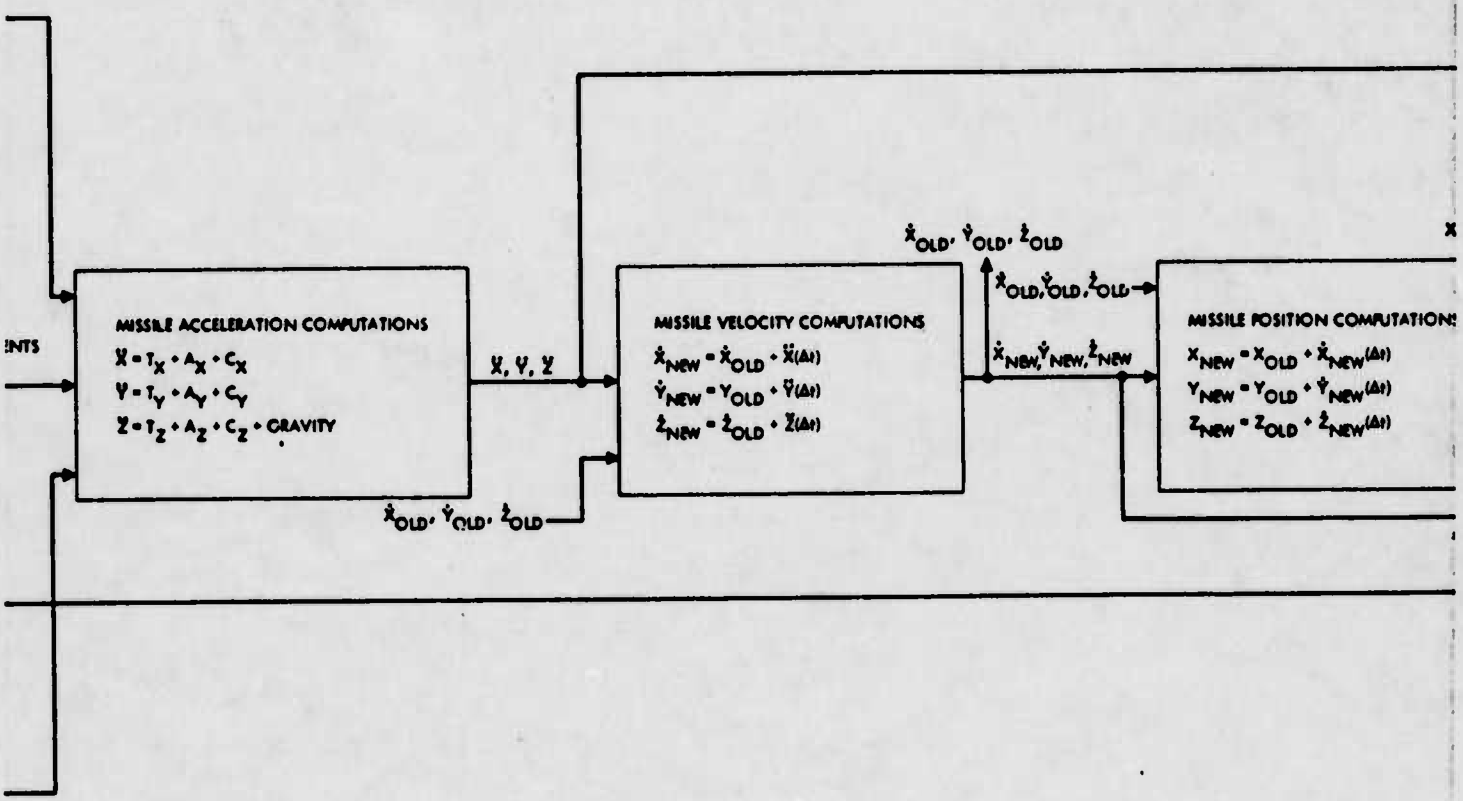


Figure 3-1. Dynamics



TIME
THRUST COMPUTATIONS
AUXILIARY COMPUTATIONS: MISSILE WEIGHT
REFER TO SECTION 3.1, THRUST
THRUST COMPONENTS
MISSILE WEIGHT
MISSILE AIMPOINT
REFERENCE ANGLES
GYRO POSITION AND RATE COMPUTATIONS (AZIMUTH AND ELEVATION)
AUXILIARY COMPUTATIONS: FOV CHECK SEEKER LOOK ANGLE CHECK
REFER TO SECTION 3.1, GYRO POSITION AND RATE COMPUTATIONS
COMMANDED ACCELERATION COMPUTATIONS
AUXILIARY COMPUTATIONS: ANGLE OF ATTACK
REFER TO SECTION 3.1, CHORD
COMMANDED COMPONENTS
ANGLE OF ATTACK
RANGE RATE TO AIMPOINT
CHORD ACCELERATION COMPUTATIONS
AUXILIARY COMPUTATIONS: NONE
REFER TO SECTION 3.1, COMMANDED ACCELERATION
CHORD COMPONENTS
MISSILE VELOCITY


ENTS
MISSILE ACCELERATION COMPUTATIONS
$\ddot{X} = T_X + A_X + C_X$
$\ddot{Y} = T_Y + A_Y + C_Y$
$\ddot{Z} = T_Z + A_Z + C_Z + \text{GRAVITY}$
$\ddot{X}, \ddot{Y}, \ddot{Z}$
$\dot{X}_{OLD}, \dot{Y}_{OLD}, \dot{Z}_{OLD}$
MISSILE VELOCITY COMPUTATIONS
$\dot{X}_{NEW} = \dot{X}_{OLD} + \ddot{X}(\Delta t)$
$\dot{Y}_{NEW} = \dot{Y}_{OLD} + \ddot{Y}(\Delta t)$
$\dot{Z}_{NEW} = \dot{Z}_{OLD} + \ddot{Z}(\Delta t)$
$\dot{X}_{OLD}, \dot{Y}_{OLD}, \dot{Z}_{OLD}$
$\dot{X}_{NEW}, \dot{Y}_{NEW}, \dot{Z}_{NEW}$
MISSILE POSITION COMPUTATIONS
$X_{NEW} = X_{OLD} + \dot{X}_{NEW}(\Delta t)$
$Y_{NEW} = Y_{OLD} + \dot{Y}_{NEW}(\Delta t)$
$Z_{NEW} = Z_{OLD} + \dot{Z}_{NEW}(\Delta t)$
X

Fi

7

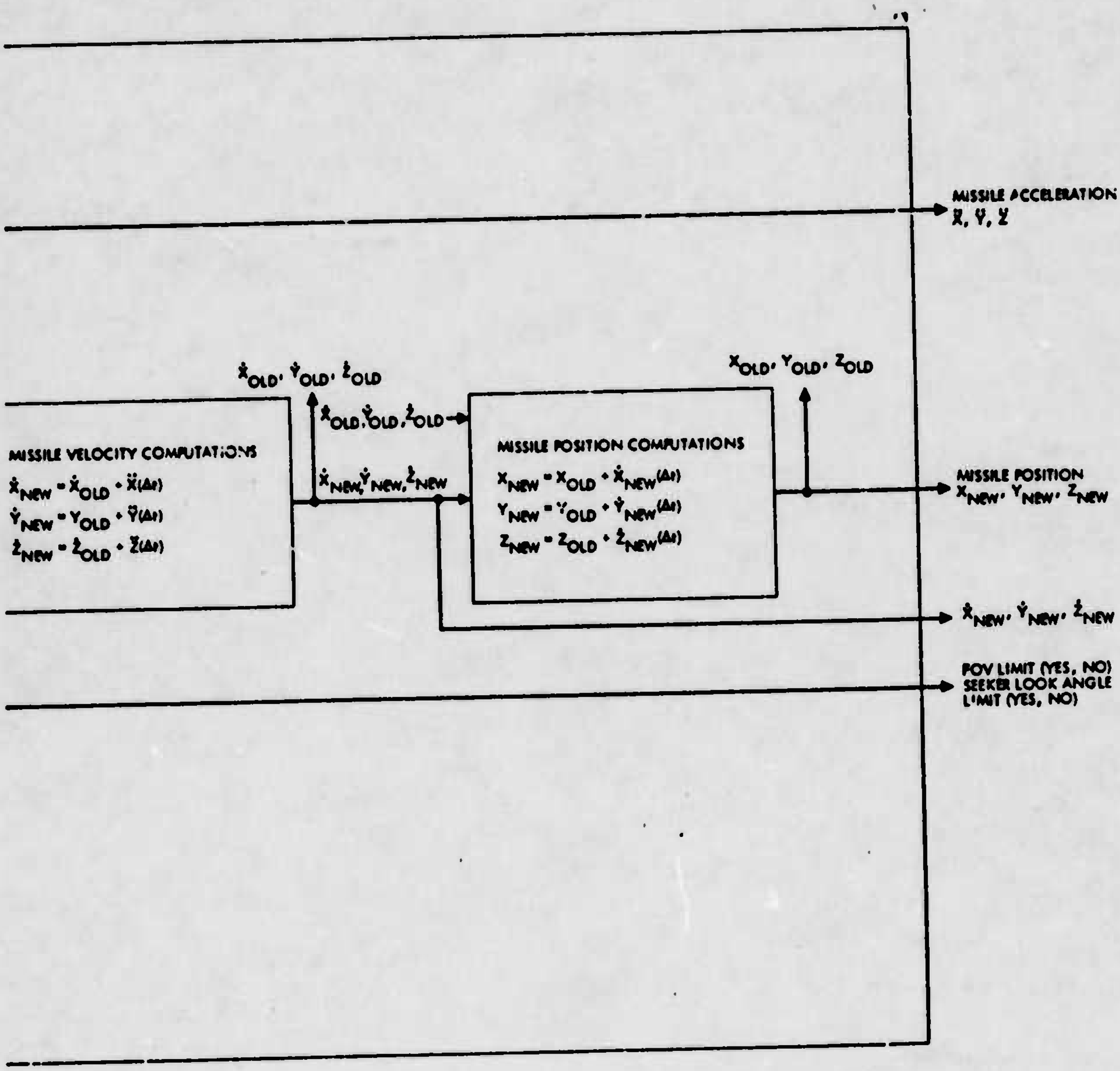


Figure 3-2. Missile dynamics block diagram

Table 3-2. Executive routine showing missile/dynamics computations

```
PROGRAM EPICS       74/74   OPT=1                           FTN 4.2+P380         03/31/75  13.38.19.

      PROGRAM EPICS(INPUT,OUTPUT,TAPE12,TAPE7,TAPE9,TAPE13,TAPE11,TAPE14 EPICS     2
     1          ,TAPE5=INPUT,TAPE6=OUTPUT)                              EPICS     3
      REAL JF,JIM                                                       EPICS     4
      DIMENSION MF(20),MTS(2),NF(20),RTLF(9,20),REL(12),IJDIR(8)        EPICS     5
      DIMENSION SF(20),HF(20),JF(20),VF(20)                             EPICS     6
      DIMENSION RF(20),RDF(20),TTGF(20),XFD(10,20)                      EPICS     7
      COMMON /BLK1/ST(10),TL(10),SIM(10),MO(10),VMD(8),CDO(8),CNT(8),CNA EPICS    8
     +(10),ALPH(10,8),LL,SA,ALTM,FOV,AS,TMAX,D1,TS,TB,A6,NM,TGU,RNK,GKO, EPICS    9
     +ISNOM,MISTYPE(2),BLURC,MMIN,DDD(9)                                EPICS    10
      COMMON /BLK2/FMS,AM,VMAX,XS,XN,YS,ZS,IAD,TGTALT,VELT,PANG(30),RINT EPICS    11
     +(30)                                                              EPICS    12
      COMMON /BLK7/IFTYPE(2),MK,DF,RL,TSURN,MFO,CDF,FIRAND(5),RIFI,MG,VF EPICS    13
     +LARE,TTN(100),JIM(100)                                            EPICS    14
      COMMON /BLK6/RNG(19),TAUT(19),TAUF(19)                            EPICS    15
      COMMON SSM1,SSM2,SSM3,SSM4,BIAS,IDSPLAT,MSS1,IFC                  EPICS    16
      COMMON PI,RHO,N1,N2,DF,T,YDLDT,TIM,V                              EPICS    17
      COMMON DELV,AM,RXY,IFP,TUP,TLOW                                   EPICS    18
      COMMON MFLARE,TFOTP(20),MLD,VDISP,AV,FMACH,S,,G                   EPICS    19
      COMMON IGT(20,3)                                                  EPICS    20
      COMMON MM(20),L2,N                                                EPICS    21
      COMMON TF(20),FSF(20),DTFRM(20),STMM,SIGM,SIDM,SIDPM,T,SI,SIP     EPICS    22
      COMMON VXF(20),VYF(20),VZF(20),Z(14),ZP(14),X(10),XP(10)          EPICS    23
      COMMON ZF(3,20),ZFP(3,20),XF(10,20),XFP(10,20),VT                 EPICS    24
      COMMON X20,X40,XP40,XP7,X33,XP33,IMRJST                           EPICS    25
      COMMON ALPHA,ALPHAP,TX,TY,TZ,GX,GY,GZ,AX,AY,AZ                    EPICS    26
      NAMELIST/PARAM/SSM1,SSM3,IRNDM,MFO,SA,W,TM,FOV,AS,TMAX,          MAR19     1
     +GKO,D1,TS,TB,A6,W,IOPT,MG,TM,X,XP,L,,CDF,RINT,PANG,              MAR19     2
     +CNA,MMIN,VMC,CDC,CNT,TAURM,TSU,ISMD,XGAIN,VFLARE,GL,              MAR19     3
     +RXY,AM,TIAS,TLOW,TUP,AV,VDISP,MLD,IFD,TIN,TDLOV,REFT,             MAR19     4
     +IDSPLAT,ST,TDPLOY,RF,RDF,TTGF,SUPELD,T4FMXD,LMSF,                 MAR19     5
     +LMSA,CLO,TL,SIM,VD,MTC,RUN,ALDMX,SSM2,SSM4,FMG,                   MAR19     6
     +TGTALT,VELT,XN,YN,YS,ZS                                           MAR19     7
      DATA(RJDIR(J),J=1,8)/90.,270.,0.,180.,45.,135.,225.,315./         EPICS    37
      REWIND 7                                                          MAR20     1
      REWIND 11                                                         MAR20     2
      REWIND 12                                                         MAR20     3
      REWIND 14                                                         MAR20     4
      CALL FLRCNST(RUN,IDPLOY,IOPT)                                     MAR19     8
      READ(5,17) MFLARE                                                 EPICS    43
      IF(MFLARE.EQ.0) GO TO 9                                           MAR20     5
   17 FORMAT(I1)                                                        MAR19     9
      READ(5,18) (IFDIR(J),J=1,MFLARE)                                  EPICS    43
   18 FORMAT(20I1)                                                      MAR19    10
    9 READ(5,19) (MTS(J),J=1,2)                                         EPICS    45
   19 FORMAT(2A6)                                                       EPICS    46
      CALL MSLCNST(MTS,MTP,W,T,G,)                                      MAR19    11
      CALL TGTCNST(ISNOM,TM)                                            MAR19    12
      CALL RVSTAU(TGTALT,ISNOM)                                         EPICS    49
   11 READ(5,PARAM)                                                     MAR19    13
      IF(IEND.NE.0) GO TO 9                                             EPICS    51
      IF(MTC.GT.0) GO TO 9                                              EPICS    52
      RUN=RUN+1.                                                        EPICS    53
      CALL INITIAL(D1,TS,MK,DF,VELT,TGTALT,IAD,IRNDM,MFO,CDF,IFTYPE,MIST EPICS   54
     +YPE,VFLARE,MD,XL,TSURN,VN,VMAX,RJDIR,T,,RUN,AM,W)                 EPICS    55
      CALL LEAD(GL,X,XP,RXY,AM,VL,DEL,V,G,TB,DLR,ALTMX,LMSF,LMSA,SUPELD,T EPICS  56
     +MTMXD,PT)                                                         EPICS    57
      CALL MISPAGE(X(2),RUN,MISTYPE,SSM4,L,)                            EPICS    58
```

(Table 3-2, continued)

```
      PROGRAM EPICS       74/74   OPT=1                         FTN 4.2+P380       09/01/75  13.?.19.

                                                                         EPICS      59
      GO TO 4                                                            EPICS      60
    6 T=T+DELT                                                           EPICS      61
      IF(T.LT.TMAX) GO TO 15                                             EPICS      62
      WRITE(6,14)                                                        EPICS      63
   14 FORMAT(/,2X,24HMISSILE LIFESPAN EXCEEDED,/)                        EPICS      64
      GO TO 17                                                           EPICS      65
   15 N1=N1+1                                                            MARL9      16
      CALL GYRO   (SIP,SIDP,SI,SIP,SIDM,SIDPM,SIN4,SIPM,VLIM,ALPHA,ALPH  EPICS      67
     +AP,Z(5),ZP(5),DELT,LP,TAU,S1,T3J,T)                                EPICS      68
      IF(L2.EQ.2) GO TO 18                                               EPICS      69
      CALL THRUST(ST,TL,SIN,W0,Z,ZP,ALPHA,ALPHAP,THRUST,W,T,DELT,TX,TY,TZ,G EPICS   70
     +1)                                                                 EPICS      71
      CALL COMMAND(X(5),XP(5),RT,G,W,DL,VMM,RHO,VMS,GMA,ALPM,FMACH,ALPHA EPICS      72
     +,A,PHAP,A5,SC0,?NT,SIN4,SIPM,SI76,SIDPM,RATIF,Z(5),ZP(5),DELT,TS,M EPICS      73
     +IAS,AX,AY,AZ,TM,T,SL)                                              EPICS      74
      CALL PHMRS(FMACH,VMP,C??,RHO,PI,S,Q1,Z,ZP,ALPHA,ALPHAP,VMM,W,CX,CY EPICS      75
     +,CP)                                                               EPICS      76
      X2P=TX+AX+CX                                                       EPICS      77
      X4P=TY+AY+CY                                                       EPICS      78
      XP6P=TZ+AZ+CZ-G                                                    EPICS      79
      X(2)=X(2)+X2P*DELT                                                 EPICS      80
      X(4)=X(4)+X4P*DELT                                                 EPICS      81
      XP(6)=XP(6)+XP6P*DELT                                              EPICS      82
      X(1)=X(1)+X(2)*DELT                                                EPICS      83
      X(3)=X(3)+X(4)*DELT                                                EPICS      84
      XP(5)=XP(5)+XP(6)*DELT                                             EPICS      85
      IF(XP(5).GT.0.) GO TO 20                                           EPICS      86
      WRITE(6,21) T                                                      EPICS      87
   21 FORMAT(1X,27HMISSILE HIT GROUND AT TIME ,F9.4)                     EPICS      88
      GO TO 17                                                           EPICS      89
   20 IF(NFLAG.EQ.1) GO TO 6                                             EPICS      90
      IF(IMODE.EQ.1) GO TO 12                                            EPICS      91
      IF(TR.GE.T) GO TO 6                                                MARL9      15
   12 CALL FLPATH   (NR,M,TGT,T,TF,FTS,FTSF,STMP,CF,XFP,K,XP,TRJPH,VTF,V EPICS      93
     +TF,VZF,VFP,XPP,VPPD,DEL,RF,RTF,IMODE,N,PI,T)                       EPICS      94
      IF(M.EQ.3) GO TO 6                                                 MARL9      16
      CALL FLPATH   (T,DELT,TF,TRJPH,SF,SFP,TF,VFD,HG,HF,N,RHO,CDF,G,VF,X EPICS     95
     +,YP,XFD,YF,YFP,NM,NC)                                              MARL9      17
      CALL MFLRVFL(M,XF,YFP,XFD,VF,RF,DELT)                              MARL9      18
      CALL MFLANG(XF,N,ZF,PI)                                            MARL9      19
      CALL MFLANG(VFP,N,TFP,PI)                                          MARL9      20
      CALL FLPIR    (T,TF,DELT,N,VF,JF,AF,IDPI,ITN,JIN,FIRING,IRNDM,PSF,P EPICS     101
     +CFI,TRUPN,VFFDM,RNS,TAJF)                                          EPICS     102
      N1=1NR,PRI                                                         EPICS     103
      IF(N1.LT.N2) GO TO 6                                               EPICS     104
    9 CONTINUE                                                           EPICS     105
      DO 7 K=1,M                                                         EPICS     106
      GO TO (13,13,7), NM(K)                                             EPICS     107
   13 IF(XFP(4,K).GT.0.) GO TO 16                                        EPICS     108
      JF(K)=0.                                                           EPICS     109
      HF(K)=0.                                                           EPICS     110
      GO TO 7                                                            EPICS     111
   16 IF(SSM2.EQ.0.) GO TO 7                                             EPICS     112
      ZF1=ZF(1,K)*M1                                                     EPICS     113
      ZF2=ZFP(1,K)*M1                                                    EPICS     114
      ZF3=ZF(2,K)*M1                                                     EPICS     115
      ZF4=ZFP(2,K)*M1
```

(Table 3-2, concluded)

```
PROGRAM EPICS      74/74   OPT=1                               FTN 4.2+P398        05/01/75  13.30.19.

          WRITE(6,402) K                                                       EPICS     116
      402 FORMAT(6X,24HNUMBER ,I4,1X,21H ARE CHARACTERISTICS(                  EPICS     117
          WRITE(6,576) XF(5,K),XF(9,K),XPO(9,K)                                EPICS     118
          WRITE(6,679) XF(7,K),XF(9,K),XPO(9,K)                                EPICS     119
          WRITE(6,576) RELF(1,K),RELF(2,K),RELF(3,K)                           EPICS     120
          WRITE(6,579) RELF(7,K),RELF(9,K),VF(K)                               EPICS     121
      579 FORMAT(2X,4HRELF=,F8.2,2X,9HRDRELF=,F9.2,1X,3HVF=,F9.2)              EPICS     122
      576 FORMAT(2X,5HRREL=,F10.2,2X,4HRRF=,F9.2,2X,5HRRFL=,F9.2)              EPICS     123
      579 FORMAT(2X,6HVXF=,F9.6,1X,6HVYF=,F9.6,2X,6HVZF=,F9.6)                 EPICS     124
      576 FORMAT(2X,3HXF=,F12.7,2X,3HYF=,F10.6,2X,3HZF=,F10.6)                 EPICS     125
          WRITE(6,999) SF(K),WF(K)                                             EPICS     126
      999 FORMAT(2X,3HSF=,F12.5,2X,7HWEIGHT=,F12.5)                            EPICS     127
          WRITE(6,692) JF(K),HF(K)                                             EPICS     128
      692 FORMAT(2X,13HINTENSITY=,F12.5,2X,11HIRRADIANCE=,F12.5)               EPICS     129
          WRITE(6,996) ZF1,ZF2,ZF3,ZF4                                         EPICS     130
      996 FORMAT(2X,7HSI(XY)=,F12.5,2X,7HRT(XZ)=,F12.5,2X,10HSIDOT(XY)=,F12.   EPICS     131
         *6,2X,10HSIDOT(XZ)=,F12.5)                                            EPICS     132
        7 CONTINUE                                                             EPICS     133
        5 CALL TGTVM   (T,TM,DELT,AM,G,FMS,VM,VT,X(4),X(5),XP(4),X(7),X(9),X   MAR23       6
         *P(3),XP7,XM3,YP3,ZP3)                                                EPICS     135
        6 VMM=SQRT(X(2)*X(2)+X(4)*X(4)+XP(6)*XP(6))                            EPICS     136
          CALL MTRVEL  (X,XP,V20,X63,XP63,X70,X73,XP93,T,VMM,VM,VT,VTD,REL,    MAR20       7
         *L2,TTG,RMISS,MSG)                                                    EPICS     138
          IF(L2.EQ.2) GO TO 10                                                 EPICS     139
          CALL MTSTANG   (X,ALPHA,V67,X73,XM3,X73,7,PI)                        MAR20       8
          CALL MTSTANG   (XP,ALPHAP,X63,XP3,XM3,XP3,ZP,PI)                     MAR20       9
          CALL TGTIR   (X(7),X(9),XP(9),WT,REL(1),RFL(2),REL(3),REL(10),PI,R   MAR20      10
         *INT,PANS,RNS,TAUT,AT,TINT)                                           EPICS     143
          CALL MSLAMMT (ZF,ZFP,Z,ZP,RELF,REL,AT,A,WF,N,HF,KF,LT,SI,SIP,SIN,    MAR20      11
         *SIDP,PA,RADOT,FOV,LP,MMIN,LL,T)                                      EPICS     149
          IF(L2.EQ.2) GO TO 10                                                 EPICS     146
          IF(N1.LT.N2) GO TO 5                                                 EPICS     147
          N1=A                                                                 MAR20      12
          IF(SSW3.NE.A.) GO TO 22                                              EPICS     149
          GO TO 23                                                             EPICS     163
       22 NN1=N-1                                                              MAR19      21
          WRITE(A) OUN,NN1,XP(3),XP(5),(XFP(A,NN),NN=1,NN1),X(3),X(5),         MAR19      22
         *(XF(B,NN),NN=1,NN1),X(1),X(4),(XF(6,NN),NN=1,NN1),SI,Z(3),(ZF(1,NN   MAR19      23
         *),NN=1,NN1),SIP,ZP(3),(ZFP(1,NN),NN=1,NN1),((IGT(NN,N1),NN=1,NN1),   MAR19      24
         *N1=1,2),(JF(NN),NN=1,NN1),(RELF(7,NN),NN=1,NN1),(RELF(A,NN),NN=1,N   MAR19      25
         *N1),MT,REL(10),REL(11),T                                             MAR19      26
       23 IF(T.GT.0.) GO TO 32                                                 EPICS     169
          IF(SSW6.GT.0.) GO TO 32                                              EPICS     170
          CALL PAGE2(IAC,VELT,TGTALT,ITYPE,7,ZP,SI,SIP,SID,SIDP,REL,X,XP,XP    EPICS     171
         *0,X6D,XP6D,X70,X30,XP93)                                             EPICS     172
       32 CALL OUTPUT(X,XP,Z,ZP,ALPHA,ALPHAP,RA,RADOT,SINW,SIPW,SIGW,SIGPW,X   EPICS     173
         *ZD,X6D,XP6D,VT,VMM,FMACH,REL,RMISS,TTG,THRUST,W,MT,T,SSW2,TX,TY,TZ   EPICS     174
         *,AX,AY,AZ,CX,CY,CZ)                                                  EPICS     175
          GO TO (5,1), L2                                                      EPICS     176
        1 IF(MTF.GT.0) GO TO 3                                                 EPICS     177
          GO TO 11                                                             EPICS     178
       10 TMISS=RMISS                                                          EPICS     179
          CALL CLOSAPP   (REL,TTG,DELT,RMISS,XP,T,TMAX,XM,YM,ZM,TTGT)          MAR20      13
          CALL PHCOMP(REL,Z(7),PHK,YS,XS,X4,ZS,PH)                             EPICS     181
          CALL FINPAGE(XM,YM,ZM,REL,RMISS,TMISS,TTGT,TTG,PH)                   EPICS     182
          IF(SSW6.EQ.0.) GO TO 30                                              EPICS     183
          WRITE(13) RMISS,PH,TMISS,REL(13)                                     EPICS     184
          ENDFILE 13                                                           EPICS     185
       30 IF(SSW7.NE.0.) GO TO 29                                              EPICS     186
          IF(IDSPLAY.EQ.0) GO TO 11                                            EPICS     187
          ENDFILE 8                                                            EPICS     188
          WRITE(8,28) RMISS,T,PH                                               EPICS     189
       28 FORMAT(3E12.5)                                                       EPICS     190
       29 ENDFILE 8                                                            EPICS     191
          GO TO 11                                                             EPICS     192
        2 IF(SSW6.EQ.0.) GO TO 31                                              EPICS     193
          ENDFILE 13                                                           EPICS     194
          REWIND 13                                                            EPICS     195
       31 STOP                                                                 EPICS     196
          END                                                                  EPICS     197
```

Gyro Position and Rate Computations

Because there is a time lag associated with the gyro and the forward tracking loop, the aimpoint location, determined in subroutine aimpoint, is not tracked precisely by the missile. The actual value of $\dot{\psi}$ (aimpoint rate) which is required to command proper acceleration, is not fed into the guidance unit of the missile until several time constants later. This effect is modeled in the program by a simple one time constant delay, which is given by:

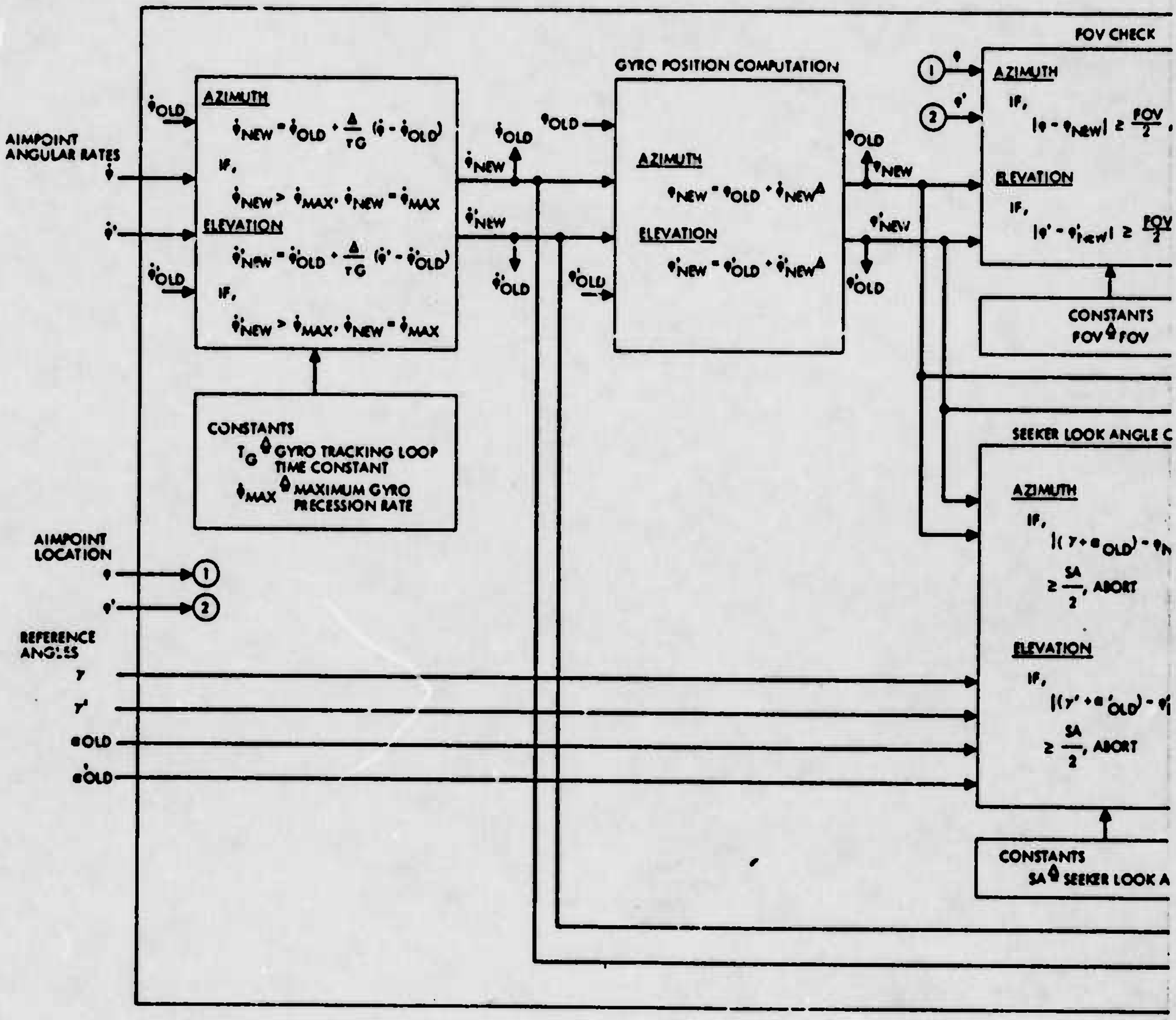
$$\dot{\psi}_{new} = \dot{\psi}_{old} + \left(\frac{\Delta}{T_G}\right)(\dot{\psi} - \dot{\psi}_{old})$$

where,

- Δ = integration step size
- $\dot{\psi}_{new}$ = present gyro rate
- $\dot{\psi}_{old}$ = previous gyro rate
- $\dot{\psi}$ = aimpoint LOS rate
- T_G = time constant of the forward tracking loop

Since the simulation uses a two-plane geometry, a computation for is made for both the horizontal and vertical plane using the equation described above. If the gyro rate computed from this equation becomes greater than maximum precession rate, it is set to the maximum value. The gyro position is obtained by an integration of the rate. Figure 3-3 shows the computations performed in this subroutine.

In addition to computing gyro rate and position, this subroutine also checks the missile FOV and seeker look angle limits. If either of these limits are exceeded, the program goes into an abort mode and the point of closest approach is subsequently computed along with the probability of hit. Table 3-3 contains a listing of this subroutine.


AIMPOINT ANGULAR RATES
AZIMUTH
IF,
ELEVATION
CONSTANTS
GYRO TRACKING LOOP TIME CONSTANT
MAXIMUM GYRO PRECESSION RATE
GYRO POSITION COMPUTATION
AZIMUTH
ELEVATION
FOV CHECK
AZIMUTH
ELEVATION
CONSTANTS
AIMPOINT LOCATION
REFERENCE ANGLES
SEEKER LOOK ANGLE C
AZIMUTH
ELEVATION
ABORT
CONSTANTS

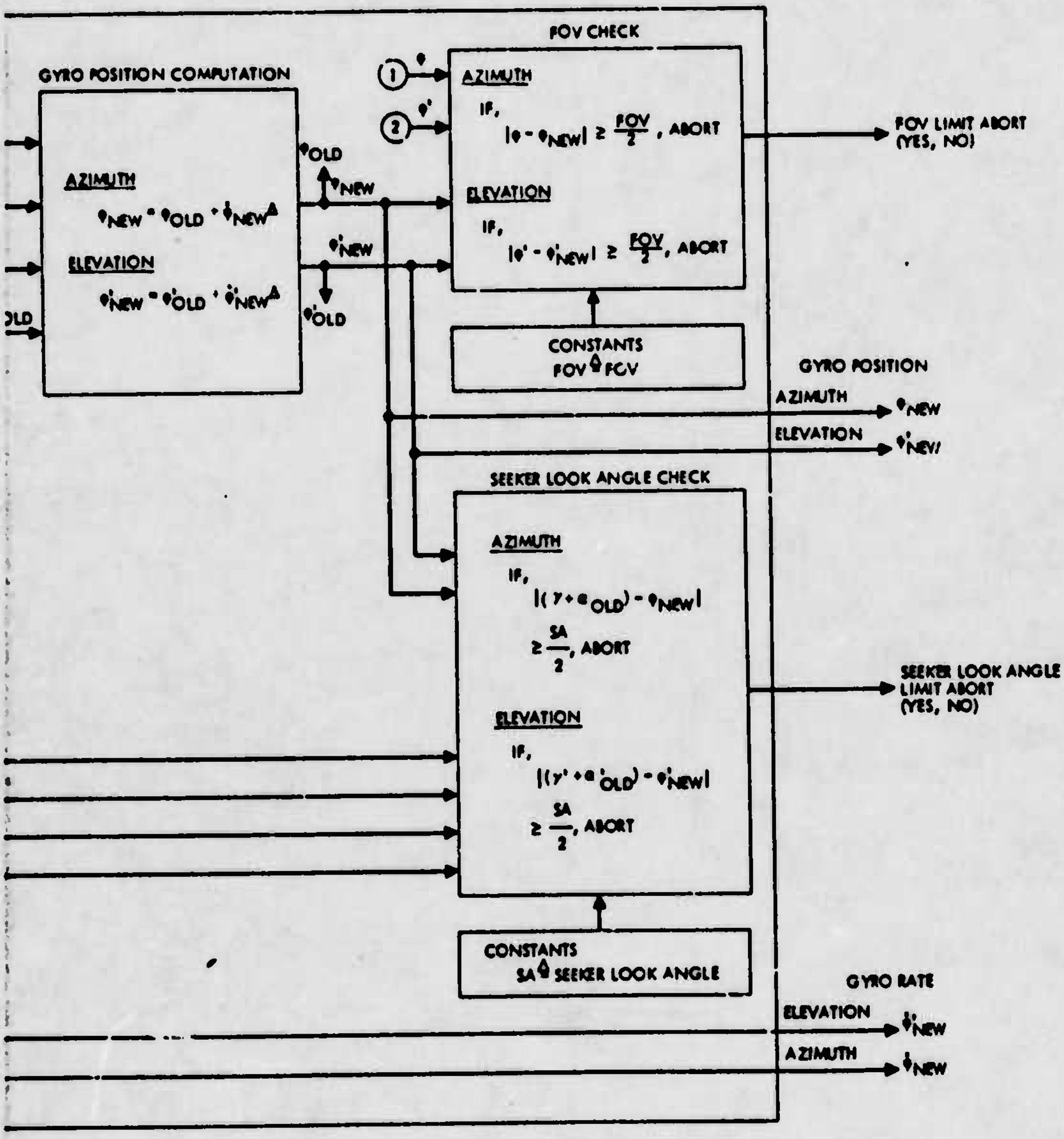


Figure 3-3. Gyro position and rate computations

Table 3-3. Gyro position and rate computation subroutine

```
SUBROUTINE GYROCMP     74/74    OPT=1                         FTN 4.2+P380       05/01/75  13.30.47.

          SUBROUTINE GYROCMP(SIO,SIOP,SI,SIP,SION,SIOPN,SIN,SIPN,SIOMAX,ALP  MAR19       38
         +HA,ALPHAP,GAM,GAMP,DELT,L2,FOV,SA,TGU,V)                           GYROCOMP     3
          IF(T.GT.DELT) GO TO 7                                              GYROCOMP     4
          SION=SIO                                                           GYROCOMP     5
          SIOPN=SIOP                                                         GYROCOMP     6
          SIN=SI                                                             GYROCOMP     7
          SIPN=SIP                                                           GYROCOMP     8
        7 SION=SION+DELT/TGU*(SIO-SION)                                      GYROCOMP     9
          IF(SION.LE.SIOMAX) GO TO 1                                         GYROCOMP    10
          SION=SIGN(SIOMAX,SION)                                             GYROCOMP    11
        1 SIOPN=SIOPN+DELT/TGU*(SIOP-SIOPN)                                  GYROCOMP    12
          IF(SIOPN.LE.SIOMAX) GO TO 2                                        GYROCOMP    13
          SIOPN=SIGN(SIOMAX,SIOPN)                                           GYROCOMP    14
        2 SIN=SIN+SIPN*DELT                                                  GYROCOMP    15
          SIPN=SIPN+SIOPN*DELT                                               GYROCOMP    16
          IF(ABS(SI-SIN).LT.FOV/2.) GO TO 3                                  GYROCOMP    17
          WRITE(6,8)                                                         MAR19       39
        8 FORMAT(1X,10HABORT COND)                                           MAR19       40
          WRITE(6,9) SI,SIN,FOV                                              MAR19       41
        9 FORMAT(1X,3HSI=,E12.6,1X,4HSIN=,E12.6,1X,4HFOV=,E12.6)             MAR19       42
          L2=2                                                               GYROCOMP    20
          RETURN                                                             GYROCOMP    21
        3 IF(ABS(SIP-SIPN).LT.FOV/2.) GO TO 4                                GYROCOMP    22
          WRITE(6,8)                                                         MAR19       43
          WRITE(6,10) SIP,SIPN,FOV                                           MAR19       44
          L2=2                                                               GYROCOMP    25
          RETURN                                                             GYROCOMP    26
        4 IF(ABS(GAM+ALPHA-SIN).LT.SA/2.) GO TO 5                            GYROCOMP    27
          WRITE(6,8)                                                         MAR19       45
          WRITE(6,11) GAM,ALPHA,SIN,SA                                       MAR19       46
          L2=2                                                               GYROCOMP    30
          RETURN                                                             GYROCOMP    31
        5 IF(ABS(GAMP+ALPHAP-SIPN).LT.SA/2.) GO TO 6                         GYROCOMP    32
          WRITE(6,8)                                                         MAR19       47
          WRITE(6,12) GAMP,ALPHAP,SIPN,SA                                    MAR19       48
       10 FORMAT(1X,4HSIP=,E12.6,1X,5HSIPN=,E12.6,1X,4HFOV=,E12.6)           MAR19       49
       11 FORMAT(1X,4HGAM=,E12.6,1X,6HALPHA=,E12.6,1X,4HSIN=,E12.6,1X,3HSA=, MAR19       50
         +E12.6)                                                             MAR19       51
       12 FORMAT(1X,5HGAMP=,E12.6,1X,7HALPHAP=,E12.6,1X,5HSIPN=,E12.6,1X,3HS MAR19       52
         +A=,E12.6)                                                          MAR19       53
          L2=2                                                               GYROCOMP    35
        6 RETURN                                                             GYROCOMP    36
          END                                                                GYROCOMP    37
```

Thrust

The propulsion system of a missile is completely defined by thrust-time history, specific impulse (delivered) and motor weight drop (if applicable) data. Figure 3-4 shows the computations performed in subroutine thrust and Table 3-4 contains a listing of this subroutine.

