

N O T I C E

THIS DOCUMENT HAS BEEN REPRODUCED FROM
MICROFICHE. ALTHOUGH IT IS RECOGNIZED THAT
CERTAIN PORTIONS ARE ILLEGIBLE, IT IS BEING RELEASED
IN THE INTEREST OF MAKING AVAILABLE AS MUCH
INFORMATION AS POSSIBLE

80-FM-29
(Supersedes 76-FM-87, Rev. 2)

JSC-16522
(Supersedes JSC-11786)

APR 29 1980

MCC Level C Formulation Requirements

Shuttle TAEM Guidance and Flight Control
(Optional TAEM Targeting)

(NASA-TM-80796) MCC LEVEL C FORMULATION
REQUIREMENTS. SHUTTLE TAEM TARGETING (NASA)
86 p HC A05/MF A01 CSCL 22B

N80-25370

Unclas
G3/16 22958

Supersedes T80-10421

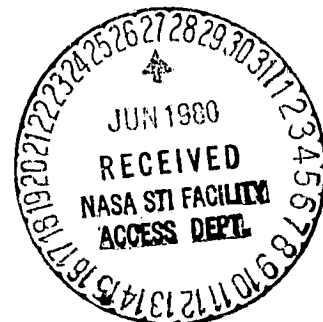
Mission Planning and Analysis Division

April 1980

NASA

National Aeronautics and
Space Administration

Lyndon B. Johnson Space Center
Houston, Texas



80-FM-29
(Supersedes 76-FM-87, Rev. 2)

JSC-16522
(Supersedes JSC-11786)

SHUTTLE PROGRAM

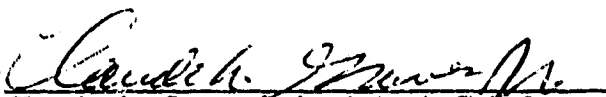
MCC LEVEL C FORMULATION REQUIREMENTS

SHUTTLE TAEM GUIDANCE AND FLIGHT CONTROL
(OPTIONAL TAEM TARGETING)

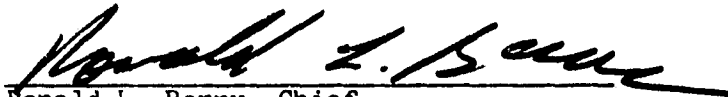
FM 4

By Gilbert L. Carman and Moises N. Montez
Flight Analysis Branch

Approved:


Claude A. Graves, Assistant Chief
Flight Analysis Branch

Approved:


Ronald L. Berry, Chief
Mission Planning and Analysis Division

Mission Planning and Analysis Division
National Aeronautics and Space Administration
Lyndon B. Johnson Space Center
Houston, Texas
April 1980

CONTENTS

Section	Page
1.0	1
<u>SUMMARY</u>	1
2.0	1
<u>INTRODUCTION</u>	1
3.0	2
<u>COORDINATE SYSTEMS</u>	2
4.0	2
<u>TAEM GUIDANCE</u>	2
4.1	3
DETAILED REQUIREMENTS	3
4.1.1	4
<u>TAEM Guidance (TGEXEC)</u>	4
4.1.2	4
<u>TAEM Guidance Initialization (TGINIT)</u>	4
4.1.3	5
<u>XHAC Function (TGXiAC)</u>	5
4.1.4	6
<u>Groundtrack Computation (GTP)</u>	6
4.1.5	10
<u>TAEM Reference, Dynamic Pressure, and Spiral Adjust Function (TGCOMP)</u>	10
4.1.6	13
<u>Phase Transition, MEP, and OHALRT Function (TGTRAN)</u>	13
4.1.7	16
<u>Normal Acceleration Command Function (TGNZC)</u>	16
4.1.8	18
<u>Speedbrake Command Function (TGSBC)</u>	18
4.1.9	19
<u>Roll Command Function (TGPHIC)</u>	19
4.1.10	21
<u>Mid Value (MIDVAL)</u>	21
4.1.11	21
<u>Resolve to 180 Degrees (RES180)</u>	21
4.2	22
SEQUENCING REQUIREMENTS	22
4.3	22
INTERFACE REQUIREMENTS	22
5.0	22
<u>TAEM DIGITAL AUTOPILOT (TDAP)</u>	22
5.1	22
REQUIREMENTS OVERVIEW	22
5.2	22
FUNCTION MODULES	22
5.3	23
INTERFACE REQUIREMENTS	23
5.4	24
SEQUENCING REQUIREMENTS	24
5.5	24
INITIALIZATION REQUIREMENTS	24
5.6	24
METHOD	24
6.0	25
<u>REFERENCE</u>	25
APPENDIX - GUIDANCE AND FLIGHT CONTROL INPUT PARAMETER CALCULATIONS	A-1

TABLES

Table		Page
I	TAEM GUIDANCE INPUTS	26
II	TAEM GUIDANCE OUTPUTS	28
III	TAEM GUIDANCE CONSTANTS	30
IV	TDAP INPUTS	41
V	TDAP OUTPUTS	42
VI	TDAP CONSTANTS	43

FIGURES

Figure		Page
1	TAEM guidance groundtrack predictor geometry	46
2	GTP geometry for acquisition phase	47
3	Resolve to 180 degrees (RES180)	48
4	TAEM guidance flow	49
5	TDAP detailed formulation	69
6	FILTER detailed formulation	72
7	SMOOTH detailed formulation	73
8	EIGEN detailed formulation	74

1.0 SUMMARY

This document describes the Level C requirements for the Shuttle Orbiter terminal area energy management (TAEM) guidance and flight control functions to be incorporated into the Mission Control Center (MCC) entry profile planning processor. This processor will be used for preentry evaluation of the entry through landing maneuvers, and will include a simplified three-degree-of-freedom model of the body rotational dynamics that is necessary to account for the effects of attitude response on the trajectory dynamics. This simulation terminates at TAEM-autoland interface.

The TAEM guidance mode is initiated based upon Earth-relative velocity. The TAEM guidance controls the Orbiter to altitude and dynamic pressure profiles that are functions of range-to-go. Angle of attack is used to control the altitude, and speedbrake deflection is used to control dynamic pressure. Excess energy or insufficient energy conditions result in use of S-turns or delayed entry into the final approach phase, respectively. Roll angle is used for lateral trajectory control and for the S-turns.

The body rotational dynamics simulation is a simplified model that assumes that the commanded rotational rates can be achieved within the integration interval. Thus, the rotational dynamics simulation is essentially a simulation of the autopilot commanded rates and integration of these rates to determine the Orbiter attitude.

It is assumed that certain data are available as input to this processor. It is assumed that the state vector is available in runway coordinates and the Orbiter heading is known with respect to the runway. Also, it is assumed that altitude of the vehicle center of mass with respect to the runway, altitude rate, horizontal component of velocity, flightpath angle, and body roll and pitch attitudes are available as input data. Derivation of these data is presented in appendix A.

2.0 INTRODUCTION

This document defines the TAEM guidance and body rotational dynamics models required for the MCC simulation of the TAEM mission phase. This simulation begins at the end of the entry phase and terminates at TAEM-autoland interface. Revision 2 incorporates the TAEM optional TAEM targeting (OTT) guidance. Implementation of this logic in the MCC must be coordinated with the flight software implementation of the OTT guidance.

The TAEM guidance is simulated in detail. The rotational dynamics simulation is a simplified model and it assumes that the commanded rotational rates can be achieved in the integration interval. Thus, the rotational dynamics simulation is essentially a simulation of the autopilot commanded rates and integration of these rates to determine Orbiter attitude. The rotational dynamics simulation also includes a simulation of the speedbrake deflection. The body flap and elevon deflections are computed in the Orbiter aerodynamic simulation.

3.0 COORDINATE SYSTEMS

The TAEM guidance requires input data referenced to two primary coordinate systems. The first is the topodetic system which has its x-, y-, and z-axes oriented in the north, east, and down directions, respectively. The origin of this system is the vehicle center of mass, and the downward direction is normal to the Earth ellipsoid model. The second is the runway coordinate system. This system is obtained by constructing a topodetic system with its origin at the center of the runway threshold and performing a rotation about its z-axis through the runway azimuth from true north. Both of these systems are Earth-fixed rotating systems, with the topodetic being redefined at each time of interest.

Pitch and roll attitudes are the angles of a yaw-pitch-roll Euler rotation sequence from topodetic to body coordinates.

4.0 TAEM GUIDANCE

The TAEM guidance will nominally guide the Orbiter to an altitude and dynamic pressure versus range profile. It controls energy by modulating drag through speedbrake deflection commands to null out the dynamic pressure errors and by nulling out altitude errors through the normal load factor command. If excess energy exists and the heading alinement turn angle is less than the predetermined value PSSTRN, the logic executes an S-turn to dissipate additional energy.

If an extremely low energy condition exists, the minimum entry point (MEP) is automatically selected to reduce the required range to go.

This revision incorporates the modifications necessary for optional TAEM targeting capability. This capability allows the selection of either straight-in (nearest heading alinement cone (HAC)) or overhead (farthest initial HAC) approach modes. In the overhead mode, the left or right HAC is selected at entry interface to be on the opposite side of the runway centerline of the Shuttle position. This selection remains latched, independent of Shuttle position, for the remainder of the simulation unless a low energy downmode to straight-in is performed by TAEM guidance or the option to manually toggle the selected HAC is exercised.

The straight-in mode selects the nearest HAC continuously. In order to correctly calculate the HAC turn angle, PSHA, under all possible situations, an approach mode flag, OVHD, is included to designate the selected approach, either overhead (OVHD = 1) or straight-in (OVHD = 0). This flag is common to both the entry range predictor (EGRT) and TAEM guidance and is initialized to 1 in the mission constants table to nominally give an overhead approach. The straight-in mode may be selected by setting the mode flag to zero (OVHD = 0) by MED input.

While in the overhead mode, TAEM guidance has the capability to downmode to straight-in if the energy state is sufficiently low. This automatic downmode feature may be overridden by setting the automatic HAC override flag (ORAHAC = 1) by MED input to latch the overhead status independent of energy. The automatic

downmode request flag (OHALRT) may still be set, but will not be acted upon if ORAHAC = 1.

There are three possible sources of approach mode changes that may occur subsequent to initialization. Since the approach mode and automatic downmode inhibit flags may be designated independently for both primary and secondary runways for redesignation cases, the OVHD flag can change status at redesignation, which may occur any time during entry or TAEM. The second source of mode change is the low energy downmode to straight-in, which can occur only in the TAEM guidance phase. The runway dependent mode flags (OVHD and ORAHAC) are selected as a function of the runway identifier flag (RWID), which is input from the runway selection/redesignation logic.

A change of approach mode may also be exercised by input of the manual HAC toggle velocity (VTOGL). When the Shuttle velocity becomes below this input value, the current approach mode (OVHD or STRT) will be reversed.

For all guidance phases (S-turn, acquisition, heading alinement, and prefinal), (IPHASE = 0, 1, 2, 3, respectively) the guidance normal load factor command is based on the altitude and altitude rate errors from the reference profile, and the speedbrake command is based on the dynamic pressure error. All guidance phases are internally determined.

On the lateral axis, if the guidance is in the S-turn phase, a constant bank angle is sent to the flight control system. In the acquisition phase, the guidance commands a bank angle that is proportional to the Orbiter heading deviation from the tangent point on the selected heading alinement cone. In the heading alinement phase, the guidance commands a bank angle which assures that the Orbiter follows the heading alinement cylinder. During the prefinal phase, it commands a bank angle from a linear combination of Orbiter lateral deviation and deviation rate from the runway centerline.

4.1 DETAILED REQUIREMENTS

The following subsections (4.1.1 to 4.1.11) define the detailed software requirements for the functions and subfunctions that constitute the TAEM guidance program (ref. 1). These subsections present the equations and the logic performed by function. The input and output variables are summarized in tables I and II. A summary of all constants is shown in table III. The values of the constants in this table are for a typical OFT-1 trajectory, and are expected to change from mission to mission.

The functions and subfunctions of TAEM guidance are:

- TAEM guidance (TGEXEC)
- TAEM guidance initialization (TGINIT)
- XHAC function (TGXHAC)
- Groundtrack predictor (GTP)
- Resolve to 180 degrees routine (RES180)
- References and dynamic pressure function (TGCOMP)
- Phase transition and MEP function (TGTRAN)

Normal acceleration command function (TGNZC)
 Speedbrake command function (TGSEC)
 Roll command function (TGPHIC)

The function and routine used in more than one other function or subfunction are:

MID VALUE (MIDVAL)

4.1.1 TAEM Guidance (TGEXEC)

This function is the TAEM guidance executive routine. On the first pass, the initialization flag, IRESET, is set to 1, the initialization function (TGINIT) is executed first. Next the XHAC (TGXHAC) and the groundtrack predictor (GTP) functions are called. The TAEM reference and dynamic pressure function (TGCOMP) are then executed. The TAEM phase transition and MEP function (TGTRAN) are then called. The guidance commands are then generated by calling the normal acceleration command (TGNZC), the speedbrake command (TGSEC), and the roll command (TGPHIC) functions. TAEM guidance is then exited.

4.1.2 TAEM Guidance Initialization (TGINIT)

The TGINIT function initializes or computes several parameters used in TAEM guidance. These operations are performed either on the first pass (IRESET = 1) or when the primary runway is redesignated (RWID \neq RWIDO). These operations are:

IF IRESET = 1, RWIDO = RWID

IRESET = 0

ISR = RFTC/DTG

MEP = 0

RF = RFO

DSBI = 0.0

OHALRT = 0

PHILIM = PHILM1

DNZUL = DNZUC1

DNZLL = DNZLC1

QBARF = QBAR

QBD = 0.0

IPHASE = 1

TG_END = 0

4.1.3 XHAC Function (TGXHAC)

This function calculates the appropriate Y runway position and the X runway coordinate of the heading alignment cone center (XHAC) and the prefinal initiation range value (RPRED3) as a function of surface wind conditions, weight, and a nominal or minimum entry point (MEP) situation.

The operations performed are:

$$XFTC = XA(IGI) + HFTC(IGS)/TGGS(IGS)$$

$$XALI = XA(IGI) + HALI(IGS)/TGGS(IGS)$$

$$XMPE = XA(IGI) + HMEP(IGS)/TGGS(IGS)$$

where the subscript IGI is selected based on a MED input GI_change and the subscript IGS is automatically selected based on the input Orbiter mass. GI change is a function of surface wind conditions.

IGI = 1 if GI_change = 0

IGI = 2 if GI_change = 1

GI_change is set to zero if the surface winds are less than 50 percent of the design surface winds, and set to 1 for winds greater than 50 percent of the design winds.

If the Orbiter mass is greater than WT_GS1, then IGS = 2. Otherwise, IGS = 1. The value for WT_GS1 is selected such that the glide slope change occurs when Orbiter payloads are greater than 32 000 pounds.

The HAC X position (XHAC) based on the minimum entry point flag (MEP) and the prefinal initiation range value (RPRED3) are calculated as follows:

$$XHAC = XFTC$$

IF MEP = 1, then XHAC = XMPE

$$RPRED3 = -XHAC + DR3$$

4.1.4 Groundtrack Computation (GTP)

This section describes the equations used by the groundtrack predictor subprogram in mathematical symbols. The detailed formulation of the groundtrack predictor subprogram is given at the end of this section.

An essential part of the TAEM guidance concept is the prediction of groundtrack range to runway threshold. The method for computing groundtrack range is illustrated in figure 1. As shown, the Orbiter velocity vector is initially pointing away from tangency to the heading alignment cone (spiral in ground plane). The acquisition turn is defined as the turn that will align the Orbiter heading tangent to the heading alignment cone. The distance d_{AC} along the acquisition turn is computed from geometrical equations and an estimate of the acquisition turn radius. The distance d_1 is computed from geometrical equations relating the position of the Orbiter after the acquisition turn to the tangent point on the cone. The distance in the heading alignment turn is computed from the turn angle in the heading alignment phase and the alignment spiral radius. The distance d_T is a fixed distance from runway threshold for each of two possible downrange locations of the alignment cone. The two possible downrange locations of the MEP alignment cones are referred to as the nominal and minimum locations. The two possible crossrange locations with respect to the runway centerline of the heading alignment cones are referred to as the straight-in and overhead HAC locations.

The groundtrack predictor estimates range-to-go by computing the segments d_{AC} , d_1 , d_{HAC} , and d_T . These segments are added to give the total predicted range. The terms that no longer apply as the approach progresses are either set to zero or dropped from the summations.

No additional range component is required for the S-turn phase because the predicted groundtrack during this phase is computed as if the S-turn were to cease immediately.

During phase three (prefinal approach) and after the vehicle downrange position with respect to the runway is within a predetermined value with respect to the HAC, the estimated range-to-go is based on a straight line to the runway threshold; thus

$$R_{PRED} = \sqrt{X^2 + Y^2}$$

The geometry for computing the segments d_{AC} and d_1 is given in figure 2. The angle $\Delta\psi$ is the difference between the Orbiter heading ψ and the heading ψ_T to the heading alignment cone tangency point. The heading alignment cone is a spiral in the ground plane described by $RTURN = R_F + R_1 \psi_{HA} + R_2 \psi_{HA}^2$. Given an estimate of ψ_{HA} , the spiral acquisition is simplified by assuming a circle with center at R_F and radius of $RTURN$. The vector from the Orbiter to the center of the selected heading alignment cone is computed as

$$X_{CIR} = X_{HAC} - X$$

$$Y_{CIR} = Y_{SGN} \text{ RF} - Y$$

The distance to the center of the heading alignment cone is

$$R_{CIR} = \sqrt{X_{CIR}^2 + Y_{CIR}^2}$$

and the straight line distance to the tangency point is

$$R_{TAN} = \sqrt{R_{CIR}^2 - R_{TURN}^2}$$

which is then limited so that $R_{TAN} > 0$.

The heading ψ_C to the center of the heading alignment cone is given by

$$\psi_C = \tan^{-1} (Y_{CIR}/X_{CIR})$$

The heading to the tangency point is then

$$\psi_T = \psi_C - Y_{SGN} \tan^{-1} (R_{TURN}/R_{TAN})$$

and the heading error is

$$\Delta\psi = \psi_T - \psi$$

An acquisition turn radius is computed as a function of a predicted average bank angle during the turn. The turn radius R_{AC} and the error angle $\Delta\psi$ are used to compute the distance, d_{AC} to be traveled during acquisition. The distance, d_1 to the tangency point from the end of acquisition is computed by using $\Delta\psi$, R_{AC} , and R_{TAN} .

The radius of the acquisition turn R_{AC} is approximated by

$$R_{AC} = V_h V / (g \tan \phi_{avg})$$

where V_h = horizontal velocity.

ϕ_{AVG} = predicted average bank angle

V = velocity magnitude

The acquisition turn arc is

$$d_{AC} = R_{AC} |\Delta c|$$

The distance d_1 is derived from the geometry of figure 2.

$$A = R_{AC} (1 - \cos (\Delta\psi))$$

$$B = R_{TAN} - R_{AC} |\sin (\Delta\psi)|$$

$$d_1 = \sqrt{A^2 + B^2}$$

GROUNDTRACK PREDICTOR SUBPROGRAM

$$XCIR = XHAC - X$$

If IPHASE = 3 and XCIR < DR4, then

$$RPRED = \text{SQRT} (X^2 + Y^2)$$

and then exit GTP. Otherwise, (IPHASE \neq 3 or XCIR \geq DR4), the remaining GTP logic is processed.

Equation set 1 is executed as

- 1.1 SIGNY = SIGN (Y)
- 1.2 YCIR = YSGN RF - Y
- 1.3 RCIR = SQRT (XCIR² + YCIR²)

IF: RCIR > RTURN

THEN: RTAN = SQRT (RCIR² - RTURN²)

ELSE: RTAN = 0

The heading error (from a heading tangent to the HAC), HAC turn angle, spiral turn radius, and range from HAC to runway are then computed by equation set 2:

- 2.1 PSC = ARCTAN2 (YCIR, XCIR)
- 2.2 PST = (PSC - YSGN ARCTAN2(RTURN, RTAN)) RTD
- 2.3 PST = RES180 (PST)
- 2.4 DPSAC = RES180 (PST-PSD)
- 2.5 PSHAN = -PST YSGN
- 2.6 If ((PSHA > PSHARS + 1. or PSHAN < -1. or YSGN \neq SIGNY) and PSHA > 90.), then PSHAN = PSHAN + 360.
- 2.7 PSHA = PSHAN

$$2.8 \quad RTURN = RF + R1 \text{ PSHA} + R2 \text{ PSHA}^2$$

$$2.9 \quad RPRED2 = (RF \text{ PSHA} + 0.5 R1 \text{ PSHA}^2 + 0.333333 R2 \text{ PSHA}^3) \text{ DTR} - \text{XHAC}$$

If IPHASE < 2, the acquisition turn radius and arc length are then computed in equation set 3:

$$3.1 \quad \text{PHAVG} = \text{PHAVGC} - \text{PHAVGS} \text{ MACH}$$

$$3.2 \quad \text{PHAVG} = \text{MIDVAL} (\text{PHAVG}, \text{PHAVGLL}, \text{PHAVGUL})$$

$$3.3 \quad \text{RTAC} = \text{VH} \text{ V}/(\text{G} \text{ TAN} (\text{PHAVG} \text{ DTR}))$$

$$3.4 \quad \text{ARCAC} = \text{RTAC} \text{ ABS} (\text{DPSAC} \text{ DTR})$$

If IPHASE < 2, the range from the end of the acquisition turn to the HAC tangency point, RC, is next computed by equation set 4:

$$4.1 \quad A = \text{RTAC} (1. - \text{COS}(\text{DPSAC} \text{ DTR}))$$

$$4.2 \quad B = \text{RTAN} - \text{RTAC} \text{ ABS} (\text{SIN}(\text{DPSAC} \text{ DTR}))$$

$$4.3 \quad \text{RC} = \text{SQRT} (A^2 + B^2)$$

$$4.4 \quad \text{RTAN} = \text{ARCAC} + \text{RC}$$

Finally, the total range-to-go is calculated from $\text{RPRED} = \text{RPRED2} + \text{RTAN}$ and GTP is then exited.

4.1.5 TAEM Reference, Dynamic Pressure, and Spiral Adjust Function (TGCOMP)

The TGCOMP function computes energy, altitude, flightpath angle, and dynamic pressure references as a function of the predicted range (RPRED). The energy reference (EN) consists of two linear segments that are functions of DRPRED. Energy slope and intercept for this linear function are switched when DRPRED is less than the range switch point (EOW_SPT). The reference energy-over-weight (EN) is lowered if necessary to target for subsonic conditions at the HAC if the estimated range-to-go at the start of the HAC phase, RPRED2, is greater than the maximum range, R2MAX, for subsonic roll maneuvers (MACH < PHIM). The altitude reference (H ref) is a cubic curve which is tangent to the autoland steep glide slope at $\text{RPRED} = -\text{XALI}$. If RPRED is less than -XALI, the altitude reference is defined by the autoland steep glide slope. If DRPRED is greater than PBRC, the altitude reference is a linear function of DRPRED.

The reference flightpath angle is the slope of the altitude versus predicted range curve at the current predicted range. The dynamic pressure reference is a two segment linear function of predicted range.

This function also computes current energy, filtered dynamic pressure, filtered dynamic pressure rate, dynamic pressure error, and the commanded equivalent airspeed. The final turn radius, RF, of the HAC spiral is adjusted in the HAC phase as a function of altitude error from an altitude profile, HREFOH, that is lower than the nominal altitude reference. This lower altitude reference corresponds to the altitude profile for a 360-degree overhead HAC.

TGCOMP SUBPROGRAM

Upon entering TGCOMP, the following computations are made:

The current energy-over-weight is computed in equation set 1:

$$1.1 \quad \text{DRPRED} = \text{RPRED} + \text{XALI}$$

$$1.2 \quad \text{EOW} = \text{H} + \text{V} \text{ V}/(2\text{G})$$

The energy reference is computed in equation set 2. If DRPRED is less than EOW SPT(IGS), set IEL = 2. Otherwise, set IEL = 1.

$$2.1 \quad \text{EN} = \text{EN_C1}(\text{IGS}, \text{IEL}) + \text{DRPRED} \text{ EN_C2}(\text{IGS}, \text{IEL}) \\ - \text{MIDVAL} (\text{EN_C2IGS}, 1 (\text{RPRED}^2 - \text{R2MAX}), 0., \text{ESHFMX})$$

If DRPRED > PBRC(IGS), the altitude reference is computed with equation 4.1.

$$4.1 \quad \text{HREF} = \text{PBHC}(\text{IGS}) + \text{PBGC}(\text{IGS}) (\text{DRPRED} - \text{PBRC}(\text{IGS}))$$

If DRPRED ≤ PBRC(IGS), the altitude reference is computed with equations 4.2 and 4.3.

$$4.2 \quad \text{HREF} = \text{HALI}(\text{IGS}) - \text{TGGS}(\text{IGS}) \text{ DRPRED}$$

$$4.3 \quad \text{IF} (\text{DRPRED} > 0.) \text{ HREF} = \text{HREF} + \text{DRPRED} \text{ DRPRED} \\ (\text{CUBIC_C3}(\text{IGS}) + \text{DRPRED} \\ \text{CUBIC_C4}(\text{IGS}))$$

The dynamic pressure reference (QBREF), final spiral radius, altitude error, flightpath angle reference, and range error from reference altitude profile are computed in equation set 5:

- 5.1 If (DRPRED < PBRCQ(IGS)), QBREF = MIDVAL (QBRUL(IGS)
 + QBC2(IGS) DRPRED,
 QBRL(IGS), QBRUL(IGS))
- 5.2 If (DRPRED > PBRCQ(IGS)), QBREF = MIDVAL (QBRL(IGS)
 + QBC1(IGS) (DRPRED
 - PBRCQ(IGS)),
 QBRL(IGS), QBRML(IGS))
- 5.3 If IPHASE = 2 and PSHA > PSRF, then execute equation set 5.4.
- 5.4.1 HREFOH = HREF - MIDVAL (DHOH1 (DRPRED - DHOH2), 0., DHOH3)
- 5.4.2 DRF = DRFK (HREFOH - H)/(PSHA DTR)
- 5.4.3 RF = MIDVAL (RF + DRF, RFMN, RFMX)
- 5.5 HERROR = HREF - H
- 5.6 If (DRPRED > PBRC(IGS)), DHDRRF = -PBGC(IGS)
- 5.7 If (DRPRED < PBRC(IGS)), DHDRRF = -MIDVAL (-TGGS(IGS)
 + DRPRED (2 CUBIC_
 C3(IGS) + 3 CUBIC_
 C4(IGS) DRPRED),
 PBGC(IGS), -TGGS(IGS))
- 5.8 DELRNG = HERROR/DHDRRF

Filtered dynamic pressure, filtered dynamic pressure rate, dynamic pressure error, and the commanded equivalent airspeed are computed in equation set 6:

- 6.1 QBARD = MIDVAL(CQG*(QBAR - QBARF), -QBARDL, QBARDL)
- 6.2 QBARF = QBARF + QBARD*DTG

$$6.3 \quad QBD = CDEQD * QBD + CQDG * QBARD$$

$$6.4 \quad QBERR = QBREF - QBARF$$

$$6.5 \quad EAS_CMD = 17.1865 * SQRT(QBREF)$$

4.1.6 Phase Transition, MEP, and OHALRT Function (TGTRAN)

This module determines all TAEM phase transition, and sets the flag TG_END which will terminate the MCC simulation. Also, this function checks for a minimum entry point situation and switches to the MEP HAC if required. The S-turn phase (IPHASE = 0) is initiated if sufficient excess energy is present. An overhead HAC low energy situation is checked and if true, the TAEM guidance downmoding flag OHALRT is set, which results in a one-time automatic downmoding to the straight-in HAC.

The TAEM program will be entered in the acquisition phase (IPHASE = 1). In the normal situation, the Orbiter will remain in this phase until the heading alinement phase is initiated. However, if the energy state is too high, the S-turn phase will be initiated unless the predicted range is less than the minimum range allowed for an S-turn (RMINST) or the HAC turn angle, PSHA, is greater than a predetermined value, PSSTRN. The main objective of the S-turn is to rotate heading so that a longer ground path can be covered without getting into a geometry problem with respect to the HAC. This geometry problem exists only for HAC turn angles ≤ 135 degrees, but the capability for doing S-turns even with the overhead HAC is possible through proper selection of the value for PSSTRN. The direction in which the S-turn is made depends on which side of the runway centerline the HAC is located and on the Orbiter heading. The logic for selection of the turn direction is

$$S = -YSGN$$

$$\text{IF } S \psi < 0$$

$$\text{and } \psi_{HA} < 90., S = -S$$

$$S = -1 \text{ left bank}$$

$$S = +1 \text{ right bank}$$

Termination of the S-turn occurs when the current energy error becomes less than ENBIAS, i.e., $EOW < EN + ENBIAS$.

During the acquisition phase, the MEP flag will be issued whenever the current energy goes below the MEP energy versus range line and the MEP flag has not already been set (MEP = 0). Transition to the heading alinement phase from the acquisition phase occurs whenever $RCIR < P2TRNC1 RTURN$ where RCIR is the Orbiter's radial distance from the HAC center and RTURN is the HAC radius.

Transition to the prefinal phase (IPHASE = 3) occurs whenever the predicted range is less than the prefinal initiation range value (RPRED3) or if altitude is less than a predetermined value (HMIN3).

After phase 3 is initiated, the TAEM/autoland transition logic is executed to determine when the transition to autoland will take place and the MCC simulation terminated.

TGTRAN SUBPROGRAM

If IPHASE = 3, a logical test is made for termination of TAEM guidance by the statement:

If

(|HERROR| < (H DEL_H1 - DEL_H2) and
 (|Y| < (H Y_RANGE1 - Y_RANGE2) and
 (|GAMMA - GAMSGS(IGS)| < (H GAMMA_COEF1 - GAMMA_COEF2) and
 (|QBERR| < QB_ERROR2) and
 (H_REF1 > H)

or

(H < H_REF2)

then the TAEM guidance termination flag is set to one; i.e., TG_END = 1, and then the TGTRAN function is exited.

Otherwise, (IPHASE ≠ 3), a test for transition to the prefinal phase (IPHASE = 3) is made on the basis of whether the predicted range is less than the prefinal initiation range value (RPRED < RPRED3) or if altitude is less than the minimum prefinal initiation value (H < HMIN3). If this test is true, then equation set 1 is executed:

- 1.1 IPHASE = 3
- 1.2 PHIO = PHIC
- 1.3 PHILIM = PHILM3

1.4 DNZUL = DNZUC2

1.5 DNZLL = DNZLC2

If the prefinal initiation test is false, the appropriate logic based on the current TAEM phase is executed as follows:

For IPHASE = 0 (S-turn):

If the current energy (EOW) is greater than the reference energy (EN) exit TGTRAN.

If the current energy is less than the biased reference energy, (EOW < EN + ENBIAS), then execute equation set 2:

2.1 IPHASE = 1

2.2 PHILIM = PHILM1

For IPHASE = 1 (acquisition)

If the HAC turn angle is less than the S-turn initiate value, and the range to XALI is greater than the S-turn initiate minimum value (PSHA < PSSTRN and DRPRED > RMINST(IGS)) the S-turn initiation logic is executed as follows:

First the energy-over-weight value to initiate an S-turn is calculated as shown in equation 3.1:

3.1 $ES = ES1(IGS) + DRPRED \text{ EDRS}(IGS)$

Next, the test for transition to the S-turn phase is made on the basis of passing the following criteria:

EOW > S

If this test is passed, equation set 3.2 is executed:

3.2.1 IPHASE = 0

3.2.2 PHILIM = PHILMO

3.2.3 S = -YSGN

3.2.4 SPSI = S PSD

3.2.5 If $SPSI <$ and $PSHA < 90$, then $S = -S$

Next, the tests for determining an MEP or OHALRT situation are made. The energy-over-weight values used in the MEP and OHALRT tests are calculated as shown in equation set 3.3:

3.3.1 $EMEP = EMEP_C1(IGS, IEL) + DRPRED\ EMEP_C2(IGS, IEL)$

3.3.2 $EMOH = EMOHC1(IGS) + DRPRED\ EMOHC2(IGS)$

The criteria for switching to the MEP is based on passing the following test:

$EOW < EMEP$ and $MEP = 0$

If this test is true, the MEP flag is set to 1.