The thrust-time, motor weight drop-time, and specific impulse-time, profiles are stored tables in the program which are read in as part of the missile file data. The values of these variables are then found by means of a table lookup.

It is necessary to formulate this force into its X-, Y-, and Z-acceleration components. Table 3-5 shows the computational procedure for performing this operation and Figure 3-4 shows the equations used in implementing this component resolution, with the (G/W_{new}) factor accounting for the conversion of force to acceleration.

The thrust subroutine in addition to computing the thrust components also computes the missile weight. The change in missile weight during the thrust period is given by:

$$W_{new} = W_{old} - (\text{thrust}/S)\,\Delta - W_o$$

where

W_{new} = new missile weight

W_{old} = old missile weight

Δ = integration step size

S = specific impulse

W_o = motor weight drop at end of boost period (if applicable)

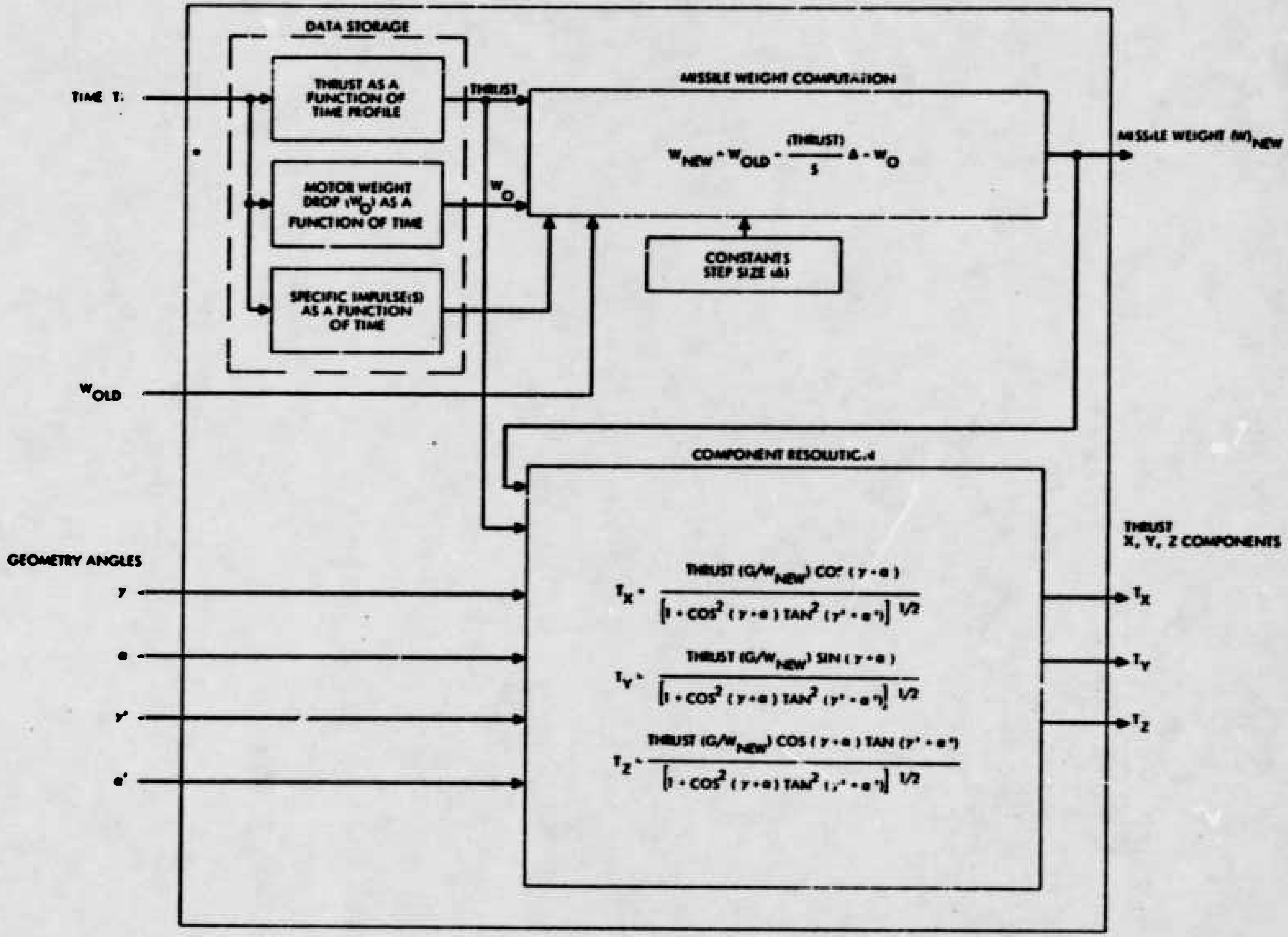


Figure 3-4. Thrust computations

Table 3.4. Thrust subroutine

```
SUBROUTINE THR        74/74    OPT=1                     FTN 4.2+P380       05/01/75  13.38.49.

      SUBROUTINE THR(ST,TL,SI,WO,Z,ZP,ALPHA,ALPHAP,THRUST,W,T,DELT,TX,TY THR      2
     +,TZ,G)                                                             THR      3
      DIMENSION Z(10),ZP(10)                                             THR      4
      DIMENSION ST(10),TL(10),SI(10),WO(10)                              THR      5
      THRUST=TLU2(T,ST,TL)                                               THR      6
      S=TLU2(T,S.,SI)                                                    THR      7
      WO=TLU2(T,ST,WO)                                                   THR      8
      W=W-THRUST*DELT/S-WO                                               THR      9
      SINGPA=SIN(Z(5)+ALPHA)                                             THR     10
      COSGPA=COS(Z(5)+ALPHA)                                             THR     11
      TANGAPP=TAN(ZP(4)+ALPHAP)                                          MARL9   94
      DEN=SQRT(1.+COSGPA**2*TANGAPP**2)*W/G                              MARL9   95
      TX=(THRUST*COSGPA)/DEN                                             THR     14
      TY=(THRUST*SINGPA)/DEN                                             THR     15
      TZ=(THRUST*COSGPA*TANGAPP)/DEN                                     MARL9   96
      RETURN                                                             THR     17
      END                                                                THR     18
```

Table 3-5. Thrust vector components

From geometry,

$$T_x = T_{xy} * \cos(\gamma + \alpha) = T_{xz} * \cos(\gamma' + \alpha')$$

$$T_{xz} = T_{xy} * \cos(\gamma + \alpha)/\cos(\gamma' + \alpha')$$

Now,

$$T^2 = T_x^{\,2} + T_y^{\,2} + T_z^{\,2}$$

$$= T_{xy}^{\,2} * \cos^2(\gamma + \alpha) + T_{xy}^{\,2} * \sin^2(\gamma + \alpha) + T_{xz}^{\,2} * \sin^2(\gamma' + \alpha')$$

$$= T_{xy}^{\,2} + T_{xz}^{\,2} * \sin^2(\gamma' + \alpha')$$

$$= T_{xy}^{\,2} + T_{xy}^{\,2} * \sin^2(\gamma' + \alpha') * \cos^2(\gamma + \alpha)/\cos^2(\gamma' + \alpha')$$

$$T_{xy} = \frac{T}{\left[1 + \cos^2(\gamma + \alpha) * \tan^2(\gamma' + \alpha')\right]^{1/2}}$$

$$T_{xz} = \frac{T * \cos(\gamma + \alpha)/\cos(\gamma' + \alpha')}{\left[1 + \cos^2(\gamma + \alpha) * \tan^2(\gamma' + \alpha')\right]^{1/2}}$$

$$T_x = \frac{T * \cos(\gamma + \alpha)}{\left[1 + \cos^2(\gamma + \alpha) * \tan^2(\gamma' + \alpha')\right]^{1/2}}$$

$$T_y = \frac{T * \sin(\gamma + \alpha)}{\left[1 + \cos^2(\gamma + \alpha) * \tan^2(\gamma' + \alpha')\right]^{1/2}}$$

$$T_z = \frac{T * \cos(\gamma + \alpha) * \tan(\gamma' + \alpha')}{\left[1 + \cos^2(\gamma + \alpha) * \tan^2(\gamma + \alpha')\right]^{1/2}}$$

Chord

The magnitude of chord force is given by the expression

$$1/2\, \rho C_c \pi (D_M/2)^2 V^2$$

where

ρ = atmospheric density

C_c = chord force coefficient (a function of mach number)

$\pi(D_M/2)^2$ = missile cross-sectional area

V = missile velocity

The atmospheric density (ρ) is modeled as an exponential function of altitude and computed only once in subroutine initial.

The chord force coefficient is a function of mach number and is found in the program by means of a table lookup.

The missile velocity needed to calculate the chord force is computed in the range and range rate computation subroutine.

The chord force is directed opposite the thrust vector and therefore has the same X-, Y-, and Z-unit vector components as the thrust vector.

Figure 3-5 shows the overall computations of subroutine thrust and Table 3-6 contains a listing of this subroutine.

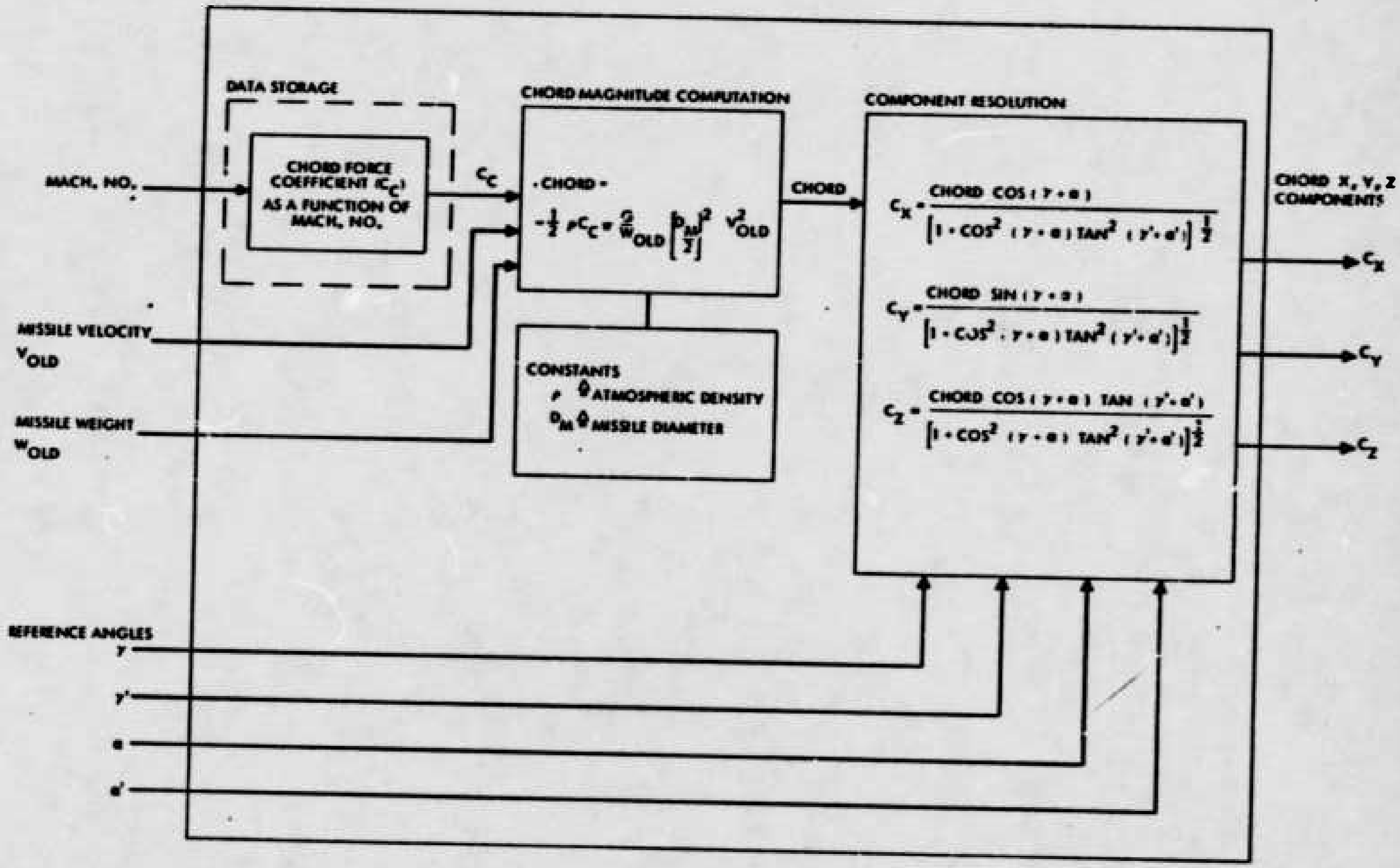


Figure 3-5. Chord acceleration computations

Table 3-6. Chord subroutine

```
SUBROUTINE CHORD      74/74   OPT=1                      FTN 4.2+P398       05/01/78  13.30.55.

      SUBROUTINE CHORD(FMACH,VMC,COD,RHO,PI,G,D1,Z,ZP,ALPHA,ALPHAP,VM,W, CHORD    2
     +CX,CY,CZ)                                                      CHORD    3
      DIMENSION VMC(8),COD(8),Z(16),ZP(16)                           CHORD    4
      VMM=VM/FMACH                                                   CHORD    5
      CC=TLU2(VMM,VMC,COD)                                           CHORD    6
      DIA=D1/12.                                                     CHORD    7
      CORD=-(.5*RHO*CC*PI*G/W)*(DIA*DIA)/4.*(VM*VM)                  CHORD    8
      COSGPA=COS(Z(9)+ALPHA)                                         CHORD    9
      SINGPA=SIN(Z(9)+ALPHA)                                         CHORD   10
      TANGAPP=TAN(ZP(9)+ALPHAP)                                      MAR19   57
      DEN=SQRT(1.+COSGPA**2*TANGAPP**2)                              MAR19   58
      CX=(CORD*COSGPA)/DEN                                           CHORD   13
      CY=(CORD*SINGPA)/DEN                                           CHORD   14
      CZ=(CORD*COSGPA*TANGAPP)/DEN                                   MAR20   17
      RETURN                                                         CHORD   15
      END                                                            CHORD   17
```

Commanded Acceleration

Commanded Force

The commanded force results from the horizontal and vertical guidance commands generated in the missile. The commanded force is based on a proportional navigation system with navigation parameter. It is assumed that the missile has its control surfaces or flippers biased to account for any lift force and to fly straight and level during normal flight conditions. The commanded acceleration in the vertical and horizontal plane is given by the expression:

$$X_n = -\frac{\Lambda \dot{r}_{xy} \dot{\psi}_{new}}{\cos(\gamma - \psi_{new} + \alpha)} \qquad \text{(horizontal)}$$

$$X_n' = \frac{\Lambda \dot{r}_{XZ} \dot{\psi}'_{new}}{\cos(\gamma' - \psi'_{new} + \alpha')} + G*BIAS \qquad \text{(vertical)}$$

where,

Λ = navigation parameter

$\dot{\psi}_{new}, \dot{\psi}'_{new}$ = gyro rates - horizontal and vertical

ψ_{new}, ψ'_{new} = gyro position - horizontal and vertical

α, α' = angle of attack - horizontal and vertical

γ, γ' = missile body angles - horizontal and vertical

$\dot{r}_{XY}, \dot{r}_{XZ}$ = aimpoint range rates - horizontal and vertical

BIAS = gravity bias term

The derivation of this guidance law is given in the following subsection.

To account for gravity bias missile systems, i.e., missiles which have their horizontal control surfaces biased in such a manner as to effectively null out gravity, the vertical commanded acceleration includes a gravity bias term, G*BIAS.

BIAS is the variable which controls the amount of gravity bias the missile is to have.

To compute the commanded acceleration, it is necessary to determine $\dot{r}_{XY}$ and $\dot{r}_{XZ}$ in terms of $\dot{r}$ (the closing rate along the LOS). Table 3-7 shows this computation.

Figure 3-6 shows the commanded acceleration computation and also shows that commanded acceleration is aerodynamically and structurally limited.

The commanded force is limited aerodynamically to be less than the maximum lift force given in the same table to be $1/2*\bar{p}*C_{NMAX}*\pi*(D_M/2)^2*V^2$. C_{NMAX} is a function of mach number as incidated in the figure.

A limit is set in the missile's autopilot or guidance unit to prevent over-maneuvering against the target. This is a g-limit and is designed by the term A5 in the program.

The g-limit is a limit internal to the missile whereas the aerodynamic limit is an external limit. At any given time and for any given missile, only one or the other constraint will be dominant. These two limits represent constraints on commanded acceleration X(5), XP(5)) and are also the only state constraints in the program.

The missile does not respond instantaneously to the commanded force. There is a delay associated with time for target information to go through the signal processor and guidance unit and finally to reach the control surface actuators. This delay is also modeled as a one time constant delay. The resulting equations are:

$$X(5)_{new} = X(5)_{old} + \frac{\Delta}{T_S}(G1 - X(5)_{old})$$

$$XP(5)_{new} = XP(5)_{old} + \frac{\Delta}{T_S}(G1' - XP(5)_{old})$$

Table 3-7. Range rate vector components

From geometry,

$$\dot{r}_x = \dot{r}_{xy} * \cos(\psi) = \dot{r}_{xz} * \cos(\psi')$$

$$\dot{r}_{xz} = \dot{r}_{xy} * \cos(\psi)/\cos(\psi')$$

Now,

$$\dot{r}^2 = \dot{r}_x^{\,2} + \dot{r}_y^{\,2} + \dot{r}_z^{\,2}$$

$$= \dot{r}_{xy}^{\,2} * \cos^2(\psi) + \dot{r}_{xy}^{\,2} * \sin^2(\psi) + \dot{r}_{xz}^{\,2} * \sin^2(\psi')$$

$$= \dot{r}_{xy}^{\,2} + \dot{r}_{xz}^{\,2} * \sin^2(\psi')$$

$$= \dot{r}_{xy}^{\,2} + \dot{r}_{xy}^{\,2} * \sin^2(\psi') * \cos^2(\psi)/\cos^2(\psi')$$

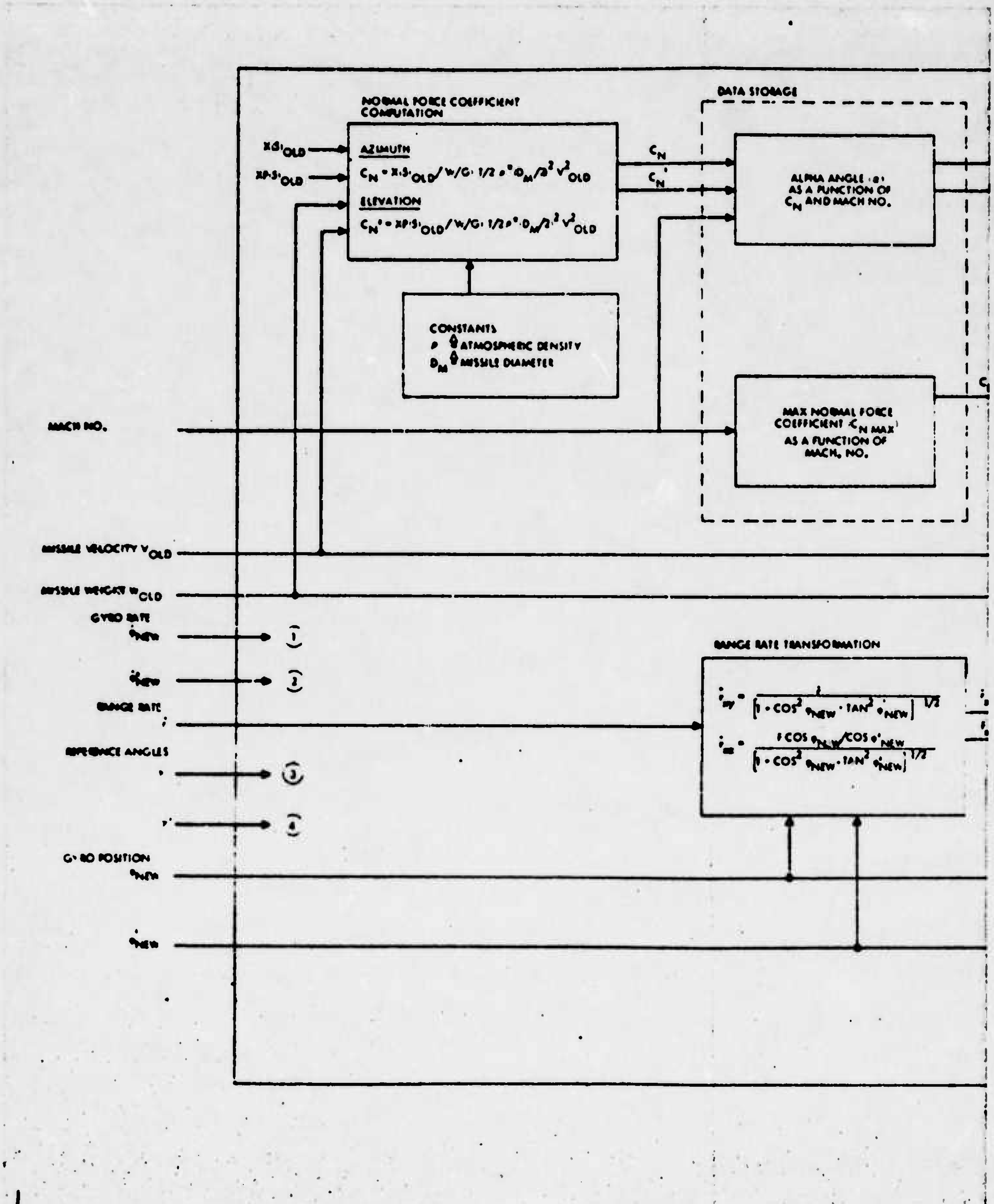
$$\dot{r}_{xy} = \frac{\dot{r}}{\left[1 + \cos^2(\psi) * \tan^2(\psi')\right]^{1/2}}$$

$$\dot{r}_{xz} = \frac{\dot{r} * \cos(\psi)/\cos(\psi')}{\left[1 + \cos^2(\psi) * \tan^2(\psi')\right]^{1/2}}$$

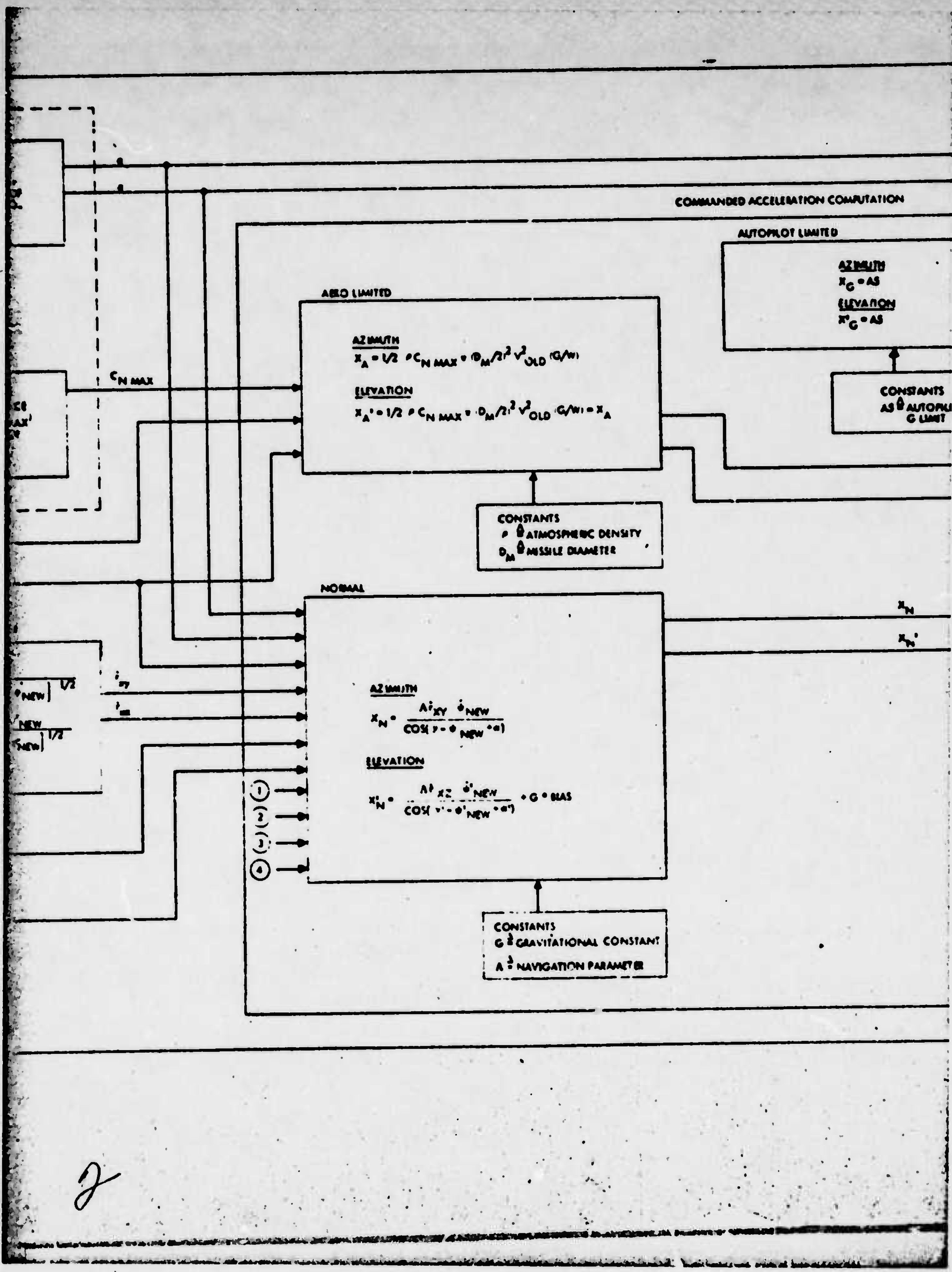
$$\dot{r}_x = \frac{\dot{r} * \cos(\psi)}{\left[1 + \cos^2(\psi) * \tan^2(\psi')\right]^{1/2}}$$

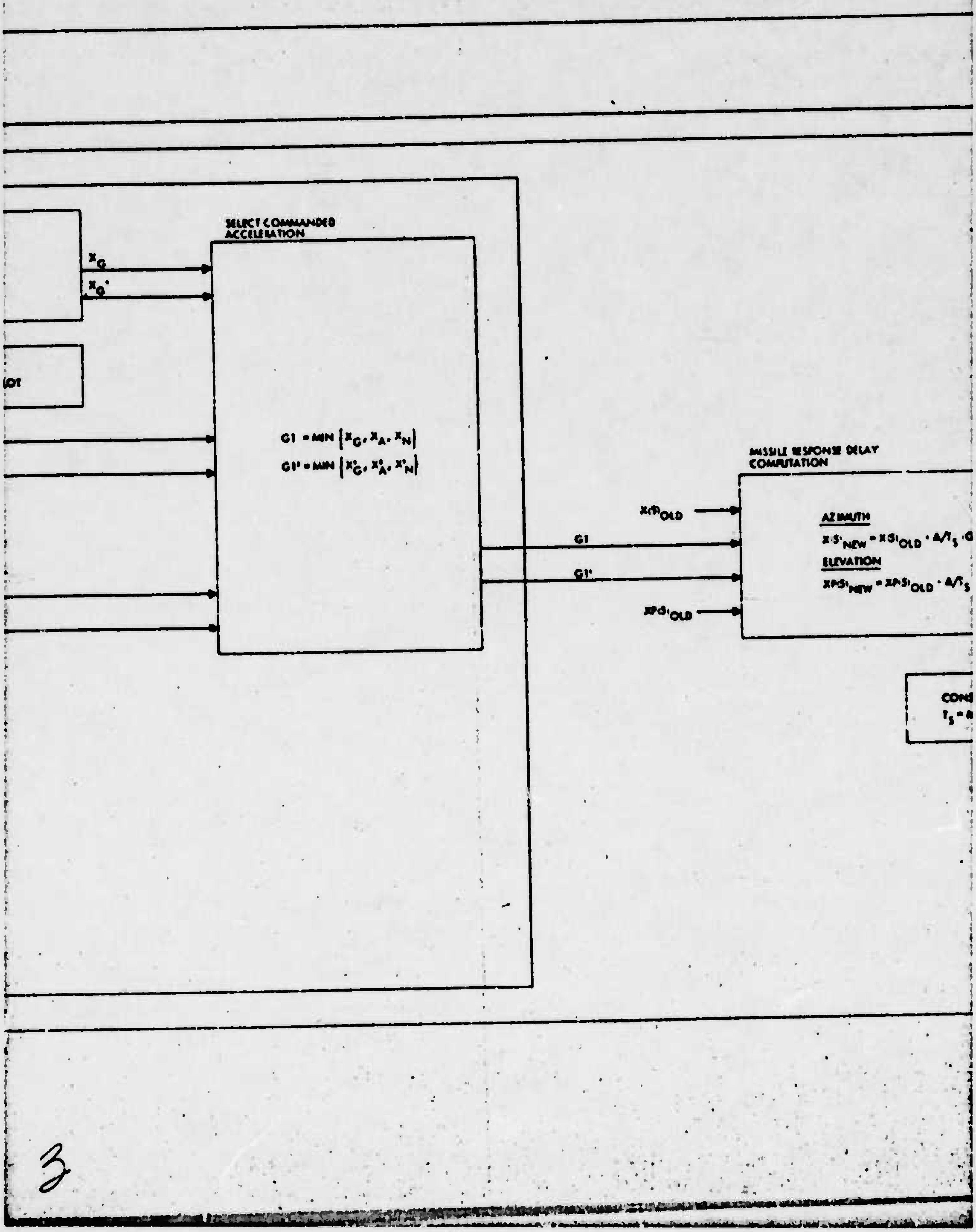
$$\dot{r}_y = \frac{\dot{r} * \sin(\psi)}{\left[1 + \cos^2(\psi) * \tan^2(\psi')\right]^{1/2}}$$

$$\dot{r}_z = \frac{\dot{r} * \cos(\psi) * \tan(\psi')}{\left[1 + \cos^2(\psi) * \tan^2(\psi')\right]^{1/2}}$$



PRECEDING PAGE BLANK NOT FILMED


COMMANDED ACCELERATION COMPUTATION
AUTOPILOT LIMITED
AZIMUTH
ELEVATION
CONSTANTS
AERO LIMITED
AZIMUTH
ELEVATION
CONSTANTS
ATMOSPHERIC DENSITY
MISSILE DIAMETER
NORMAL
AZIMUTH
ELEVATION
CONSTANTS
GRAVITATIONAL CONSTANT
NAVIGATION PARAMETER
1
2
3
4



SELECT COMMANDED ACCELERATION
G1 = MIN {XG, XA, XN}
G1' = MIN {X'G, X'A, X'N}
G1
G1'
MISSILE RESPONSE DELAY COMPUTATION
AZIMUTH
ELEVATION

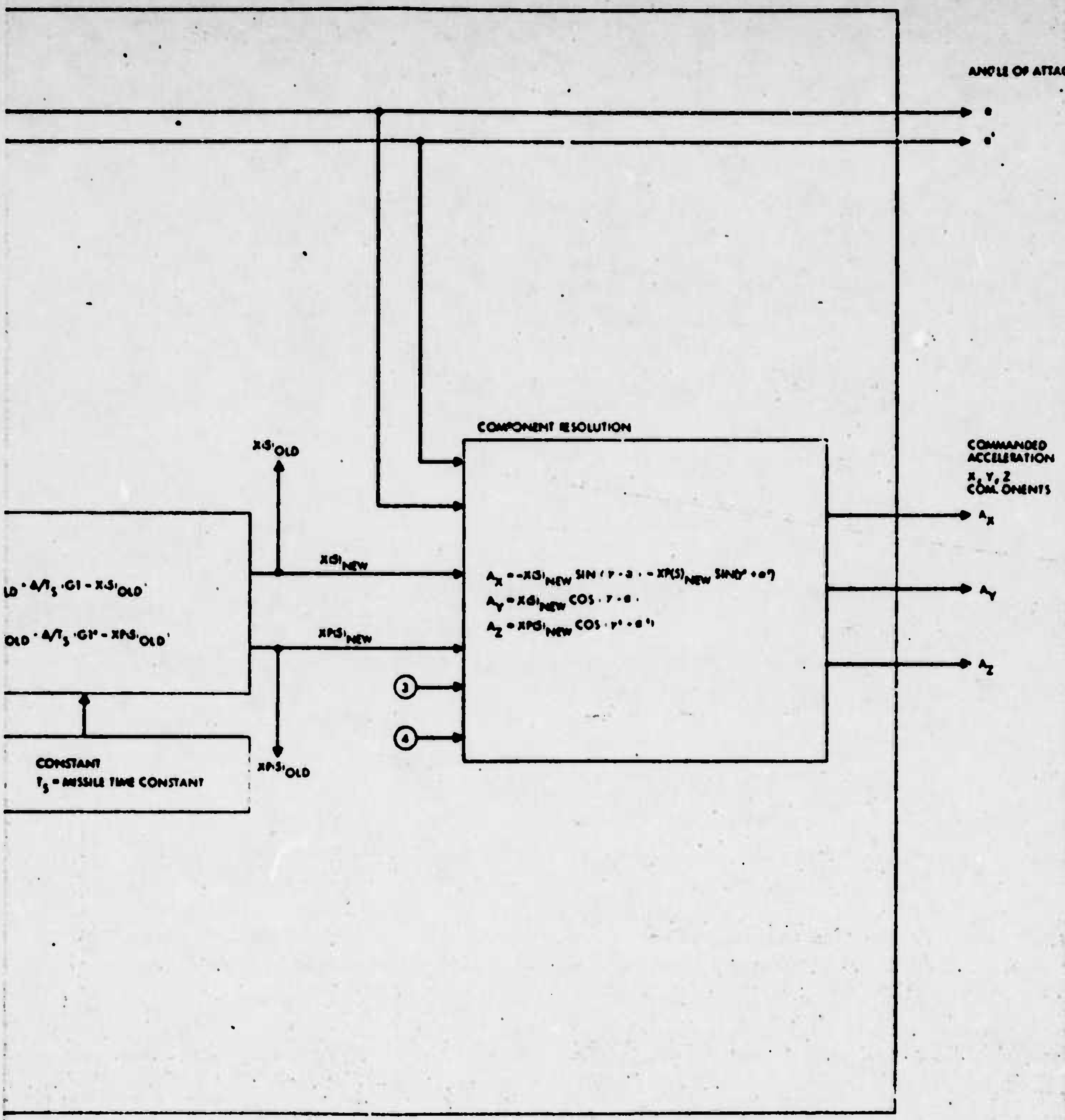


Figure 3-6. Commanded acceleration computations

where,

$G1$ = commanded acceleration required (horizontal plane)

$G1'$ = commanded acceleration required (vertical plane)

$X(5)$ = actual acceleration at the control surfaces

$XP(5)$ = actual acceleration at the control surfaces (vertical)

T_s = missile time constant

The commanded force is in a direction normal to the thrust vector. It's X-, Y-, and Z-vector components are:

$$A_X = -\sin(\gamma + \alpha) * X(5)_{new} - \sin(\gamma + \alpha) \times XP(5)_{new}$$

$$A_Y = \cos(\gamma + \alpha) * X(5)_{new}$$

$$A_Z = \cos(\gamma' + \alpha') * XP(5)_{new}$$

Table 3-8 contains a listing of this subroutine.

The missile angle of attack is also computed in this subroutine. One angle of attack (alpha angle) is computed for the missile in the vertical plane and one in the horizontal plane. Curves of alpha angle as a function of Mach No. and normal force coefficient are generally available from missile specifications. This data is tabularized into a two dimensional array of alpha as a function of mach number and normal force coefficient for use in the program. The procedure for determining alpha in each plane is to compute the normal force coefficient, based on the achieved accelerations, and the mach number and then do a table lookup for alpha.