Next, the test for determining an OHALRT situation is made as follows:

If $EOW < EMOH$ and $PSHA > PSOHAL$

and $RPRED > RMOH$, then $OHALRT = 1$

The test for transition to the heading alignment phase is made. If $RCIR < 2TRNC1\ RTURN$:

Equation set 3.5 is executed.

3.5.1 $IPHASE = 2$

3.5.2 $PHILIM = PHILM2$

For $IPHASE = 2$ (heading alignment):

No calculations are presently required.

4.1.7 Normal Acceleration Command Function (TGNZC)

The function TGNZC computes the incremental normal acceleration command, NZC, to the FCS for all TAEM phases. The command is generated from a computed altitude rate error term (HDERR), which is formulated from the altitude error input from

TGCOMP, a computed desired altitude rate term (HDREF), and the current, sensed altitude rate input from navigation. The upper and lower limits on the command are based on energy-over-weight, dynamic pressure, and TAEM phase considerations. A filter limiting the rate of change of the limited command (DNZCL) is implemented to smooth the command for all TAEM phases except prefinal (IPHASE = 3).

Upon entering TGNZC, the unlimited normal acceleration command (DNZC) and the upper and lower dynamic pressure limits (QBMXNZ, QBMNNZ) are computed using equation set 1.

- 1.1 $GDH = MIDVAL (GDHC - GDHS H, GDHLL, GDHUL)$
- 1.2 $HDREF = VH DHDRRF$
- 1.3 $HDERR = HDREF - HDOT$
- 1.4 $DNZC = DNZCG GDH (HDERR + HDREQG GDH HERROR)$

Generate minimum dynamic pressure profile

- 1.5 If $MACH < QMACH2$,
 then $MXQBWT = MIDVAL (QBWT1 + QBMSL1(MACH - QMACH1), QBWT2, QBWT1)$
 Else $MXQBWT = MIDVAL (QBWT2 + QBMSL2(MACH - QMACH2), QBWT2, QBWT3)$
- 1.6 $QBALL = MXQBWT \text{ weight}$
- 1.7 $QBMNNZ = QBALL/AMAX1 (COSPHI, CPMIN)$

Generate maximum dynamic pressure profile

- 1.8 IF $MACH > BM1$ then
 $QBMXNZ = MIDVAL (QBMX2$
 $+ QBMXS2(MACH - QBM2), QBMX2, QBMX3)$
 Else $QBMXNZ = MIDVAL(QBMX2 + QBMXS1(MACH - QBM1), QBMX2, QBMX1)$

In the transonic energy region for an overhead HAC approach, the maximum dynamic pressure profile is lowered to target for subsonic conditions at HA initiation.

1.9 If $EQLOWL < EOW$ and $EOW < EQLOWU$ and $PSHA > PSOHQB$,
 then $QBMXNZ = MIDVAL (QBREF2(IGS) - PQBWRR(RPRED2 - R2MAX$
 $+ (EOW - EN)/PEWRR, QBMNNZ, QBMXNZ)$

Next, the upper and lower limits based on the minimum and maximum dynamic pressure ($QBNZUL, QBNZLL$) are computed using equation set 2:

2.1 $QBNZUL = -(QBG1 (QBMNNZ - QBARF) - QBD) QBG2$

2.2 $QBNZLL = -(QBG1 (QBMXNZ - QBARF) - QBD) QBG2$

If $IPHASE = 3$, then $NZC = MIDVAL (DNZC, QBNZLL, QBNZUL)$ and then exit $TGNZC$. Otherwise, the limited and filtered normal acceleration command (NZC) is computed using equation set 3:

3.1 $EMAX = EN + EDELNZ(IGS) MIDVAL (DRPRED$
 $/DEL_R_EMAX(IGS), EDEL C1, EDEL C2)$

3.2 $EMIN = EN - EDELNZ(IGS)$

3.3 $EOWNZUL = (GEUL GDH(EMAX - EOW) + HDERR)GEHDUL GDH$

3.4 $EOWNZLL = (GELL GDH(EMIN - EOW) + HDERR)GEHDLL GDH$

3.5 $DNZCL = MIDVAL (DNZC, EOWNZLL, EOWNZUL)$

3.6 $DNZCL = MIDVAL (DNZCL, QBNZLL, QBNZUL)$

3.7 $DNZCD = MIDVAL ((DNZCL - NZC) CQG, - DNZCDL, DNZCDL)$

3.8 $NZC = NZC + DNZCD DTG$

Finally, for all phases, NZC is limited between the upper and lower phase dependent limits ($DNZLL, DNZUL$) by $NZC = MIDVAL (NZC, DNZLL, DNZUL)$.

4.1.8 Speedbrake Command Function (TGSCB)

This module performs the calculations required to generate the speedbrake command to the FCS for all TAEM phases. The upper limit on the speedbrake command ($DSBLIM$) is Mach-dependent and is set to 65 degrees for supersonic flight and to 98.6 degrees for subsonic flight. During the S-turn phase ($IPHASE = 0$), the speedbrake command is simply $DSBLIM$. For all other phases, the command is based on a nominal speedbrake command ($DSBNOM$), plus proportional and integral terms

based on the dynamic pressure error (QBERR) input from TGCOMP. The integral component of the command (DSBI) is held at last computed value whenever the previous unlimited computed speedbrake command equals or exceeds either the upper or lower speedbrake command limit. Additionally, if the energy-over-weight error with respect to the nominal reference becomes greater than a predetermined value, DEMXSB, the unlimited speedbrake command is reset to zero. Finally, for all phases, the FCS speedbrake command, DSBC_AT, is calculated based on the unlimited command and the calculated limits.

If MACH is greater than DSBCM, the speedbrake command issued to the FCS, DSBC_AT, is set to a constant value of DSBSUP and TGSBC is exited. Otherwise, DSBC_AT is calculated from the following:

First, the command limits are calculated using equation set 1.

$$1.1 \text{ DSBCLL} = \text{MIDVAL} (\text{DSBSUP} + \text{DSBLLS}(\text{MACH} - \text{DSBCM}), 0., \text{DSBSUP})$$

$$1.2 \text{ DSBCUL} = \text{MIDVAL} (\text{DSBSUP} + \text{DSBULS}(\text{MACH} - \text{DSBCM}), \text{DSBSUP}, \text{DSBLIM})$$

Next, the unlimited speedbrake command, DSBC, is calculated using equation set 2. If IPHASE = 0, equation 2.1 is used; otherwise, (IPHASE ≠ 0) equations 2.2 through 2.5 are used.

$$2.1 \text{ DSBC} = \text{DSBLIM}$$

$$2.2 \text{ DSBE} = \text{GSBE QBERR}$$

2.3 If $\text{DSBC} < \text{DSBCUL}$ and $\text{DSBC} > \text{DSBCLL}$, then

$$\text{DSBI} = \text{MIDVAL} (\text{DSBI} + \text{GSBI QBERR DTG}, -\text{DSBIL}, \text{DSBIL})$$

$$2.4 \text{ DSBC} = \text{DSBNOM} - \text{DSBE} - \text{DSBI}$$

2.5 If $\text{EN} - \text{EOW} > \text{DEMXXSB}$, then $\text{DSBC} = 0$

Finally, the speedbrake command issued to the FCS, DSBC_AT, is calculated using equation set 3:

$$3.1 \text{ DSBC_AT} = \text{MIDVAL} (\text{DSBC}, \text{DSBCLL}, \text{DSBCUL})$$

4.1.9 Roll Command Function (TGPHIC)

The function TGPHIC computes the roll command, PHIC_AT, for the lateral axis command to the FCS for all TAEM phases. If the guidance is in the S-turn phase (IPHASE = 0), a constant roll command equal to S PHILIMIT is issued to the FCS where S is the sign of the roll command calculated in TGTRAN, and PHILIMIT is

the maximum roll command allowed. During the acquisition phase (IPHASE = 1), a roll command is given which is proportional to the Orbiter's heading deviation from tangency to the nearest HAC. In the heading alinement phase (IPHASE = 2), the roll command is generated to assure that the Orbiter performs a turn which follows the heading alinement cone. In the prefinal phase (IPHASE = 3), the roll command is generated from a linear combination of the Orbiter's lateral deviation and deviation rate from the runway centerline. The roll command limit, PHILIMIT, is a Mach-dependent schedule with supersonic and phase dependent constraints.

Upon entering TOPHIC, the roll command limit is calculated using

$$\text{PHILIMIT} = \text{MIDVAL} (\text{PHILMSUP} + \text{PHILS} (\text{MACH} - \text{PHIM}), \text{PHILMSUP}, \text{PHILIM})$$

Next, the unlimited roll command, PHIC, is generated on the basis of the current TAEM phase as follows:

1. IPHASE = 0 (S-turn):

1.1 $\text{PHIC} = \text{S PHILIMIT}$

2. IPHASE = 1 (acquisition):

2.1 $\text{PHIC} = \text{GPHI DPSAC}$

3. IPHASE = 2 (heading alinement):

3.1 $\text{RERRC} = \text{RCIR} - \text{RTURN}$

3.2 If $\text{RERRC} > \text{RERRLM}$, then execute equation set 3.3; otherwise execute equation set 3.4.

3.3.1 $\text{PHIC} = \text{GPHI DPSAC}$

3.3.2 If $\text{PHILIMIT} > \text{PHILM1}$, then $\text{PHILIMIT} = \text{PHILM1}$

3.4.1 $\text{RDOT} = -(\text{XCIR XDOT} + \text{YCIR YDOT})/\text{RCIR}$

3.4.2 $\text{PHIP2C} = (\text{VH VH} - \text{RDOT RDOT}) \text{RTD}/(\text{G RTURN})$

3.4.3 $\text{RDOTRF} = -\text{VH} (\text{R1} + 2. \text{R2 PSHA}) \text{RTD}/\text{RTURN}$

3.4.4 $\text{PHIC} = \text{YSGN AMAX1} (0., \text{PHIP2C} + \text{GR RERRC} \\ + \text{GRDOT} (\text{RDOT} - \text{RDOTRF}))$

4. IPHASE = 3 (prefinal):

4.1 $\text{YERRC} = \text{MIDVAL} (-\text{GY Y}, -\text{YERRLM}, \text{YERRLM})$

- 4.2 PHIC = YERRC - GYDOT YDOT
- 4.3 If ABS (PHIC) > PHILMC, then PHILIMIT = PHILM4
- 4.4 If ISR > 0 then
 - 4.4.1 DPHI = (PHIC - PHIO)/ISR
 - 4.4.2 ISR = ISR - 1
 - 4.4.3 PHIC = PHIO + DPHI
 - 4.4.4 PHIO = PHIC

Finally, after the phase dependent unlimited roll command is calculated, the roll command to the FCS, PHIC_AT, is calculated using equation set 5:

$$5.1 \text{ PHIC_AT} = \text{MIDVAL} (\text{PHIC}, -\text{PHILIMIT}, \text{PHILIMIT})$$

4.1.10 Mid Value (MIDVAL)

The MIDVAL function selects the middle valued of three input arguments.

- a. Detailed Requirements.- The function MIDVAL is defined as follows:

$$\text{MIDVAL} (X1, X2, X3) = X1 + X2 + X3 - \text{AMAX1} (X1, X2, X3) - \text{AMIN1} (X1, X2, X3)$$

4.1.11 Resolve to 180 Degrees (RES180)

The function of RES180 is to resolve the input angle to lie within the range ± 180 degrees.

Calling RES180 with the statement

$$\text{ANG} = \text{RES180} (\text{ANG})$$

causes the function RES180 to execute the logic shown in figure 3.

4.2 SEQUENCING REQUIREMENTS

TAEM guidance is to be executed at a rate of 1.04 Hz (0.96 sec). The guidance detailed flow is shown in figure 4.

4.3 INTERFACE REQUIREMENTS

TAEM guidance input and output parameters are listed in table I and table II, respectively.

It is assumed that service routines in the Mission Control Center will compute all of the required inputs to the TAEM guidance and flight control system simulations. The state vector in runway coordinates and the Orbiter heading with respect to the runway centerline are required input data. Also, required input data are the dynamic pressure, true airspeed, altitude rate, geodetic altitude above the runway, horizontal component of ground relative velocity, body roll, and pitch angles. The dynamic pressure and true airspeed data requirements will be satisfied by using the true airspeed and air density from the environment to compute dynamic pressure. Derivation of these data is presented in appendix A.

It is also assumed that the output parameters consisting of predicted range, TAEM phase counter, equivalent airspeed command, altitude error from reference glide slope, filtered dynamic pressure, and the energy deficiency alert flag will be sent to the display processor.

The output parameters consisting of the normal load factor command, roll angle command, and speedbrake angle command are sent to the TAEM digital autopilot.

5.0 TAEM DIGITAL AUTOPILOT (TDAP)

5.1 REQUIREMENTS OVERVIEW

The TAEM digital autopilot (TDAP) is the first order simulation of the Shuttle Orbiter flight control system, which performs the function of controlling the body attitude and speedbrake responses during the TAEM guidance phase of the entry trajectory. Formulation requirements for this function are presented in figures 5 through 8, and are discussed in detail in the subsequent sections.

5.2 FUNCTION MODULES

The module TDAP consists of the following principal subfunction modules.

- a. The roll channel calculates body roll rate using the roll attitude and TAEM guidance roll angle command.

- b. The pitch channel calculates body pitch rate as a function of normal load factor, attitudes, true airspeed, and guidance normal load factor command.
- c. The yaw channel calculates the body yaw rate required to produce coordinated turns and null lateral load factors.
- d. The attitude integrator converts body rates to a rotation matrix and updates the body to inertial attitude matrix.
- e. The speedbrake channel calculates speedbrake actuator rates using the guidance deflection command and hinge moment constraints, and updates the speedbrake deflection.

The following utility subroutines are used by TDAP:

- a. FILTER generalized first order filter
- b. SMOOTH guidance command smoother
- c. EIGEN eigen vector rotation of an orthogonal matrix
- d. MIDVAL applies upper and lower limits
- e. MAMP multiplies two matrices
- f. CROSS vector cross product
- g. Unit unit vector calculations
- h. AMIN1 select most negative argument
- i. AMAX1 select most positive argument
- j. SIN trigonometric sine
- k. COS trigonometric cosine
- l. MOD modular counter conversion

Formulation is supplied in this document for the functions of TDAP and the first three utility subroutines in the above list.

5.3 INTERFACE REQUIREMENTS

Inputs required by TDAP are listed in table IV. Outputs are listed in table V. Constants required are shown in table VI.

5.4 SEQUENCING REQUIREMENTS

TDAP is to be executed at a rate of 2.08 Hz (0.48 sec) during the TAEM guidance phase after guidance execution.

5.5 INITIALIZATION REQUIREMENTS

The internal variable LOOP in TDAP must have the value zero on the first pass execution. The speedbrake deflection, body to inertial attitude matrix, body roll rate, and yaw rate must be available at the initial execution pass as the last pass values from the entry guidance phase so they may be subsequently updated by TDAP.

5.6 METHOD

The MCC entry profile planning processor is a three-degree-of-freedom simulation in the sense that it includes complete second order equations of motion for three dimensional translational components only. The rotational equations of motion are first order and assume that commanded body attitude rates are achieved instantaneously. This assumption eliminates the requirement to calculate rotational moments and accelerations for the purpose of integrating rates and attitudes as would be done in a six-degree-of-freedom simulation. The formulation for attitude control consists of calculating desired body rates using the guidance roll and load factor commands. This formulation is derived from the level C flight software requirements for the entry flight control system (reference) and provides virtually the same trajectory and maneuvering characteristics as a six-degree-of-freedom simulation.

The first calculations upon entering TDAP are to obtain the trigonometric functions of pitch, roll, and angle-of-attack attitude angles that are used subsequently.

The roll channel calculations are performed next by first calculating the scheduled roll rate gain and roll rate limit. The slow rate TAEM roll command is then smoothed by linear extrapolation to the faster TDAP rate. The roll attitude error is calculated and converted to a roll rate that is then limited.