Guidance Law

In summary, the guidance law implemented in infrared missile autopilots is generally of the form

$$A_T = K_g \dot{\psi}$$

Table 3-8. Commanded acceleration subroutine

```
SUBROUTINE COMMACC     74/74    OPT=1                          FTN 4.2+P380        05/01/75  13.30.51.

      SUBROUTINE COMMACC(XS,XPS,PI,G,W,D1,VM,RHO,VMC,CNA,ALPH,FMACH,ALPH COMMACCP    2
     +A,ALPHAP,AS,GKO,CNT,SINM,SIPM,SIDM,SIDPM,RDOT,GAM,GAMP,DELT,TS,DIA COMMACCP    3
     +S,AX,AY,AZ,TP,T,GL)                                                COMMACCP    4
      DIMENSION CNA(10),ALPH(10,8),VMC(8),CNT(8)                         COMMACCP    5
      RAD=PI/180.                                                        COMMACCP    6
      DIA=D1/12.                                                         COMMACCP    7
      TEMPA=(.5*RHO*PI*G/W)*(DIA*DIA)/4.*(VM*VM)                         COMMACCP    8
      CN=XS/TEMPA                                                        COMMACCP    9
      CNP=XPS/TEMPA                                                      COMMACCP   10
      VMM=VM/FMACH                                                       COMMACCP   11
      CNTEMP=ABS(CN)                                                     COMMACCP   12
      CNPTEMP=ABS(CNP)                                                   COMMACCP   13
      CALL INTOA6(VMC,8,CNA,10,ALPH,VMM,CNTEMP,ALPHA)                    COMMACCP   14
      CALL INTOC6(VMC,8,CNA,10,ALPH,VMM,CNPTEMP,ALPHAP)                  COMMACCP   15
      ALPHA=SIGN(ALPHA,CN)                                               COMMACCP   16
      ALPHAP=SIGN(ALPHAP,CNP)                                            COMMACCP   17
      ALPHA=ALPHA*RAD                                                    COMMACCP   18
      ALPHAP=ALPHAP*RAD                                                  COMMACCP   19
      CNMAX=TLU2(VMM,VMC,CNT)                                            COMMACCP   20
      COSSIN=COS(SIMM)                                                   COMMACCP   21
      TANSIPM=TAN(SIPM)                                                  COMMACCP   22
      COSSIPM=COS(SIPM)                                                  COMMACCP   23
      TEMP=SQRT(1.+COSSIN*COSSIN*TANSIPM*TANSIPM)                        COMMACCP   24
      RODTL=-500.                                                        COMMACCP   25
      RODTK=AMIN1(RDOT,RODTL)                                            MARZA      16
      RDKY=RODTK/TEMP                                                    COMMACCP   27
      RDKZ=(RODTK*COSSIN/COSSIPM)/TEMP                                   COMMACCP   28
      XG=AS                                                              COMMACCP   29
      XGP=XG                                                             COMMACCP   30
      XA=CNMAX*TEMPA                                                     COMMACCP   31
      XAP=XA                                                             COMMACCP   32
      GSA=GAM-SINM+ALPHA                                                 COMMACCP   33
      GSAP=GAMP-SIPM+ALPHAP                                              COMMACCP   34
      IF(GL.EQ.0.) GO TO 5                                               COMMACCP   35
      CNREF=TLU2(1.5,VMC,CNT)                                            COMMACCP   36
      XN=(GKO*VM*SIDM)*CNMAX/CNREF                                       COMMACCP   37
      XNP=(GKO*VM*SIDPM)*CNMAX/CNREF+G*BIAS                              COMMACCP   38
      GO TO 6                                                            COMMACCP   39
    5 XN=-(GKO*RDKY*SIDM)/COS(GSA)                                       COMMACCP   40
      XNP=-(GKO*RDKZ*SIDPM)/COS(GSAP)+G*BIAS                             COMMACCP   41
    6 G1=AMIN1(XG,ABS(XN))                                               MARZC      15
      G1=SIGN(G1,XN)                                                     COMMACCP   43
      G1P=AMIN1(XGP,ABS(XNP))                                            MARZD      15
      G1P=SIGN(G1P,XNP)                                                  COMMACCP   45
      IF(T.GT.TP) GO TO 1                                                COMMACCP   46
      AX=0.                                                              COMMACCP   47
      AY=0.                                                              COMMACCP   48
      AZ=0.                                                              COMMACCP   49
      GO TO 2                                                            COMMACCP   50
    1 XS=XS+DELT/TS*(G1-XS)                                              COMMACCP   51
      IF(ABS(XS).LE.XA) GO TO 3                                          COMMACCP   52
      XS=SIGN(XA,XS)                                                     COMMACCP   53
    3 XPS=XPS+DELT/TS*(G1P-XPS)                                          COMMACCP   54
      IF(ABS(XPS).LE.XAP) GO TO 4                                        COMMACCP   55
      XPS=SIGN(XAP,XPS)                                                  COMMACCP   56
    4 GA=GAM+ALPHA                                                       COMMACCP   57
      GAP=GAMP+ALPHAP                                                    COMMACCP   58
      AX=-XS*SIN(GA)-XPS*SIN(GAP)                                        COMMACCP   59
      AY=XS*COS(GA)                                                      COMMACCP   60
      AZ=XPS*COS(GAP)                                                    COMMACCP   61
    2 RETURN                                                             COMMACCP   62
      END                                                                COMMACCP   63
```

where,

A_T = commanded acceleration normal to body axis

$\dot{\psi}$ = LOS rate

K_g = navigation gain

Because the navigation gain (K_g) varies considerably with tactical conditions (altitude, launch speed, target speed, etc.) it is difficult to decide upon a fixed value for this constant when working with one missile only. When many missiles are to be evaluated, as is the case of the missile, target, and flare simulation, this guidance law becomes too difficult to implement and can lead to a large variation in missile trajectory. In order to avoid this problem, the missile, target, and flare simulation uses an ideal guidance law given by

$$A_T = \frac{\Lambda r \ddot{\psi}}{\cos(\gamma - \psi + \alpha)}$$

where these parameters are defined in Table 1. It should be pointed out that this law is impossible to implement in IR missile hardware due to the fact that missile and target range rate information is required. However, it does enable the simulation to remain generic and to evaluate missile performance under ideal conditions.

The guidance law of the form $K_g\dot{\psi}$ can also be implemented in the simulation, particularly when statistical variations in missile trajectory are desired. This can be accomplished easily by varying the navigation gain over its range of values.

The commanded missile acceleration normal to the line-of-sight (LOS) for ideal proportional navigation is given by

$$A_N = \Lambda V_C \dot{\psi}$$

Figure 3-7 shows the reference geometry and Table 3-9 defines the glossary of terms.

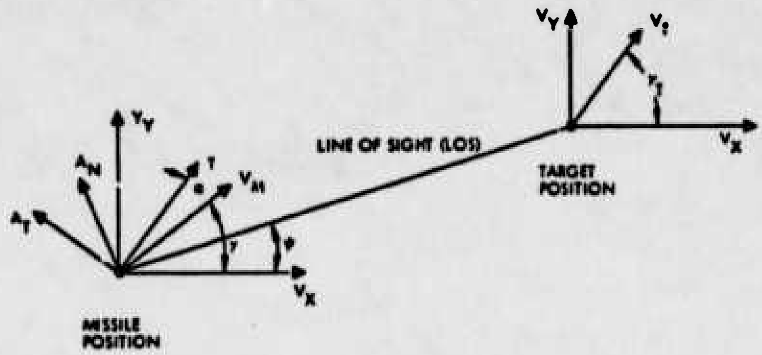


Figure 3-7. Missile and target reference geometry

In actual practice, the commanded acceleration is directed normally to the body axis and given by

$$A_T = \frac{\Lambda \dot{r} \dot{\psi}}{\cos \alpha \cos (\alpha - \psi)}$$

This relationship is the equation for implementing the proportional navigation law in the ideal case, and is presently the equation used in the missile, target, and flare simulation.

From the infrared missile hardware point of view, this equation is impossible to implement because range rate information is not available to the missile. The equation generally implemented in the autopilot of infrared missiles is

$$A_T = K_g \dot{\psi} \tag{3-8}$$

where

$$K_g = \frac{\Lambda V_c}{\cos (\gamma - \psi + \alpha)} \tag{3-9}$$

Table 3-9. Glossary of terms

V_x, V_y	=	inertial reference axes
V_T	=	target velocity
γ_T	=	angle between target velocity and inertial axis
V_m	=	missile velocity
γ	=	angle between missile velocity and inertial axis
ψ	=	angle between LOS and inertial axis
$\dot{\psi}$	=	LOS rate
α	=	angle of attack
T	=	missile body axis direction
$\dot{r}$	=	range rate along LOS
V_c	=	missile/target closing velocity
A_N	=	missile acceleration normal to LOS
A_T	=	missile acceleration normal to body axis
Λ	=	navigation parameter
K_g	=	navigation gain

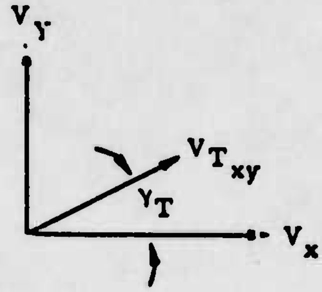
The parameter K_g is set before launch and is a function of altitude, target speed, and launch speed. This constant is normally chosen to be three times the maximum closing velocity achieved by the missile during flight. This ensures that the minimum value of the navigation parameter will be 3. The values during flight will be higher, ranging up to 5 or more at the end of flight. The constant value of 4 used in the simulation is an approximation of the average value during an actual flight. This is somewhat optimistic from the missile point of view.

3.2 TARGET EQUATIONS

The program allows for four different types of flight paths to be flown by the target aircraft. These include: straight and level, circular turn, straight acceleration, and turn and tangential acceleration.

All target motion is considered to take place only in the horizontal X-Y plane, and the equations governing each flight path are given in Tables 3-10 through 3-13.

Table 3-10. Straight and level flight



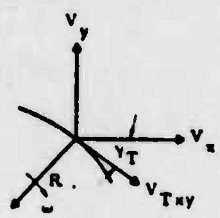
Equations

$$V_{Tx} = V_T \cos \gamma_T$$

$$V_{Ty} = V_T \sin \gamma_T$$

$$V_{Tz} = 0$$

Table 3-11. Circular turn



V_x, V_y, V_z	=	reference axes (V_z not shown)
R	=	turn radius
V_{Txy}	=	total velocity of aircraft in xy plane
γ_T	=	angle between aircraft velocity vector (xy plane only) and V_x reference axis
ω	=	rate of angular rotation of aircraft in the turn
V_{Tx}, V_{Ty}, V_{Tz}	=	components of aircraft velocity along reference axes
M_g	=	Integral number of g's that aircraft pulls in the turn
M_g	>	0 for left turn
M_g	<	0 for right turn
T_M	=	time maneuver begins
g	=	gravitational constant
T	=	time

Equations

$$M_g g = \frac{V_{T_{xy}}^2}{R}$$

$$R = \frac{V_{T_{xy}}^2}{M_g g}$$

$$\omega = \frac{V_{Txy}}{R} = \frac{M_g g}{V_{Txy}}$$

If $T \cdot T_M$, then

$$\gamma_T = \gamma_T$$

Otherwise,

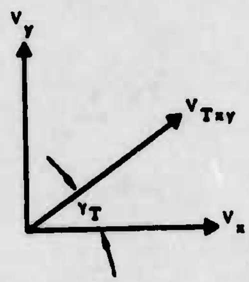
$$\gamma_T = \gamma_T(T_M) + \omega(T - T_M)$$

$$V_{Tx} = V_{Txy} \cos \gamma_T$$

$$V_{Ty} = V_{Txy} \sin \gamma_T$$

$$V_{Tz} = 0$$

Table 3-12. Straight acceleration



A_m = maximum forward acceleration of aircraft in g's

$V_m(h)$ = maximum aircraft speed at altitude (h)

V_T = total velocity of aircraft

Δ_T = size of integration step

Note. All previous definitions are applicable.

__Equations__

$$V_{Tx}\ (\text{new}) = V_{Tx}\ (\text{old}) + \dot{V}_x \Delta t$$

$$V_{Ty}\ (\text{new}) = V_{Ty}\ (\text{old}) + \dot{V}_y \Delta t$$

$$V_{Tz} = 0$$

If $T < T_M$ or $V_T \geq V_M(h)$, then

$$\dot{V}_x = \dot{V}_y = 0$$

Otherwise,

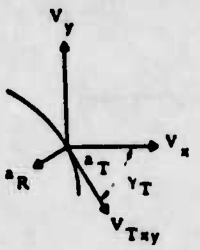
(Note: The target mode of operation is changed from military power to afterburner at $t = T_M$)

$$\dot{V}_x = A_m g \cos \gamma_T$$

$$\dot{V}_y = A_m g \sin \gamma_T$$

$$\dot{V}_z = 0$$

Table 3-13. Turn and acceleration



a_R = radial component of aircraft acceleration

a_T = tangential component of aircraft acceleration

Equations

$$V_{Txy}(new) = V_{Txy}(old) + \dot{V}_{Txy} g \Delta t$$

$$\gamma_T(new) = \gamma_T(old) + \dot{\gamma}_T \Delta t$$

$$V_{Tx} = V_{Txy} \cos \gamma_T$$

$$V_{Ty} = V_{Txy} \sin \gamma_T$$

$$V_{Tz} = 0$$

when

$$T < T_M$$

$$\dot{\gamma}_T = 0$$

$$\dot{V}_{Txy} = 0$$

$$T > T_M$$

(Note: The target mode of operation is changed from military power to afterburner at $t = T_M$.)

If

$$V_T \leq V_M(h)$$

then

$$\dot{V}_{Txy} = a_T = A_M g$$

$$\dot{\gamma}_T = \frac{a_R}{V_{Txy}} = \frac{M_g g}{V_{Txy}}$$

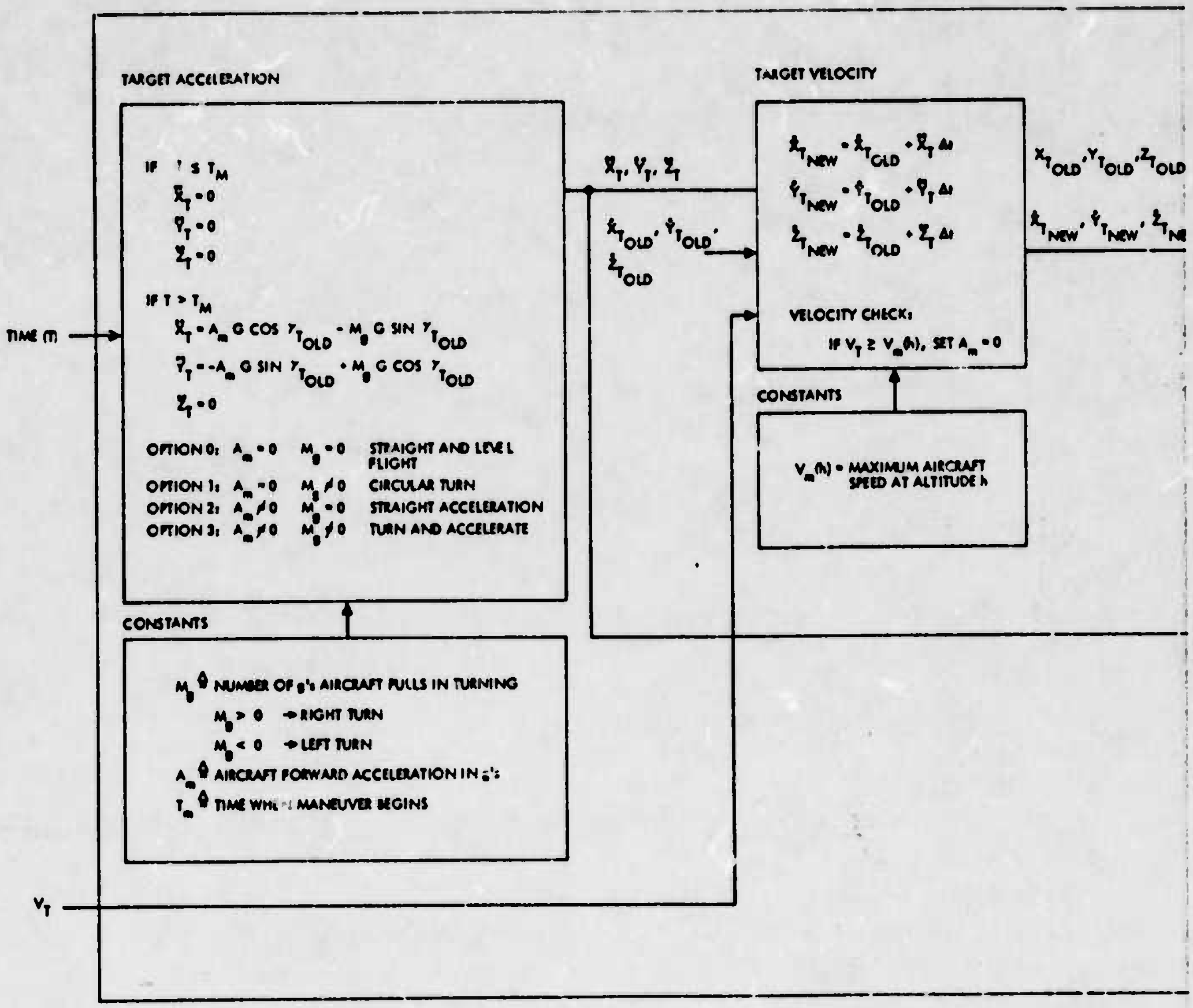
If not,

$$\dot{V}_{Txy} = 0$$

$$\dot{\gamma}_T = \frac{M_g g}{V_{Txy}}$$

The target altitude and velocity is an initial input into the program. The target heading relative to the inertial axes is determined in the program by missile aspect angle at launch. The additional aircraft data required to effect the maneuvers includes the maximum number of g's the aircraft can pull in a turn, the maximum aircraft acceleration forward, and the maximum speed it can obtain.

Figure 3-8 shows the basic equations used in the program to simulate target motion. Target maneuvers are controlled through NAMELIST input variables TM, AM, FMG. The time of initiation of maneuver is set by TM. Linear acceleration AM, and target turns FMG, are input as the magnitude of the acceleration(s) in terms of the number of g's. A positive sign for FMG will generate a right turn and a negative sign a left turn. Table 3-14 contains a listing of this subroutine.


TARGET ACCELERATION
IF T ≤ T_M
Ẍ_T = 0
Ÿ_T = 0
Z̈_T = 0
IF T > T_M
Ẍ_T = A_m G COS γ_TOLD − M_g G SIN γ_TOLD
Ÿ_T = −A_m G SIN γ_TOLD + M_g G COS γ_TOLD
Z̈_T = 0
OPTION 0: A_m = 0 M_g = 0 STRAIGHT AND LEVEL FLIGHT
OPTION 1: A_m = 0 M_g ≠ 0 CIRCULAR TURN
OPTION 2: A_m ≠ 0 M_g = 0 STRAIGHT ACCELERATION
OPTION 3: A_m ≠ 0 M_g ≠ 0 TURN AND ACCELERATE
TIME (T)
CONSTANTS
M_g ≜ NUMBER OF g's AIRCRAFT PULLS IN TURNING
M_g > 0 → RIGHT TURN
M_g < 0 → LEFT TURN
A_m ≜ AIRCRAFT FORWARD ACCELERATION IN g's
T_m ≜ TIME WHEN MANEUVER BEGINS
V_T
Ẍ_T, Ÿ_T, Z̈_T
Ẋ_TOLD, Ẏ_TOLD, Ż_TOLD
TARGET VELOCITY
Ẋ_TNEW = Ẋ_TOLD + Ẍ_T Δt
Ẏ_TNEW = Ẏ_TOLD + Ÿ_T Δt
Ż_TNEW = Ż_TOLD + Z̈_T Δt
VELOCITY CHECK:
IF V_T ≥ V_m(h), SET A_m = 0
CONSTANTS
V_m(h) = MAXIMUM AIRCRAFT SPEED AT ALTITUDE h
X_TOLD, Y_TOLD, Z_TOLD
Ẋ_TNEW, Ẏ_TNEW, Ż_TNE

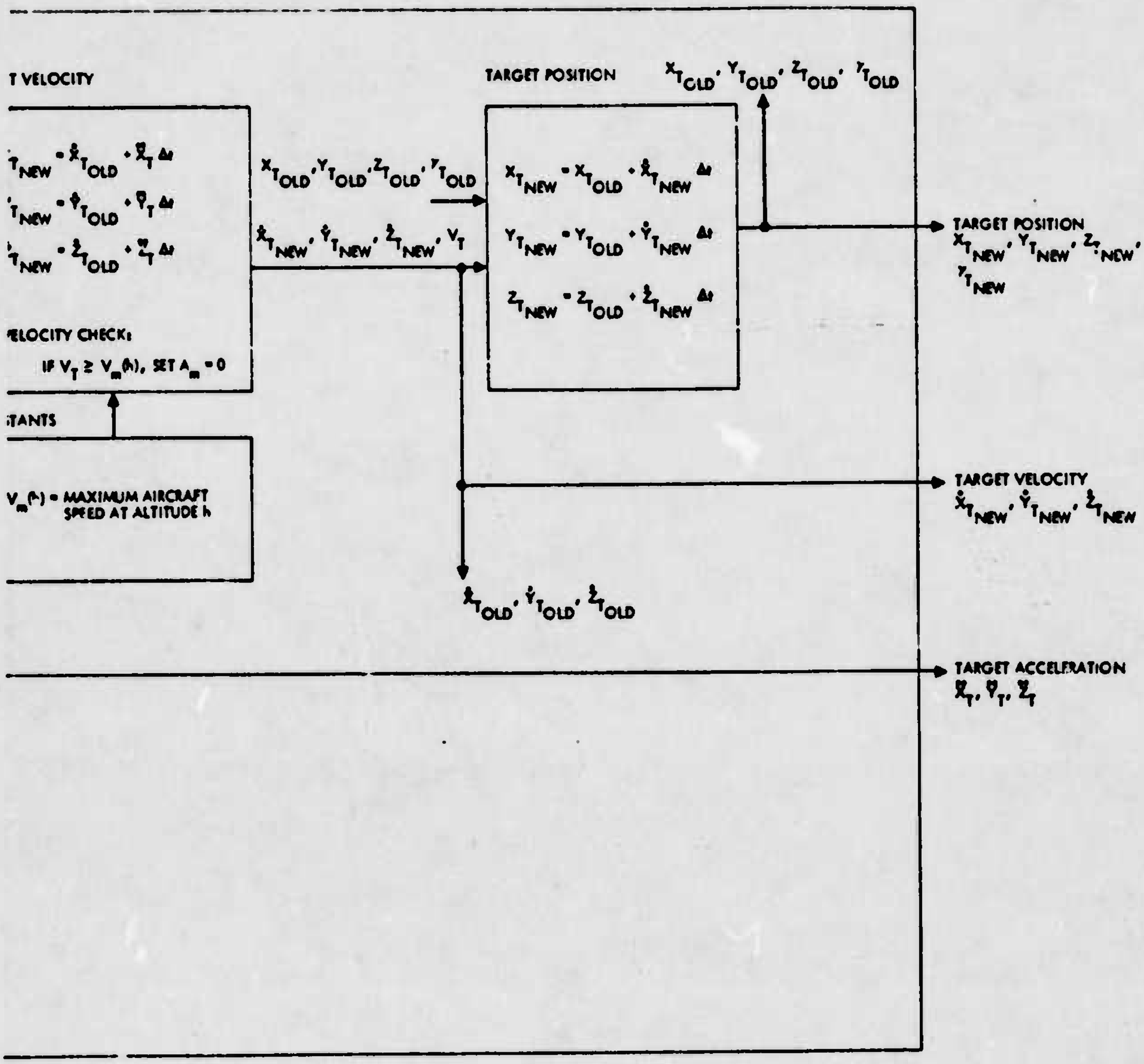


Figure 3-8. Target dynamics

Table 3-14. Target motion subroutine

```
SUBROUTINE TGDYN      74/74   OPT=1                    FTN 4.2+P380       09/01/75  13.31.09.

          SUBROUTINE TGDYN(T,TM,DELT,AM,G,FMG,VM,VT,X6,X8,XP8,X7,X9,XP9,X   MAR19      69
         +70,X90,XP90,GAMT)                                                 TGTDYNAM    3
          IF(T.GT.TM) GO TO 1                                               TGTDYNAM    4
          X70=0.                                                            TGTDYNAM    5
          X90=0.                                                            TGTDYNAM    6
          XP90=0.                                                           TGTDYNAM    7
          GO TO 2                                                           TGTDYNAM    8
        1 X70=AM*G*COS(GAMT)-FMG*G*SIN(GAMT)                                TGTDYNAM    9
          X90=-AM*G*SIN(GAMT)+FMG*G*COS(GAMT)                               TGTDYNAM   10
          XP90=0.                                                           TGTDYNAM   11
        2 X7=X7+X70*DELT                                                    TGTDYNAM   12
          X9=X9+X90*DELT                                                    TGTDYNAM   13
          XP9=XP9+XP90*DELT                                                 TGTDYNAM   14
          IF(VT.GE.VM) AM=0.                                                TGTDYNAM   15
          X6=X6+X7*DELT                                                     TGTDYNAM   16
          X8=X8+X9*DELT                                                     TGTDYNAM   17
          XP8=XP8+XP9*DELT                                                  TGTDYNAM   18
          RETURN                                                            TGTDYNAM   19
          END                                                               TGTDYNAM   20
```

3.3 FLARE CONTROL AND DYNAMICS

Figure 3-9 shows a block diagram of the major computations performed in this section. Basically, the flare control portion of this problem is responsible for determining the flare status (not dispensed, burning, or extinguished), the times at which flares are ejected, and the initial velocity and position of the flares at ejection. The flare dynamics updates the flare trajectory for those flares which have been dispensed and are still burning.

Flare Control

The flare status as indicated in Figure 3-10 is controlled by the variable NM(k), with NM(k) = 1 indicating that the flare has not yet been dispensed, NM(k) = 2 indicating that the flare has been dispensed and is still burning, and NM(k) = 3 indicating that the flare has been extinguished. The flare initial deployment strategy is also under the control of this portion of the program. Figure 3-10 shows that flares can be dispensed as a function of (1) time after missile launch, (2) missile and target range, (3) time to go, and (4) missile and target range rate.

Once a flare has been deployed, the flare's initial velocity is computed based on the target velocity plus the dispenser ejection velocity. This includes the ejection direction. The flare's position is computed based on target position plus dispenser location relative to engine(s) position. A listing of this subroutine is contained in Table 3-15.

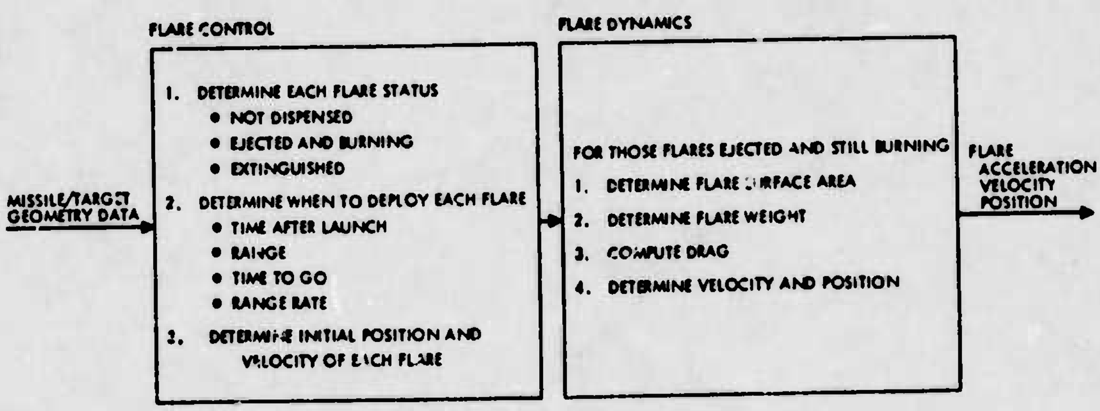
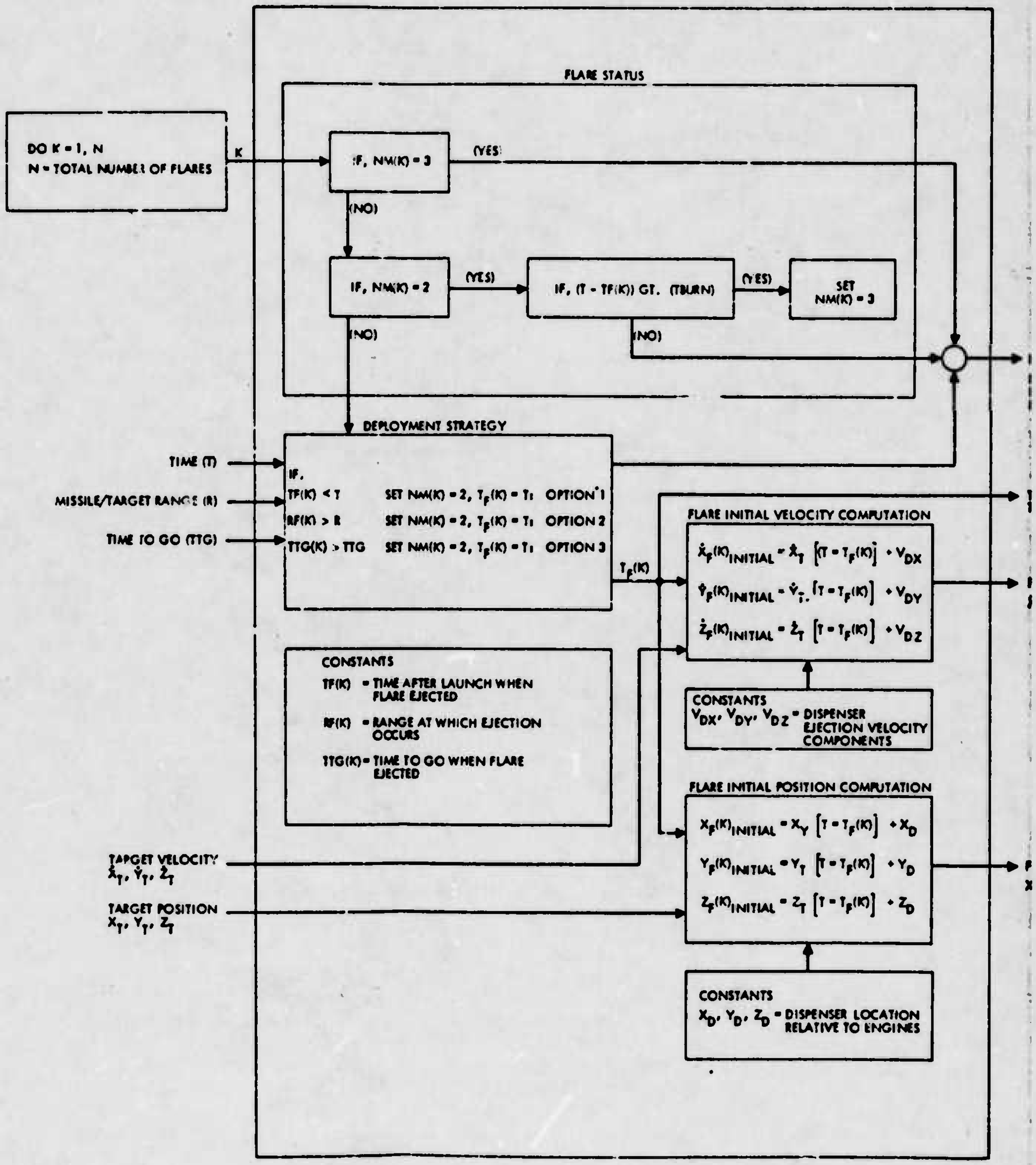


Figure 3-9. Flare control and dynamics overview



FLARE STATUS
DO K = 1, N
N = TOTAL NUMBER OF FLARES
K
IF, NM(K) = 3
(YES)
(NO)
IF, NM(K) = 2
(YES)
IF, (T - TF(K)) GT. (TBURN)
(YES)
SET NM(K) = 3
(NO)
(NO)
DEPLOYMENT STRATEGY
TIME (T)
MISSILE/TARGET RANGE (R)
TIME TO GO (TTG)
IF,
TF(K) < T SET NM(K) = 2, T_F(K) = T OPTION 1
RF(K) > R SET NM(K) = 2, T_F(K) = T OPTION 2
TTG(K) > TTG SET NM(K) = 2, T_F(K) = T OPTION 3
T_F(K)
CONSTANTS
TF(K) = TIME AFTER LAUNCH WHEN FLARE EJECTED
RF(K) = RANGE AT WHICH EJECTION OCCURS
TTG(K) = TIME TO GO WHEN FLARE EJECTED
FLARE INITIAL VELOCITY COMPUTATION
CONSTANTS
V_DX, V_DY, V_DZ = DISPENSER EJECTION VELOCITY COMPONENTS
FLARE INITIAL POSITION COMPUTATION
TARGET VELOCITY
TARGET POSITION
CONSTANTS
X_D, Y_D, Z_D = DISPENSER LOCATION RELATIVE TO ENGINES

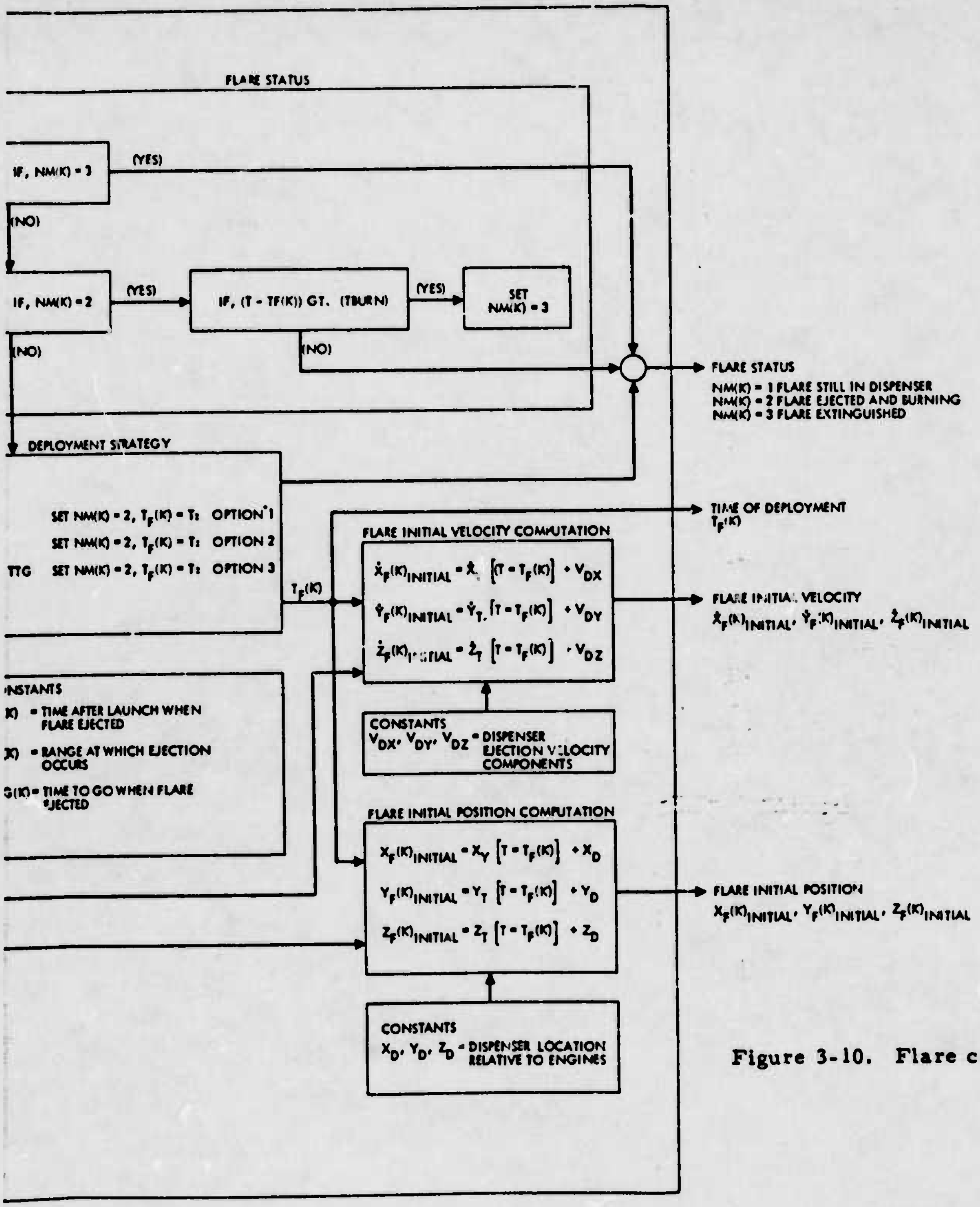


Figure 3-10. Flare control

Table 3-15. Flare control subroutine

```
SUBROUTINE FLRCTRL     74/74    OPT=1                       FTN 4.2+P387        05/31/75  17.38.56.

          SUBROUTINE FLRCTRL(NM,N,IGT,T,TF,TTG,TTGF,SSW2,XF,XFP,X,XP,TBURN      MAR19      58
         +,VXF,VYF,VZF,XO,YO,ZO,REL,RF,RDF,IDPLOY,M,PI,Z)                      FLRCONTR    3
          DIMENSION NM(20),IGT(20,2),TF(20),SF(20),WF(20),XF(10,20)            FLRCONTR    4
          DIMENSION XFP(10,20),X(10),XP(10),VXF(20),VYF(20),VZF(20),REL(12)    FLRCONTR    5
          DIMENSION RF(20),RDF(20),TTGF(20),Z(14)                              FLRCONTR    6
          IF(N.EQ.21) GO TO 9                                                  FLRCONTR    7
          GO TO (2,3,4), IDPLOY                                                FLRCONTR    8
        2 IF(TF(N).GT.T) GO TO 13                                              FLRCONTR    9
          GO TO 9                                                              FLRCONTR   10
        3 IF(RF(N).LE.REL(10)) GO TO 10                                        FLRCONTR   11
          GO TO 12                                                             FLRCONTR   12
        4 IF(N.GT.1) GO TO 22                                                  FLRCONTR   13
          IF(TTG) 10,10,22                                                     FLRCONTR   14
       22 IF(TTGF(N).LE.TTG) GO TO 13                                          FLRCONTR   15
       12 TF(N)=T                                                              FLRCONTR   16
        9 N=N+1                                                                FLRCONTR   17
          M=N-1                                                                FLRCONTR   18
          GT=Z(7)+SIGN(PI/2.,VYF(M))                                           FLRCONTR   19
          VXF(M)=ABS(VYF(M))*COS(GT)                                           FLRCONTR   20
          VYF(M)=ABS(VYF(M))*SIN(GT)                                           FLRCONTR   21
          GO TO 20                                                             FLRCONTR   22
       10 IF(N.EQ.1) GO TO 21                                                  FLRCONTR   23
       20 DO 1 K=1,M                                                           FLRCONTR   24
          GO TO (7,6,1), NM(K)                                                 FLRCONTR   25
        6 IF((T-TF(K)).LT.(TBURN-.1)) GO TO 1                                  FLRCONTR   26
          NM(K)=3                                                              FLRCONTR   27
          IGT(K,2)=1                                                           FLRCONTR   28
          IF(SSW2.EQ.0.) GO TO 1                                               FLRCONTR   29
          IF(XFP(8,K).LE.0.) GO TO 1                                           FLRCONTR   30
          WRITE(6,14) K,T                                                      FLRCONTR   31
       14 FORMAT(/,6X,14HFLARE NUMBER  ,I4,2X,22HEXTINGUISHED AT TIME  ,F6.3   FLRCONTR   32
         +)                                                                    FLRCONTR   33
          GO TO 1                                                              FLRCONTR   34
        7 NM(K)=2                                                              FLRCONTR   35
          IGT(K,1)=1                                                           FLRCONTR   36
          IGT(K,2)=0                                                           FLRCONTR   37
          XF(7,K)=X(7)+VXF(K)                                                  FLRCONTR   38
          XF(9,K)=X(9)+VYF(K)                                                  FLRCONTR   39
          XFP(9,K)=XP(9)+VZF(K)                                                FLRCONTR   40
          XFP(6,K)=X(5)                                                        FLRCONTR   41
          XFP(7,K)=X(7)+VXF(K)                                                 FLRCONTR   42
          XF(6,K)=X(6)+XO                                                      FLRCONTR   43
          XF(8,K)=X(8)+YO                                                      FLRCONTR   44
          XFP(6,K)=XP(8)+ZO                                                    FLRCONTR   45
          IF(SSW2.EQ.0.) GO TO 1                                               FLRCONTR   46
          WRITE(6,670) K,T                                                     FLRCONTR   47
      670 FORMAT(/,6X,14HFLARE NUMBER  ,I4,2X,17HEJECTED AT TIME  ,F6.3)       FLRCONTR   48
          WRITE(6,677) K                                                       FLRCONTR   49
      677 FORMAT(6X,7HNUMBER ,I4,1X,14HFLARE POSITION)                         FLRCONTR   50
          WRITE(6,676) XF(6,K),XF(8,K),XFP(8,K)                                FLRCONTR   51
          WRITE(6,672) K                                                       FLRCONTR   52
      672 FORMAT(6X,7HNUMBER ,I4,1X,22HFLARE INITIAL VELOCITY)                 FLRCONTR   53
          WRITE(6,675) XF(7,K),XF(9,K),XFP(9,K)                                FLRCONTR   54
      676 FORMAT(2X,3HXF=,F10.2,2X,3HYF=,F13.4,2X,3HZF=,F13.4)                 FLRCONTR   55
      675 FORMAT(2X,4HVXF=,F9.4,2X,4HVYF=,F9.4,2X,4HVZF=,F5.4)                 FLRCONTR   56
          WRITE(6,13) K,T                                                      FLRCONTR   57
       13 FORMAT(/,6X,14HFLARE NUMBER  ,I4,2X,17HIGNITED AT TIME  ,F6.3)       FLRCONTR   58
        1 CONTINUE                                                             FLRCONTR   59
       21 RETURN                                                               FLRCONTR   60
          END                                                                  FLRCONTR   61
```

Flare Dynamics

In any missile, target, and flare simulation, it is vital that the trajectory time histories of the flares under consideration be known. Table 3-16 shows the two forces acting on the flare, drag and gravity, as well as the magnitude of each. The following equation governing the trajectory time history can then be written:

$$\dot{\vec{V}}_F = -\left(\frac{1}{2}\ pA_s C_D (G/W_F) V_F \vec{V}_F + G\right)$$

where

p = atmospheric density

C_D = drag coefficient

A_s = cross-sectional area

W_F = weight of flare

V_F = velocity of flare

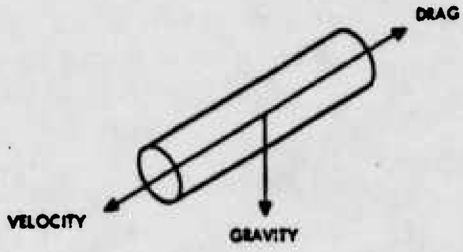
G = gravitational constant

The major difficulties in solving this equation are that both the cross-sectional area and the weight are time dependent functions, while the cross-sectional area is spatially dependent if the flare is tumbling (as it normally does).

To deal with the spatial orientation problem, it is assumed that a cross-sectional area averaged over all surfaces will account for this orientation of the flare as it tumbles. Then, the cross-sectional area (SFB) presented to the wind stream will be the average of longitudinal (A2) and axial cross sectional area or

$$SFB = \frac{A1 + A2}{2}$$

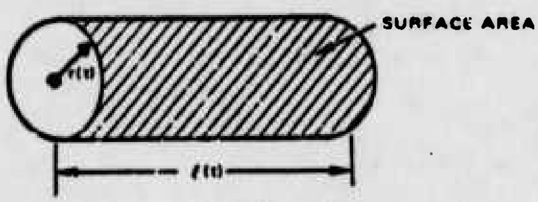
Table 3-16. Forces acting on the Kth flare



FORCE	MAGNITUDE	
	HORIZONTAL PLANE	VERTICAL PLANE
GRAVITY	0	$W_F(K)$
DRAG	$1/2 \cdot \rho \cdot C_D \cdot S_F(K) \cdot V_F^2(K)$	

$W_F(K)$ = WEIGHT OF Kth FLARE
$V_F(K)$ = VELOCITY OF Kth FLARE
$S_F(K)$ = SURFACE AREA OF Kth FL[illegible] NORMAL TO VELOCITY VECTOR
C_D = DRAG COEFFICIENT

Now consider cylindrical flares such as the MK-46, MK-49, and ALA-17. Assume that the linear burn rate is constant, and that the area of the cylinder walls is much greater than that of the ends of the cylinder. This is a good assumption in view of the shapes of the flares being investigated. However, if the area of the ends of the cylinders are neglected and the definitions in the sketch below are used, the perimeter surface area can be expressed as



$A(t) = 2\pi r(t)\, \ell(t)$

On the assumption that the linear burn rate is constant and that the radius of the cylinder is much shorter than the length of the cylinder, it follows that dr/dt = constant. The solution to this equation is

$$r(t) = r_o(1 - t/t_B)$$

where

r_o = initial radius of flare

t_B = burn time

If a similar expression for $\ell(t)$ is assumed, which is not necessarily the exact solution for this term but one that should result in an adequate approximation, the expression for the $\ell(t)$ becomes

$$\ell(t) = \ell_o (1 - t/t_B)$$

For cylindrical type flares,

$$A1(t) = \pi r^2(t)$$

and

$$A2(t) = 2r(t)\ell(t)\pi$$

The average cross-sectional area is

$$SFB(t) = \frac{A1(t) + A2(t)}{2} = \frac{\pi r^2(t)}{2} + \frac{2r(t)\,\ell(t)\,\pi}{2}$$

$$= \left[\frac{\pi r_o^2}{2} + r_o \ell_o \pi\right](1 - t/t_B)^2$$

The weight of the flare is generally specified in terms of both a total weight (WFO) and a grain weight (WG). The grain weight is changing as a function of time. If it is assumed for simplicity that the grain weight changes with a similar expression as the cross-sectional area, then the expression for the flare weight is as shown in Table 3-17. This table shows the basic computations the simulation uses to determine the flare trajectory. Figure 3-11 shows a flow diagram of the computations of this subroutine including component resolution of all acceleration, velocity, and position components. Table 3-18 contains a listing of this subroutine.

Table 3-17. Drag

Drag accelerations: $1/2\ \rho\ C_D G S_F(k) V^2 / W_V(k)$

1. 2.

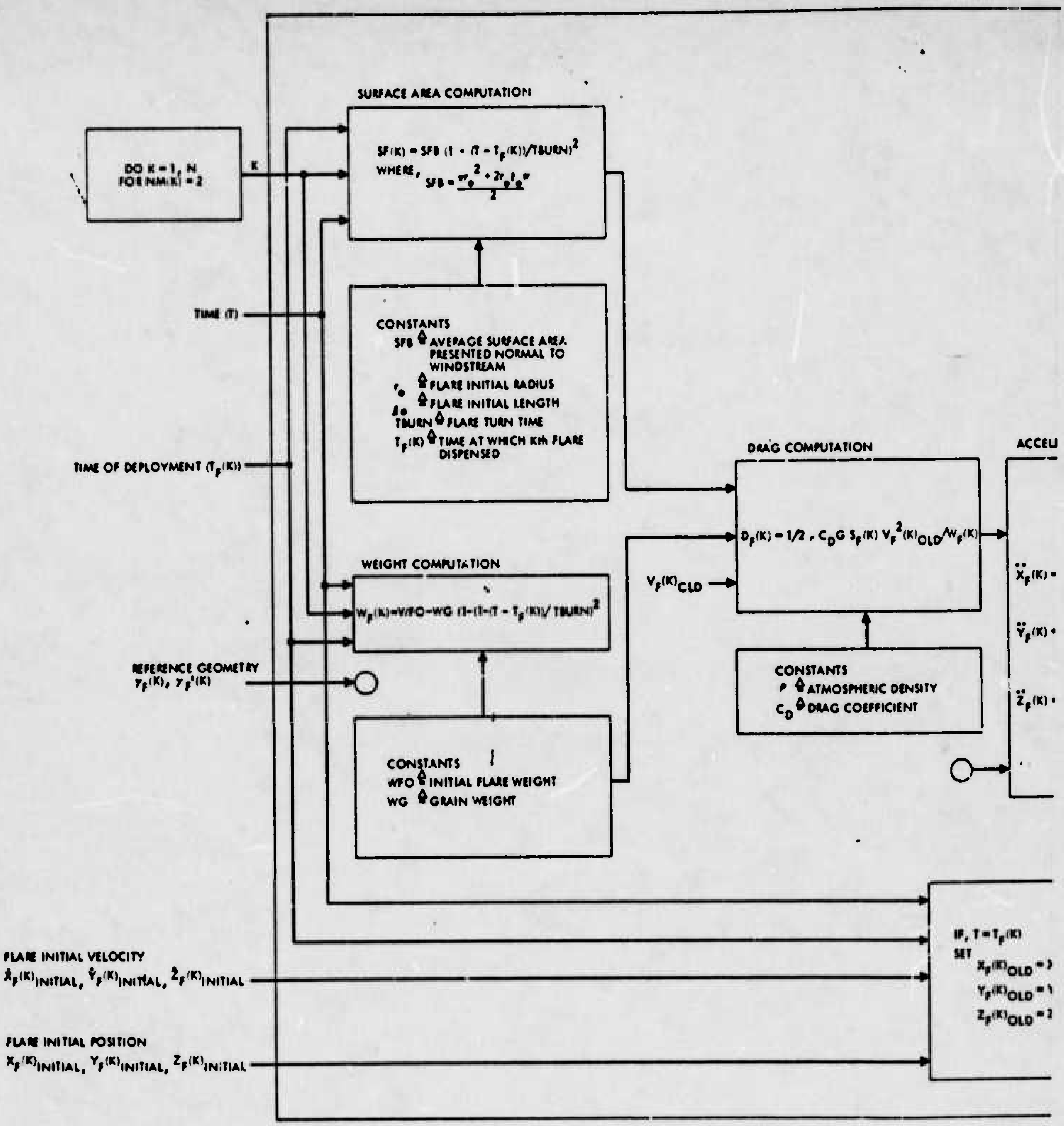
1. $A1 = \pi r_o^2$

$A2 = 2 r_o \ell_o \pi$

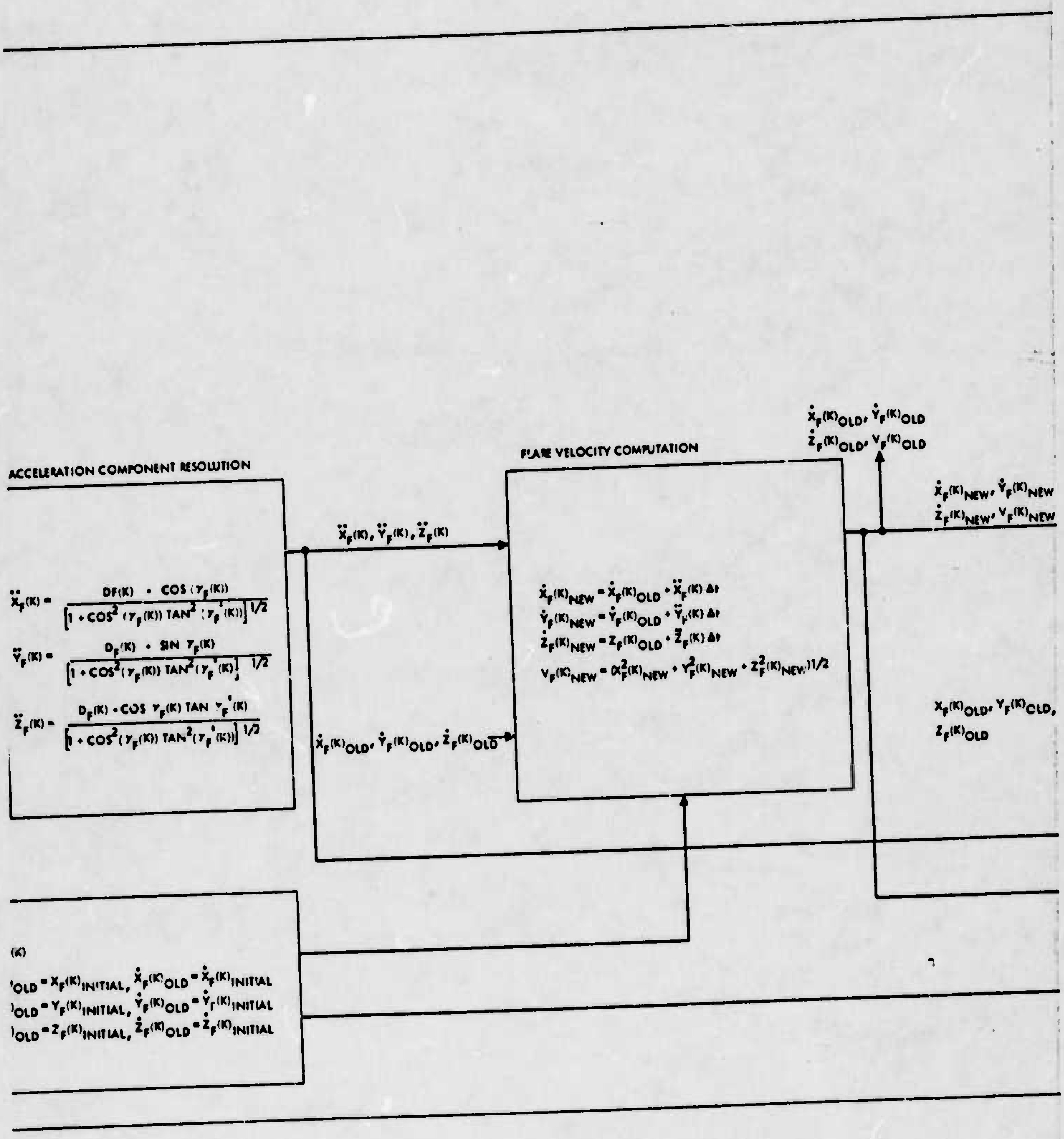
$SFB = (A1 + A2)/2$

$SF(k) = SFB\ (1 - (T - T_F(k)/TBURN)^2$

2. $W_F(k) = WFO - WG\left(1 - (1 - (T - T_F(k))/TBURN)^2\right)$



SURFACE AREA COMPUTATION
SF(K) = SFB (1 - (T - T_F(K))/TBURN)^2
WHERE, SFB = (πr_o^2 + 2r_o ℓ_o π)/2
DO K = 1, N
FOR NM(K) = 2
K
TIME (T)
CONSTANTS
SFB ≙ AVERAGE SURFACE AREA PRESENTED NORMAL TO WINDSTREAM
r_o ≙ FLARE INITIAL RADIUS
ℓ_o ≙ FLARE INITIAL LENGTH
TBURN ≙ FLARE TURN TIME
T_F(K) ≙ TIME AT WHICH Kth FLARE DISPENSED
TIME OF DEPLOYMENT (T_F(K))
DRAG COMPUTATION
D_F(K) = 1/2 ρ C_D G S_F(K) V_F^2(K)_OLD/W_F(K)
ACCEL
WEIGHT COMPUTATION
W_F(K) = WFO - WG (1 - (1 - (T - T_F(K))/TBURN)^2
V_F(K)_OLD
Ẍ_F(K) =
Ÿ_F(K) =
Z̈_F(K) =
REFERENCE GEOMETRY
γ_F(K), γ_F'(K)
CONSTANTS
ρ ≙ ATMOSPHERIC DENSITY
C_D ≙ DRAG COEFFICIENT
CONSTANTS
WFO ≙ INITIAL FLARE WEIGHT
WG ≙ GRAIN WEIGHT
IF, T = T_F(K)
SET
X_F(K)_OLD =
Y_F(K)_OLD =
Z_F(K)_OLD =
FLARE INITIAL VELOCITY
Ẋ_F(K)_INITIAL, Ẏ_F(K)_INITIAL, Ż_F(K)_INITIAL
FLARE INITIAL POSITION
X_F(K)_INITIAL, Y_F(K)_INITIAL, Z_F(K)_INITIAL


ACCELERATION COMPONENT RESOLUTION
$\ddot{X}_F(K) = \frac{DF(K) \cdot COS(\gamma_F(K))}{[1 + COS^2(\gamma_F(K)) TAN^2(\gamma_F'(K))]^{1/2}}$
$\ddot{Y}_F(K) = \frac{D_F(K) \cdot SIN\ \gamma_F(K)}{[1 + COS^2(\gamma_F(K)) TAN^2(\gamma_F'(K))]^{1/2}}$
$\ddot{Z}_F(K) = \frac{D_F(K) \cdot COS\ \gamma_F(K)\ TAN\ \gamma_F'(K)}{[1 + COS^2(\gamma_F(K)) TAN^2(\gamma_F'(K))]^{1/2}}$
$\ddot{X}_F(K), \ddot{Y}_F(K), \ddot{Z}_F(K)$
$\dot{X}_F(K)_{OLD}, \dot{Y}_F(K)_{OLD}, \dot{Z}_F(K)_{OLD}$
FLARE VELOCITY COMPUTATION
$\dot{X}_F(K)_{NEW} = \dot{X}_F(K)_{OLD} + \ddot{X}_F(K)\,\Delta t$
$\dot{Y}_F(K)_{NEW} = \dot{Y}_F(K)_{OLD} + \ddot{Y}_F(K)\,\Delta t$
$\dot{Z}_F(K)_{NEW} = \dot{Z}_F(K)_{OLD} + \ddot{Z}_F(K)\,\Delta t$
$V_F(K)_{NEW} = (\dot{X}^2_F(K)_{NEW} + \dot{Y}^2_F(K)_{NEW} + \dot{Z}^2_F(K)_{NEW})^{1/2}$
$\dot{X}_F(K)_{OLD}, \dot{Y}_F(K)_{OLD}$
$\dot{Z}_F(K)_{OLD}, V_F(K)_{OLD}$
$\dot{X}_F(K)_{NEW}, \dot{Y}_F(K)_{NEW}$
$\dot{Z}_F(K)_{NEW}, V_F(K)_{NEW}$
$X_F(K)_{OLD}, Y_F(K)_{OLD},$
$Z_F(K)_{OLD}$
(K)
$_{OLD} = X_F(K)_{INITIAL}, \dot{X}_F(K)_{OLD} = \dot{X}_F(K)_{INITIAL}$
$_{OLD} = Y_F(K)_{INITIAL}, \dot{Y}_F(K)_{OLD} = \dot{Y}_F(K)_{INITIAL}$
$_{OLD} = Z_F(K)_{INITIAL}, \dot{Z}_F(K)_{OLD} = \dot{Z}_F(K)_{INITIAL}$

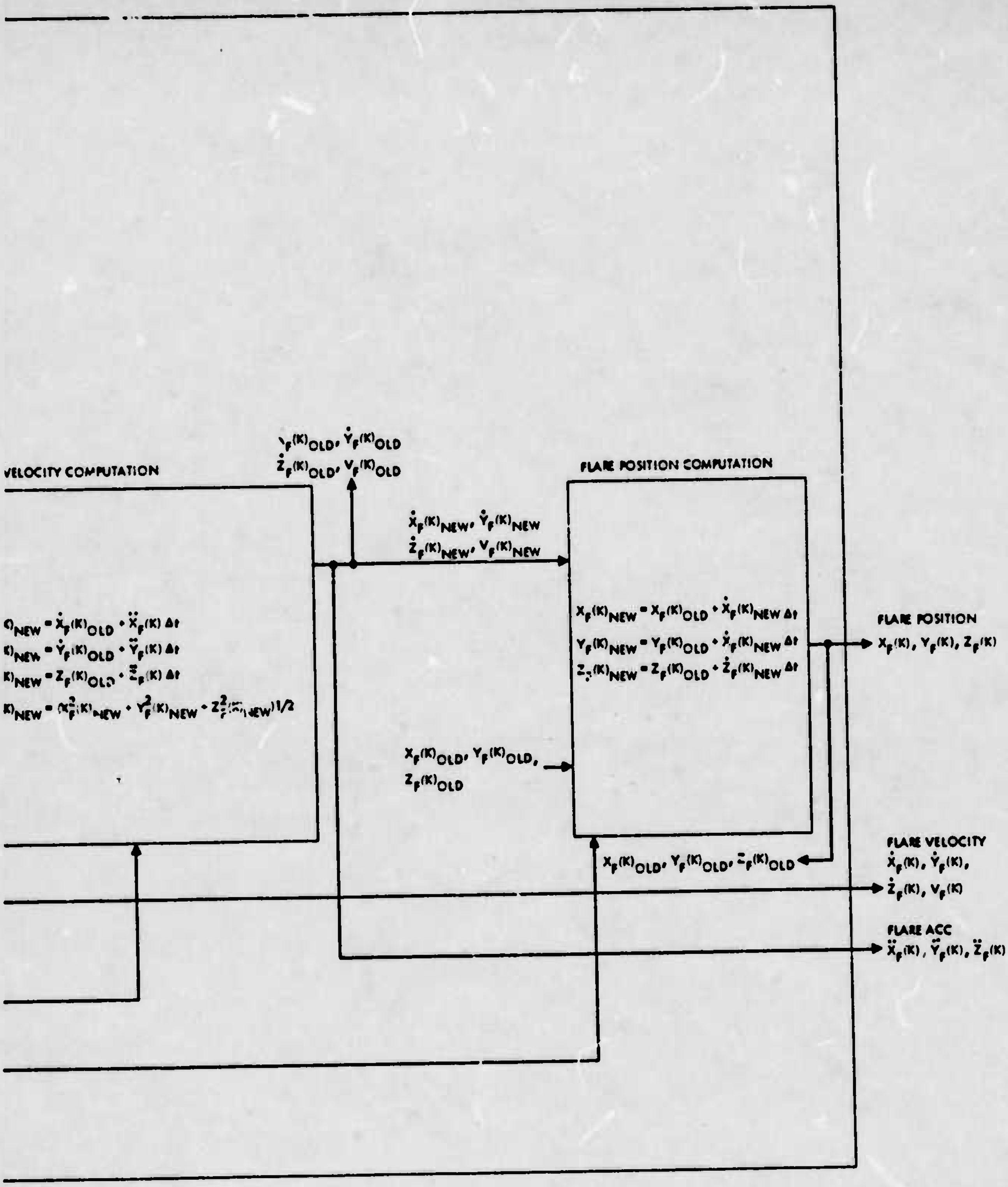


Figure 3-11. Flare dynamics, Kth flare

Table 3-18. Flare dynamics subroutine

```
SUBROUTINE FLRDYN      74/74    OPT=1                     FTN 4.2+P381        75/11/79  13.31.59.

      SUBROUTINE FLRDYN(T,DELT,TF,TAURN,SF,SF0,XFL,WFO,WG,WF,N,RHO,COF     NOR19       01
     *,G,VF,X,XP,XFD,XF,XFP,NN,NC)                                          FLRDYNAM     2
      DIMENSION X(10),XP(10),NN(20),TF(20),WF(20),VF(20)                   FLRDYNAM     4
      DIMENSION XFD(10,20),XF(10,20),XFP(10,20),SF(20)                     FLRDYNAM     5
      DO 1 K=1,N                                                           FLRDYNAM     6
      GO TO (2,5,2), NN(K)                                                 FLRDYNAM     7
    5 IF(XFP(9,K).LE.0.) GO TO 1                                           FLRDYNAM     8
      A=1.-(T-TF(K))/TAURN                                                 FLRDYNAM     9
      SF(K)=SF0*A*A                                                        FLRDYNAM    10
      GO TO (17,17,17,18), NC                                              FLRDYNAM    11
   18 A1=SF0/3.                                                            FLRDYNAM    12
      A2=XFL*XFL/144.                                                      FLRDYNAM    13
      AREA=(A1+A2)/2.                                                      FLRDYNAM    14
      SFK=AREA*A*A                                                         FLRDYNAM    15
      WF(K)=WFO*A*A*A                                                      FLRDYNAM    16
      GO TO 19                                                             FLRDYNAM    17
   17 SFK=SF(K)                                                            FLRDYNAM    18
      WF(K)=WFO-WG*(1.-A*A)                                                FLRDYNAM    19
   19 CONST=-.5*RHO*CDF*G                                                  FLRDYNAM    20
      DF=CONST*SFK*VF(K)*VF(K)/WF(K)                                       FLRDYNAM    21
      TANGFP=XFP(9,K)/XF(7,K)                                              FLRDYNAM    22
      B=SQRT(XF(7,K)*XF(7,K)+XF(9,K)*XF(9,K))                              FLRDYNAM    23
      COSGF=XF(7,K)/B                                                      FLRDYNAM    24
      SINGF=XF(9,K)/B                                                      FLRDYNAM    25
      A=COSGF*COSGF*TANGFP*TANGFP                                          FLRDYNAM    26
      XFD(1,K)=DF*COSGF/SQRT(1.+A)                                         FLRDYNAM    27
      XFD(2,K)=DF*SINGF/SQRT(1.+A)                                         FLRDYNAM    28
      XFD(3,K)=DF*COSGF*TANGFP/SQRT(1.+A)-G                                FLRDYNAM    29
      DO 3 IV=1,6                                                          FLRDYNAM    30
      XF(IV,K)=X(IV)                                                       FLRDYNAM    31
    3 XFP(IV,K)=XP(IV)                                                     FLRDYNAM    32
      XFP(6,K)=XFP(8,K)+XF(7,K)*DELT                                       FLRDYNAM    33
      XFP(7,K)=XFP(7,K)+XFD(1,K)*DELT                                      FLRDYNAM    34
      XF(7,K)=XF(7,K)+XFD(1,K)*DELT                                        FLRDYNAM    35
      XF(9,K)=XF(9,K)+XFD(2,K)*DELT                                        FLRDYNAM    36
      XFP(9,K)=XFP(9,K)+XFD(3,K)*DELT                                      FLRDYNAM    37
      VF(K)=SQRT(XF(7,K)*XF(7,K)+XF(9,K)*XF(9,K)+XFP(9,K)*XFP(9,K))        FLRDYNAM    38
      XF(6,K)=XF(5,K)+XF(7,K)*DELT                                         FLRDYNAM    39
      XF(8,K)=XF(8,K)+XF(9,K)*DELT                                         FLRDYNAM    40
      XFP(8,K)=XFP(8,K)+XFP(9,K)*DELT                                      FLRDYNAM    41
    1 CONTINUE                                                             FLRDYNAM    42
    2 RETURN                                                               FLRDYNAM    43
      END                                                                  FLRDYNAM    44
```

4. ANGLE AND RANGE COMPUTATIONS

In this section, the missile and target and missile and flare computations are performed to determine angle, angle rate, range, and range rate. Section 4.1 discusses the angle computations, and Section 4.2 discusses the range computations.

4.1 ANGLE AND ANGLE RATE COMPUTATIONS

Tables 4-1 and 4-2 show how the angles between the missile and target and their corresponding rates shown in the geometry are computed along with the variable name associated with each in the computer program. Subroutine MTGTANG in the program is responsible for performing these computations. See Table 4-3 for a listing of this subroutine. Figures 4-1 and 4-2 show the computational procedures for these calculations in block diagram form.

Figures 4-3 and 4-4 describe the angles and angular computations for the angles between the missile and flare (K^{th} flare). Subroutine MFLANG is used to compute these angular variables. See Table 4-4 for a program listing.

Table 4-1. Angle and angle rate computations between target and missile, horizontal plane

Angle/ Angle Rate	Variable Name	Computation
σ	Z(1)	TAN^{-1}[X(3)/X(1)]
$\dot{\sigma}$	Z(2)	[X(1)*X(4)-X(2)*X(3)]/[X(3)*X(3)+X(1)*X(1)]
ψ	Z(3)	TAN^{-1}[[X(8)-X(3)]/[X(6)-X(1)]]
$\dot{\psi}$	Z(4)	[X(6)-X(1)]*[X(9)-X(4)]-[X(8)-X(3)]*[X(7)-X(2)]/[[X(6)-X(1)]^2+[X(8)-X(3)]^2]
γ	Z(5)	TAN^{-1}[X(4)/X(2)]
$\dot{\gamma}$	Z(6)	[$\dot{X}$(4)*X(2)-X(4)*$\dot{X}$(2)]/[X(2)*X(2)+X(4)*X(4)]
γ_T	Z(7)	TAN^{-1}[X(9)/X(7)]
$\dot{\gamma}_T$	Z(8)	[$\dot{X}$(9)*X(7)-$\dot{X}$(7)*X(9)]/[X(7)*X(7)+X(9)*X(9)]
$\gamma-\sigma$	Z(9)	Z(5)-Z(1)
σ_T	Z(10)	TAN^{-1}[X(8)/X(6)]
σ_T	Z(11)	[X(6)*X(9)-X(7)*X(8)]/[X(6)*X(6)+X(8)*X(8)]
$\alpha+\gamma-\psi$	Z(12)	ALPHA+Z(5)-Z(3)

Table 4-2. Angle and angle rate computations between target and missile, vertical plane

Angle/ Angle Rate	Variable Name	Computation
σ'	ZP(1)	TAN^{-1}[XP(3)/XP(1)]
$\dot{\sigma}'$	ZP(2)	[XP(1)*XP(4)-XP(2)*XP(3)]/[XP(3)*XP(3)+XP(1)*XP(1)]
ψ'	ZP(3)	TAN^{-1}[XP(8)-XP(3)]/[XP(6)-XP(1)]]
$\dot{\psi}'$	ZP(4)	[XP(6)-XP(1)]*[XP(9)-XP(4)]-[XP(8)-XP(3)]*[XP(7)-XP(2)]/ [[XP(6)-XP(1)]^2+[XP(8)-XP(3)]^2]
γ'	ZP(5)	TAN^{-1}[XP(4)/XP(2)]
γ'	ZP(6)	[$\dot{X}$P(4)*XP(2)-XP(4)*$\dot{X}$P(2)]/[XP(2)*XP(2)+XP(4)*XP(4)]
γ'_T	ZP(7)	TAN^{-1}[XP(9)/XP(7)]
$\dot{\gamma}'_T$	ZP(8)	[$\dot{X}$P(9)*XP(7)-$\dot{X}$P(7)*XP(9)]/[XP(7)*XP(7)+XP(9)*XP(9)]
$\gamma'-\sigma'$	ZP(9)	ZP(5)-ZP(1)
σ'_T	ZP(10)	TAN^{-1}[XP(8)/XP(6)]
$\dot{\sigma}'_T$	ZP(11)	[XP(6)*XP(9)-XP(7)*XP(8)]/[XP(6)*XP(6)+XP(8)*XP(8)]
$\alpha'+\gamma'-\psi'$	ZP(12)	ALPHAP+ZP(5)-ZP(3)

Table 4-3. Missile and target angle and angle rate computations subroutines

```
SUBROUTINE MTGTANG     74/74    OPT=1                    FTN 4.2+P349        05/01/75  13.31.14.

      SUBROUTINE MTGTANG(X,ALPHA,X60,X20,X30,X70,Z,PI)                   MAR19      77
      DIMENSION X(10),Z(14)                                              MISSTGTA    3
      IF(X(1).NE.0.) GO TO 1                                             MISSTGTA    4
      Z(1)=PI/2.                                                         MISSTGTA    5
      GO TO 2                                                            MISSTGTA    6
    1 Z(1)=ATAN2(X(3),X(1))                                              MISSTGTA    7
    2 Z(2)=X(3)**2+X(1)**2                                               MAR20      10
      IF(Z(2).NE.0.0) Z(2)=(X(1)*X(4)-X(2)*X(3))/Z(2)                    MAR20      19
      Y=X(8)-X(3)                                                        MISSTGTA    9
      XX=X(6)-X(1)                                                       MISSTGTA   10
      IF(XX.NE.0.) GO TO 3                                               MISSTGTA   11
      Z(3)=PI/2.                                                         MISSTGTA   12
      GO TO 4                                                            MISSTGTA   13
    3 Z(3)=ATAN2(Y,XX)                                                   MISSTGTA   14
      IF(Z(3).GE.-PI/2.) GO TO 4                                         MISSTGTA   15
      Z(3)=2.*PI+Z(3)                                                    MISSTGTA   16
    4 D1=X(6)-X(1)                                                       MISSTGTA   17
      D2=X(8)-X(3)                                                       MISSTGTA   18
      Z(4)=((X(6)-X(1))*(X(9)-X(4))-(X(8)-X(3))*(X(7)-X(2)))/(D1*D1+D2*D MISSTGTA   19
     +2)                                                                 MISSTGTA   20
      IF(X(2).NE.0.) GO TO 5                                             MISSTGTA   21
      Z(5)=PI/2.                                                         MISSTGTA   22
      GO TO 6                                                            MISSTGTA   23
    5 Z(5)=ATAN2(X(4),X(2))                                              MISSTGTA   24
    6 Z(6)=(X40*X(2)-X(4)*X20)/(X(2)*X(2)+X(4)*X(4))                     MISSTGTA   25
      IF(X(9).NE.0.) GO TO 11                                            MISSTGTA   26
      Z(7)=0.                                                            MISSTGTA   27
      GO TO 8                                                            MISSTGTA   28
   11 IF(X(7).NE.0.) GO TO 7                                             MISSTGTA   29
      Z(7)=PI/2.                                                         MISSTGTA   30
      GO TO 8                                                            MISSTGTA   31
    7 Z(7)=ATAN2(X(9),X(7))                                              MISSTGTA   32
    8 Z(8)=(X90*X(7)-X70*X(9))/(X(7)*X(7)+X(9)*X(9))                     MISSTGTA   33
      Z(9)=Z(5)-Z(1)                                                     MISSTGTA   34
      IF(X(6).NE.0.) GO TO 9                                             MISSTGTA   35
      Z(10)=PI/2.                                                        MISSTGTA   36
      GO TO 10                                                           MISSTGTA   37
    9 Z(10)=ATAN2(X(8),X(6))                                             MISSTGTA   38
   10 Z(11)=(X(6)*X(9)-X(7)*X(8))/(X(6)*X(6)+X(8)*X(8))                  MISSTGTA   39
      Z(12)=ALPHA+Z(5)-Z(3)                                              MISSTGTA   40
      RETURN                                                             MISSTGTA   41
      END                                                                MISSTGTA   42
```

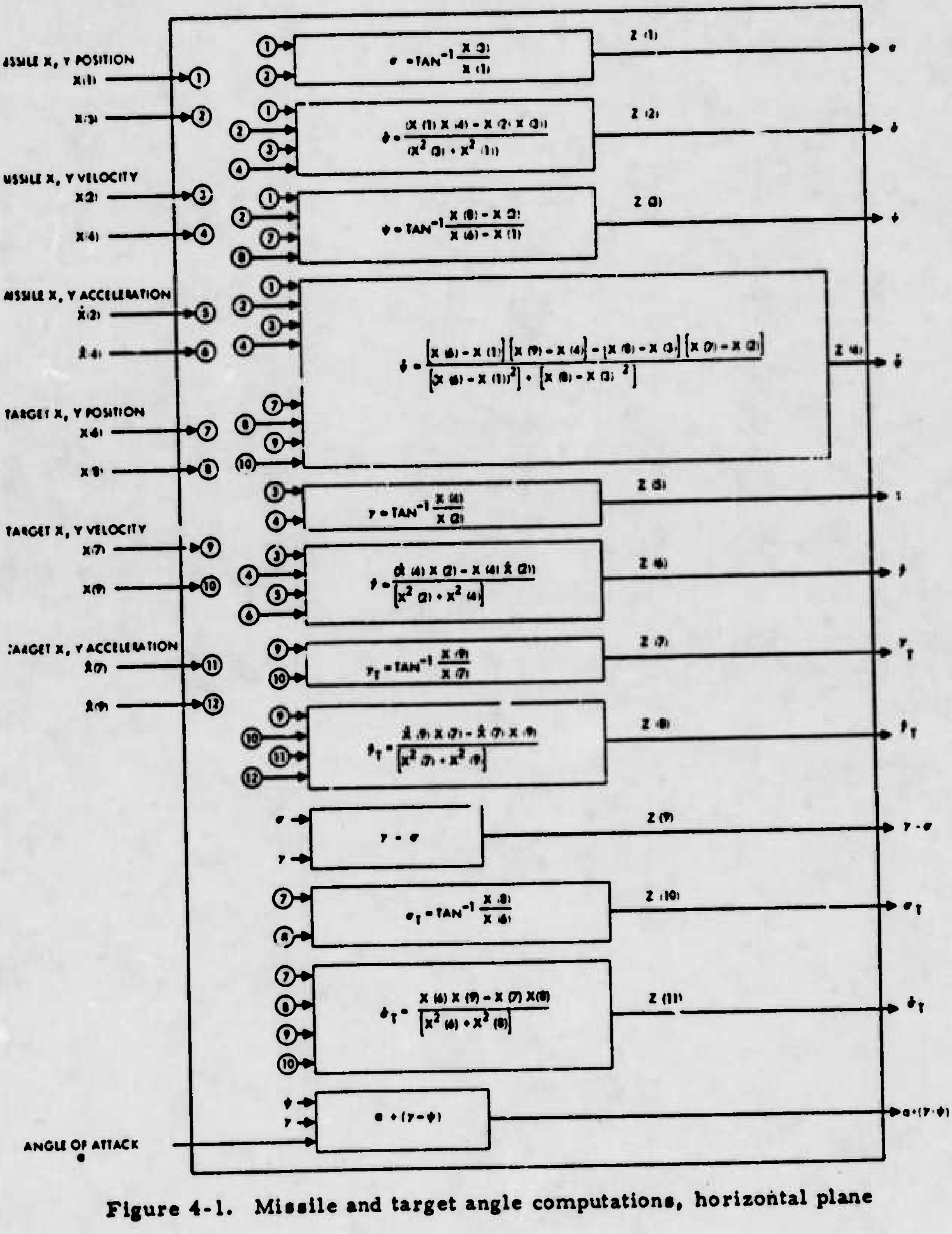
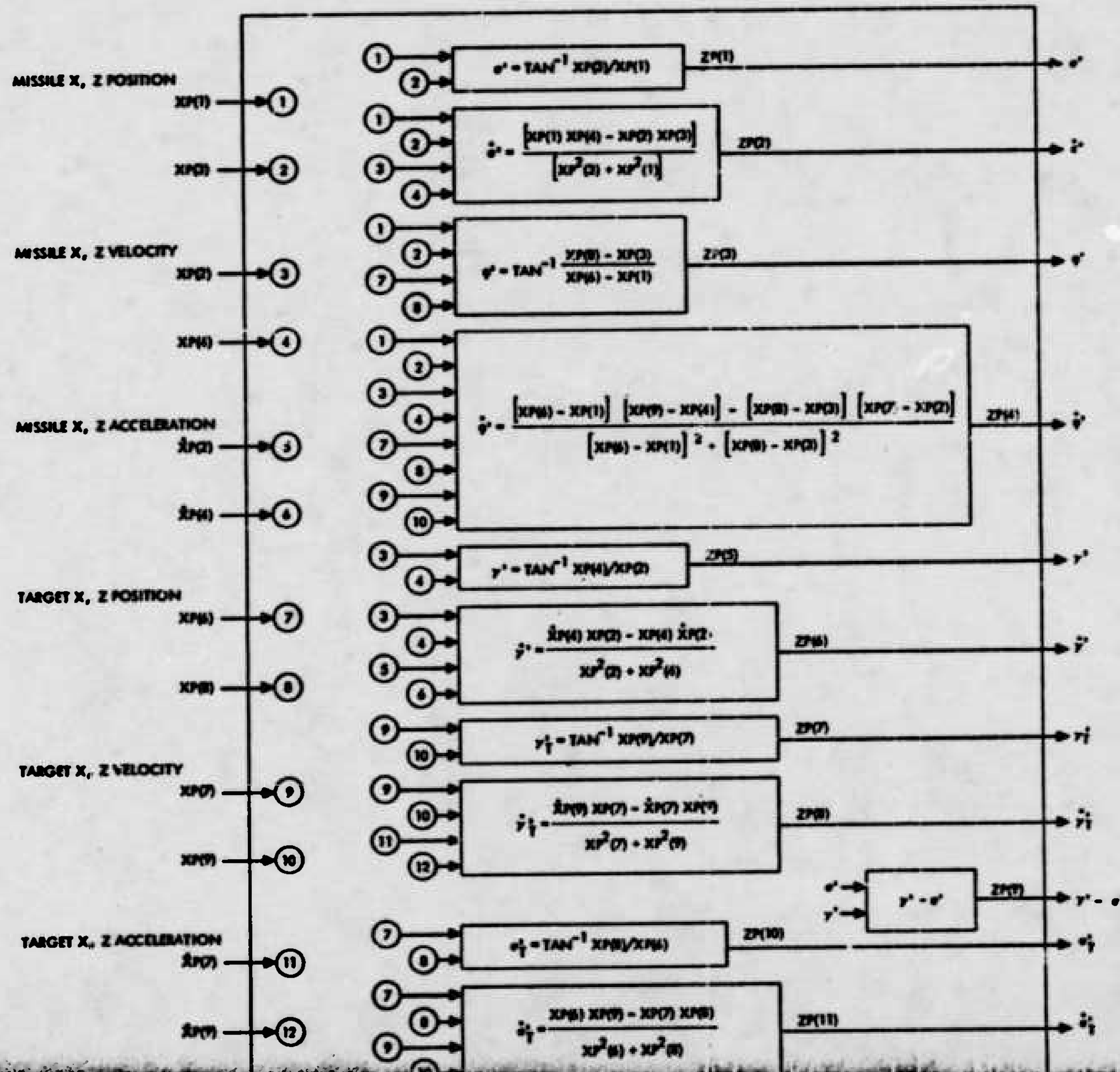