The pitch channel begins with calculation of the open loop coordinated turn pitch rate and smoothing it through a first order lag filter. The TAEM NZ command is linearly smoothed, and the load factor bias for equilibrium turn compensation is calculated. The smoothed NZ command is converted to a C* command that will produce the pitch rate required to maintain a constant load factor at the desired level. The C* command and turn compensation terms are summed to produce the total desired load factor, and load factor error is calculated.

The load factor error is converted to a negative pitch rate command, smoothed through a first order lag filter, summed with the coordinated turn pitch rate, and filtered again to obtain the unlimited negative pitch rate. Angle-of-attack limits are obtained from their scheduled profiles, and the maximum allowable pitch rates are determined by the proximity from the angle-of-attack limits and

load factor limits. The most restrictive of these limits are then applied to obtain the final pitch rate.

The yaw channel begins with calculation of the lateral load factor yaw rate gain. The yaw rate required to maintain a coordinated turn maneuver about the X-stability axis is determined from body roll rate and angle of attack. The sensed body lateral load factor is smoothed through a first order filter, converted to a yaw rate, and summed with the coordination term to obtain the final yaw rate.

To integrate the vehicle attitude, body rates are assumed constant over the computation interval and, therefore, form the eigen vector in body coordinates about which the body to inertial attitude matrix is to be rotated. This matrix rotation is performed in the subfunction EIGEN by constructing the transformation matrix which, when premultiplied by the present attitude matrix, will produce the direction cosine matrix that represents the attitude resulting from an eigen rotation through the total angle determined by the desired rate times the time interval. The columns of this matrix, which represent the updated body axes in the inertial frame, are then normalized to preserve orthogonality.

The final function of TDAP is the first order simulation of speedbrake actuator response to the guidance commanded deflection. The principal considerations in determining the actuator rate include the flight software priority rate limiting and actuator ability to overcome the aerodynamic hinge moment. Since the Orbiter speedbrake actuator system gain is sufficient to produce time constants less than the TDAP cycle interval, a gain of $1/DT^2$ is used to prevent an oscillatory characteristic. The maximum opening rate is determined as a square root function of hinge moment. Maximum closing rate is constant at a value selected by the deflection in relation to the software soft stop position. The deflection error is converted to a rate and limited. This rate is integrated rectangularly to obtain the speedbrake deflection which is then limited to the position constraints.

The first order filter subfunction utilizes the nodal implementation of the generalized S-transform $(C_1S + C_2)/(S + a)$. Conversion to the equivalent Z-transform is performed on the initial pass to obtain the required coefficients for the recursive difference equation.

Detailed formulation for the function modules TDAP, FILTER, SMOOTH, and EIGEN is shown in figures 5 through 8.

6.0 REFERENCE

1. Space Shuttle Orbital Flight Test Level C Functional Subsystem Software Requirements. Part A - Guidance, SD 76-SH-0001B, Nov. 19, 1976.
Part C - Flight Control - Entry, SD 76-SH-0007, Nov. 26, 1976.

TABLE I.- TAEM GUIDANCE INPUTS

<u>Description</u>	<u>Type</u>	<u>Source</u>	<u>Unit</u>	<u>Internal name</u>
Geodetic altitude of vehicle center of mass above RW	F	GCOMP ^a	ft	H
Negative z component of velocity in topodetic coordinates	F	GCOMP	fps	HDOT
x-component of position in runway coordinates	F	GCOMP	ft	X
y-component of position in runway coordinates	F	GCOMP	ft	Y
Magnitude of Earth relative velocity vector	F	GCOMP	fps	V
Topodetic horizontal component of Earth relative velocity	F	GCOMP	fps	VH
x-component of velocity in runway coordinates	F	GCOMP	fps	XDOT
Course with respect to centerline	F	GCOMP	deg	PSD
Mach number	F	GCOMP	--	MACH
Dynamic pressure	F	GCOMP	psf	QBAR
Cosine of body roll Euler angle	F	GCOMP	--	COSPFI
Secant of body pitch Euler angle	F	GCOMP	--	SECTH
Orbiter mass	F	AERO	slugs	WEIGHT
True airspeed	F	GCOMP	fps	TAS
y-component of velocity in runway coordinates	F	GCOMP	fps	YDOT

^aGuidance and flight control input parameter calculations (appendix).

TABLE I.- Concluded

<u>Description</u>	<u>Type</u>	<u>Source</u>	<u>Unit</u>	<u>Internal name</u>
Topodetic Earth relative flight-path angle	F	GCOMP	deg	GAMMA
Flag indicating glide slope desired	I	MED ^b	--	GI_CHANGE
HAC radius	F	EGRT	ft	RTURN
HAC turn angle	F	EGRT	deg	PSHA
OVHD/STRT approach mode flag	I	MED ^c	n.d.	OVHD(I)
Automatic downmode inhibit flag	I	MED ^d	n.d.	ORAHAC(I)
R/L cone indicator	F	EGRT	n.d.	YSGN
Selected runway ID flag	I	SITE ^e selection	n.d.	RWID
Velocity to toggle OVHD/STRT HAC status	F	MED ^f	fps	VTOGL

^bMED input based on surface wind conditions (initialized in mission constants table to zero).

^cMED input to select approach mode for each runway (overhead = 1, straight-in = 0), I = 1 for primary RW, I = 2 for secondary RW. Initialized to 1 in mission constants table.

^dMED input to inhibit automatic downmode to straight-in for each runway (inhibit = 1), initialized in mission constants table to zero.

^eRWID = 1 for primary runway, RWID = 2 for secondary runway.

^fMED input to simulate manual HAC toggle. Initialized to zero in mission constants table.

TABLE II.- TAEM GUIDANCE OUTPUTS

<u>Description</u>	<u>Type</u>	<u>Unit</u>	<u>Internal name</u>	<u>Destination</u>
Commanded body normal load factor increment from equilibrium	F	g's	NZC	TDAP ^a
Roll angle command	F	deg	PHIC_AT	TDAP
Range error from reference altitude profile	F	ft	DEL RNG	DSP ^b
Heading error to HAC tangency	F	deg	DPSAC	DSP
Speedbrake angle command (hinge line)	F	deg	DSBC_AT	TDAP
E/W at which the MEP is selected	F	ft	EMEP	DSP
Energy per unit weight	F	ft	EOW	DSP
E/W at which an S-turn is initiated	F	ft	ES	DSP
Predicted range to runway threshold	F	ft	RPRED	DSP
TAEM phase counter	I	--	IPHASE	DSP
TAEM guidance termination flag	I	--	TG_END	DSP ^c
Equivalent airspeed command	F	knots	EAS_CMD	DSP
Altitude error from reference profile	F	ft	HERROR	DSP
Filtered dynamic pressure	F	psf	QBARF	DSP

^aTAEM digital autopilot.

^bDisplays.

^cThis flag will terminate the MCC simulation.

TABLE II.- Concluded

<u>Description</u>	<u>Type</u>	<u>Unit</u>	<u>Internal name</u>	<u>Destination</u>
TAEM automatic downmode flag	I	--	OHALRT	DSP ^d
Minimum entry point flag	I	--	MEP	DSP

^dSet to 1 when automatic downmode to straight-in has occurred, otherwise zero.

TABLE III.- TAEM GUIDANCE CONSTANTS

Symbol	Description	Value	Unit
CDEQD	Gain in calculation of QBD	0.68113143	--
CPMIN	Minimum value CosPHI	0.707	--
CQDG	Gain used in calculation of QBD	0.31886857	--
CQG	Gain used in calculation of DNZCD and QBARD	0.5583958	--
CUBIC_C3(1) ^a	Quadratic term coefficient in HREF CURVE FIT	-4.7714787E-7	ft/ft ²
CUBIC_C3(2)	Quadratic term coefficient in HREF CURVE FIT	TBD	ft/ft ²
CUBIC_C4(1)	Cubic term coefficient in HREF CURVE FIT	-2.4291527E-13	ft/ft ³
CUBIC_C4(2)	Cubic term coefficient in HREF CURVE FIT	TBD	ft/ft ³
DEL_H1	Altitude error coefficient	0.19	--
DEL_H2	Altitude error coefficient	900.	ft
DEL_R_EMAX(1)	Constant used in computing E max	54000.	ft
DEL_R_EMAX(2)	Constant used in computing E max	TBD	ft
DNZCG	Gain used to compute DNZC	0.01	g/s
DNZLC1	Phases 0, 1, and 2 NZC lower limit	-0.5	g
DNZLC2	Phase 3, NZC lower limit	-0.5	g
DNZUC1	Phases 0, 1, and 2 NZC upper limit	0.5	g

^aArray dimension 2.

TABLE III.- Continued

Symbol	Description	Value	Unit
DNZUC2	Phase 3 NZC upper limit	0.5	g
DR3	Delta range value for RPRED3	8000.	ft
DSBCM	Mach at which speedbrake modulation begins	0.95	--
DSBLIM	Maximum value of DSBC	98.6	deg
DSBSUP	Supersonic fixed speedbrake deflection	65.	deg
DSBIL	Limit on DSBI	20.	deg
DSBNOM	Nominal speedbrake command	65.	deg
DTG	TAEM guidance cycle interval	0.96	sec
DTR	Degrees-to-radian conversion factor	0.0174533	rad/deg
EDEL C1	Constant used in determination of EMAX	1.	ND
EDEL C2	Constant used in determination of EMAX	1.	ND
EDELNZ(1)	Energy delta from the nominal energy line slope for the S-turn	4000.	ft
EDELNZ(2)	Energy delta from the nominal energy line slope for the S-turn	TBD	ft
EDRS(1)	Slope for S-turn energy line	.69946182	ft ² /ft
EDRS(2)	Slope for S-turn energy line	TBD	ft ² /ft
EMEP_C1(1,1) ^b	Y-intercept of MEP energy line	-3263.	ft
EMEP_C1(2,1)	Y-intercept of MEP energy line	-3263	ft

^b2 x 2 array given column wise.

TABLE III.- Continued

<u>Symbol</u>	<u>Description</u>	<u>Value</u>	<u>Unit</u>
EMEP_C1(1,2)	Y-intercept of MEP energy line	12088.	ft
EMEP_C1(2,2)	Y-intercept of MEP energy line	12088.	ft
EMEP_C2(1,1)	Slope of MEP energy line	.51554944	ft ² /ft
EMEP_C2(2,1)	Slope of MEP energy line	TBD	ft ² /ft
EMEP_C2(1,2)	Slope of MEP energy line	.265521	ft ² /ft
EMEP_C2(2,2)	Slope of MEP energy line	TBD	ft ² /ft
EN_C1(1,1)	Y-intercept of nominal energy line	949.	ft ² /ft
EN_C1(2,1)	Y-intercept of nominal energy line	949.	ft ² /ft
EN_C1(1,2)	Y-intercept of nominal energy line	15360.	ft ² /ft
EN_C1(2,2)	Y-intercept of nominal energy line	15360.	ft ² /ft
EN_C2(1,1)	Slope of nominal energy line	.6005	ft ² /ft
EN_C2(2,1)	Slope of nominal energy line	TBD	ft ² /ft
EN_C2(1,2)	Slope of nominal energy line	.46304	ft ² /ft
EN_C2(2,2)	Slope of nominal energy line	TBD	ft ² /ft
ES1(1)	Y-intercept of S-turn energy line	4523.	ft
ES1(2)	Y-intercept of S-turn energy line	4523.	ft

TABLE III.- Continued

<u>Symbol</u>	<u>Description</u>	<u>Value</u>	<u>Unit</u>
EOW_SPT(1)	Range at which to change slope and intercept on the MEP and nominal energy line	76068.0	ft
EOW_SPT(2)	Range at which to change slope and intercept on the MEP and nominal energy line	TBD	ft
G	Earth gravitational constant	32.174	ft/s ²
GAMMA_COEF1	Flightpath error coefficient	0.0007	deg/ft
GAMMA_COEF2	Flightpath error coefficient	3.0	deg
GAMSGS(1)	A/L steep glideslope angle	-22.	deg
GAMSGS(2)	A/L steep glideslope angle	TBD	deg
GDHC	Constant for computing GDH	2.0	--
GDHLL	GDH lower limit	0.3	--
GDHS	Slope for computing GDH	7.0E-5	ft ⁻¹
GDHUL	GDH upper limit	1.0	--
GEHLL	Gain used in computing EOWNZLL	0.01	g/fps
GEHDUL	Gain used in computing EOWNZUL	0.01	g/fps
GELL	Gain used in computing EOWNZLL	0.1	sec ⁻¹
GEUL	Gain used in computing EOWNZUL	0.1	sec ⁻¹
GPHI	Heading error gain for computing PHIC	2.5	--
GR	Gain on RCIR in computing HA roll angle command	0.005	deg/ft

TABLE III.- Continued

<u>Symbol</u>	<u>Description</u>	<u>Value</u>	<u>Unit</u>
GRDOT	Gain on dRCIR/dt in computing HA roll angle command	0.2	deg/fps
GSBE	Speedbrake proportional gain on QBERR	1.5	deg/psf
GSBI	Gain on QBERR integral in computing speedbrake command	0.1	deg/psf-s
GY	Gain on Y in computing PFL roll angle command	0.07	deg/ft
GYDOT	Gain on YDOT in computing PFL roll angle command	0.7	deg/fps
H_REF1	Altitude reference	10000.	ft
H_REF2	Altitude reference	5000	ft
HALI(1)	Altitude at A/L steep glideslope at MEP	10018.	ft
HALI(2)	Altitude at A/L steep glideslope at MEP	TBD	ft
HDREQG	Gain used to compute HDREQ	0.1	--
HFTC(1)	Altitude of A/L steep glideslope at nominal entry point	12018.	ft
HFTC(2)	Altitude of A/L steep glideslope at nominal entry point	TBD	ft
MACHAD ^a	Mach no. to use air data	0.75	--
PBGC(1)	Linear coefficient of range for HREF and lower limit for DHDRRF	0.1112666	--

^aUsed in GCOMP calculations (appendix).