Figure 4-1. Missile and target angle computations, horizontal plane



MISSILE X, Z POSITION
XP(1)
XP(2)
MISSILE X, Z VELOCITY
XP(3)
XP(4)
MISSILE X, Z ACCELERATION
TARGET X, Z POSITION
XP(6)
XP(8)
TARGET X, Z VELOCITY
XP(7)
XP(9)
TARGET X, Z ACCELERATION
ZP(1)
ZP(2)
ZP(3)
ZP(4)
ZP(5)
ZP(6)
ZP(7)
ZP(8)
ZP(10)
ZP(11)

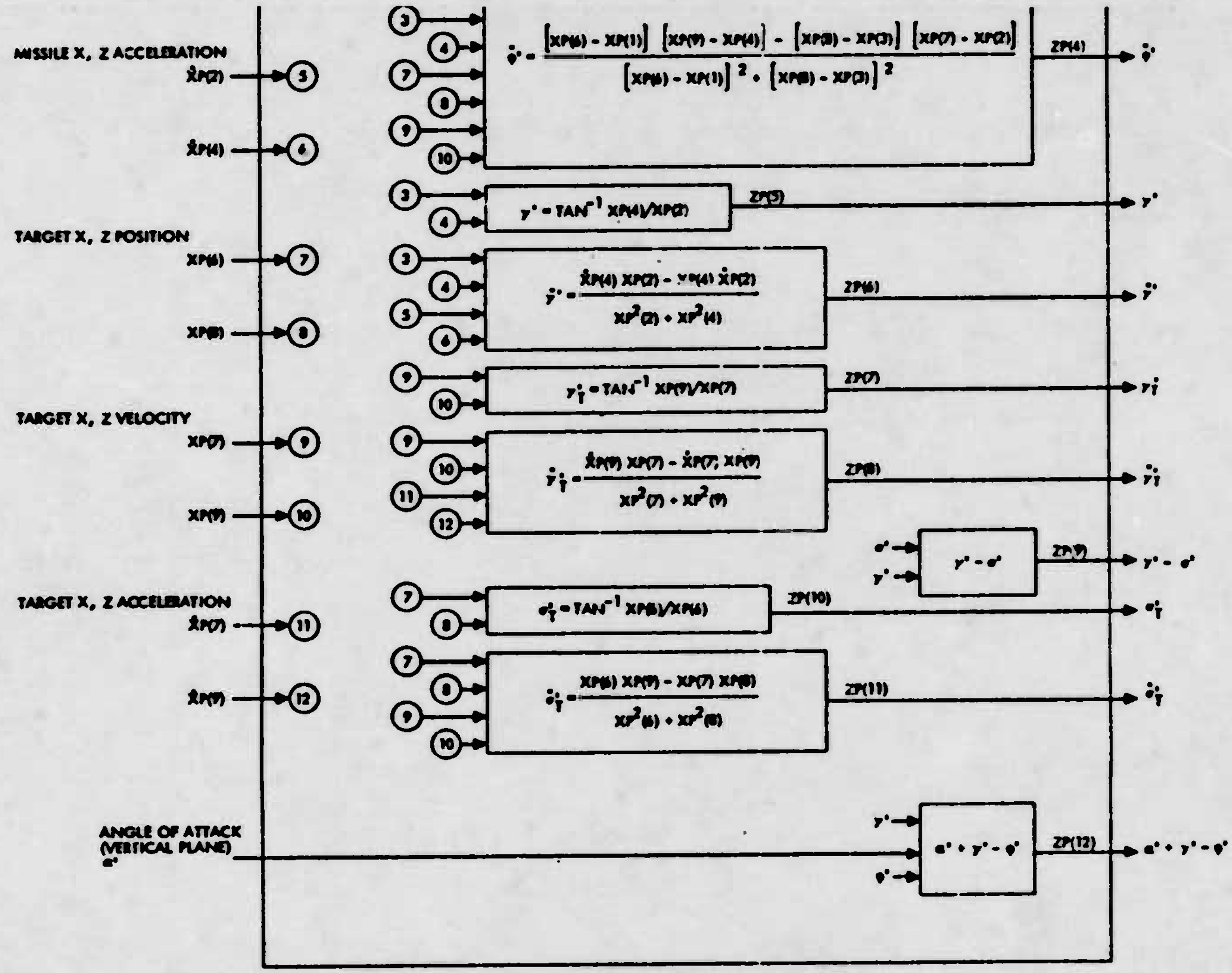


Figure 4-2. Missile and target angle computations, vertical plane

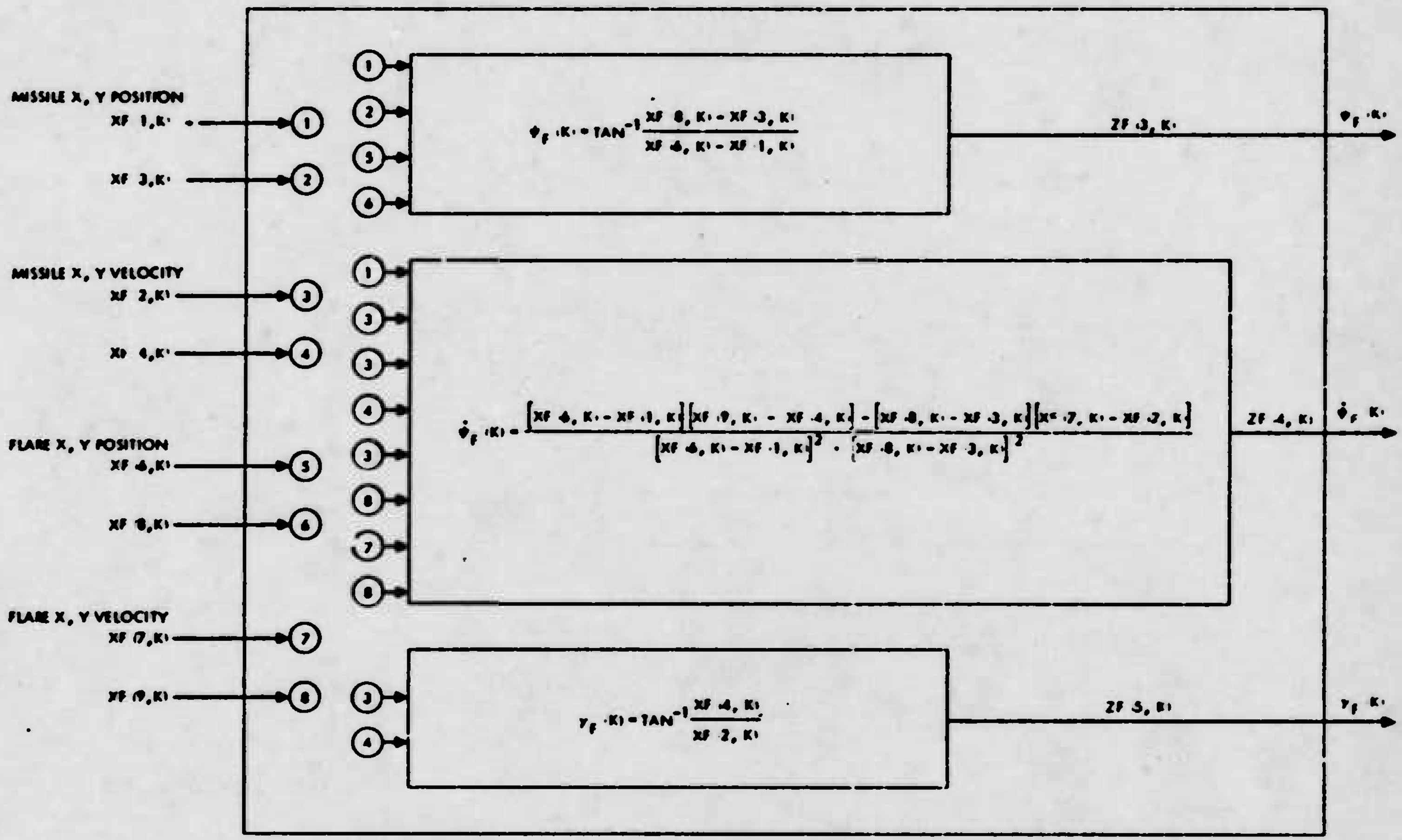


Figure 4-3. Missile and flare angle computations, horizontal plane

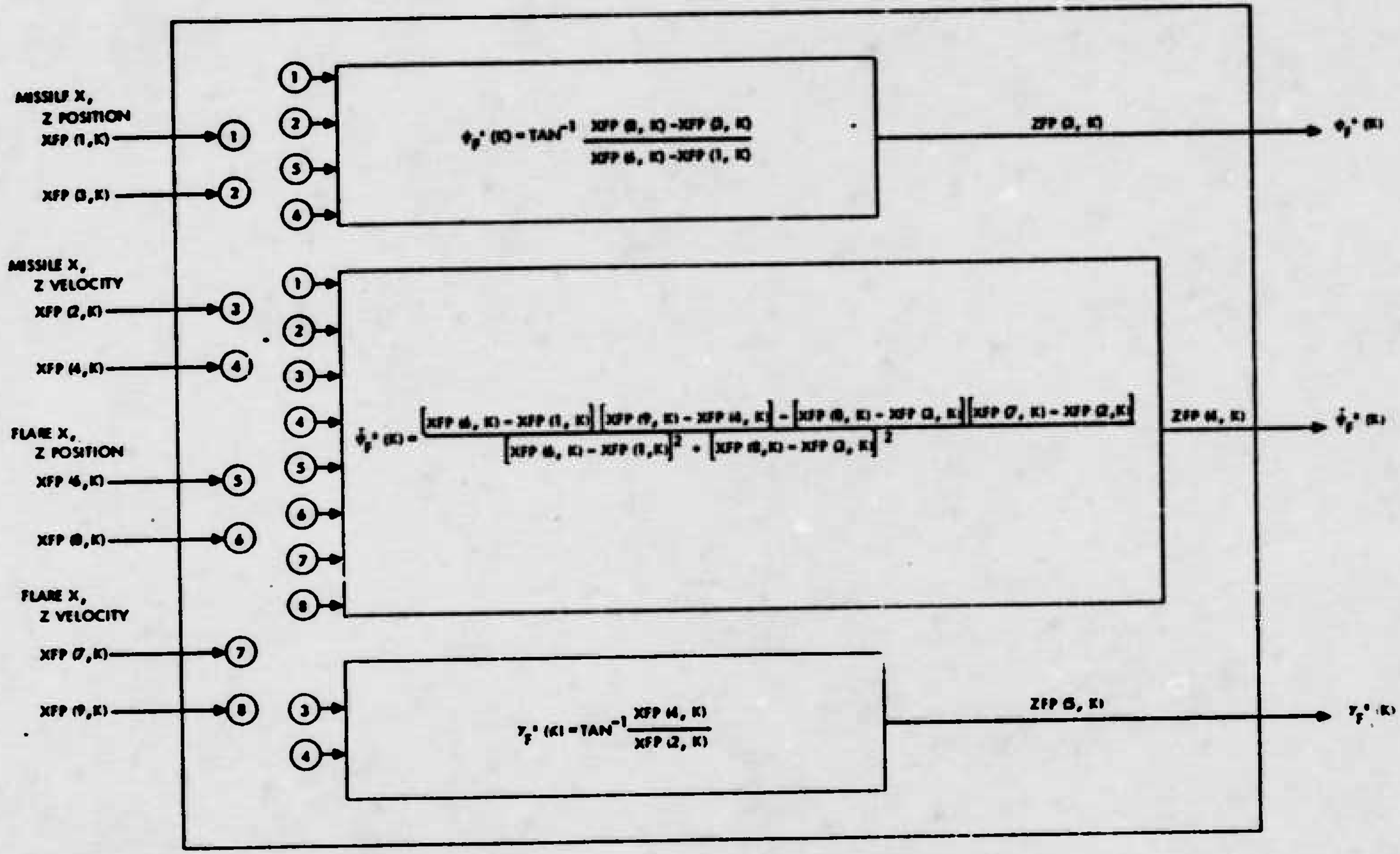


Figure 4-4. Missile and flare angle computations, vertical plane

Table 4-4. Missile and flare angle and angle rate computations subroutine

```
SUBROUTINE MFLANG      74/74    OPT=1                           FTN 4.2+P380        05/01/75  13.31.05.

                 SUBROUTINE MFLANG(XFP,N,ZF,PI)                                      MAR19       61
                 DIMENSION XFP(16,20),ZF(3,20)                                       MISSFLRA     3
                 DO 1 K=1,N                                                          MISSFLRA     4
                 Y=XFP(8,K)-XFP(3,K)                                                 MISSFLRA     5
                 X=XFP(6,K)-XFP(1,K)                                                 MISSFLRA     6
                 IF(X.NE.0.) GO TO 6                                                 MISSFLRA     7
                 ZF(1,K)=PI/2.                                                       MISSFLRA     8
                 GO TO 9                                                             MISSFLRA     9
               6 ZF(1,K)=ATAN2(Y,X)                                                  MISSFLRA    10
                 IF(ZF(1,K).GE.-PI/2.) GO TO 9                                       MISSFLRA    11
                 ZF(1,K)=2.*PI+ZF(1,K)                                               MISSFLRA    12
               9 A=XFP(9,K)-XFP(4,K)                                                 MISSFLRA    13
                 B=XFP(7,K)-XFP(2,K)                                                 MISSFLRA    14
                 ZF(2,K)=(X*A-Y*B)/(X*X+Y*Y)                                         MISSFLRA    15
                 IF(XFP(9,K).NE.0.) GO TO 2                                          MISSFLRA    16
                 ZF(3,K)=0.                                                          MISSFLRA    17
                 GO TO 3                                                             MISSFLRA    18
               2 IF(XFP(7,K).NE.0.) GO TO 8                                          MISSFLRA    19
                 ZF(3,K)=PI/2.                                                       MISSFLRA    20
                 GO TO 3                                                             MISSFLRA    21
               8 ZF(3,K)=ATAN2(XFP(9,K),XFP(7,K))                                    MISSFLRA    22
               3 CONTINUE                                                            MISSFLRA    23
               1 RETURN                                                              MISSFLRA    24
                 END                                                                 MISSFLRA    25
```

4.2 RANGE AND RANGE RATE COMPUTATIONS

Figure 4-5 shows the computational procedure for determining the missile velocity missile acceleration, missile mach number, the target velocity, target acceleration, the components of relative position velocity and acceleration between the missile and the target, and the relative range and range rate. In addition to these computations, this subroutine also calculates the miss distance and time-to-go parameters. There is a check in the computations to determine if the range rate is postive after missile thrusting is terminated. This thrusting termination period is nominally set at 5 seconds. If this constraint is violated, the program goes into an abort. Tables 4-5 and 4-6 contain listings of these subroutines.

Figure 4-6 shows how the range between the missile and an arbitrary flare (K^{th} flare) as well as their corresponding rates are computed.

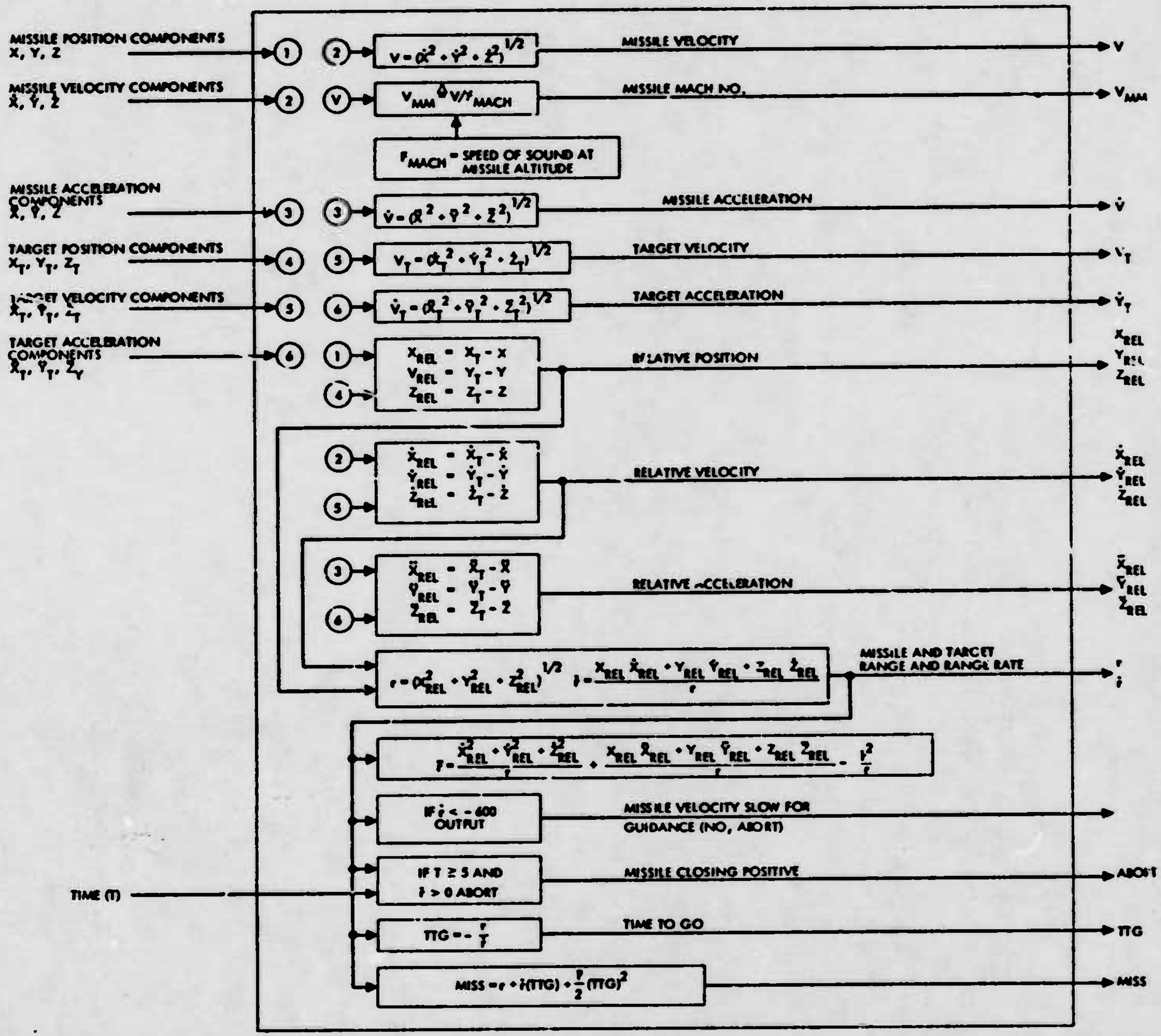


Figure 4.5. Missile and target range and velocity computations

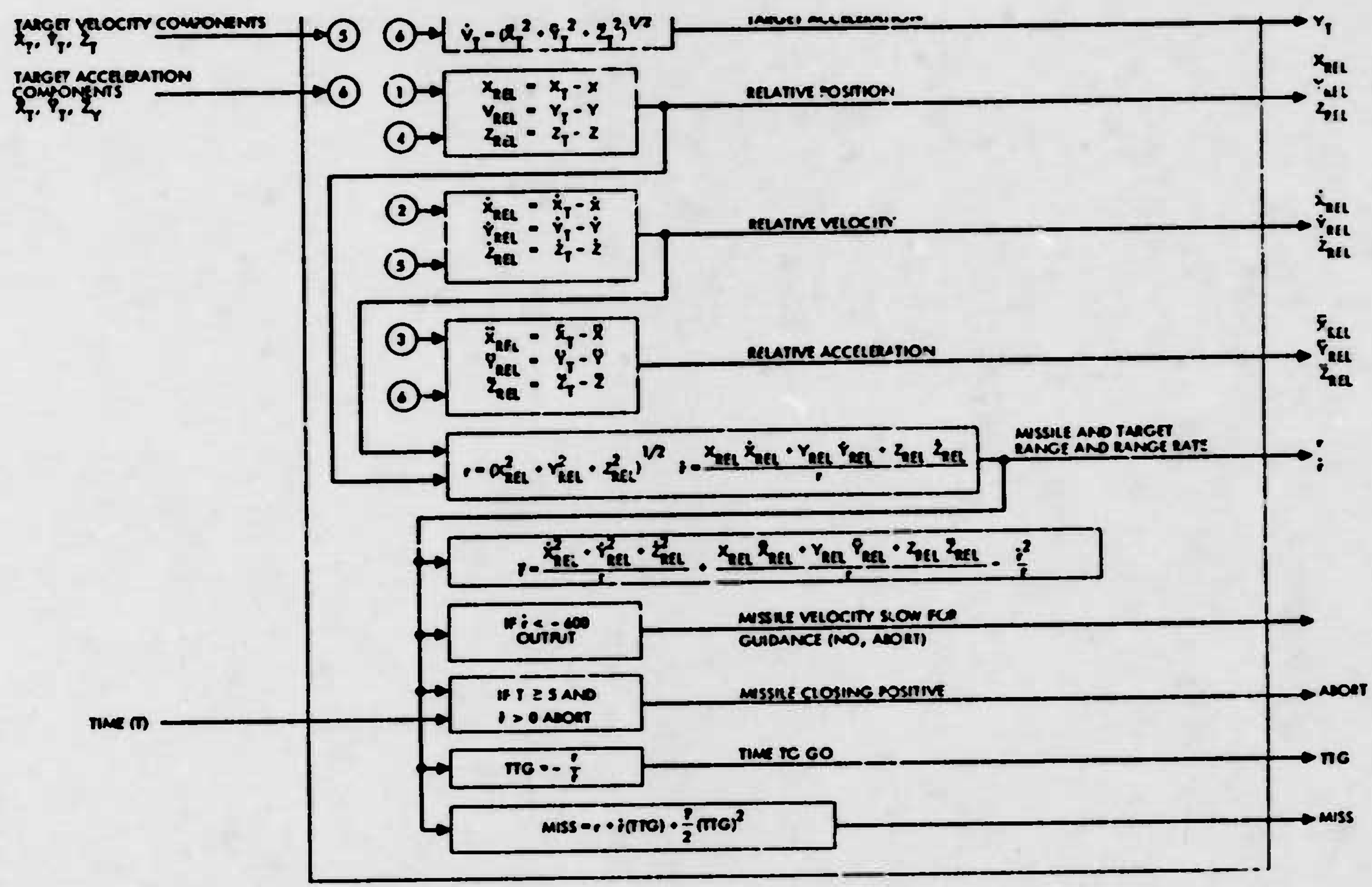


Figure 4.5. Missile and target range and velocity computations

Table 4-5. Range and range rate between missile and target subroutines

```
SUBROUTINE MTRVEL     74/74    OPT=1                           FTN 4.2+P393        05/01/75  13.31.11.

      SUBROUTINE MTRVEL(X,XP,X20,X60,XP60,X73,X90,XP90,T,VM,VM0,VT,VT0  MAR19      66
     *,REL,LP,TTG,RMISS,MSG1)                                      MTRNGVEL    3
      DIMENSION X(10),XP(10),REL(12)                               MTRNGVEL    4
      XP(1)=X(1)                                                   MTRNGVEL    5
      XP(2)=X(2)                                                   MTRNGVEL    6
      XP(6)=X(6)                                                   MTRNGVEL    7
      XP(7)=X(7)                                                   MTRNGVEL    8
      VM=SQRT(X(3)*X(3)+X(4)*X(4)+XP(4)*XP(4))                     MTRNGVEL    9
      VM0=SQRT(X23*X20+X60*X60+XP60*XP60)                          MTRNGVEL   10
      VT=SQRT(X(7)*X(7)+X(9)*X(9)+XP(9)*XP(9))                     MTRNGVEL   11
      VT0=SQRT(X73*X70+X90*X90+XP93*XP93)                          MTRNGVEL   12
      REL(1)=X(6)-X(1)                                             MTRNGVEL   13
      REL(2)=X(8)-X(3)                                             MTRNGVEL   14
      REL(3)=XP(8)-XP(3)                                           MTRNGVEL   15
      REL(4)=X(7)-X(2)                                             MTRNGVEL   16
      REL(5)=X(9)-X(4)                                             MTRNGVEL   17
      REL(6)=XP(9)-XP(4)                                           MTRNGVEL   18
      REL(7)=X70-X20                                               MTRNGVEL   19
      REL(8)=X90-X60                                               MTRNGVEL   20
      REL(9)=XP90-XP60                                             MTRNGVEL   21
      REL(10)=SQRT(REL(1)*REL(1)+REL(2)*REL(2)+REL(3)*REL(3))      MTRNGVEL   22
      AA=REL(1)*REL(4)+REL(2)*REL(5)+REL(3)*REL(6)                 MTRNGVEL   23
      REL(11)=AA/REL(10)                                           MTRNGVEL   24
      A=REL(4)*REL(4)+REL(5)*REL(5)+REL(6)*REL(6)                  MTRNGVEL   25
      B=REL(1)*REL(7)+REL(2)*REL(8)+REL(3)*REL(9)                  MTRNGVEL   26
      REL(12)=A/REL(10)+B/REL(10)-REL(11)*REL(11)/REL(10)          MTRNGVEL   27
      IF(REL(11).GE.-400.) GO TO 6                                 MTRNGVEL   28
      GO TO 7                                                      MTRNGVEL   29
    6 MSG1=0                                                       MTRNGVEL   30
      GO TO 2                                                      MTRNGVEL   31
    7 MSG1=MSG1+1                                                  MTRNGVEL   32
      IF(MSG1.GT.1) GO TO 2                                        MTRNGVEL   33
      WRITE(6,1)                                                   MTRNGVEL   34
    1 FORMAT(1X,45HMISSILE VELOCITY SLOW FOR GUIDANCE (NO ABORT))  MTRNGVEL   35
    2 IF(T.GE.5..AND.REL(11).GT.0.) GO TO 5                        MTRNGVEL   36
      GO TO 3                                                      MTRNGVEL   37
    5 WRITE(6,4)                                                   MTRNGVEL   38
    4 FORMAT(1X,26HMISSILE CLOSING POSITIVE)                       MAR19      67
      WRITE(6,8) REL(1),REL(2),REL(3),REL(4),REL(5),REL(6)         MAR19      68
    8 FORMAT(1X,61HREL(1),REL(2),REL(3),REL(4),REL(5),REL(6),/,6(1X,E12.  MAR19  69
     *6))                                                          MAR19      70
      WRITE(6,9) X(8),XP(8),XP(9)                                  MAR19      71
    9 FORMAT(1X,5HX(8)=,E12.5,1X,6HXP(8)=,E12.5,1X,6HXP(9)=,E12.5)  MAR19     72
      WRITE(6,10) X(1),X(2),X(3),X(4),X(6),XP(4),XP(9),X(9),X(7)   MAR19      73
   10 FORMAT(1X,64HX(1),X(2),X(3),X(4),X(5),XP(4),XP(9),X(9),X(7),/,9(1X  MAR19  74
     *,E12.4))                                                     MAR19      75
      WRITE(6,11) T,REL(11),LP                                     MAR19      76
   11 FORMAT(1X,2HT=,E12.6,1X,8HREL(11)=,E10.3,1X,3HLP=,I2)        MTRNGVEL   64
      LP=2                                                         MTRNGVEL   65
    3 TTG=-AA/A                                                    MTRNGVEL   66
      RMISS=REL(10)+REL(11)*TTG+REL(12)*.5*TTG*TTG                 MTRNGVEL   67
      RETURN                                                       MTRNGVEL   68
      END
```

Table 4-6. Range and range rate between missile and flare subroutine

```
SUBROUTINE MFLRVEL     74/74    OPT=1                              FTN 4.2+P390       05/01/75  13.31.02.

          SUBROUTINE MFLRVEL(N,XF,VFD,XFD,VF,AF,RELF)                              MAR19      62
          DIMENSION XF(13,23),XFP(13,23),XFD(13,23),RELF(8,23)                     MISSFLRN    3
          DIMENSION VF(23),AF(23)                                                  MISSFLRN    4
          DO 1 K=1,N                                                               MISSFLRN    5
          VF(K)=SQRT(XF(7,K)*XF(7,K)+XF(8,K)*XF(8,K)+XFD(9,K)*XFD(9,K))            MISSFLRN    6
          AF(K)=SQRT(XFD(1,K)*XFD(1,K)+XFD(2,K)*XFD(2,K)+XFD(3,K)*XFD(3,K))        MISSFLRN    7
          RELF(1,K)=XF(4,K)-XF(1,K)                                                MISSFLRN    8
          RELF(2,K)=XF(5,K)-XF(2,K)                                                MISSFLRN    9
          RELF(3,K)=XFP(9,K)-XFP(3,K)                                              MISSFLRN   10
          RELF(4,K)=XF(7,K)-XF(2,K)                                                MISSFLRN   11
          RELF(5,K)=XF(9,K)-XF(6,K)                                                MISSFLRN   12
          RELF(6,K)=XFP(9,K)-XFP(6,K)                                              MISSFLRN   13
          RELF(7,K)=SQRT(RELF(1,K)*RELF(1,K)+RELF(2,K)*RELF(2,K)+RELF(3,K)*R       MISSFLRN   14
         +ELF(3,K))                                                                MISSFLRN   15
          RELF(8,K)=(RELF(1,K)*RELF(4,K)+RELF(2,K)*RELF(5,K)+RELF(3,K)*RELF(       MISSFLRN   16
         +6,K))/RELF(7,K)                                                          MISSFLRN   17
        1 CONTINUE                                                                 MISSFLRN   18
        2 RETURN                                                                   MISSFLRN   19
          END                                                                      MISSFLRN   20
```

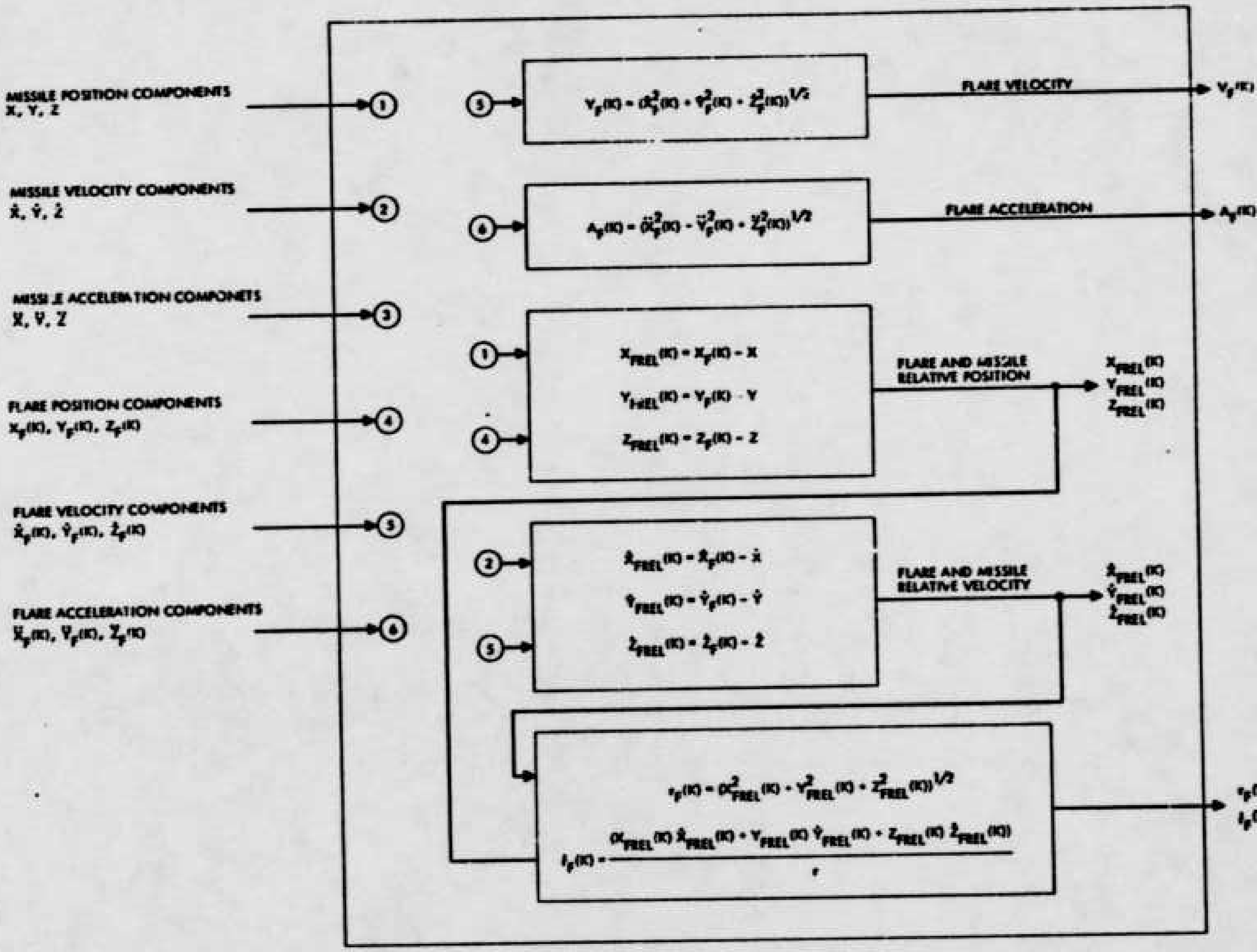


Figure 4-6. Missile and flare range and velocity computations

5. TARGET AND FLARE IRRADIANCE COMPUTATIONS

This portion of the program determines the irradiance from the target and flare at the missile dome. Section 5.1 describes the target irradiance computations, and Section 5.2 describes the flare irradiance computations.

5.1 TARGET IRRADIANCE COMPUTATIONS

In determining the irradiance at the missile from the target there are several major factors which must be determined as indicated in Figure 5-1.

First, the aspect angle at which the missile views the aircraft must be determined. This is found from the dot product relationship.

$$V_T \cdot r = V_T\, r \cos \theta$$

or

$$\dot{X}_T\, r_X + \dot{Y}_T\, r_Y + \dot{Z}_T r_Z = V_T\, r \cos \theta$$

where,

V_T = target velocity

r = missile/target range

$\dot{X}_T, \dot{Y}_T, \dot{Z}_T$ = X, Y, Z components of target velocity

r_X, r_Y, r_Z = X, Y, Z components of missile/target range

θ = aspect angle.

The target intensity data is stored in the program as a function of polar angle. Once the aspect viewing angle has been determined the value of the target intensity is found by means of a table look up on this data.

Data on atmospheric attenuation is stored in the program as a function of range and black body temperature. Based on the aircraft tailpipe temperature and the missile/target range, the atmospheric attenuation is found by means of a table look up.

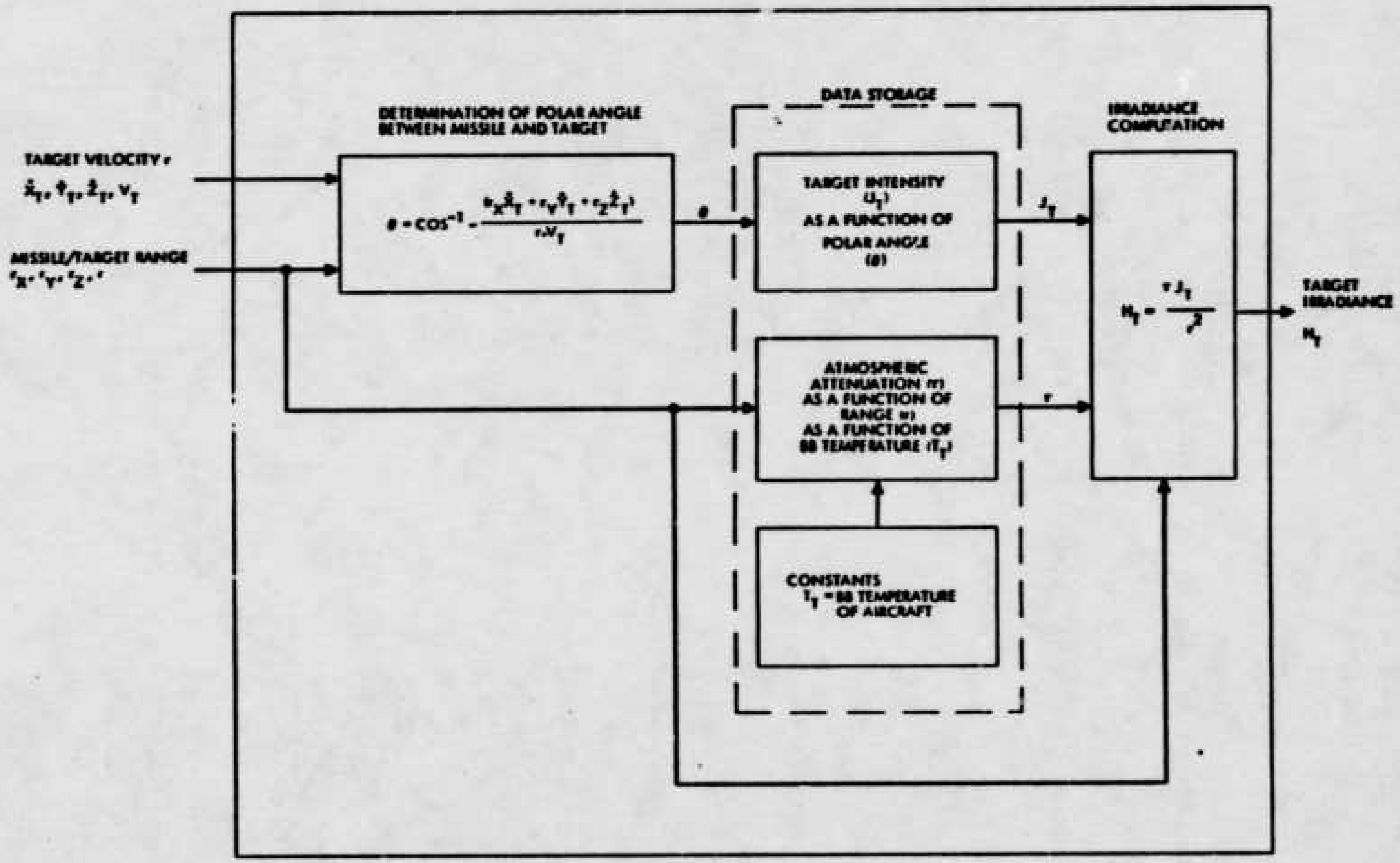


Figure 5-1. Target irradiance computation

The irradiance from the target is then computed, as indicated in Figure 5-1, based on the target intensity, atmospheric attenuation and missile/target range. A listing of this subroutine is contained in Table 5-1. Note: The EPICS program consists of ASDIR II in conjunction with the SPKINT subroutine and M/T/CM. As we have just seen, M/T/CM contains an atmospheric transmission file. Thus, when using ASDIR II (or any other program which generates spectral radiant intensity) it is important to use a true (unattenuated) spectral radiant intensity.

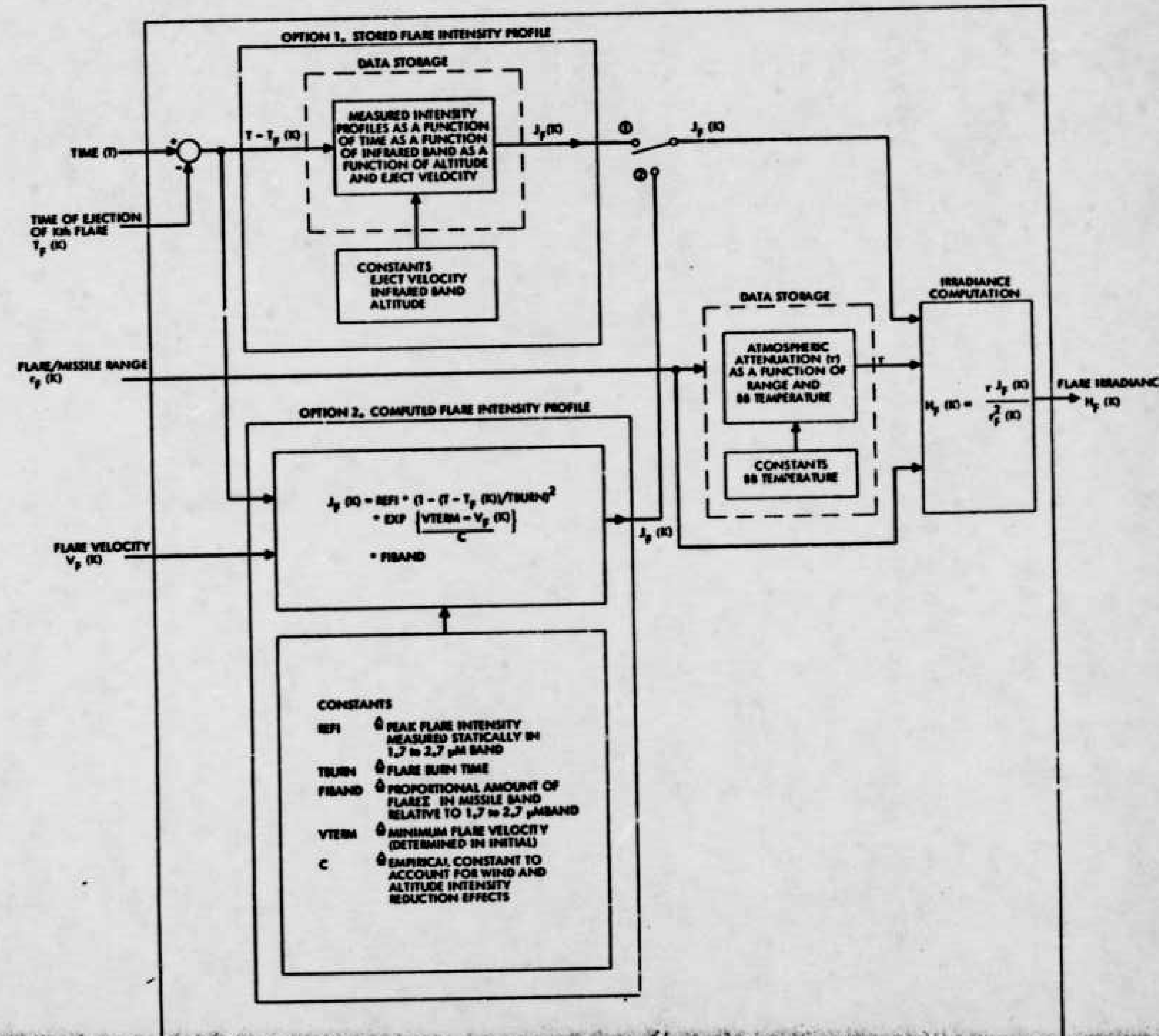
If it is desired to use a different atmospheric transmission model, this latter model must be used (in conjunction with target temperature, altitude, range and optical waveband) to generate a new atmospheric file in the M/T/CM program.

Table 5-1. Target irradiance subroutine

```
SUBROUTINE TGTIR      74/74   OPT=1                     FTN 4.2+P380       05/01/75  13.31.22.

        SUBROUTINE TGTIR(XDT,YDT,ZDT,VT,RX,RY,RZ,R,PI,RINT,RANG,RNG,TAU     MAP19     FA
       +T,AT,TINT)                                                          TGTIRRAD   3
        DIMENSION RINT(30),RANG(70),RNG(19),TAUT(19)                        TGTIRRAD   4
        WPCM2=1E.*(.3048*.3048)                                             TGTIRRAD   5
        A=-(RX*XDT+RY*YDT+RZ*ZDT)/(R*VT)                                    TGTIRRAD   6
        IF(ABS(A).GT.1.) GO TO 2                                            TGTIRRAD   7
        GO TO 3                                                             TGTIRRAD   8
      2 A=SIGN(1.,A)                                                        TGTIRRAD   9
      3 AT=ACOS(A)*180./PI                                                  TGTIRRAD  10
        TINT=TLU2(AT,RANG,RINT)                                             TGTIRRAD  11
        TAUS=TLU2(R,RNG,TAUT)                                               TGTIRRAD  12
        HT=(TAUS*TINT*WPCM2)/(R*R)                                          TGTIRRAD  13
        RETURN                                                              TGTIRRAD  14
        END                                                                 TGTIRRAD  15
```

5.2 FLARE IRRADIANCE COMPUTATIONS

The program allows for two options in determining the flare intensity profile as indicated in Figure 5-2. Option 1 uses a table look up procedure for determing the flare intensity as a function of time. This procedure is generally used when dynamic, in flight intensity profiles are available for altitude and windstream conditions at or near those being considered. When no dynamic, in flight data exists, option 2 can be used to generate an intensity profile which accounts for altitude and windstream degradation factors. Basically this model consists of four factors. First, a peak flare intensity value (REFI) measured statically and referenced to the 1.7-2.7μ band is



OPTION 1. STORED FLARE INTENSITY PROFILE
DATA STORAGE
MEASURED INTENSITY PROFILES AS A FUNCTION OF TIME AS A FUNCTION OF INFRARED BAND AS A FUNCTION OF ALTITUDE AND EJECT VELOCITY
CONSTANTS EJECT VELOCITY INFRARED BAND ALTITUDE
TIME (T)
TIME OF EJECTION OF Kth FLARE T_F (K)
$T - T_F$ (K)
J_F (K)
FLARE/MISSILE RANGE r_F (K)
DATA STORAGE
ATMOSPHERIC ATTENUATION (τ) AS A FUNCTION OF RANGE AND BB TEMPERATURE
CONSTANTS BB TEMPERATURE
IRRADIANCE COMPUTATION
$H_F (K) = \frac{\tau J_F (K)}{r_F^2 (K)}$
FLARE IRRADIANCE H_F (K)
OPTION 2. COMPUTED FLARE INTENSITY PROFILE
$J_F (K) = REFI * (1 - (T - T_F (K))/TBURN)^2 * EXP \left[\frac{VTERM - V_F (K)}{C}\right] * FIBAND$
FLARE VELOCITY V_F (K)
CONSTANTS
REFI ≜ PEAK FLARE INTENSITY MEASURED STATICALLY IN 1.7 to 2.7 μM BAND
TBURN ≜ FLARE BURN TIME
FIBAND ≜ PROPORTIONAL AMOUNT OF FLARES IN MISSILE BAND RELATIVE TO 1.7 to 2.7 μMBAND
VTERM ≜ MINIMUM FLARE VELOCITY (DETERMINED IN INITIAL)
C ≜ EMPIRICAL CONSTANT TO ACCOUNT FOR WIND AND ALTITUDE INTENSITY REDUCTION EFFECTS

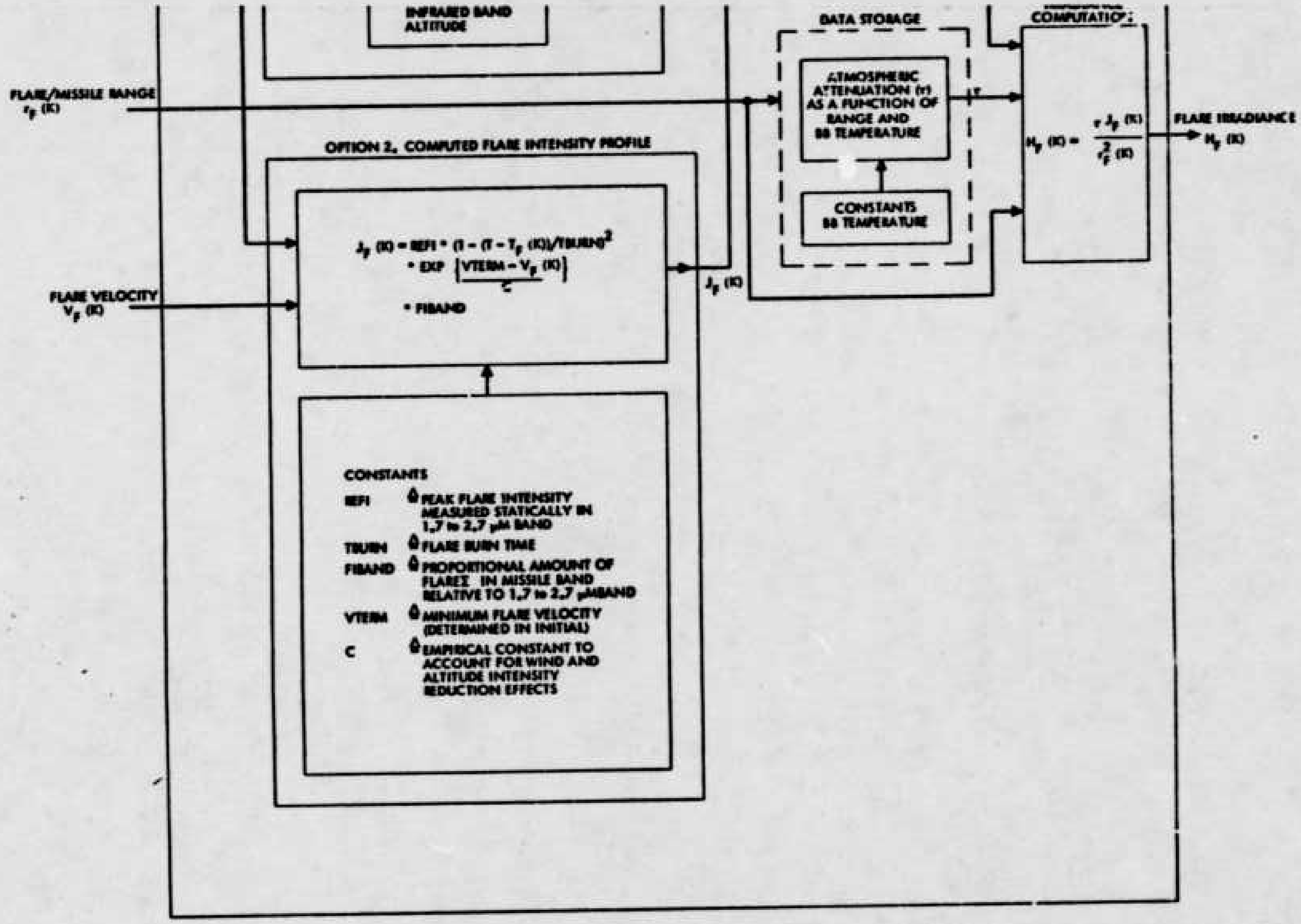


Figure 5-2. Flare irradiance computations

required. Second, the variation in the flare is burning surface area as a function of time is required. The expression shown is for a cylindrical type whose linear burn rate is assumed to be constant. (See reference 1.) Next, the factor

$$\exp\left\{\frac{V\ term - V_F\ (K)}{C}\right\}\ \text{accounts for}$$

altitude and windstream degradation. This is an empirical expression found to curve fit measured, dynamic flare data quite well. The term V term represents the minimum flare velocity and is determined from trajectory parameters in the initialization portion of the program. (See also Fifth Quarterly Report, IRCM Simulation Study). Finally, the term FIBAND is used to proportion out the flare energy in the missile band being considered relative to the reference band (1.7-2.7μ).

Once the flare intensity has been determined, the atmospheric attenuation is found by means of a table look up, and the irradiance value is computed. A listing of this subroutine is contained in Table 5-2.

Table 5-2. Flare irradiance subroutine

```
SUBROUTINE FLRIR       74/74   OPT=1                          FTN 4.2+P393)      05/01/79  13.31.07.

      SUBROUTINE FLRIR(T,TF,RELF,N,VF,JF,HF,IOPT,TIM,JIM,FIBAND,IBNDM   MAR19      64
     *,FSF,REFI,TBURN,VTERM,RNG,TAUF)                                   FLRIRRAD    3
      REAL JF,JIM                                                       FLRIRRAD    4
      DIMENSION FIBAND(9),VTERM(20),FSF(20),RNG(19),TAUF(19)            FLRIRRAD    5
      DIMENSION TF(20),RELF(9,20),VF(20),JF(20),HF(20),TIM(180),JIM(180) FLRIRRAD   6
      WPCM2=1./(.32808*.32808)                                          FLRIRRAD    7
      DO 1 K=1,N                                                        FLRIRRAD    8
      TD=T-TF(K)                                                        FLRIRRAD    9
      IF(TD.LT.(TBURN-.01)) GO TO 8                                     FLRIRRAD   10
      JF(K)=0.                                                          FLRIRRAD   11
      GO TO 6                                                           FLRIRRAD   12
    8 A=1.-TD/TBURN                                                     FLRIRRAD   13
      GO TO (3,4), IOPT                                                 FLRIRRAD   14
    3 JF(K)=TLU2(TD,TIM,JIM)*FIBAND(IBNDM)*FSF(K)                       FLRIRRAD   15
      GO TO 6                                                           FLRIRRAD   16
    4 IF((VTERM(K)-VF(K)).GE.0.) GO TO 7                                FLRIRRAD   17
      JF(K)=REFI*A*A*EXP((VTERM(K)-VF(K))/35.)*FIBAND(IBNDM)/1000.*FSF(K FLRIRRAD  18
     *)                                                                 FLRIRRAD   19
      GO TO 6                                                           FLRIRRAD   20
    7 JF(K)=REFI*A*A*FIBAND(IBNDM)/1000.*FSF(K)                         FLRIRRAD   21
    6 TAU=TLU2(RELF(7,K),RNG,TAUF)                                      FLRIRRAD   22
      HF(K)=TAU*JF(K)*WPCM2/(RELF(7,K)*RELF(7,K))                       FLRIRRAD   23
    1 CONTINUE                                                          FLRIRRAD   24
    2 RETURN                                                            FLRIRRAD   25
      END                                                               FLRIRRAD   26
```

6. AIMPOINT DETERMINATION

The primary functions of this portion of the program are to

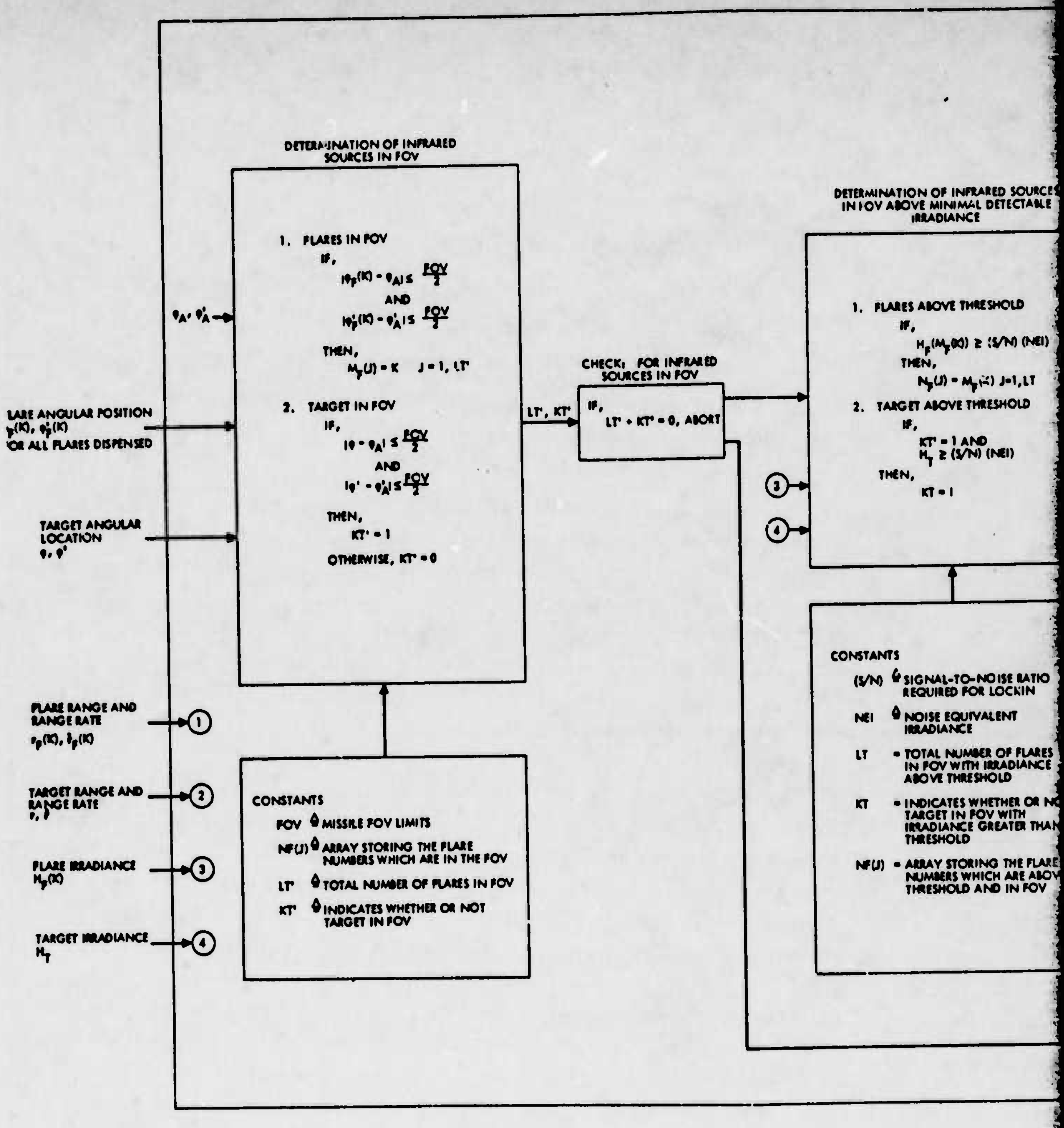
1. Determine which infrared sources are within the missile FOV.
2. Determine which of these sources within the missile FOV have irradiance levels above the minimum detectable.
3. Determine the missile aimpoint, on the basis of the missile's signal processing and source irradiance levels -- within FOV and above minimum detectable.

Figure 6-1 describes the computational procedure for determining the aimpoint in block diagram form. A further detailed description of these computations is contained in the following paragraphs. A listing of this subroutine is contained in Table 6-1.

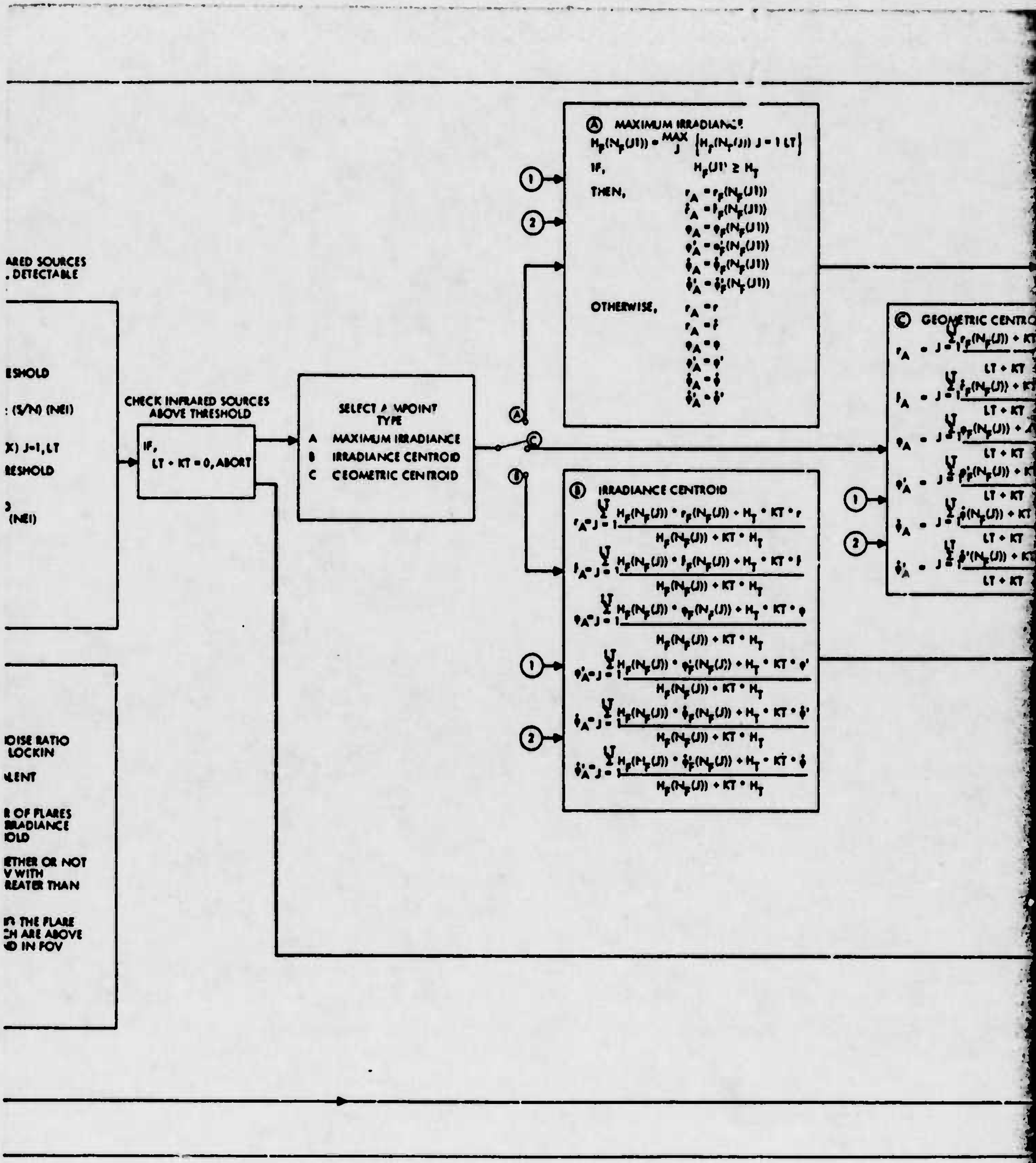
Table 6-2 shows the equations used to check which ignited flares are within the FOV of the missile and they are stored in an array $N_F(J)$ for further aimpoint processing. Similarity, the target is checked to see if it is within the FOV and the information on whether it is or not is stored in the variable KT'.

The total number of flares in the FOV is indicated by the variable LT'. The sum of the variables (LT' + KT') indicates the total number of IR sources in the FOV. If there are none, then the program will go into an abort mode due to the fact that there are no infrared sources in the missile FOV.

If there is at least one infrared source in the FOV, the program determine which IR sources have irradiance levels above the minimum detectable by the missile. The flares which meet this criteria are stored in an array $N_F(J)$ for further aimpoint processing with their total number in the array being indicated by the variable LT. Similarily, the target is checked to see if it is above threshold level and the information on whether it is or not is stored in the variable KT. If the sum of these variable (LT + KT) equals zero, then the program will abort due to the fact that there are no IR sources within the missile FOV above threshold value. If there is at one source which meet. this criteria the program will then determine the aimpoint.



DETERMINATION OF INFRARED SOURCES IN FOV
1. FLARES IN FOV
IF,
AND
THEN,
J = 1, LT'
2. TARGET IN FOV
IF,
AND
THEN,
KT' = 1
OTHERWISE, KT' = 0
LARE ANGULAR POSITION
OR ALL FLARES DISPENSED
TARGET ANGULAR LOCATION
FLARE RANGE AND RANGE RATE
TARGET RANGE AND RANGE RATE
FLARE IRRADIANCE
TARGET IRRADIANCE
1
2
3
4
CONSTANTS
FOV MISSILE FOV LIMITS
NF(J) ARRAY STORING THE FLARE NUMBERS WHICH ARE IN THE FOV
LT' TOTAL NUMBER OF FLARES IN FOV
KT' INDICATES WHETHER OR NOT TARGET IN FOV
LT', KT'
CHECK: FOR INFRARED SOURCES IN FOV
IF,
LT' + KT' = 0, ABORT
DETERMINATION OF INFRARED SOURCES IN FOV ABOVE MINIMAL DETECTABLE IRRADIANCE
1. FLARES ABOVE THRESHOLD
IF,
≥ (S/N) (NEI)
THEN,
J=1,LT
2. TARGET ABOVE THRESHOLD
IF,
KT' = 1 AND
≥ (S/N) (NEI)
THEN,
KT = 1
CONSTANTS
(S/N) SIGNAL-TO-NOISE RATIO REQUIRED FOR LOCKIN
NEI NOISE EQUIVALENT IRRADIANCE
LT = TOTAL NUMBER OF FLARES IN FOV WITH IRRADIANCE ABOVE THRESHOLD
KT = INDICATES WHETHER OR NO TARGET IN FOV WITH IRRADIANCE GREATER THAN THRESHOLD
NF(J) = ARRAY STORING THE FLARE NUMBERS WHICH ARE ABOV THRESHOLD AND IN FOV



ARED SOURCES
, DETECTABLE
ESHOLD
: (S/N) (NEI)
X) J=1, LT
RESHOLD
(NEI)
OISE RATIO
LOCKIN
LENT
R OF FLARES
RRADIANCE
OLD
ETHER OR NOT
V WITH
REATER THAN
THE FLARE
CH ARE ABOVE
D IN FOV
CHECK INFRARED SOURCES ABOVE THRESHOLD
IF,
LT + KT = 0, ABORT
SELECT AIMPOINT TYPE
A MAXIMUM IRRADIANCE
B IRRADIANCE CENTROID
C GEOMETRIC CENTROID
A MAXIMUM IRRADIANCE
IF,
THEN,
OTHERWISE,
B IRRADIANCE CENTROID
C GEOMETRIC CENTRO

Fi

7

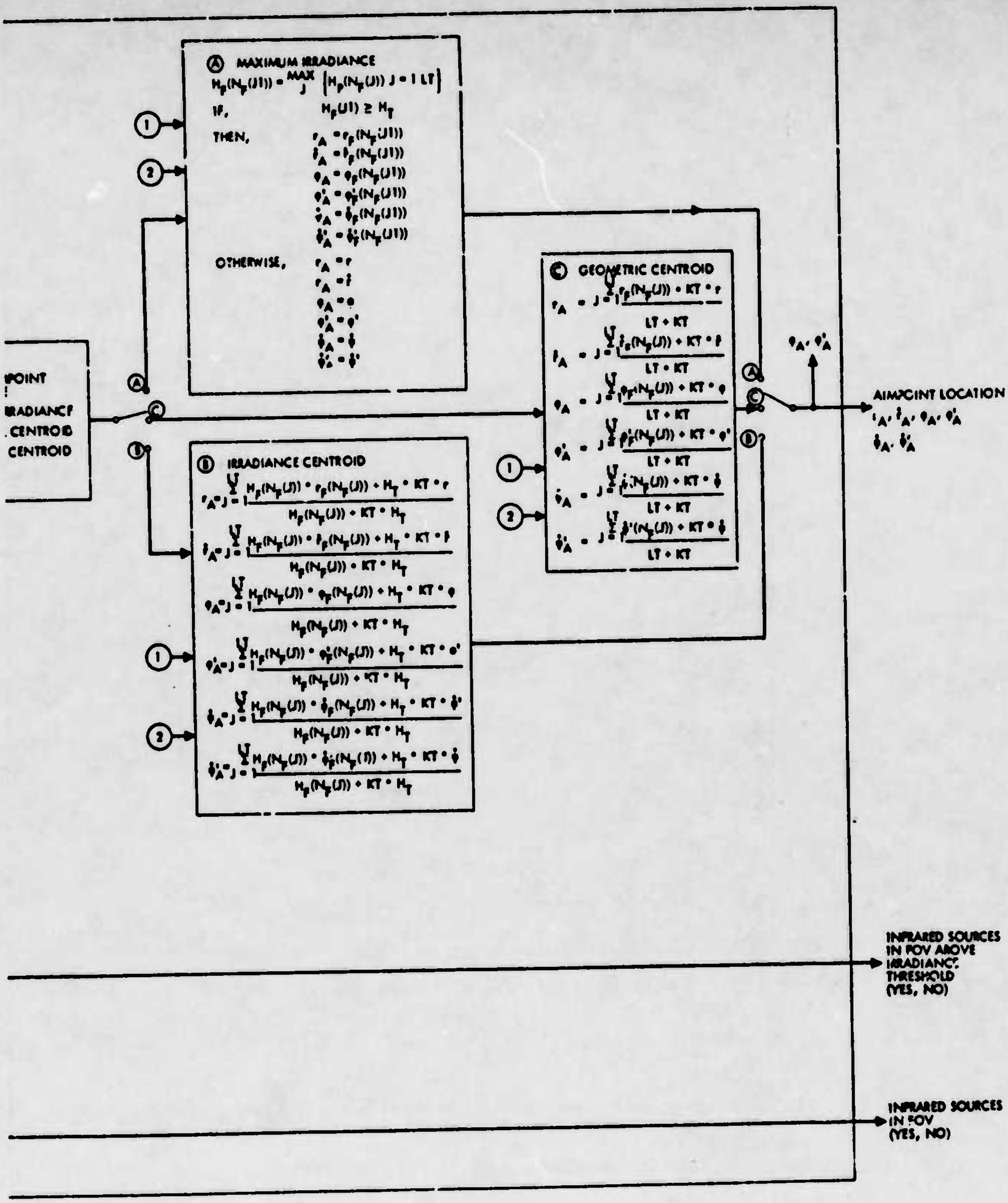


Figure 6-1. Missile aimpoint determination

3

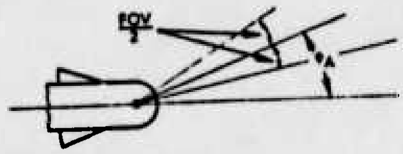
Table 6-1. Aimpoint determination subroutine

```
SUBROUTINE MSLAMPT     74/74    OPT=1                          FTN 4.2+P393       09/01/75  13.31.23.

          SUBROUTINE MSLAMPT(ZF,ZFP,Z,ZP,RELP,REL,HT,4.,HF,N,NF,KT,.T,SI,SI   MAR19       79
         *P,SIO,SIOP,LA,RADOT,FOV,L2,HMIN,.L,T)                               MSLAIMPT     3
          DIMENSION Z(16),ZP(16),ZF(3,20),ZFP(3,20),REL(12),RELP(4,20)        MSLAIMPT     4
          DIMENSION HF(20),NF(20),HP(20)                                      MSLAIMPT     5
          IF(T.GT.0.) GO TO 23                                                MSLAIMPT     6
          SI=Z(3)                                                             MSLAIMPT     7
          SIP=ZP(3)                                                           MSLAIMPT     8
       23 LTP=0                                                               MSLAIMPT     9
          IF(N.NE.1) GO TO 2                                                  MSLAIMPT    10
          GO TO 22                                                            MSLAIMPT    11
        2 DO 1 K=1,N                                                          MSLAIMPT    12
          IF((ABS(ZF(1,K)-SI).LE.FOV/2.).AND.(ABS(ZFP(1,K)-SIP).LE.FOV/2.))   MSLAIMPT    13
         *GO TO 4                                                             MSLAIMPT    14
          GO TO 1                                                             MSLAIMPT    15
        4 LTP=LTP+1                                                           MSLAIMPT    16
          NF(LTP)=K                                                           MSLAIMPT    17
        1 CONTINUE                                                            MSLAIMPT    18
       22 IF((ABS(Z(3)-SI).LE.FOV/2.).AND.(ABS(ZP(3)-SIP).LE.FOV/2.)) GO TO   MSLAIMPT    19
         *5                                                                   MSLAIMPT    20
          KTP=0                                                               MSLAIMPT    21
          GO TO 6                                                             MSLAIMPT    22
        5 KTP=1                                                               MSLAIMPT    23
        6 IF((LTP+KTP).NE.0) GO TO 7                                          MSLAIMPT    24
          WRITE(6,3) T                                                        MSLAIMPT    25
        3 FORMAT(/,2X,33HNO IR TARGETS IN THE FOV AT TIME ,F8.4)              MSLAIMPT    26
          WRITE(6,24) LTP,KTP                                                 MAR19       93
       24 FORMAT(1X,4HLTP=,I1,4HKTP=,I1)                                      MAR19       91
          L2=2                                                                MSLAIMPT    28
          RETURN                                                              MSLAIMPT    29
        7 LT=0                                                                MSLAIMPT    30
          KT=0                                                                MSLAIMPT    31
          M=0                                                                 MSLAIMPT    32
          ML=0                                                                MSLAIMPT    33
          JT=0                                                                MSLAIMPT    34
          DO 8 J=1,LTP                                                        MSLAIMPT    35
          IF(LTP.EQ.0)  GO TO 9                                               MAR19       92
          M=NF(J)                                                             MSLAIMPT    37
          IF(HF(M).LT.HMIN) GO TO 9                                           MSLAIMPT    38
          LT=LT+1                                                             MSLAIMPT    39
          HP(LT)=M                                                            MSLAIMPT    40
          MI=HF(M)                                                            MSLAIMPT    41
          IF(ML.GT.HF(M)) GO TO 9                                             MSLAIMPT    42
          ML=HF(M)                                                            MSLAIMPT    43
          JT=M                                                                MSLAIMPT    44
        9 IF(KTP.NE.1.OR.HT.LT.HMIN) GO TO 8                                  MSLAIMPT    45
          KT=1                                                                MSLAIMPT    46
        8 CONTINUE                                                            MSLAIMPT    47
          IF((LT+KT).NE.0) GO TO 10                                           MSLAIMPT    48
          WRITE(6,21)                                                         MSLAIMPT    49
       21 FORMAT(/,2X,29HNO IR SOURCES ABOVE THRESHOLD)                       MSLAIMPT    50
          WRITE(6,29) LT,LTP,KT,HC,HMIN                                       MAR19       87
       29 FORMAT(1X,3HLT=,I1,1X,4HLTP=,I1,1X,3HKT=,I1,1X,3HHT=,E12.5,1X,5HHM  MAR19       86
         *IN=,E12.6)                                                          MAR19       95
          L2=2                                                                MSLAIMPT    52
          RETURN                                                              MSLAIMPT    53
       10 IF(LT.EQ.0) GO TO 14                                                MSLAIMPT    54
          GO TO (11,12,13), LL                                                MSLAIMPT    55
```

(Table 6-1, concluded)

```
SUBROUTINE MSLAIMPT    74/74   OPT=1                    FTN 4.2+P380      05/01/75  13.31.23.

      11 A=1
         B=FLOAT(KT)
         C=FLOAT(LT-KT)
         GO TO 19
      12 B=FLOAT(KT)*MT
         C=FLOAT(KT)*MT+M
      16 RA=0
         RADOT=0.
         SI=0.
         SIP=0.
         SID=0.
         SIDP=0.
         DO 15 J=1,LT
         J1=MF(J)
         GO TO (17,18), LL
      18 A=AF(J1)
      17 RA=RA+RELF(7,J1)*A
         RADOT=RADOT+RELF(9,J1)*A
         SI=SI+ZF(1,J1)*A
         SIP=SIP+ZFP(1,J1)*A
         SID=SID+ZF(2,J1)*A
      15 SIDP=SIDP+ZFP(2,J1)*A
         RA=(RA+B*REL(10))/C
         RADOT=(RADOT+B*REL(11))/C
         SI=(SI+B*Z(3))/C
         SIP=(SIP+B*ZP(3))/C
         SID=(SID+B*Z(4))/C
         SIDP=(SIDP+B*ZP(4))/C
         ML=M
         GO TO 19
      13 IF(KT.EQ.0) GO TO 20
         IF(ML.GT.MT) GO TO 20
      14 RA=REL(10)
         RADOT=REL(11)
         SI=Z(3)
         SIP=ZP(3)
         SID=Z(4)
         SIDP=ZP(4)
         ML=MT
         GO TO 19
      20 RA=RELF(7,JTM)
         RADOT=RELF(9,JTM)
         SI=ZF(1,JTM)
         SIP=ZFP(1,JTM)
         SID=ZF(2,JTM)
         SIDP=ZFP(2,JTM)
      19 RETURN
         END
```

Table 6-2. Missile FOV



1. FLARES IN THE FOV

IF, $\left|\phi_F(K) - \phi_A\right| \le \frac{FOV}{2}$

AND

$\left|\phi_F'(K) - \phi_A'\right| \le \frac{FOV}{2}$

THEN, $N_F(J) = K$

$N_F(J) \triangleq$ ARRAY STORING THE FLARES WHICH ARE IN THE FOV

LT = TOTAL NO. OF FLARES IN THE FOV

2. TARGET IN THE FOV

IF, $\left|\phi - \phi_A\right| \le \frac{FOV}{2}$

AND

$\left|\phi - \phi_A'\right| \le \frac{FOV}{2}$

THEN, KT = 1 OTHERWISE, KT = 0

The aimpoint type can be based on

1. geometric centroid
2. irradiance centroid
3. maximum irradiance

of IR sources within the FOV above the minimum detectable irradiance level. In general, con-scan, FM signal processing missiles are max irradiance trackers and spin-scan AM signal processing missiles are irradiance centroid trackers.

The aimpoint determines the angles ψ_A, ψ_A', angle rates, $\dot{\psi}_A$, $\dot{\psi}_A'$ and the range rate $\dot{r}_A$ which represent the direction, direction rates and the range rate from the missile to an apparent target within the FOV. The equations used to calculate ψ_A, ψ_A', $\dot{\psi}_A$, $\dot{\psi}_A'$ and $\dot{r}_A$ for each aimpoint condition are shown in Table 6-3.

These aimpoint variables are fed back into the dynamics portion of the program to determine gyro position and rate and ultimately to determine the missile guidance commands.