TABLE III.- Continued

<u>Symbol</u>	<u>Description</u>	<u>Value</u>	<u>Unit</u>
PBGC(2)	Linear coefficient of range for HREF and lower limit for DHDRRF	TBD	--
PBHC(1)	Altitude reference for DRPRED = PBRC	78161.826	ft
PBHC(2)	Altitude reference for DRPRED = PBRC	TBD	ft
PBRC(1)	Maximum range for cubic altitude reference	256527.82	ft
PBRC(2)	Maximum range for cubic altitude reference	TBD	ft
PBRCQ(1)	Range breakpoint for QBREF	89971.082	ft
PBRCQ(2)	Range breakpoint for QBREF	TBD	ft
PHAVGC	Constant for computing PHAVG	63.33	deg
PHAVGLL	Lower limit for PHAVG	30.	deg
PHAVGS	Slope for computing PHAVG	13.33	deg
PHAVGUL	Upper limit for PHAVG	50.	deg
PHILMO	Roll command limit S-turn	50.	deg
PHILM1	Roll command limit acquisition phase	50.	deg
PHILM2	Roll command limit heading alinement phase	60.	deg
PHILM3	Roll command limit prefinal phase	30.	deg
PHILMSUP	Supersonic roll angle command limit	30.	deg
PHIM	Mach at which supersonic roll command limit is removed	0.95	--

TABLE III.- Continued

<u>Symbol</u>	<u>Description</u>	<u>Value</u>	<u>Unit</u>
P2TRNC1	Constant used in Phase 2 transition logic	1.1	--
QB_ERROR2	Dynamic pressure error bound	24.	psf
QBARDL	Limit on QBARD	5.	psf/s
QBC1(1)	Slope of QBREF for DRPRED > PBRQC	3.6086999E-4	psf/ft
QBC1(2)	Slope of QBREF for DRPRED > PBRQC	TBD	psf/ft
QBC2(1)	Slope of QBREF for DRPRED < PBRQC	-1.1613301E-3	psf/ft
QBC2(2)	Slope of QBREF for DRPRED < PBRQC	TBD	psf/ft
QBG1	Gain used to compute QBNZUL and QBNZLL	0.1	s ⁻¹
QBG2	Gain used to compute QBNZUL and QBNZLL	0.125	s-g/psf
QBMXS2	Slope of QBMXNZ with MACH > OBM2	0.0	psf
QBMX1	Constant for computing QBMXNZ	340.	psf
QBMX2	Constant for computing QBMXNZ	300.	psf
QBMX3	Constant for computing QBMXNZ	300.	psf
QBM1	Mach breakpoint for computing QBMXNZ	1.05	--
QBM2	Mach breakpoint for computing QBMXNZ	1.7	--
QBRL(1)	QBREF lower limit	180.	psf
QBRL(2)	QBREF lower limit	TBD	psf

TABLE III.- Continued

Symbol	Description	Value	Unit
QBRML(1)	QBREF middle limit	220.	psf
QBRML(2)	QBREF middle limit	TBD	psf
QBRUL(1)	QBREF upper limit	285.	psf
QBRUL(2)	QBREF upper limit	TBD	psf
RERRLM	Limit of RERRC	7000.	ft
RFTC	Roll fader time constant	5.	S
RMINST(1)	Minimum range allowed to initiate S-turn phase	122204.6	ft
RMINST(2)	Minimum range allowed to initiate S-turn phase	TBD	ft
RTD	Radians-to-degrees conversion factor	57.29578	deg/rad
TGGS(1)	Tan of steep glideslope for autoland	-0.40402623	--
TGGS(2)	Tan of steep glideslope for autoland	TBD	--
WT_GS1	Maximum shuttle orbiter weight	8000.	slugs
XA(1)	Steep glideslope intercept	-5000.	ft
XA(2)	Steep glideslope intercept	TBD	ft
YERRLM	Limit on YERRC	280.	deg
Y_RANGE1	Crossrange coefficient	0.18	--
Y_RANGE2	Crossrange coefficient	800.	ft

TABLE III.- Continued

<u>Symbol</u>	<u>Description</u>	<u>Value</u>	<u>Unit</u>
DR4	Range from HAC for phase 3	2000.	ft
HMEP(1)	Minimum entry point altitude	6000.	ft
HMEP(2)	Minimum entry point altitude	6000.	ft
DEMFSB	Maximum EOW error for speedbrakes out	10000.	ft
DHOH1	Altitude reference dev/spiral slope	0.11	n.d.
DHOH2	Altitude reference dev/spiral range bias	35705.	ft
DHOH3	Altitude reference dev/spiral max shift of altitude	6000.	ft
EQLOWU	Upper EOW of region for OVHD that QBMXNZ is lowered	85000.	ft
ESHFMX	Max shift of EN target at HAC	20000.	ft
PHILS	Constant used to calculate PHILIMIT	-300.	deg
PEWRR	Partial of EN with respect to range at R2MAX	0.52	n.d.
PQBWRR	Partial of QBAR with respect to range with constraints of E = EN and MACH = PHIM	0.006	psf/ft
PSHARS	PSHA reset value	270.	deg
PSRF	Maximum PSHA for adjusting RF	90.	deg
PSOHAL	Minimum PSHA to issue OHALRT	200.	deg
PSOHQB	Minimum PSHA for which QBMXNZ is lowered	0.	deg
PSSTRN	Maximum PSHA for S-turn	200.	deg

TABLE III.- Continued

Symbol	Description	Value	Unit
QBREF2(1)	QBREF at R2MAX	185.	psf
QBREF2(2)	QBREF at R2MAX	185.	psf
QBMSL1	Slope of MXQBWT with Mach	-0.0288355	psf/slug
QBMSL2	Slope of M QBWT with Mach	0.00570829	psf/slug
QBWT1	Constant for computing QBLL	0.0233521	psf/slug
QBWT2	Constant for computing QBLL	0.01902763	psf/slug
QBWT3	Constant for computing QBLL	0.03113613	psf/slug
QMACH1	Mach break point for MXQBWT	0.89	n.d.
QMACH2	Mach break point for MXQBWT	1.15	n.d.
RFMN	Minimum HAC spiral turn radius on final	5000.	ft
RFMX	Maximum HAC spiral turn radius on final	14000.	ft
RFO	Initial HAC spiral turn radius on final	14000.	ft
DNZCDL	NZC rate limit	0.1	g/sec
DRFK	RF adjust gain ($-0.8/\tan 15^\circ$)	-3.	n.d.
DSBLLS	Constant used to calculate DSBCLL	650.	deg
DSBULS	Constant used to calculate DSBCUL	-336.	deg
EMOHC1(1)	Constant EOW used to compute EMNOH	-3894.	ft
EMOHC1(2)	Constant EOW used to compute EMNOH	-3894.	ft
EMOHC2(1)	Slope of EMNOH with range	0.51464	n.d.

TABLE III.- Concluded

<u>Symbol</u>	<u>Description</u>	<u>Value</u>	<u>Unit</u>
EMOHC2(2)	Slope of EMNOH with range	0.51464	n.d.
ENBIAS	EOW bias for S-turn off	0.	ft
EQLOWL	Lower EOW of region for OVHD that QBMXNZ is lowered	60000.	ft
RMOH	Minimum RPRED to issue OHALRT	273500.	ft
R1	Linear coefficient of spiral	0.	ft/deg
R2	Quadratic coefficient of spiral	0.093	ft/deg ²
R2MAX	Maximum range on HAC to be subsonic with nominal QBAR	115000.	ft
PHILM4	Bank limit for large bank command	60.	deg
PHILMC	Large bank command	100.	deg
QBMXS1	Slope of QBMXNZ with Mach < QBM1	-400.	psf
HMIN3	Minimum altitude for prefinal	7000.	ft

TABLE IV.- TDAP INPUTS

<u>Name</u>	<u>Description</u>	<u>Source</u>	<u>Unit</u>
THETA	Topodetic to body pitch angle	GCOMP ^b	deg
PHI	Topodetic to body roll angle	GCOMP	deg
NZC	Commanded body normal load factor increment from equilibrium	TAEM	g
TAS	True airspeed	GCOMP	fps
NZ	Body normal load factor	GCOMP	g
MACH	MACH number	GCOMP	--
ALPHA	Angle of attack	GCOMP	deg
NY	Body lateral load factor	GCOMP	g
PHIC	Roll angle command	TAEM	deg
DSBC	Speedbrake command (hinge line)	TAEM	deg
DELBA ^a	Speedbrake deflection	AERO	deg
HMSB	Speedbrake hinge moment	AERO	in-lb
CBIA ^a	Body to inertial attitude matrix	EDAP ^c	--
p ^a	Body roll rate	EDAP	deg/sec
r ^a	Body yaw rate	EDAP	deg/sec

^aRequired only for initialization.

^bGuidance and flight control input parameter calculations.

^cEntry guidance phase autopilot.

TABLE V.- TDAP OUTPUTS

<u>Name</u>	<u>Description</u>	<u>Destination</u>	<u>Unit</u>
P	Body roll rate	MON ^a	deg/sec
Q	Body pitch rate	MON	deg/sec
R	Body yaw rate	MON	deg/sec
CBI	Body to inertial attitude matrix	GCOMP	--
DELB	Speedbrake deflection (hinge line)	AERO	deg
DEOTRC	Speedbrake rate	MON	deg/sec

^aNot presently used except for possible checkout monitoring.

TABLE VI.- TDAP CONSTANTS

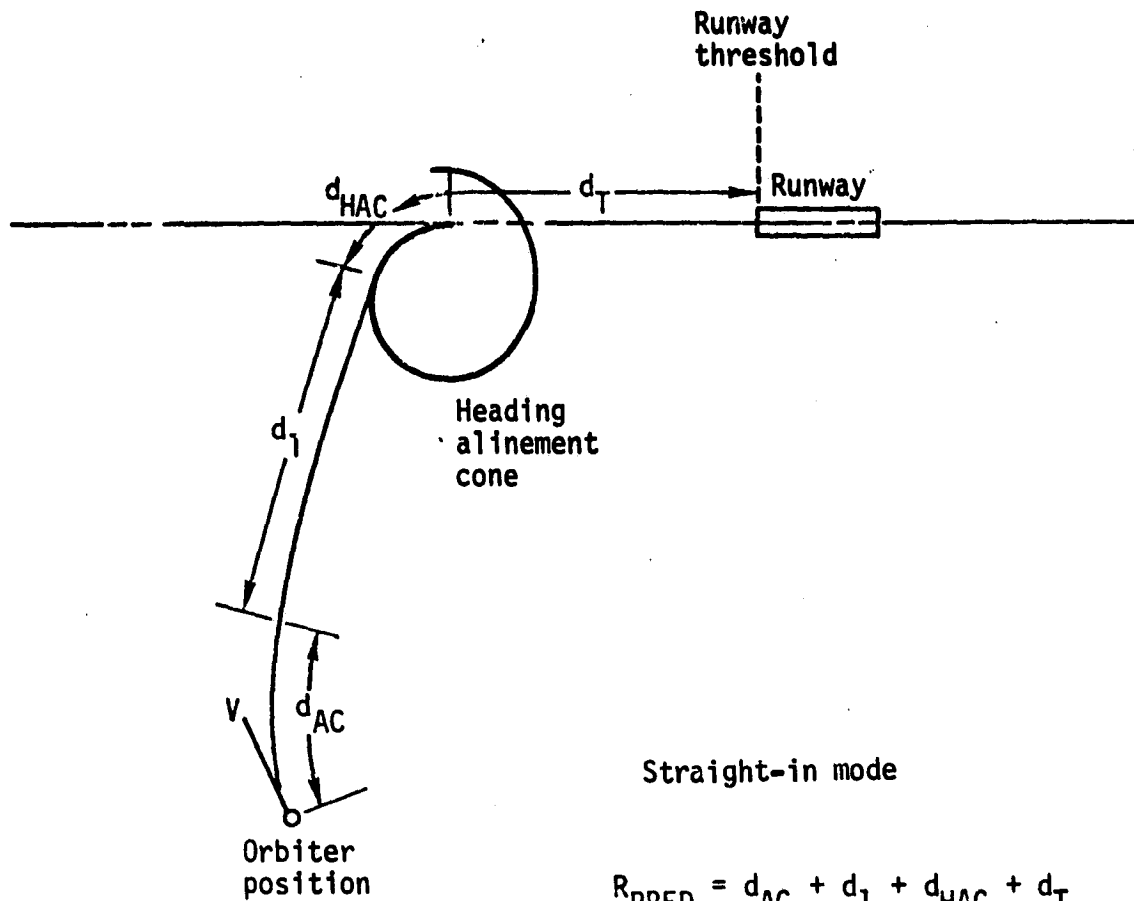
<u>Name</u>	<u>Description</u>	<u>Value</u>	<u>Unit</u>
AMNI	Y-intercept of minimum alpha profile	-16.	deg
AMNLL	Minimum alpha lower limit	-100.	deg
AMNS	Slope of minimum alpha profile	20.	deg/Mach
AMNUL	Minimum alpha upper limit	-100.	deg
AMXLL	Maximum alpha lower limit	100.	deg
AMXML	Maximum alpha middle limit	100.	deg
AMXM1	Maximum alpha profile first breakpoint	0.6	Mach
AMXM2	Maximum alpha profile second breakpoint	1.5	Mach
AMXS1	Lower slope of maximum alpha profile	-25.	deg/Mach
AMXS2	Upper slope of maximum alpha profile	10.	deg/Mach
AMXUL	Maximum alpha upper limit	100.	deg
CPMIN	Lower limit on cosine PHI	0.5	--
DSBLIM	Maximum speedbrake deflection limit	98.6	deg
DT2	TDAP time interval	0.48	sec
DTG	TAEM guidance interval	0.96	sec
DTR	Degrees to radians conversion	0.0174533	rad/deg
GBDOT	Gain to scale compensated yaw rate	1.	--
GF11	Filter 1 first coefficient	0.	--

TABLE VI.- Continued

<u>Name</u>	<u>Description</u>	<u>Value</u>	<u>Unit</u>
GF12	Filter 1 second coefficient	1.	--
GF13	Filter 1 third coefficient	1.	--
GF21	Filter 2 first coefficient	0.2	--
GF22	Filter 2 second coefficient	2.	--
GF23	Filter 2 third coefficient	2.	--
GF31	Filter 3 first coefficient	2.	--
GF32	Filter 3 second coefficient	2.	--
GF33	Filter 3 third coefficient	2.	--
GF41	Filter 4 first coefficient	0.	--
GF42	Filter 4 second coefficient	5.	--
GF43	Filter 4 third coefficient	5.	--
GPI	Y-intercept of GPBANK profile	4.4	(deg/s)/deg
GPLL	GPBANK lower limit	0.5	(deg/s)/deg
GPS	Slope of GPBANK profile	-3.25	(deg/s)/(deg-Mach)
GPUL	GPBANK upper limit	1.8	(deg/s)/deg
GQA	Alpha limit pitch rate gain	1.	(deg/s)/deg
GQN	Pitch rate gain	3.36	(deg/s)/g
GQNL	Load factor limit pitch gain	4.	(deg/s)/g
GRPHI	Cross-over velocity gain	-1845.06	(deg/ft)/g-sec ²
GYI	Y-intercept of GRAY profile	23.6	(deg/s)/g

TABLE VI.- Concluded

<u>Name</u>	<u>Description</u>	<u>Value</u>	<u>Unit</u>
GYLL	GRAY lower limit	3.2	(deg/s)/g
GYS	Slope of GRAY profile	-6.8	(deg/s)/(g-Mach)
GYUL	GRAY upper limit	10.	(deg/s)/g
HMS	Speedbrake stall hinge moment	2.40E6	in-lb
NZMAX	Maximum normal load factor	100.	g
NZMIN	Minimum normal load factor	-100.	g
PCI	Y-intercept of PCLIM profile	30.	deg/sec
PCLL	PCLIM lower limit	5.	deg/sec
PCS	Slope of PCLIM profile	-16.667	deg/(sec-Mach)
PCUL	PCLIM upper limit	20.	deg/sec
SBLOW	Minimum speedbrake deflection for Mach \leq SBMSW	5.	deg
SBMID	Minimum speedbrake deflection for Mach $>$ SBMSW	15.	deg
SBMSW	Mach number to switch minimum speedbrake deflection	0.6	Mach
SBRLC	Speedbrake maximum closing rate	10.86	deg/sec
SBRLO	Speedbrake maximum opening rate	6.1	deg/sec
SBRS	Speedbrake maximum closing rate below soft stop	1.	deg/sec
SBSOFT	Speedbrake soft stop	12.	deg
TPLIM	Upper limit on tangent PHI	1.	--
VCO	Cross-over velocity	549.125	fps

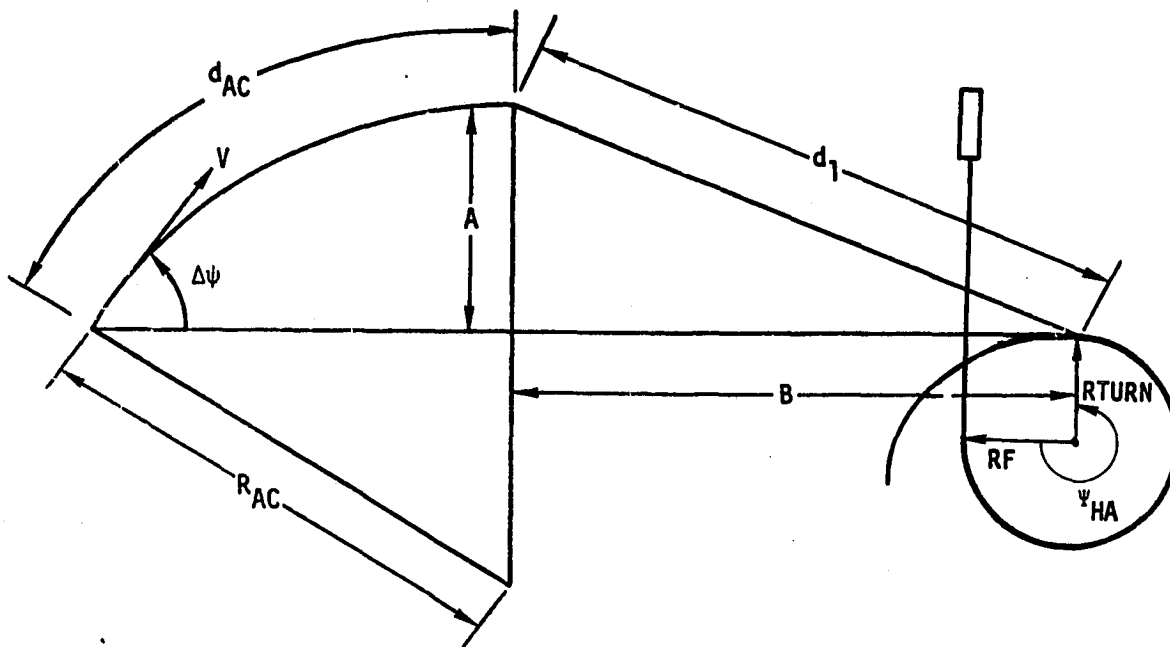


Straight-in mode

$$R_{PRED} = d_{AC} + d_1 + d_{HAC} + d_T$$

(Computed groundtrack range)

Figure 1.- TAEM guidance groundtrack predictor geometry.