Table 6-3. Missile aimpoint

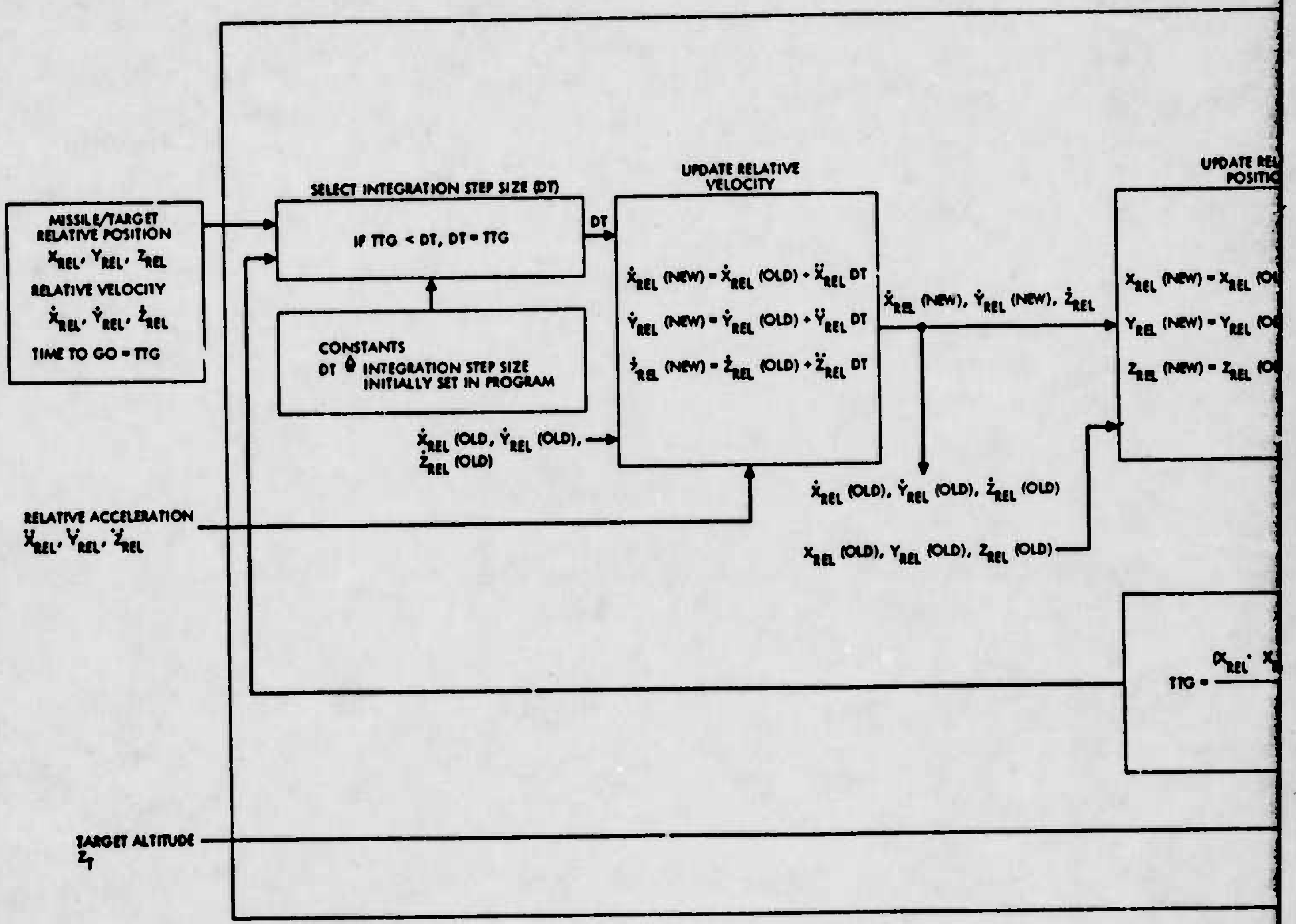
Geometric Centroid	Irradiance Centroid	Maximum Irradiance
$r_A = \dfrac{\sum_{J=1}^{LT} r_F(N_F(J)) + KT * r}{LT + KT}$	$r_A = \dfrac{\sum_{J=1}^{LT} H_F(N_F(J)) * r_F(N_F(J)) + H_T * KT * r}{H_F(N_F(J)) + KT * H_T}$	$H_F(N_F(J1)) = \underset{J}{MAX} \{H_F(N_F(J))\ J = 1, LT\}$ If, $H_T(J1) \geq H_T$ Then, $r_A = r_F(N_F(J1))$ $\dot{r}_A = \dot{r}_F(N_F(J1))$ $\psi_A = \psi_F(N_F(J1))$ $\psi'_A = \psi'_F(N_F(J1))$ $\dot{\psi}_A = \dot{\psi}_F(N_F(J1))$ $\dot{\psi}'_A = \dot{\psi}'_F(N_F(J1))$ Otherwise, $r_A = r$ $r_A = \dot{r}$ $\psi_A = \psi$ $\psi'_A = \psi'$ $\dot{\psi}_A = \dot{\psi}$ $\dot{\psi}'_A = \dot{\psi}'$
$\dot{r}_A = \dfrac{\sum_{J=1}^{LT} \dot{r}_F(N_F(J)) + KT * \dot{r}}{LT + KT}$	$\dot{r}_A = \dfrac{\sum_{J=1}^{LT} H_F(N_F(J)) * \dot{r}_F(N_F(J)) + H_T * KT * \dot{r}}{H_F(N_F(J)) + KT * H_T}$	
$\psi_A = \dfrac{\sum_{J=1}^{LT} \psi_F(N_F(J)) + KT * \psi}{LT + KT}$	$\psi_A = \dfrac{\sum_{J=1}^{LT} H_F(N_F(J)) * \psi_F(N_F(J)) + H_T * KT * \psi}{H_F(N_F(J)) + KT * H_T}$	
$\psi'_A = \dfrac{\sum_{J=1}^{LT} \psi'_F(N_F(J)) + KT * \psi'}{LT + KT}$	$\psi'_A = \dfrac{\sum_{J=1}^{LT} H_F(N_F(J)) * \psi'_F(N_F(J)) + H_T * KT * \psi'}{H_F(N_F(J)) + KT * H_T}$	
$\dot{\psi}_A = \dfrac{\sum_{J=1}^{LT} \dot{\psi}(N_F(J)) + KT * \dot{\psi}}{LT + KT}$	$\dot{\psi}_A = \dfrac{\sum_{J=1}^{LT} H_F(N_F(J)) * \dot{\psi}_F(N_F(J)) + H_T * KT * \dot{\psi}}{H_F(N_F(J)) + KT * H_T}$	
$\dot{\psi}'_A = \dfrac{\sum_{J=1}^{LT} \dot{\psi}'(N_F(J)) + KT * \dot{\psi}'}{LT + KT}$	$\dot{\psi}'_A = \dfrac{\sum_{J=1}^{LT} H_F(N_F(J)) * \dot{\psi}'_F(N_F(J)) + H_T * KT * \dot{\psi}'}{H_F(N_F(J)) + KT * H_T}$	

7. CLOSEST APPROACH COMPUTATION

This subroutine is called when the missile no longer has a source within the seeker FOV. The missile and target trajectories are projected foreward in time, assuming missile and target accelerations remain constant, at the last value, before loss of tracking.

Figure 7-1 shows a block diagram of the computation. The trajectory is projected forward until the missile passes the target (normal termination), begins to diverge, hits the ground, or exceeds the maximum missile lifetime.

The computed time to go (TTG) is the time remaining to closest approach assuming constant missile and target velocities. It is equal to the projection of the range in the relative velocity direction, divided by the magnitude of the relative velocity. Table 7-1 gives the closest approach computation subroutine.


MISSILE/TARGET RELATIVE POSITION
X_{REL}, Y_{REL}, Z_{REL}
RELATIVE VELOCITY
$\dot{X}_{REL}$, $\dot{Y}_{REL}$, $\dot{Z}_{REL}$
TIME TO GO = TTG
SELECT INTEGRATION STEP SIZE (DT)
IF TTG < DT, DT = TTG
DT
CONSTANTS
DT ≜ INTEGRATION STEP SIZE INITIALLY SET IN PROGRAM
UPDATE RELATIVE VELOCITY
$\dot{X}_{REL}$ (NEW) = $\dot{X}_{REL}$ (OLD) + $\ddot{X}_{REL}$ DT
$\dot{Y}_{REL}$ (NEW) = $\dot{Y}_{REL}$ (OLD) + $\ddot{Y}_{REL}$ DT
$\dot{Z}_{REL}$ (NEW) = $\dot{Z}_{REL}$ (OLD) + $\ddot{Z}_{REL}$ DT
$\dot{X}_{REL}$ (OLD, $\dot{Y}_{REL}$ (OLD), $\dot{Z}_{REL}$ (OLD)
$\dot{X}_{REL}$ (NEW), $\dot{Y}_{REL}$ (NEW), $\dot{Z}_{REL}$
$\dot{X}_{REL}$ (OLD), $\dot{Y}_{REL}$ (OLD), $\dot{Z}_{REL}$ (OLD)
X_{REL} (OLD), Y_{REL} (OLD), Z_{REL} (OLD)
RELATIVE ACCELERATION
$\ddot{X}_{REL}$, $\ddot{Y}_{REL}$, $\ddot{Z}_{REL}$
UPDATE REL
POSITIO
X_{REL} (NEW) = X_{REL} (OL
Y_{REL} (NEW) = Y_{REL} (O
Z_{REL} (NEW) = Z_{REL} (O
TTG =
TARGET ALTITUDE
Z_T

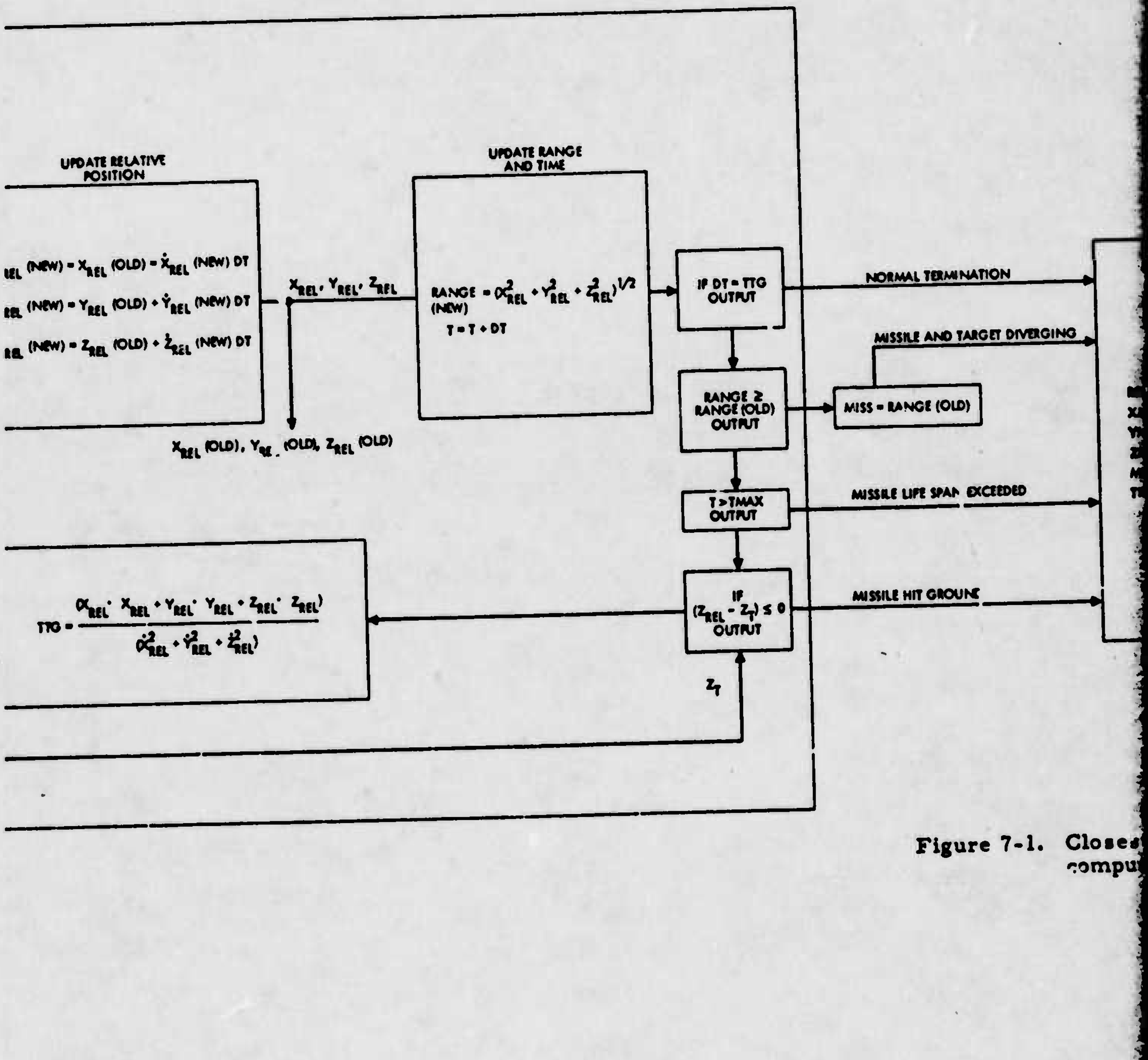


Figure 7-1. Closes[t] compu[t]

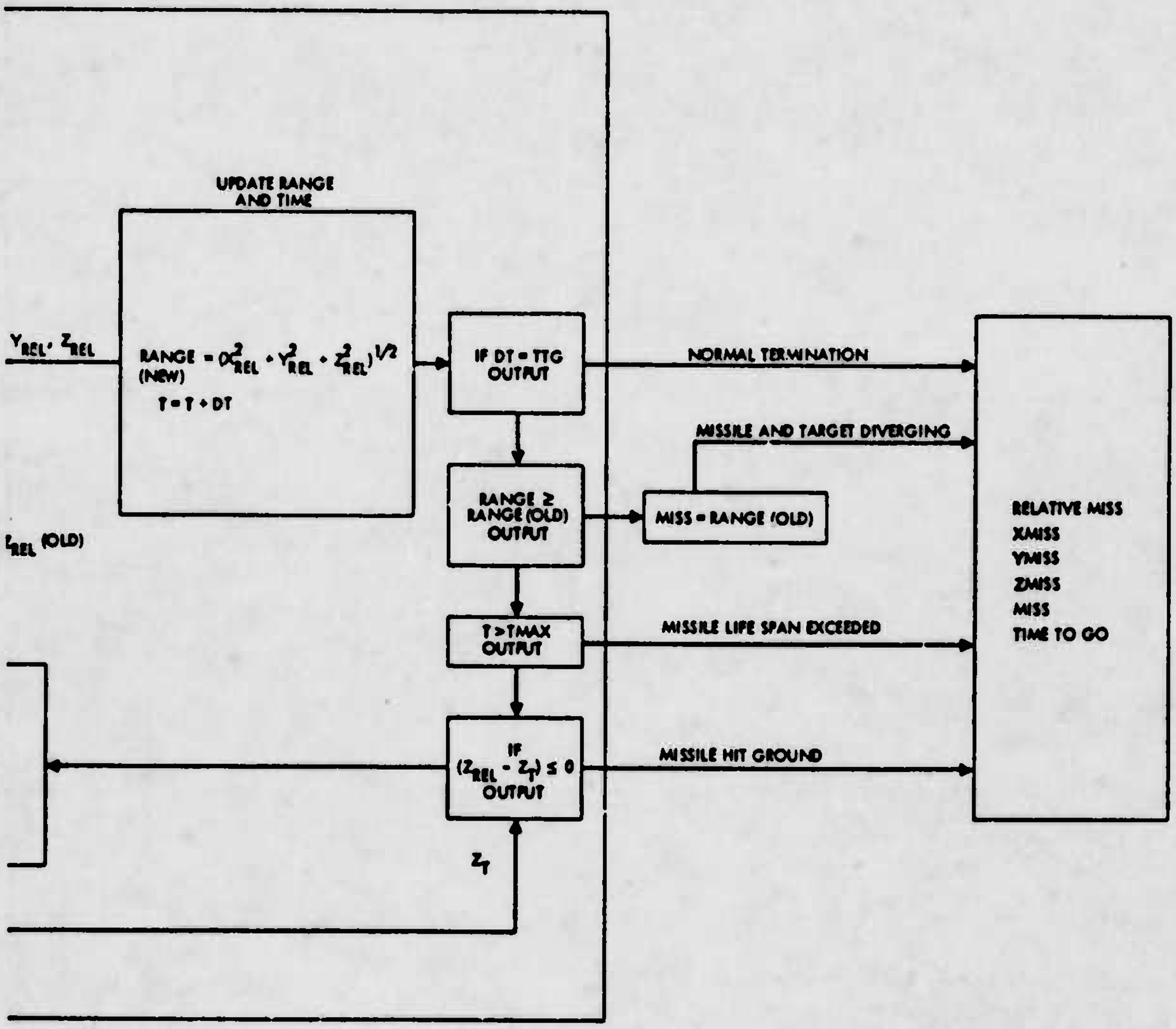


Figure 7-1. Closest approach computation

Table 7-1. Closest approach computation

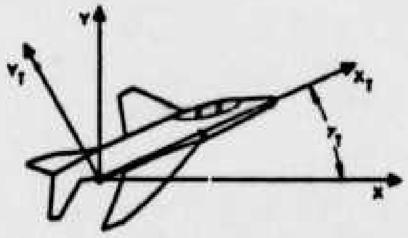
```
SUBROUTINE CLOSAPP    74/74    OPT=1                 FTN 4.2+P384        09/01/75  13.31.49.

        SUBROUTINE CLOSAPP(REL,TTG,DELT,RMISS,XP,T,TMAX,XM,YM,ZM,ITGT(
        DIMENSION REL(12),XP(18)
        XM=REL(1)
        YM=REL(2)
        ZM=REL(3)
        TTGT=TTG
        IT=1
      5 IF(TTG.GE.DELT) GO TO 1
        DELT=TTG
      1 REL(4)=REL(4)+REL(7)*DELT
        REL(5)=REL(5)+REL(8)*DELT
        REL(6)=REL(6)+REL(9)*DELT
        REL(1)=REL(1)+REL(4)*DELT
        REL(2)=REL(2)+REL(5)*DELT
        REL(3)=REL(3)+REL(6)*DELT
        RSQR=REL(1)*REL(1)+REL(2)*REL(2)+REL(3)*REL(3)
        VRSQ=REL(4)*REL(4)+REL(5)*REL(5)+REL(6)*REL(6)
        RDRDT=REL(1)*REL(4)+REL(2)*REL(5)+REL(3)*REL(6)
        T=T+DELT
        CMISS=SQRT(RSQR)
        IF(DELT.EQ.TTG) GO TO 2
        TTG=-RDRDT/VRSQ
        IF(CMISS.GT.RMISS.AND.IT.EQ.1) GO TO 6
        IF(CMISS.GT.RMISS.AND.IT.GT.1) GO TO 2
        IF(T.GT.TMAX) GO TO 7
        RMISS=CMISS
        IT=IT+1
        IF((REL(9)-XP(9)).GT.0.) GO TO 5
        WRITE(6,8)
      8 FORMAT(1X,19HMISSILE HIT GROUND)
        GO TO 2
      6 WRITE(6,3)
      3 FORMAT(1X,29HMISSILE AND TARGET DIVERGING)
        GO TO 2
      7 WRITE(6,9)
      9 FORMAT(1X,25HMISSILE LIFE SPAN EXCEEDED)
      2 RETURN
        END
```

8. PROBABILITY OF HIT

It is necessary when evaluating thousands of computer runs (1) to use the probability of hit (P_H) as the only measure of effectiveness in the simulation and (2) to have a simple means of computing P_H so as to keep overall program complexity and computation time to a minimum. The approach taken here to calculate P_H utilizes the following assumptions and definitions:

1. The missile aimpoint is located at the geometric centroid of all the aircraft's tailpipes and thus the point of missile closest approach to the aircraft is relative to the tailpipe.
2. The tailpipes of the aircraft are symmetrically located about the vertical and wing axes of the aircraft.
3. The point of closest approach of the missile to the aircraft is defined to be the warhead detonation point.
4. Warhead detonation inside a volume defined by the aircraft dimensions will have a probability of hit (P_H) equal to one.
5. If the missile is a hit-to-kill missile, a detonation outside this volume will have a $P_H = 0$.
6. For proximity fused missiles, a warhead lethality zone around the aircraft volume will be assumed.
7. A warhead (proximity fused) detonation outside this lethality zone will have a $P_H = 0$. A detonation between the two zones will have P_H linearly proportional to the detonation point distance from the aircraft volume.

The missile, target, and flare simulation program provides miss distance information in inertial coordinates, therefore, it is necessary to perform a coordinate transformation to obtain the miss distance in terms of aircraft coordinates. Figure 8-1 shows the equations used to perform this transformation with the coordinates of the miss vector being (XMISS, YMISS, ZMISS) in the inertial system and (XRT, YRT, ZRT) in the aircraft system.



$$XRT = XMISS * \cos \gamma_T - YMISS * \sin \gamma_T$$
$$YRT = -XMISS * \sin \gamma_T + YMISS * \cos \gamma_T$$
$$ZRT = ZMISS$$

Figure 8-1. Coordinate transformation, inertial-to-aircraft coordinate

The aircraft coordinate system has the X_T and Z_T axes along the longitudinal and vertical axes of the aircraft and X_T axis along the aircraft wing. This coordinate system shown in Figure 8-2 has the tailpipe at its center and the aircraft dimensions defined relative to this point.

For proximity fused missiles, a warhead lethality zone around the aircraft volume is assumed. This zone is simply determined by adding to each aircraft dimension the warhead's effective kill radius (M_R).

If warhead detonation occurs within the aircraft volume, $P_H = 1$ is assumed; if it lies outside the warhead lethality zone, $P_H = 0$ is assumed. If warhead detonation lies between the two zones, P_H is assumed to be linearly proportional to the distance from the outer boundary of the aircraft volume to the detonation point. For the case of hit-to-kill missiles, the missile effective kill radius (M_R) is set equal to zero making the aircraft volume and warhead lethality zone coincident.

The equations and logic required to implement this calculation are as follows:

If $\quad (X_N + M_R) \leq XRT \text{ and } XRT \leq -(-X_B + M_R)$

Then, $\quad P_X = 0$

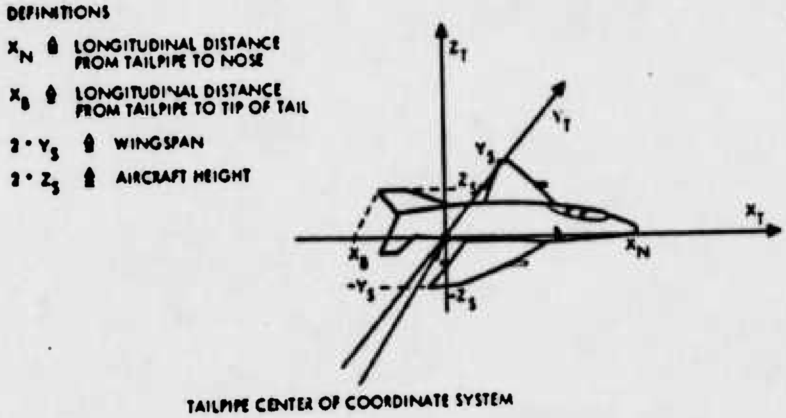


Figure 8-2. Aircraft coordinate system

If $-(X_B) \le XRT \le X_N$

Then, $P_X = 1$

If $XRT \ge X_N$

Then, $$PX = \frac{(X_N + M_R) - XRT}{M_R}$$

Otherwise, $$P_X = \frac{(X_B + M_R) + XRT}{M_R}$$

If $ABS\ (YRT) \ge (Y_S + M_R)$

Then, $P_Y = 0$

If $ABS\ (YRT) \ge Y_S$

Then, $P_Y = 1$

Otherwise, $P_Y = \frac{(Y_S + M_R) - ABS\ (YRT)}{M_R}$

If ABS (ZRT) $\geq (Z_S + M_R)$

Then, $P_Z = 0$

If ABS (ZRT) $\leq Z_S$

Then, $P_Z = 1$

Otherwise, $P_Z = \frac{(Z_S + M_R) - ABS\ (ZRT)}{M_R}$

Finally, $P_H = P_X * P_Y * P_Z$

Figure 8-3 shows a block diagram of these computations. Table 8-1 gives the probability of hit subroutine.

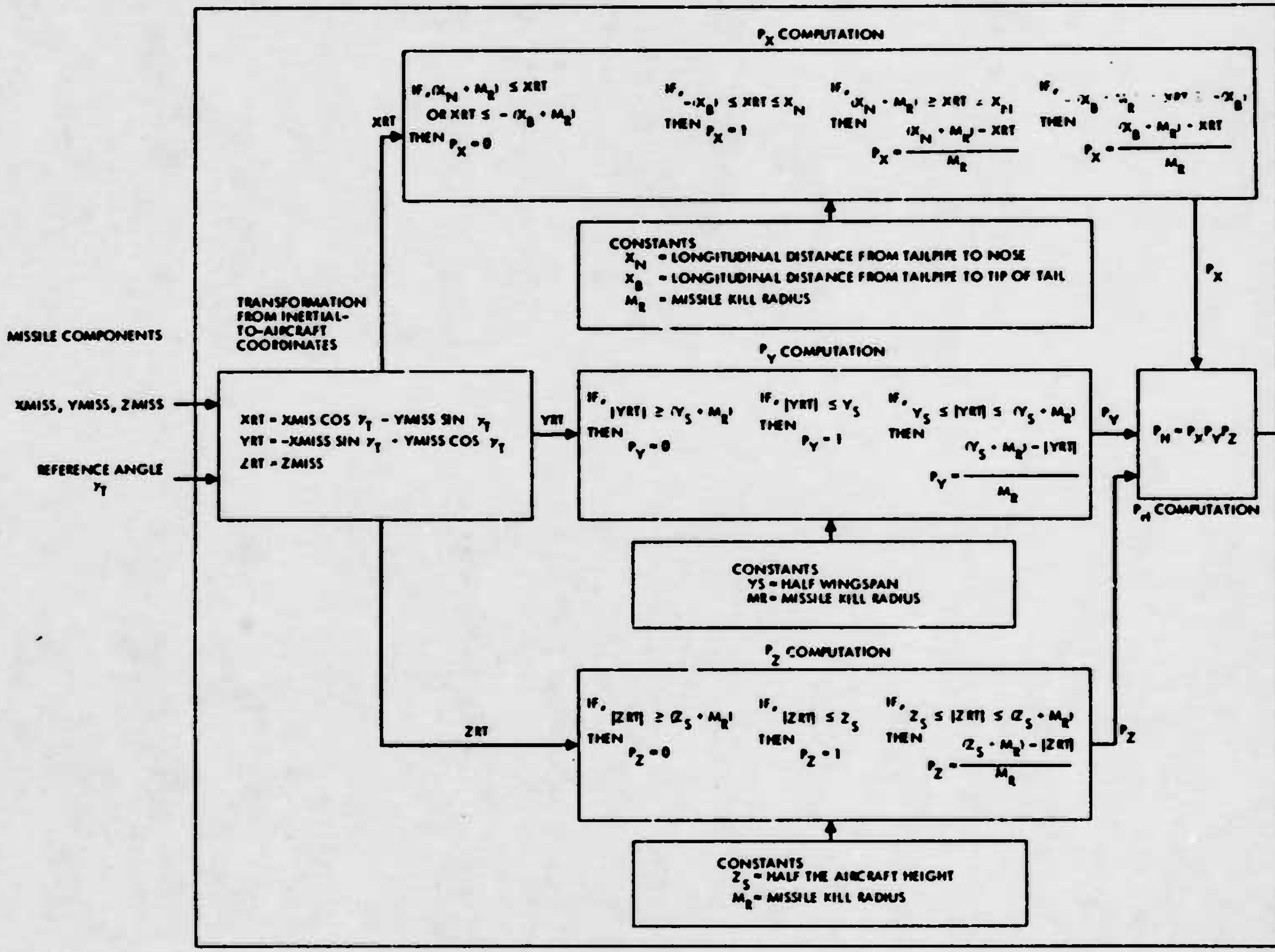


Figure 8-3. Probability of hit computation

Table 8-1. Probability of hit subroutine

```
SUBROUTINE PHCOMP     74/74   OPT=1                                  FTN 4.2+P380        05/21/75  13.31.51.

      SUBROUTINE PHCOMP(REL,GAMT,RMK,YS,XB,XM,ZS,PH)
      DIMENSION REL(12)
      XM=REL(1)
      YM=REL(2)
      ZM=REL(3)
      XRT=-1.*(XM*COS(GAMT)-YM*SIN(GAMT))
      YRT=-1.*(XM*SIN(GAMT)+YM*COS(GAMT))
      ZRT=-1.*ZM
      IF(XRT.LT.(XM+RMK)) GO TO 2
      PX=0.
      GO TO 1
    2 IF(XRT.GT.-(XB+RMK)) GO TO 3
      PX=0.
      GO TO 1
    3 IF(XRT.LT.-XB.OR.XRT.GT.XM) GO TO 4
      PX=1.
      GO TO 1
    4 IF(XRT.LT.XM) GO TO 5
      PX=(XM+RMK-XRT)/RMK
      GO TO 1
    5 PX=(XB+RMK+XRT)/RMK
    1 IF(ABS(YRT).LT.(YS+RMK)) GO TO 6
      PY=0.
      GO TO 7
    6 IF(ABS(YRT).GT.YS) GO TO 8
      PY=1.
      GO TO 7
    8 PY=(YS+RMK-ABS(YRT))/RMK
    7 IF(ABS(ZRT).LT.(ZS+RMK)) GO TO 9
      PZ=0.
      GO TO 10
    9 IF(ABS(ZRT).GT.ZS) GO TO 11
      PZ=1.
      GO TO 10
   11 PZ=(ZS+RMK-ABS(ZRT))/RMK
   10 PH=PX*PY*PZ
      RETURN
      END
```

9. SPECTRAL INTEGRATOR

In order to utilize the I/R target signature generated by ASDIR II* (or any other computer program which can generate apparent spectral radiant intensity $J_\lambda T_\lambda$) it is necessary to integrate $J_\lambda T_\lambda$ over the optical waveband of the missile simulated in the M/T/CM program. This integration is accomplished by use of an auxiliary routine SPKINT, which can integrate over any desired spectral region of the ASDIR-II output (maximum of 50).

Two options are available in SPKINT which can be selected at run time and extend the usefulness of this routine. These include:

(1) An atmospheric transmission table can be read in at run time and used in the integration

(2) A spectral filter can be read in and applied to the integration.

Using the optional atmospheric table and/or filter table, the integrated $J_\lambda T_\lambda$ can be determined as a function of range or, if these tables are omitted, the values of $J_\lambda T_\lambda$ as a function of range from the ASDIR-II output are used. Figure 9-1 is a block diagram of the SPKINT program and Table 9-1 is a program listing.

---

*Stone, C.W., Capt. USAF and Tate, Stanley, Planning ASDIR-II (Vols I, II, III) Deputy for Development, Aeronautical Systems Division, Wright Patterson Air Force Base, Ohio, ASD/XR-TR-75-1, January 1975.

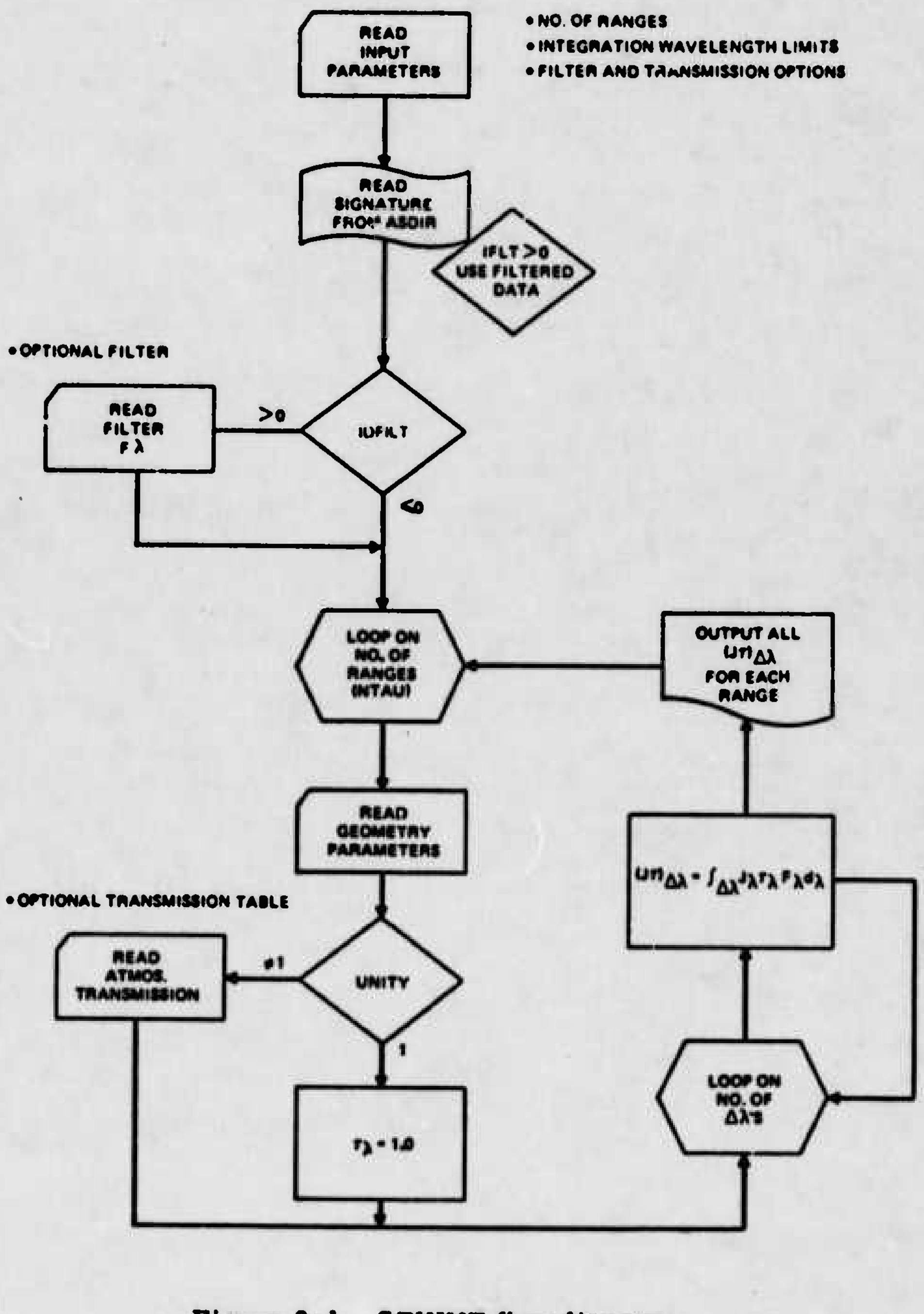


Figure 9-1. SPKINT flow diagram

Table 9-1. SPKINT program listing

```
      COMMON SFSM(50), SFTS(50), TAIS(50), WA(670), TRAN(670), ATMID(3),   0010
     *       TWT(670), TX(300), WLA(50), WLB(50), TEF(50), AIRM(50),        0020
     *       AATAN(670), TEMTAR(670), OSR(670), OSWL(100), OSRT(100),       0030
     *       STB(300), WTG(10,300), TAR(10,300), ANAM(6), LABEL(10,20),     0040
     *       NFREQ(10),RAD(10),FTAR(10,300)
C                                                                             300
C INITIALIZE DATA                                                             310
C                                                                             320
      INTEGER UNITY,SOURCE                                                   0325
      REWIND 3; REWIND 4; REWIND 7                                            330
      REWIND 10
      PI = 3.14159                                                            350
  555 READ(5,30) NSRG,IDSRT,NPRTY,MPRTY,KPLTY,KPUTAU,ITAUW,IFLT,
     *UNITY,SOURCE
      IF(NSRG.EQ.0) STOP
      OUTPUT NSRG,IDSRT,NPRTY,ITAUW,IFILT
C                                                                             360
C READ NTAU, LABEL, INPUT PARAMETERS                                          370
C                                                                             380
      IF (UNITY.EQ.1) READ(5,5) NTAU; WRITE(3,5) NTAU; GO TO 6
      IF(ITAUW.GT.0) READ(5,5) NTAU
      IF(ITAUW.LE.0) READ(3,5) NTAU
    5 FORMAT(8I5)                                                             400
    6 CONTINUE
      OUTPUT NTAU
      READ(5,10) (ANAM(I), I=1,6)                                             410
   10 FORMAT(20A4)                                                            420
      WRITE(6,20) (ANAM(I), I=1,6)                                            430
   20 FORMAT(1H1/32X,'NON-EQUISPACED SPECTRAL INTEGRATOR'//58X,'CASE'/61
     *X,6A4///)                                                               450
      IF(ITAUW.GT.0) READ(3) NFREQ(1); READ(3) (WA(I),TRAN(I), I = 1,
     *NFREQ(1)); GO TO 25
   25 CONTINUE
      NWL =NFREQ(1)
      WRITE(10,10) (ANAM(I), I=1,6)                                           530
      WRITE(10,30) KPLTY                                                      540
C                                                                             550
C READ INTEGRATION LIMITS (5 PAIRS/CARD)                                      560
C                                                                             570
      READ(5,50) (WLA(I), WLB(I), I=1,NSRG)                                   580
   50 FORMAT(10F6.0)                                                          590
C                                                                             600
C READ TARGETS (FILE 7, STORE FOR PLOTS ON FILE 10                            610
   35 READ(7,END=36) KT,(LABEL(KT,I), I = 1,20),NFREQ(KT)
       KTT = KT
      IF(KT.GT.10) OUTPUT'NO. OF TARGETS EXCEEDS 10'; STOP                    520
      IF(KT.LE.0) OUTPUT 'NO TARGETS'; STOP                                   630
      OUTPUT KT,NFREQ(KT)
      OUTPUT '==============================================='
      NTLT= NFREQ(KT)
      DO 60 I = 1,NTLT
      READ(7) WTG(KT,I), TAR(KT,I),FTAR(KT,I)
   30 FORMAT(24I3)                                                            470
      IF(IFLT.GT.0) TAR(KT,I) = FTAR(KT,I)
   60 CONTINUE
      READ(7) RAD(KT)
      IF(NPRTY.NE.0) WRITE(6,209) (LABEL(KT,I), I=1,20)
      IF(NPRTY.NE.0) WRITE(6,212)(WTG(KT,I),TAR(KT,I),I=1,NTLT)
  209 FORMAT(20A4)
```

(Table 9-1, continued)

```
      212 FORMAT(10HTOT LAMBOA  ,11X,4H TOT ,20X,'TOT LAMBOA',11X,' TOT'/(5X
     *,F7.4,4X,E12.4,10X,F7.4,4X,E12.4))
          OUTPUT !--.-----------------------------------------!
          IF(KPLTY.EQ.0) GO TO 54
          IF((KT.NE.1).AND.(SOURCE.GT.0)) GOTO 54
          WRITE(10,10) (LABEL(KT,J), J = 1,20)
          WRITE(10,55) NTLT,(WTG(KT,J),TAR(KT,J), J = 1,NTLT)
   55     FORMAT(I5/(6E12.5))                                          740
   54     CONTINUE
          GO TO 35
   36     CONTINUE
          KT = KTT
          IF(SOURCE.GT.0)KT=1
          WRITE(4,40) NTAU, KT, NSRG, KT                               500
   40     FORMAT(4I5)
          WRITE(4,446) (WLA(I),WLB(I), I = 1,NSRG)
  446     FORMAT(10F6.4)
C READ DETECTOR RESPONSE (OPTIONAL)                                    770
C                                                                      780
          IF(IOSRT.EQ.0) GO TO 80                                      790
          READ(5,70) (DSWL(I),DSRT(I), I=1,IDSRT)                      800
   70     FORMAT(5(F6.3,F6.4))                                         810
          WRITE(6,75) (DSWL(I),DSRT(I), I=1,IOSRT)                     820
   75     FORMAT(1H0,T47,'INPUT',26X,'INPUT'/40X,'WAVELENGTH',20X,'DETECTOR 830
         *RESPONSE'//(43X,F7.4,25X,E15.5))                             840
   80     CONTINUE                                                     850
C                                                                      860
C MAIN  LOOP ON NTAU                                                   870
C-------------------------------                                      880
          DO 500 J=1,NTAU                                              890
          IF(UNITY.EQ.1) READ(5,2) ATMID,HO,HS,SR,OVIR
          IF(ITAUW.GT.0) GO TO 111
          IF(UNITY.NE.1) GO TO 85                                      0894
          NWL = NTLT                                                   0896
          DO 81 I=1,NTLT                                               0898
          TRAN(I) = 1.0                                                0899
   81     WA(I) = WTG(1,I)                                             0902
    2     FORMAT(3A4,4E10.3)
        . WRITE(3,90) NWL,ATMIO,HC,HS,SR,OVIR
          NM = NWL/5
          IF(NWL.GT.(NM*5)) NM=NM+1
          MA=1
          MB=5
          DO 82 MMB=1,NM
          WRITE(3,95) (WA(MC),TRAN(MC), MC= MA,MB)
          MA = MA+5
   82     MB=MB+5
          GO TO 111                                                    0904
   85     READ(3,90) NWL,ATMIO,HO,HS,SR,OVIR                           0906
   90     FORMAT(I5,3A4,5X,F8.2,6X,F8.2,5X,F8.0,6X,F4.0)               910
          NM = NWL/5                                                   920
          IF(NWL.GT.(NM*5)) NM = NM+1                                  930
C                                                                      932
C READ  WAVELENGTH-TRANS. TABLE                                        933
C                                                                      936
          MA = 1                                                       940
          MB = 5                                                       950
          DO 100 MMB = 1,NM
          READ(3,95) (WA(MC),TRAN(MC), MC=MA,MB)                       970
   95     FORMAT(5(F6.3,F6.4))                                         9[illegible]
```