Overhead HAC mode

Figure 2.- GTP geometry for acquisition phase.

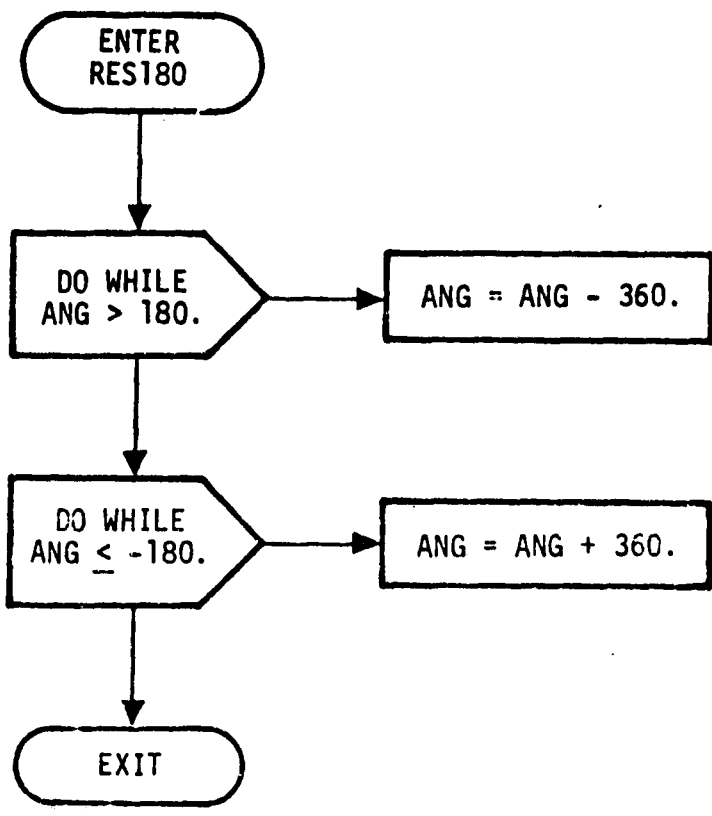


Figure 3.- Resolve to 180 degrees (RES180).

TGEXEC

* IF IRESET = 1 OR RNID # RNIDO THEN CALL TGINIT: *

* CALL TGXHAC: *

* CALL GYP: *

* CALL TGCOMP: *

* CALL TGTRAN: *

* CALL TGNZC: *

* CALL TGSBC: *

* CALL TGFHIC: *

TGINIT

IF IRESET = 1, RWIDO = RWID

```
*****  
* IPHASE = 1;  
* ISK = RFTC / DTG;  
* RF = RFO  
* MEP = 0  
* OHALRT = 0  
* DSBI = 0;  
* PHILIM = PHILMI;  
* DNZUL = DNZUCI;  
*****
```

```
*****  
* DNZLL = DNZLCI;  
* QBARF = QBAR;  
* QBD = 0;  
* TG_END = 0  
* IRESET = 0  
*****
```

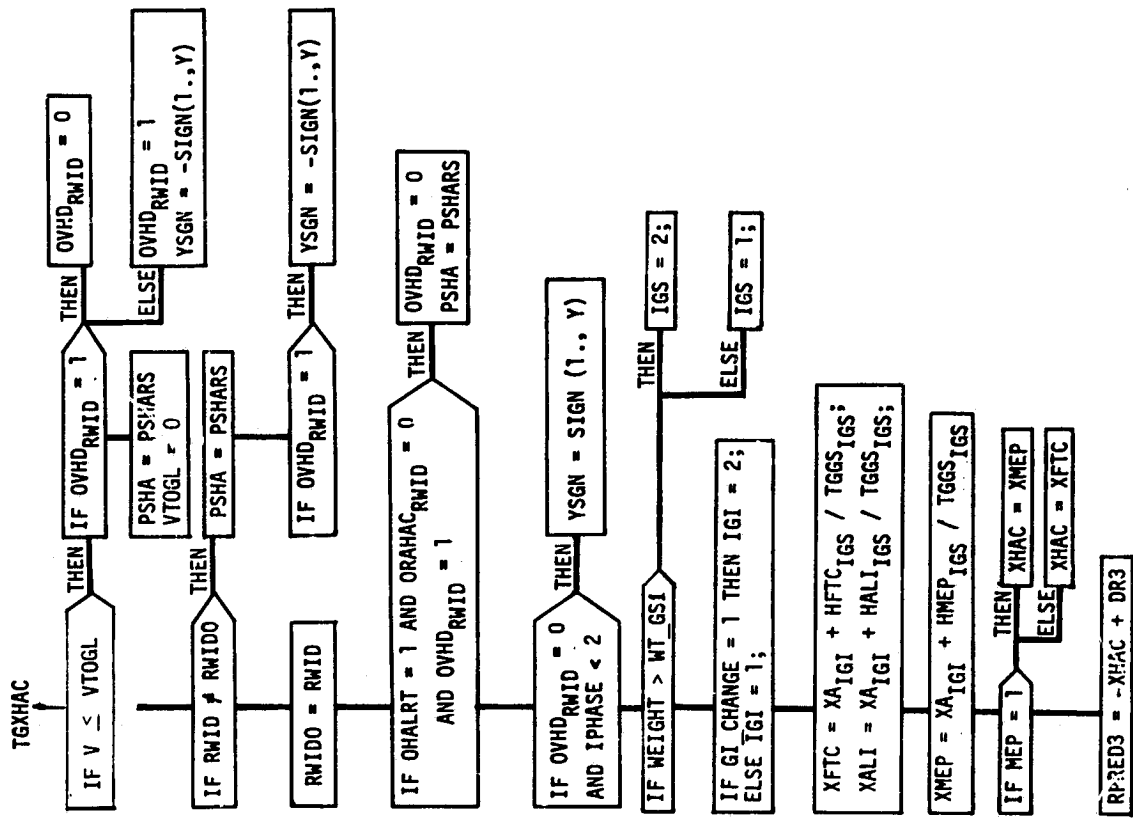



Figure 4.- Continued.

IGCOMP:

```
*****  
* DRPRED = RPRED + XALI *  
* EOW = H + V V / (2G); *  
*****
```

```
*****  
* IF DRPRED < EDM_SPT IGS *  
*****  
* THEN *****  
* IEL = 2; *  
*****  
* ELSE *****  
* IEL = 1; *  
*****
```

```
*****  
* EN = EN_C1 IGS, IEL + DRPRED EN_C2 IGS, IEL *  
* -MIDVAL(EN_C2 IGS, 1 (RPRED2 - R2MAX), 0., ESHFMX) *  
*****
```

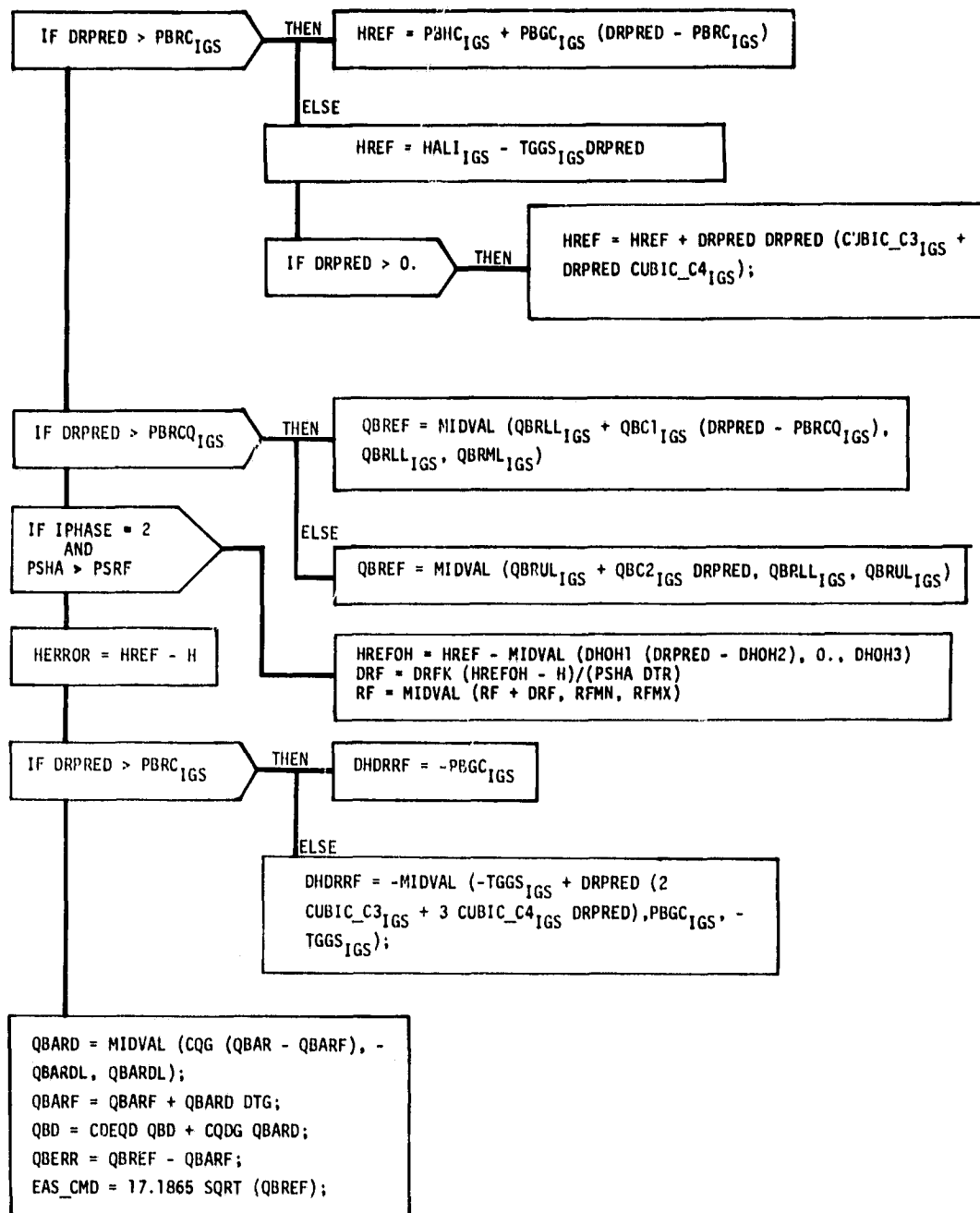


Figure 4.- Continued.

```

* TGTRAN
*
* *****
* IF IPHASE = 3
* *****
*
* *****
* THEN
* *****
*
* *****
* IF (ABS(HERROR) < (H DEL H1 - DEL H2) AND ABS(Y) <
* (H Y_RANGE1 - Y_RANGE2) AND ABS(GAMMA - GAMSGS_IGS)
* < (H GAMMA_COEF1 - GAMMA_COEF2) AND ABS(QBERR) <
* QB_ERROR2 AND H < H_REF1) OR H < H_REF2
* *****
*
* *****
* TG END = 1
* *****

```

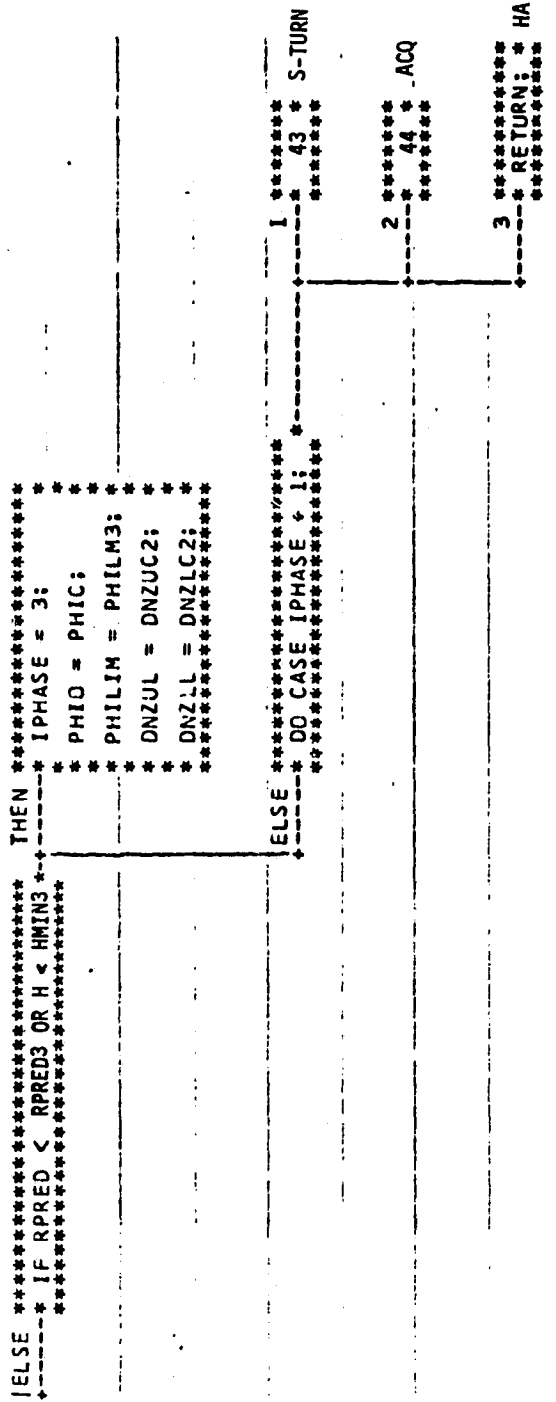
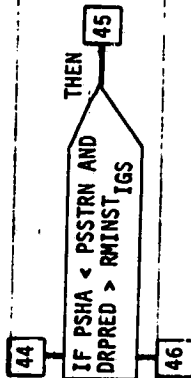


Figure 4.- Continued.

* 43 *

***** THEN *****
* IF EDW < EN + ENBIAS *
***** * IPHASE = 1; *
***** * PHILIM = PHILM1; *
***** *****



```

*****
* 45 *
*****

*****
* ES = ES1 IGS + DRPRED EDRS IGS:
*****

*****
* IF EOM > ES
*****
***** THEN
***** IPHASE = 0:
***** PHILIM = PHILIM:
***** S = -YSGN
***** SPST = S PSD:
*****
***** THEN
***** IF SPST < 0 AND PSHA < 90
*****
***** S = -S
*****

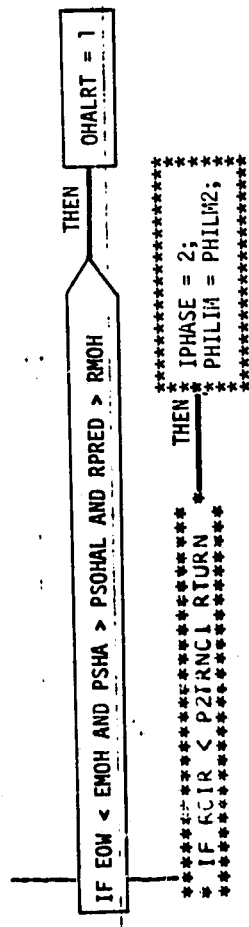
*****
* 46 *
*****

*****
* EMEP = EMEP_C1 IGS, IEL + DRPRED
*****
* EMEP_C2 IGS, IEL
*****

*****
* EMOH = EMOH1 IGS + EMOH2 IGS DRPRED
*****
* IF EOM < EMEP AND MEP = 0
***** THEN
***** MEP = 1
*****

```

Figure 4.- Continued.



ORIGINAL PAGE IS
OF POOR QUALITY

Figure 4.- Continued.

TGNZC

```

*****
* GDH = MIDVAL(GDHC - GDHS H, GDHLL, GDHUL); *
* FDRF = VH DARRF; *
* HCERR = HDREF - HDOT; *
* DNZC = CNZCG GDH (HDERR + HDREGG GDH *
* HEFROR); *
*****

```

```

IF MACH < QMACH2 THEN
  MXQBWT = MIDVAL (QBWT1 +
  QBMSL1 (MACH - QMACH1),
  QBWT2, QBWT1)

```

```

  QBL = MXQBWT WEIGHT
  ELSE
  MXQBWT = MIDVAL (QBWT2 +
  QBMSL2 (MACH - QMACH2),
  QBWT2, QBWT3)

```

```

  QBMINZ = QBL/AMAX1(COSPHI, CPMTR)

```

```

*****
* IF MACH > QB1 *
*****
  THEN QBMINZ = MIDVAL(QBMX2 + QBMS2 (MACH -
  QBMS2), QBMS2, QBMS3);
*****

```

```

  ELSE QBMINZ = MIDVAL (QBMX2 + QBMS1 (MACH - QBMS1), QBMS2, QBMS1)

```

```

IF EQLOWL < EOW AND
EOW < EQLOWU AND
PSHA > PSOHQB
  THEN
  QBMINZ = MIDVAL(QBREF2_IGS - PQBRR(RPRED2 - R2MAX)
  + (EOW - EN)/PEBRR), QBMINZ, QBMINZ)

```

```

*****
* QBZUL = -(QBG1 (QBMINZ - QBARF) - QB0) *
* QB02; *
* QBZLL = -(QBG1 (QBMINZ - QBARF) - QB0) *
* QB02; *
*****

```

```

*****
IPHASE = 3
*****
      THEN
*****
N3C = MIDVAL(DN3C, QBNZLL, QBNZUL);
*****
      ELSE
*****
EMAX = EN + EDELNZ IGS
/ DEL R_EMAX_IGS, EDELCL, EDEL2C;
*****
EMIN = EN - EDELNZ IGS;
*****
EOMNZUL = (GEUL GDH (EMAX - EOM) + HDERR)
GERDUL GDH;
*****
EOMNZLL = (GELL GDH (EMIN - EOM) + HDERR)
GERDOLL GDH;
*****
DNZCL = MIDVAL (DNZC, EOMNZLL, EOMNZUL)
DNZCL = MIDVAL (DNZCL, QBNZLL, QBNZUL)
DNZCD = MIDVAL ((DNZCL - NZC)
CQG, - DNZCDL, DNZCDL)
NZC = NZC + DNZCD DTG
*****
*****
DNZC = MIDVAL(DN3C, QBNZLL, QBNZUL);
*****

```

Figure 4.- Continued.

TGSBC:

* IF MACH > DSBCM *

THEN *****
* DSBC AT = DSBSUP *

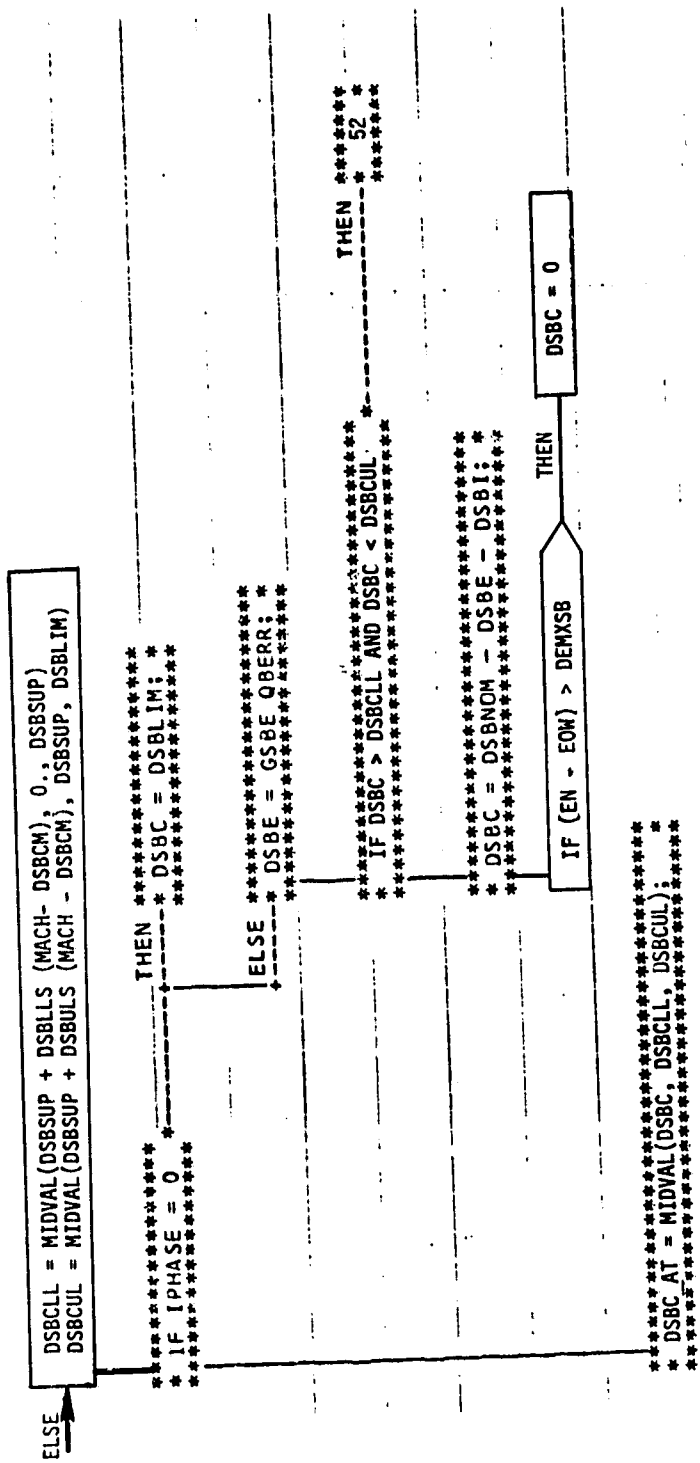


Figure 4.- Continued.

ORIGINAL PAGE IS
OF POOR QUALITY

* 52 *

* DSBI = MIDVAL(DSBI + GSBI QBERR DTG, - *
* DSBI, CSBIL); *****

TGPHIC

PHILIMIT = MIDVAL (PHILMSUP + PHILS (MACH - PHIM), PHILMSUP, PHILIM)

* DO CASE IPHASE + 1;

* PHIC = S PHILIMIT;

* PHIC = GPHI DPSAC;

RERRC = RCIR - RTURN

IF RERRC > RERRLM

THEN

IF PHILIMIT > PHILIM

THEN PHILIMIT = PHILIM

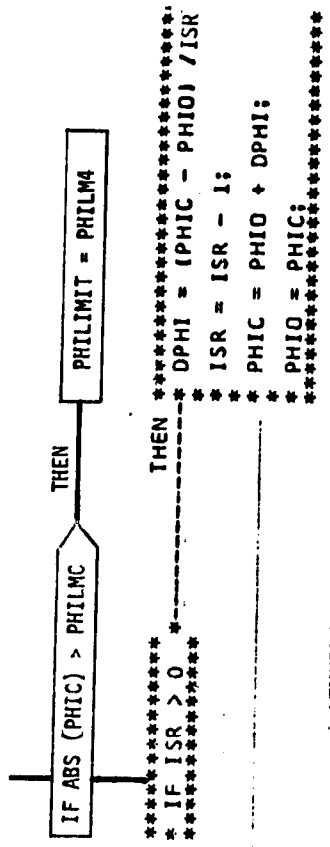
PHIC = GPHI DPSAC

RDOT = - (XCIR XDOT + YDOT YCIR)/RCIR
PHIP2C = (VH VH - RDOT RDOT) RTD/(G RTURN)
ROOTRF = -VH(R1 + 2R2 PSHA) RTD/RTURN
PHIC = YSGN AMAX1 (PHIP2C + GR RERRC +
GRDOT (RDOT - RDOTRF), 0.)

ELSE

* YERRC = MIDVAL(-GY Y, -YERRLM, YERRLM);
* PHIC = YERRC - GYDOT YDOT;

Figure 4.- Continued.



```

    *****
    * PHIC AT = MINVAL(PHIC, -PHILIMIT, PHILIMIT); *
    *****
  
```

GTP

* XCIR = XHAC - X *

* IF IPHASE = 3 AND XCIR < DR4 *

THEN *****
* RPRED = SQRT(X * Y * Y); *

ELSE

* SIGNY = SIGN(Y) *****
* YCIR = YSGN RF - Y; *
* RCIR = SQRT(XCIR * XCIR + YCIR * YCIR); *

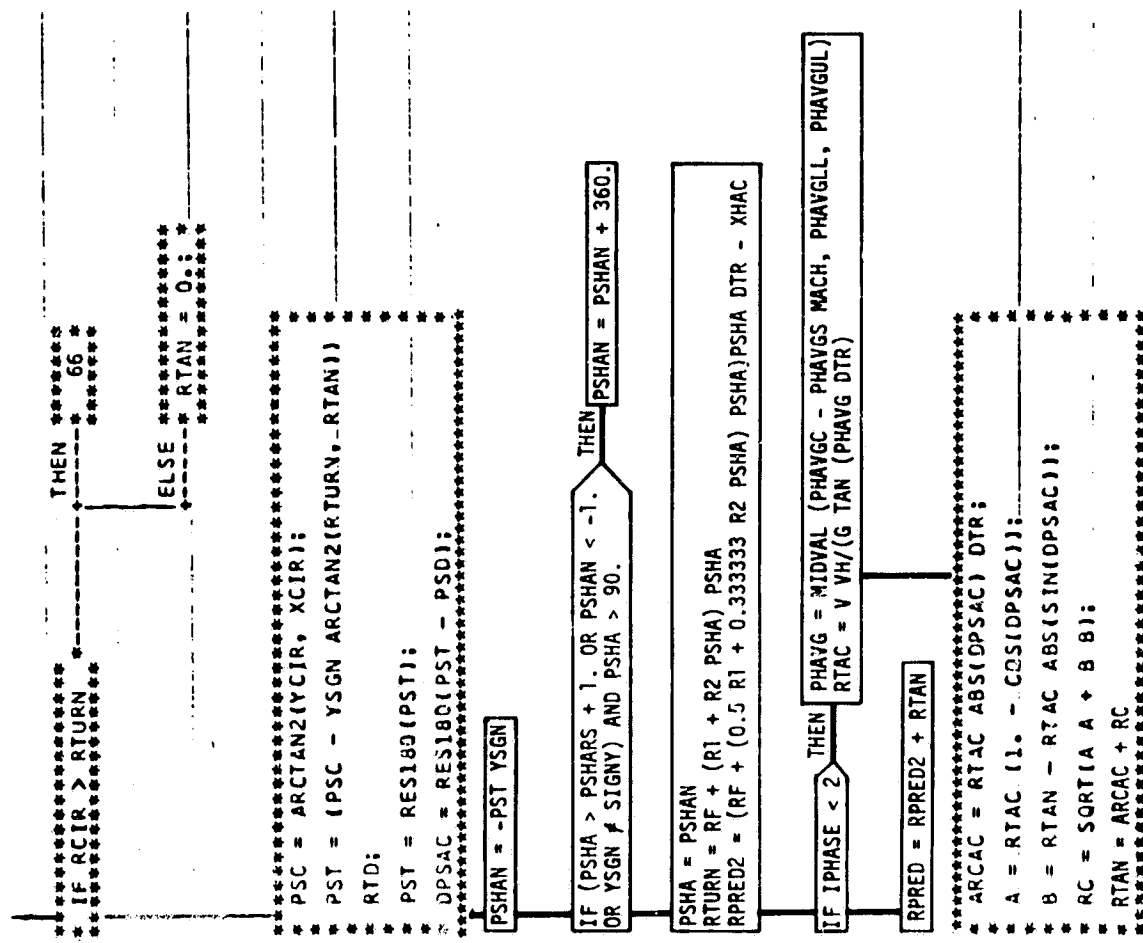


Figure 4.- Continued.

* 66 *

* RIAN = SORT URCIN RCIR - RTURN RTURNI: *

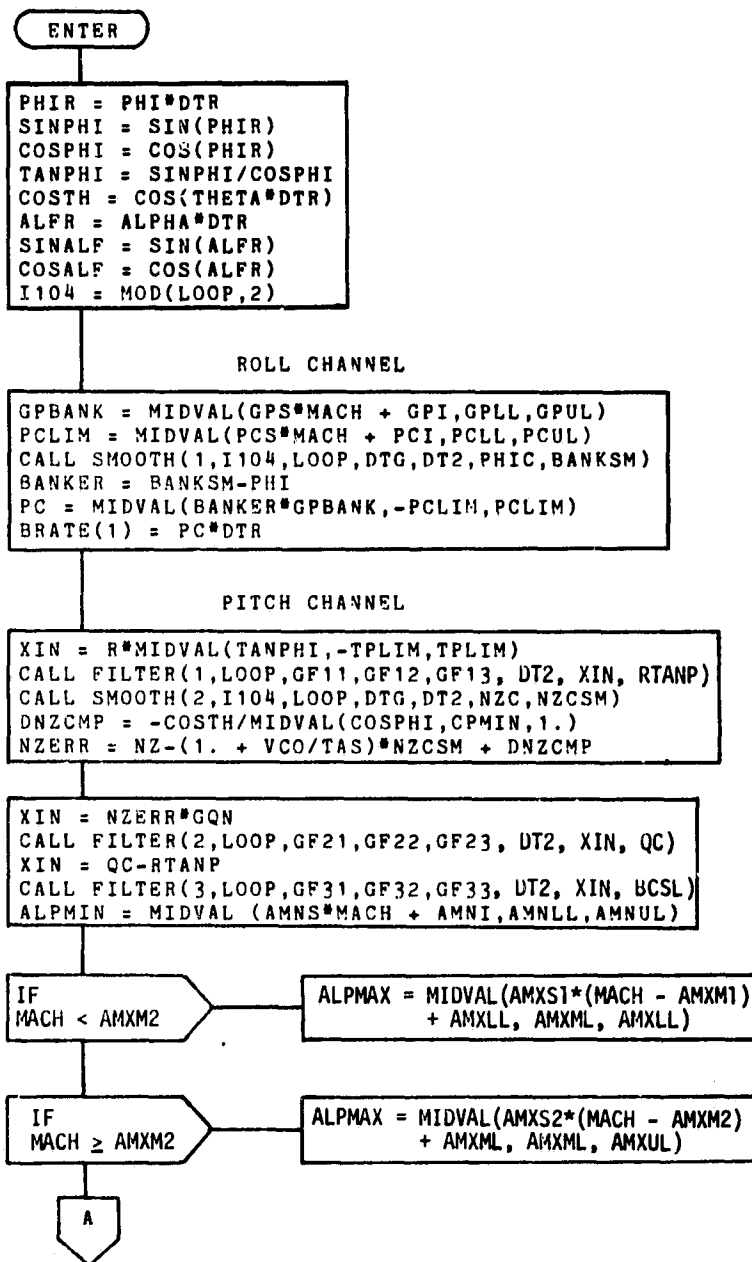


Figure 5.- TDAP detailed formulation.