(Table 9-1, continued)

```
      MA = MA+5                                                        990
  100 MB = MB+5                                                       1000
  111 CONTINUE
      IF(MPNTY.NE.0) WRITE(6,110) J, (WA(I), TRAN(I), I=1,NWL)       1010
  110 FORMAT(1H1/I3,34X,'INPUT WAVELENGTH VS. TRANSMISSION '//(4(F10.4,   1020
     *'(U)',E14.5)))                                                  1030
      WRITE(10,40) KT,NWL                                             1030
      IF(IDSNT.EQ.0) GO TO 130                                        1031
      DO 120 I=1,NWL                                                  1032
  120 DSR(I) = TLU2(WA(I),DSw! ,OS*T)                                 1033
  130 CONTINUE                                                        1034
C                                                                     1040
C LOOP ON NUMBER OF TARGETS                                           1050
C                                                                     1060
      K = J
      IF(SOURCE.GT.0)K=1
      NTLT = NFREQ(K)                                                 1080
      WRITE(10,10) (LABEL(K,I), I=1,20)                               1090
      OUTPUT WTG(K,1),WA(1),WTG(K,NTLT),WA(NWL)
      OUTPUT '-------------------------------------------------'
      IF(WTG(K,1).LT.WA(1)) OUTPUT'LOW END OF TGT LESS THAN LAMBDA(1)';   1100
     *GO TO 30C                                                       1110
      IF(WTG(K,NTLT).GT.WA(NWL))OUTPUT 'HIGH END OF TGT GREATER THAN LAM   1120
     *BDA(NWL)'; STOP                                                 1130
      IF(UNITY.NE.1) GO TO 135                                        1131
      DO 132 I=1,NTLT                                                 1132
  132 TWT(I) = TAN(K,I)
      K3=1;K4=NTLT                                                    1134
      GO TO 155                                                       1135
  135 CONTINUE                                                        1139
      DO 140 I=1,NTLT                                                 1140
      STG(I) = WTG(K,I)                                               1150
  140 TX(I) = TAR(K,I)                                                1160
      DO 150 I=1,NWL                                                  1170
  150 TWT(I) = TLU2(WA(I),STG,TX)                                     1180
      W1 = WTG(K,1) ; W2 = WTG(K,NTLT)                                1190
      CALL SCRIPT(W1,W2,K3,K4,WA,NWL)                                 1200
  155 CALL TRAP(K3,K4,WA,TWT,SUM)
       AJMAX = RAD(K)/SUM
      WRITE(6,170)                                                    1290
  170 FORMAT(///50X,'FRACTION OF TANGET PLUME IN BAND'///50X,'WAVELENGTH   1300
     * REGION',7X,'F JTG'/)
      DO 180 I=1,NSRG                                                 1320
      W1 = WLA(I)                                                     1240
      W2 = WLB(I)                                                     1250
      CALL SCRIPT(W1,W2,KA,KB,WA,NWL)                                 1260
      CALL TRAP(KA,KB,WA,TWT,SUMT)                                    1270
      SFTS(I) = SUMT/SUM                                              1280
      WRITE(6,175)WA(KA),WA(KB),SFTS(I)
  175 FORMAT(50X,F6.3,'-TO-',F6.3,4X,E12.5)                           1330
  180 CONTINUE                                                        1340
      DO 190 I=1,NWL                                                  1350
      AJMAX = 1.
  190 TEMTAR(I) = TWT(I)*AJMAX
      DO 200 I=1,NSRG                                                 1370
      W1 = WLA(I); W2 = WLB(I)
      CALL SCRIPT(W1,W2,KA,KB,WA,NWL)                                 1390
      CALLTRAP(KA,KB,WA,TEMTAR,SUMF)                                  1400
      SFSM(I) =SUMF                                                   1410
  200 IF(SFSM(I).LE.0) SFSM(I) = 1.                                   1420
```

(Table 9-1, continued)

```
      D0 210 I•1,NWL                                                    1430
      TEMTAR(I) • TEMTAR(I) • THAN(I)                                   1440
  210 IF(IDSRT.GT.0) TEMTAR(I) • TEMTAR(I)•DSH(I)                       1450
      WRITE(6,219)
  219 FORMAT(1X)
      WRITE(6,220) ATMID,HO,HS,SR,OVIR                                  1460
  220 FORMAT(22X,3A4,7X,'HO • ',F8.2,' M',5X,'HS • ',F8.2,' M',5X,'SR • 1470
     •',F8.0,' M',5X,'VH • ',F4.0,' KM'//)                              1480
      WRITE(6,230) RAD(K),WTG(K,1),WTG(K,NTLT),(LABEL(K,I),I•1,20),AJMAX 1500
  230 FORMAT(15X,'RAD • ',E12.4,' IN THE WAVELENGTH REGION',F8.4,' TO', 1510
     •F8.4,' MICRONS'//18X,'TARGET TYPE IS ',20A4//18X,'JSCAL • ',E12.4/ 1520
     •//)                                                               1830
      IF(SR.EQ.0.) WRITE(6,235); GO TO 244
  235 FORMAT(/12X,'WAVELENGTH REGION',10X,'J TAU',10X,'TAU EFF',10X,'TAU 1550
     • (TOTAL',9X/16X,'(MICRONS'//)
      WRITE(6,240)                                                      1540
  240 FORMAT(/12X,'WAVELENGTH REGION',10X,'J TAU',10X,'TAU EFF',10X,'TAU 1550
     • (TOTAL)',9X,'IRRADIANCE'/16X,9H(MICRONS)//)                      1560
  244 WRITE(10,245) (WA(I),TEMTAN(I), I • 1,NWL)
  245 FORMAT(5E14.6)                                                    1590
  250 CONTINUF                                                          1600
      D0 260 I•1,NSRG                                                   1610
      W1• WLA(I); W2•WLB(I)                                             1620
      CALL SCRIPT(W1,W2,KA,KB,WA,NWL)                                   1630
      CALL TRAP(KA,KB,WA,TEMTAR,SUM)
      AATAR(I) • SUM                                                    1650
      TEF(I) • SUM/SFSM(I)                                              1660
      TAIB(I) • TEF(I) • SFTS(I)                                        1670
      RS • SR•100.                                                      1680
      IF(RS.EQ.0)WRITE(6,256)WA(KA),WA(KB),AATAR(I),TEF(I),
     •TAIB(I);GO TO 260
  256 FORMAT(12X,F6.3,'-TO-',F6.3,7X,E12.5,7X,F9.7,10X,F9.7)
      AIRR(J) • AATAR(I)/RS/RS                                          1690
      WRITE(6,255)WA(KA),WA(KB),AATAR(I),TEF(I),TAIB(I),AIRR(J)
  255 FORMAT(12X,F6.3,'-TO-',F6.3,7X,E12.5,7X,F9.7,10X,F9.7,7X,E12.8)   1710
  260 CONTINUE                                                          1720
      IF(KPJTAJ.EQ.1) WRITE(6,265) (WA(I),TEMTAN(I),I•1,NWL)            1730
  265 FORMAT(1H1,42X,18HLAMBDA VS. J TAU   //(6E18.8))                  1740
  300 CONTINUE
      WRITE(6,350)                                                      1760
  350 FORMAT(1H1)                                                       1770
  800 CONTINUE                                                          1780
      GO TO 555
      END                                                               1790
```

Table 9-1 (concluded)

```
      FUNCTION TLU2(A,X,F)
      DIMENSION X(1),F(1)
      IF(A.LT.X(1)) TLU2=F(1);RETURN
      DO 1 I=1,50000
      IF(X(I).GT.X(I+1)) TLU2=F(I);RETURN
      IF(A.GE.X(I).AND.A.LE.X(I+1)) GO TO 2
    1 CONTINUE
    2 V=(A-X(I))/(X(I+1)-X(I))
      TLU2=F(I)+V*(F(I+1)-F(I))
      RETURN
      END
```

```
      SUBROUTINE SCRIPT(W1,W2,L1,L2,WAVE,NWL)
      DIMENSION WAVE(1)
      DO 5 I= 1,NWL
      IF(WAVE(I).GT.W1) GO TO 10
    5 CONTINUE
   10 L1 = I - 1
      IF(L1.LT.1) L1 = 1
      DO 20 I=1,NWL
      IF(WAVE(I).GT.W2) GO TO 30
   20 CONTINUE
   30 L2 = I
      IF(L2.GT.NWL) L2 = NWL
      L2  = L2 - 1
      RETURN
      END
```

```
      SUBROUTINE TRAP(L1,L2,X,Y,SUM)
      DIMENSION X(1), Y(1)
      L2 = L2-1
      SUM = 0.
      DO 10 I=L1,L2
   10 SUM = SUM + 0.5*(Y(I+1)+ Y(I)) * (X(I+1) - X(I))
      RETURN
      END
```

10. SAMPLE RUNS

A set of runs showing the interaction of the entire methodology simulation (ASDIR II, M/T/CM, and SPKINT) was made, and sample outputs are given in this section. The engine used to calculate the IR signature was the 10,000 foot default engine of ASDIR II. Engine hot part contributions were assumed using the equivalent blackbody temperature and area of 824°K and 730 cm^2 respectively (0 degrees aspect angle). The blackbody area was varied as a function of the cosine of the aspect angle which was varied at 15 degree increments from 9 to 90 degrees. Apparent $J_\lambda \tau_\lambda$ values (1.8 to 5.5 microns) were calculated for ranges of 0.0, 0.305, 1.524, 6.096, and 15.240 kilometers. These values were then integrated, by SPKINT, over five spectral intervals to generate the apparent effective $(J\tau)_{\Delta\lambda}$ values used in the M/T/CM simulation program. The integrated values of $(J\tau)_{\Delta\lambda}$ were then entered into the M/T/CM program and a typical set of missile simulation runs using a spin-scan type missile were made for aspect angles in 15 degree increments from 8 to 90 degrees, and launch range of 5,000 feet. The results of these runs were P_s = 0 for all but aspect angles of 75 and 90 degrees. At these two angles, the missile was unable to maneuver to catch the target (i.e., it was launched outside of the aerodynamic launch boundary) and P_s = 1 in these cases.

Sample outputs from these runs are shown in Tables 10-1 and 10-2 and in Figures 10-1 through 10-6.* Table 10-1 shows the ASDIR II output of $J_\lambda \tau_\lambda$ versus λ and Table 1-2 the integrated SPKINT values. Both of these cases are for 0 degrees aspect and 0 Km range. Figure 10-1 gives a plot of the spectral $J_\lambda \tau_\lambda$ for the 0 Km range case, and Figure 10-2 is a polar plot of $(J\tau)_{\Delta\lambda}$, $\Delta\lambda$ = 1.8 to 2.6μ, for three ranges. Plots of the simulated missile flight are shown in the last four figures. Missile target trajectories in the X-Z and X-Y planes are shown in Figures 10-3 and 10-4 respectively. Apparent effective intensity and effective irradiance at the missile seeker as a function of time are given in Figures 10-5 and 10-6.

---

*Figures 10-1 through 10-6 were generated by separate CALCOMP plotter routines which are not part of EPICS.

Table 10-1. ASDIR II output

...TOTAL SIGNATURE OVER THE SPECTRAL BAND 1.80 TO 5.50 MICRONS AT A RANGE OF .000 KM
FOR AN ASPECT ANGLE OF .0 DEGREES IN A NOR. ATMOSPHERE...

.VEHICLE ALTITUDE = 3.05 KM AND OBSERVER ALTITUDE = 3.05 KM

BAND CENTER MICRONS	BAND WIDTH MICRONS	APPARENT RADIANCE WATTS/STERADIAN	WAVENUMBER (CENTER)CM-1	INCREMENT CM-1	SPECTRAL RADIANCE WATTS/MICRON/SR
1.8040	.0122	.3254	5543.18	37.4	26.7557
1.8204	.0165	.4345	5493.18	50.0	26.2249
1.8372	.0169	.4239	5443.18	50.0	25.1161
1.8542	.0172	.4561	5393.18	50.0	26.5354
1.8715	.0175	.4914	5343.18	50.0	28.0581
1.8892	.0178	.5758	5293.18	50.0	32.2653
1.9072	.0182	.5720	5243.18	50.0	31.4509
1.9256	.0185	.6341	5193.18	50.0	34.2012
1.9443	.0189	.6949	5143.18	50.0	36.7628
1.9634	.0193	.7557	5093.18	50.0	39.2048
1.9829	.0197	.8140	5043.18	50.0	41.4066
2.0027	.0200	.8709	4993.18	50.0	43.4274
2.0230	.0205	.9271	4943.18	50.0	45.3084
2.0437	.0209	.9864	4893.18	50.0	47.2355
2.0648	.0213	1.0463	4843.18	50.0	49.0849
2.0863	.0217	1.1081	4793.18	50.0	50.9173
2.1083	.0222	1.1726	4743.18	50.0	52.7637
2.1308	.0227	1.2401	4693.18	50.0	54.6269
2.1537	.0232	1.3105	4643.18	50.0	56.5084
2.1771	.0237	1.3845	4593.18	50.0	58.4178
2.2011	.0242	1.4620	4543.18	50.0	60.3530
2.2256	.0247	1.5432	4493.18	50.0	62.3096
2.2506	.0253	1.6279	4443.18	50.0	64.2760
2.2763	.0259	1.7164	4393.18	50.0	66.2523
2.3025	.0265	1.8084	4343.18	50.0	68.2243
2.3293	.0271	1.9036	4293.18	50.0	70.1728
2.3567	.0278	2.0027	4243.18	50.0	72.1144
2.3848	.0284	2.0997	4193.18	50.0	73.8359
2.4136	.0291	2.1901	4143.18	50.0	75.1905
2.4431	.0299	2.2584	4093.18	50.0	75.6734
2.4733	.0306	2.2717	4043.18	50.0	74.2726
2.5043	.0314	2.2228	3993.18	50.0	70.8885
2.5360	.0322	2.0920	3943.18	50.0	65.0542
2.5686	.0330	1.6351	3893.18	50.0	49.5653
2.6020	.0339	1.6538	3843.18	50.0	48.8522
2.6363	.0347	2.0999	3793.18	50.0	60.4271
2.6715	.0357	.1523	3743.18	50.0	4.2678
2.7077	.0367	.2709	3693.18	50.0	7.3895
2.7449	.0377	.8401	3643.18	50.0	22.3007

Table 10-2. SPKINT integrated output

1.804-TO- 2.636	.16958E 00
2.672-TO- 3.580	.31717E 00
2.991-TO- 4.777	.61955E 00
3.932-TO- 4.666	.20835E 00
3.856-TO- 5.146	.38716E 00

NOR   HD = 3.05 M   HS = 3.05 M   SR = 0. M   VR = 0. KM

RAD = .2813E 03 IN THE WAVELENGTH REGION 1.8040 TO 5.4254 MICRONS

TARGET TYPE IS AS3IN TEST RUN 10KFT DEFAULT

JSCAL = .1000E 01

WAVELENGTH REGION (MICRONS	J TAU	TAU EFF	TAU (TOTAL
1.804-TO- 2.636	.46787E 02	1.0000000	.1695770
2.672-TO- 3.580	.87510E 02	1.0000000	.3171721
2.991-TO- 4.777	.17094E 03	1.0000000	.6195509
3.932-TO- 4.666	.57456E 02	1.0000000	.2083524
3.856-TO- 5.146	.10737E 03	1.0000000	.3891558

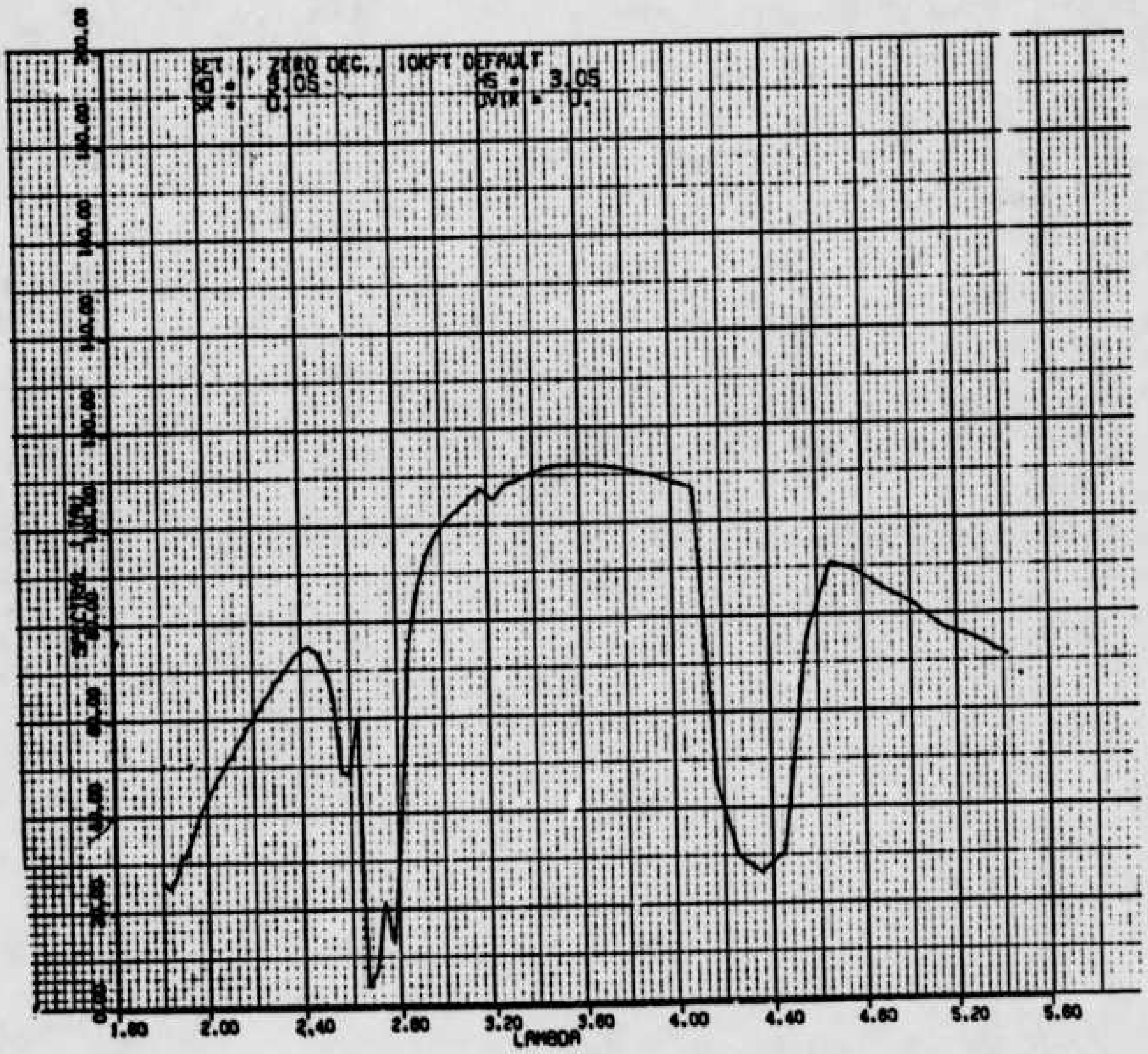


Figure 10-1. Spectral $J_\lambda \tau_\lambda$ (R = 0, aspect = 0)

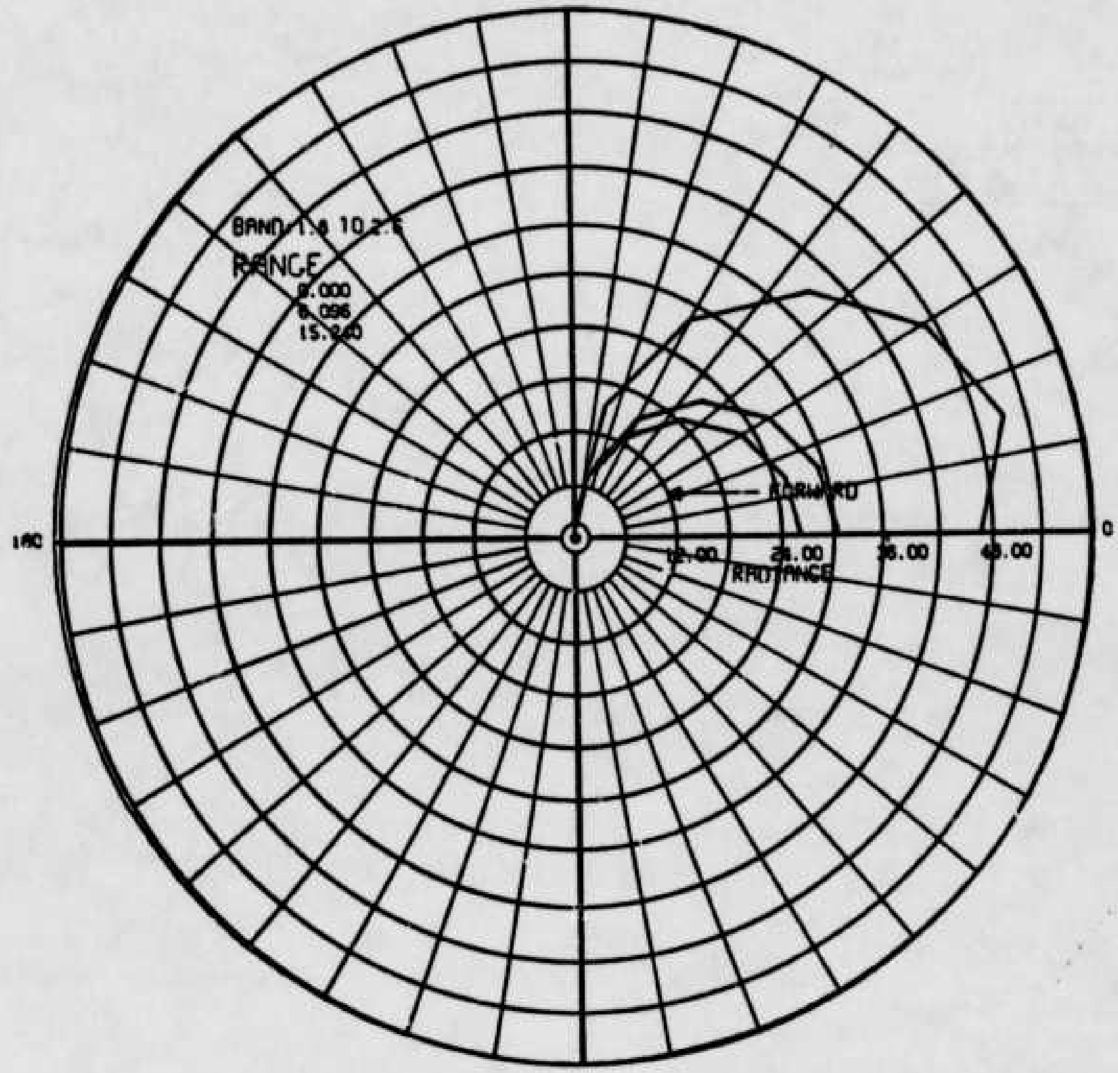


Figure 10-2. Polar plots of apparent effective radiant intensity

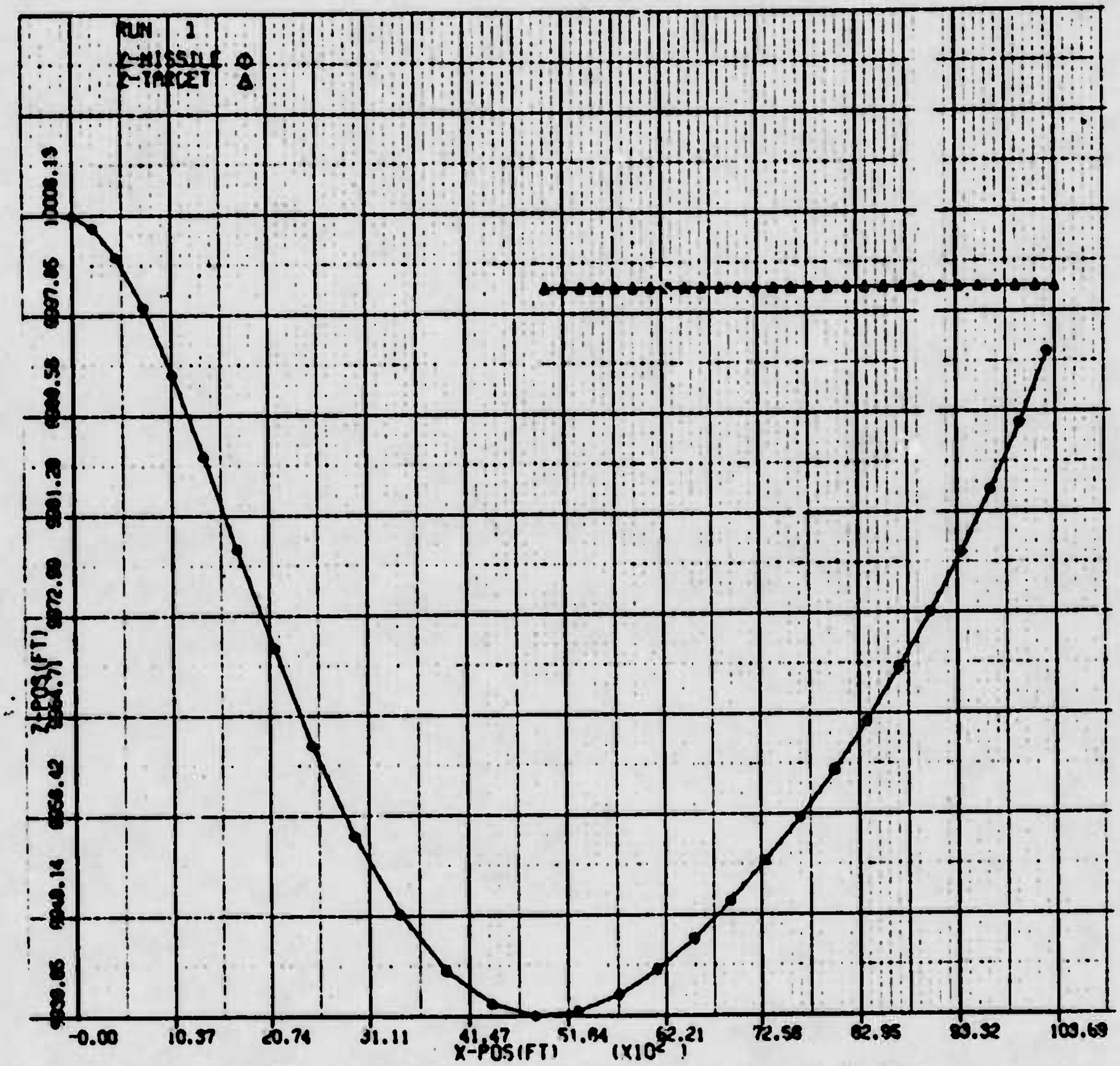


Figure 10-3. Missile - target trajectory X-Z plane

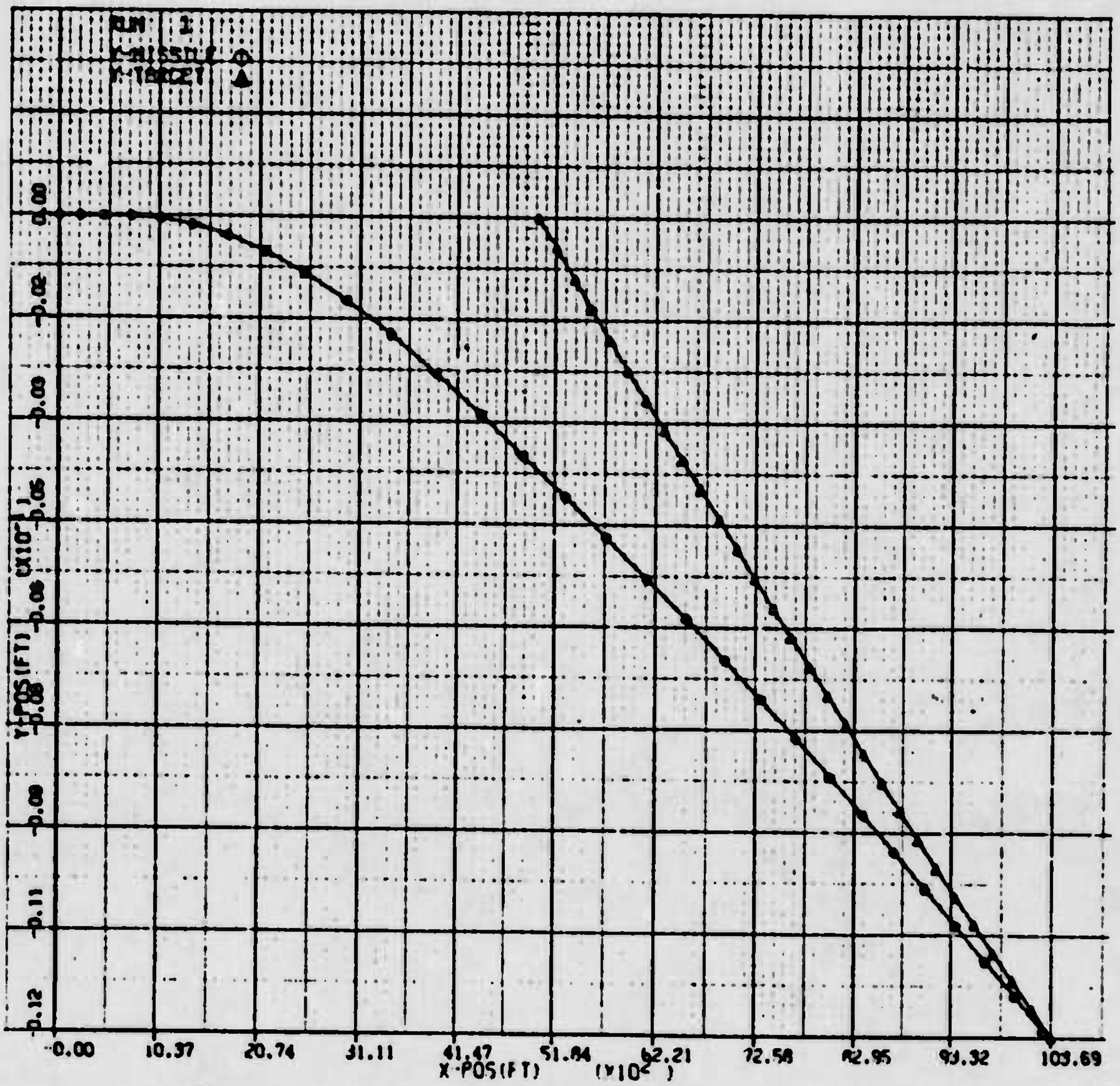


Figure 10-4. Missile-target trajectory X-Y plane

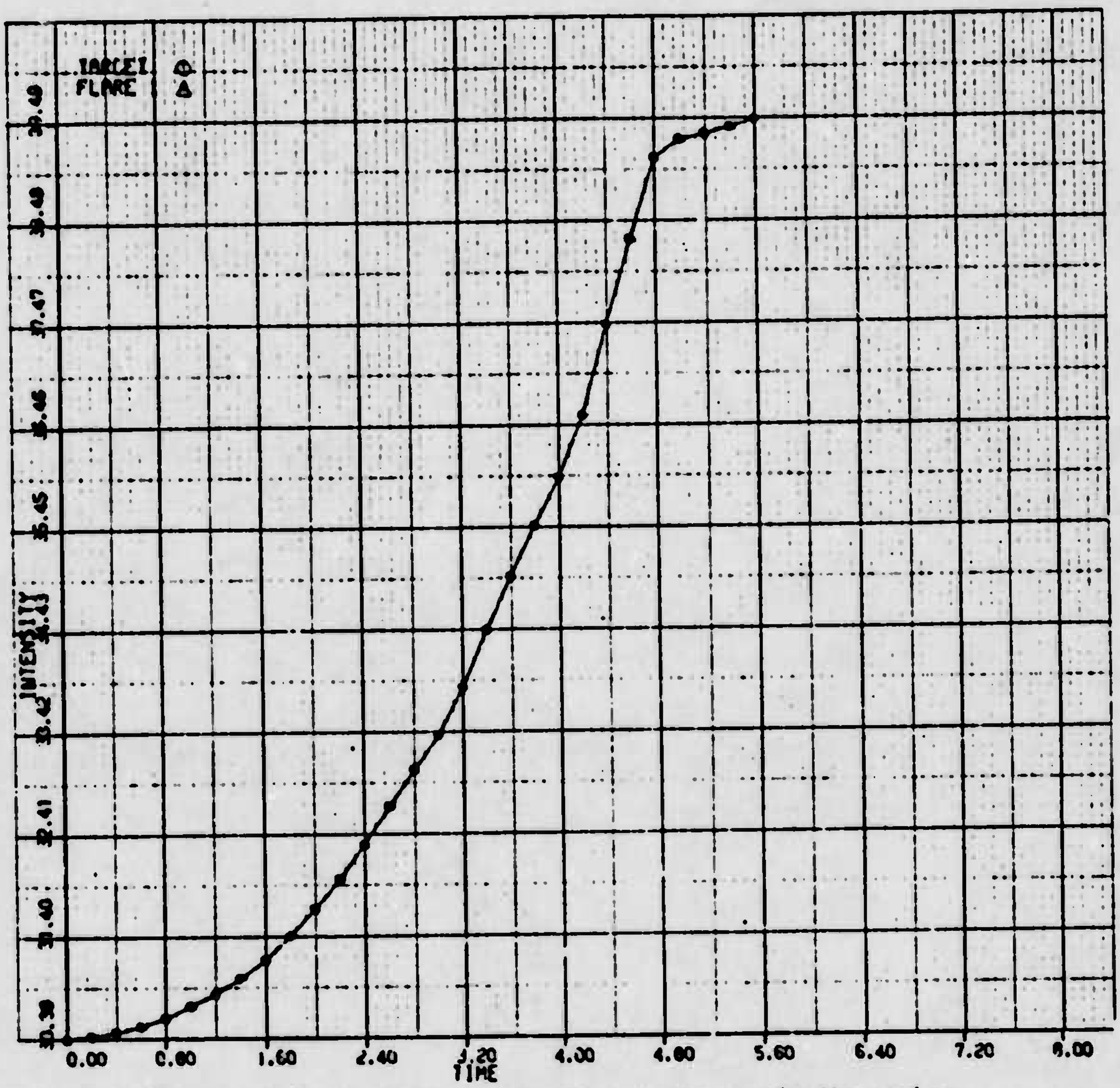


Figure 10-5. Apparent radiant intensity at missile seeker

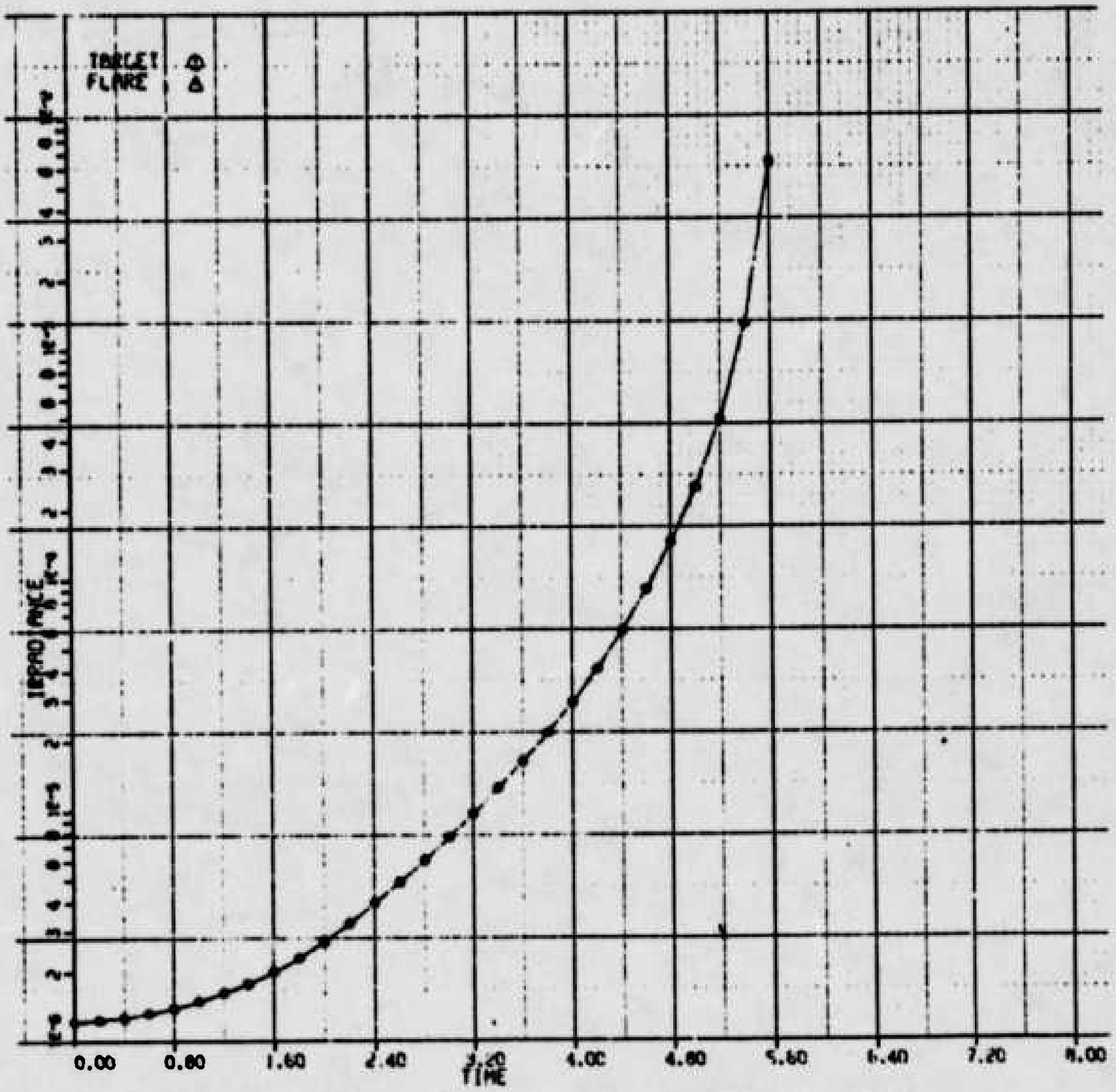


Figure 10-6. Effective irradiance at missile seecker