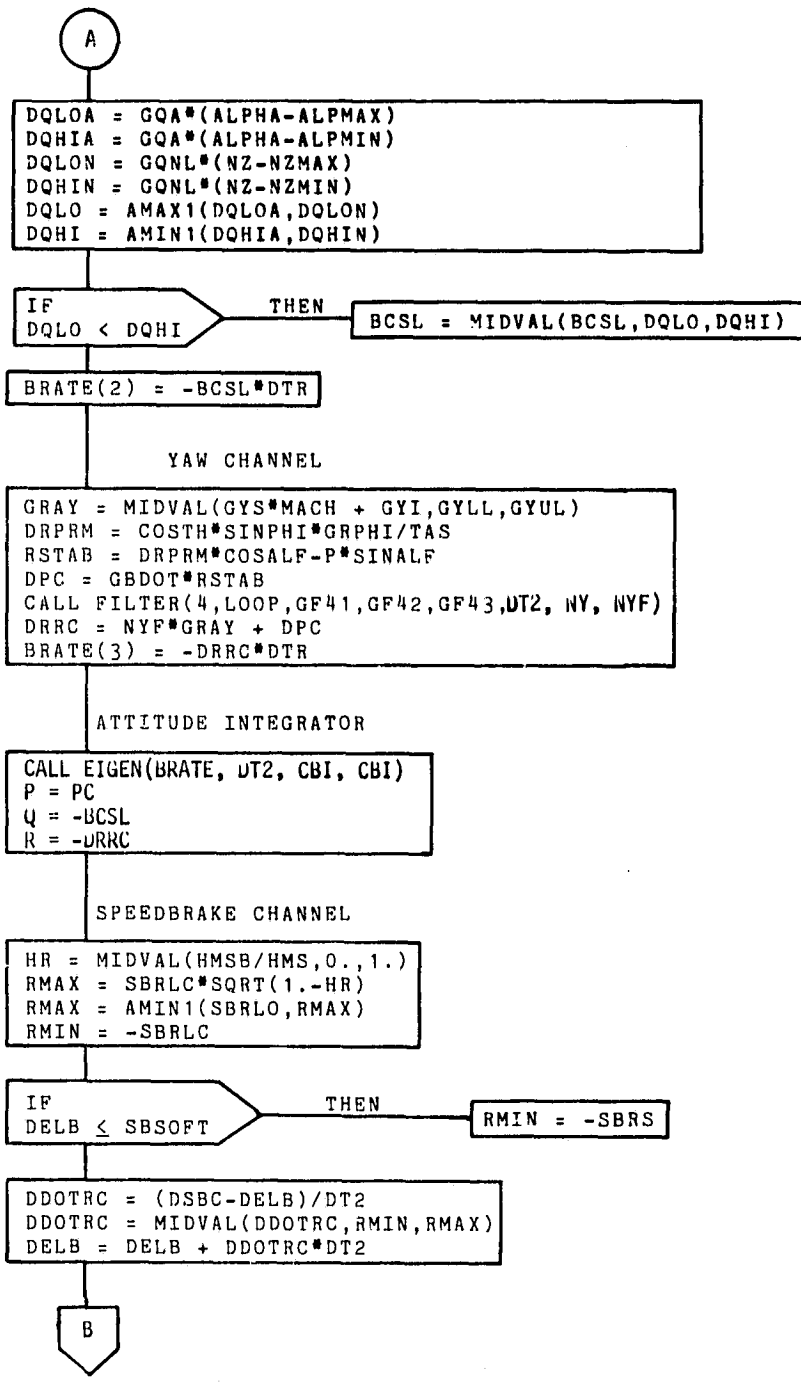


Figure 5.- Continued.

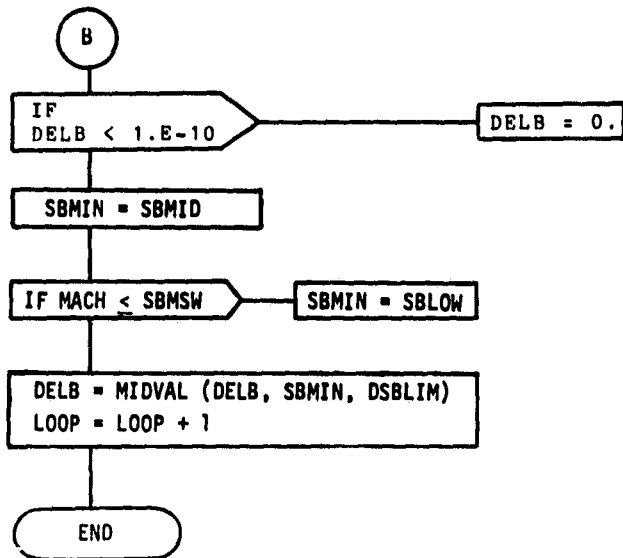


Figure 5. - Concluded.

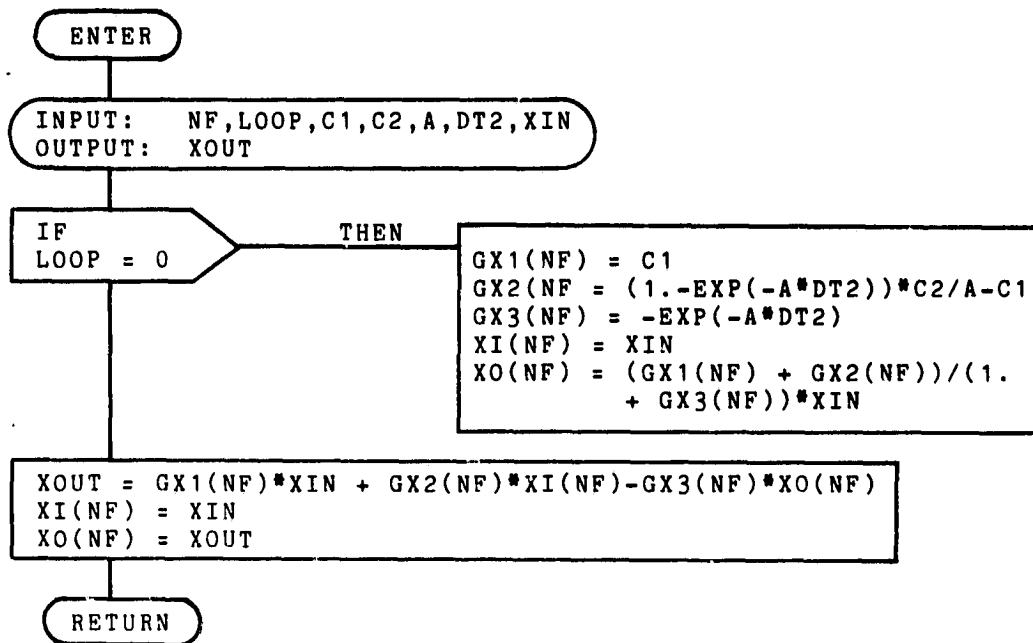


Figure 6.- FILTER detailed formulation.

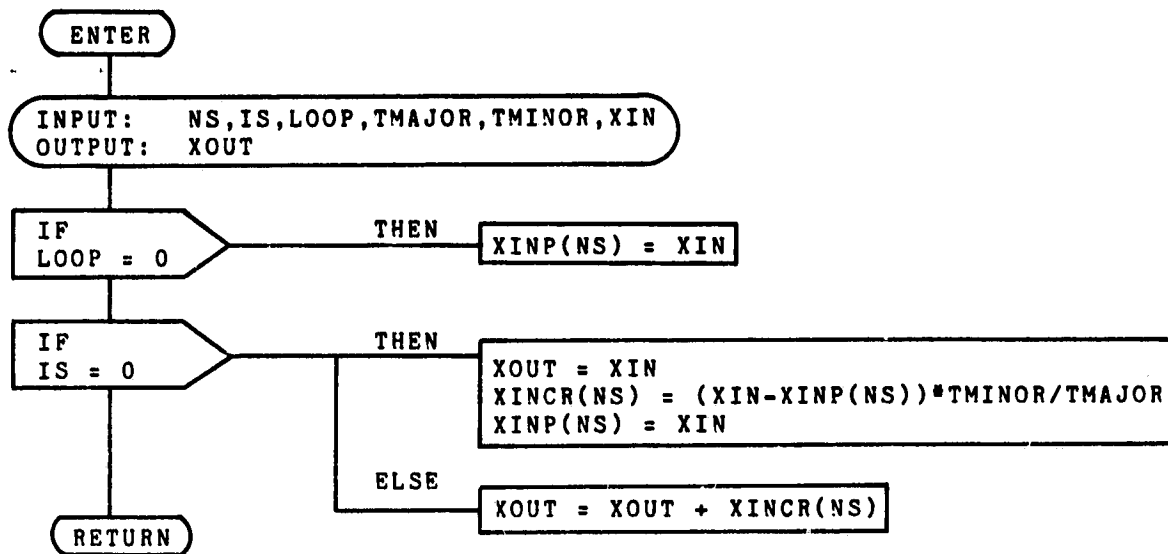


Figure 7.- SMOOTH detailed formulation.

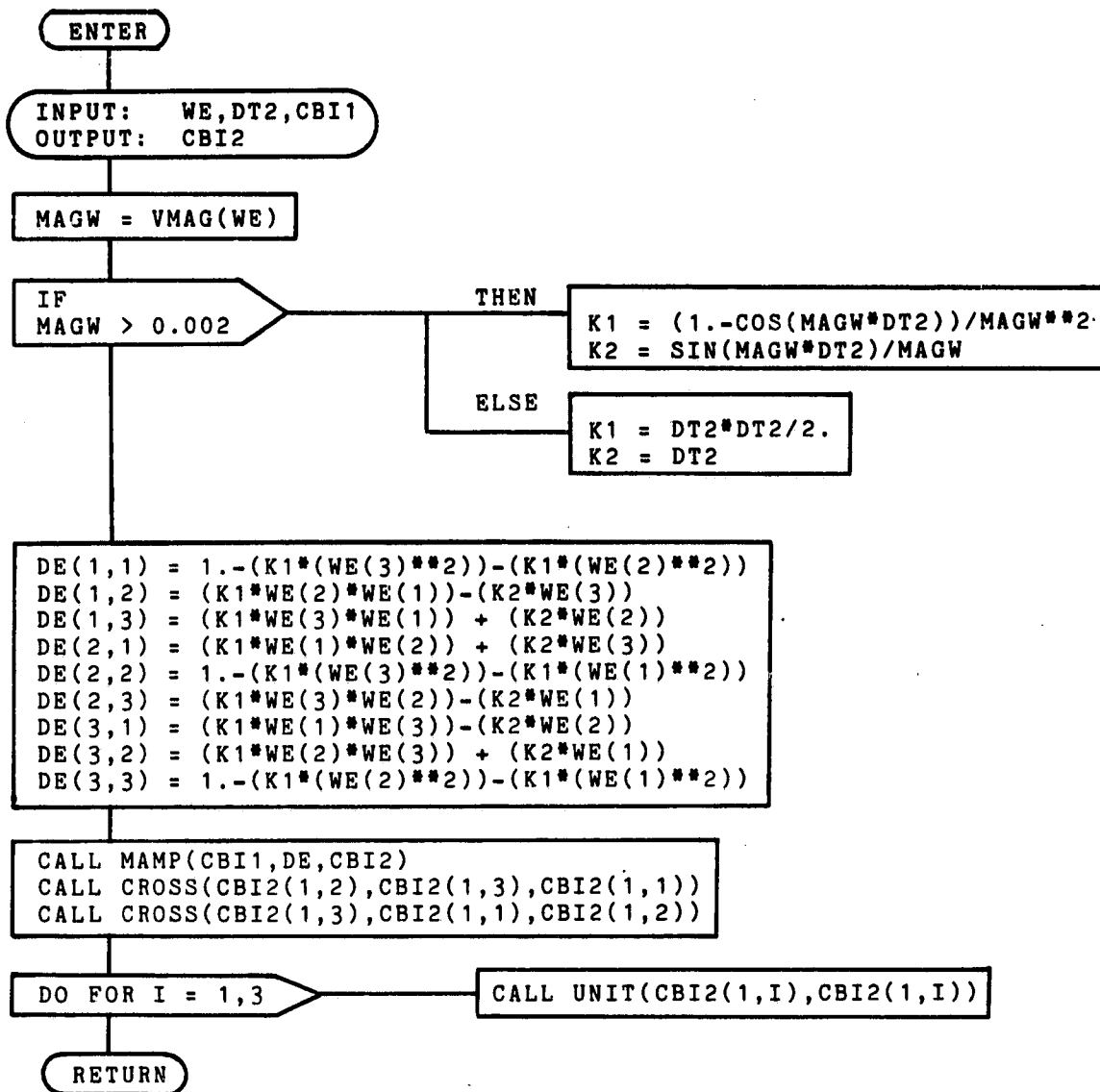


Figure 8.- EIGEN detailed formulation.

APPENDIX

GUIDANCE AND FLIGHT CONTROL INPUT PARAMETER CALCULATIONS

This appendix is included to clarify implementation requirements and contains supplementary information which describes the derivation of the TAEM guidance and flight control input parameters which are listed as GCOMP source inputs. In the flight software operating environment, these parameters are supplied by the attitude processor, air data, and navigation subsystems. Since these subsystems are not included in the MCC simulation, their function must be approximated by calculation of the necessary parameters from the error-free vehicle state. Definitions of these parameters, which are derived in the following section, are contained in the input tables of this document.

- a. State vector parameters - Since the Earth relative coordinates of the landing runway are constant, the rotation matrix from Greenwich to runway coordinates, and the position vector of the runway threshold may be calculated only on the initial pass and saved for use throughout the trajectory. This matrix is a Z,Y,Z Euler rotation sequence through the angles λ_{RW} , $-(\phi_{RW} + 90)$, and ψ_{RW} , respectively.

$$(CER) = \begin{bmatrix} -S\phi C\lambda C\psi - S\lambda S\psi & -S\phi S\lambda C\psi + C\lambda S\psi & C\psi C\phi \\ S\phi C\lambda S\psi - S\lambda C\psi & S\phi S\lambda S\psi + C\lambda C\psi & -C\phi S\psi \\ -C\phi C\lambda & -C\phi S\lambda & -S\phi \end{bmatrix}$$

$$RRE_x = \left[\frac{REQ}{\sqrt{C\phi^2 + (1-e)^2 S\phi^2}} + h_{RW} \right] C\phi C\lambda$$

$$RRE_y = \left[\frac{REQ}{\sqrt{C\phi^2 + (1-e)^2 S\phi^2}} + h_{RW} \right] C\phi S\lambda$$

$$RRE_z = \left[\frac{REQ (1-e)^2}{\sqrt{C\phi^2 + (1-e)^2 S\phi^2}} + h_{RW} \right] S\phi$$

where

$$S\phi = \sin \phi_{RW} \quad S\lambda = \sin \lambda_{RW} \quad S\psi = \sin \psi_{RW}$$

$$C\phi = \cos \phi_{RW} \quad C\lambda = \cos \lambda_{RW} \quad C\psi = \cos \psi_{RW}$$

The position and velocity vectors in Earth relative Greenwich coordinates are calculated using the time dependent inertial to Greenwich transformation matrix. If the inertial coordinate system used is based on the true date of vernal equinox and Earth polar axis, this matrix is simply a Z-axis rotation through the Greenwich hour angle, and the Earth angular rate vector is coincident with the Z-axis.

$$\begin{aligned}\overline{RE} &= (CIE) \overline{RI} \\ \overline{VE} &= (CIE) (\overline{VI} - \overline{\omega_e} \times \overline{RI})\end{aligned}$$

The position and velocity vectors in runway coordinates are:

$$\begin{aligned}\overline{RRW} &= (CER) (\overline{RE} - \overline{RRE}) \\ \overline{VRW} &= (CER) \overline{VE}\end{aligned}$$

To construct the topodetic coordinate system, the longitude and geodetic latitude of the vehicle must be obtained. This is done by first determining the quantity A by iterating three times the equation

$$A = 1 + REQ \frac{1 - (1 - e)^2}{\sqrt{RXY^2/A^2 + (1 - e)^2 RE_z^2}}$$

where

$$RXY = \sqrt{RE_x^2 + RE_y^2}$$

The starting value used for A is $1/(1 - e)^2 = 1.0067$, which is its true value for $h = 0$. Three iterations will guarantee about 7 to 8 digit accuracy.

The geodetic latitude and altitude are given by

$$\begin{aligned}\phi_D &= \text{ATAN} \left[\frac{RE_z}{(RXY/A)} \right] \\ h_D &= \frac{1 - A(1 - e)^2}{1 - (1 - e)^2} \sqrt{RXY^2/A^2 + RE_z^2}\end{aligned}$$

Longitude is easily obtained from

$$\lambda = \text{ATAN2}(\text{RE}_y, \text{RE}_x)$$

The Greenwich to topodetic transformation matrix is a Z,Y Euler rotation sequence through the angles λ , and $-(\phi_D + 90)$, respectively.

$$(\text{CET}) = \begin{bmatrix} -\sin \phi_D \cos \lambda & -\sin \phi_D \sin \lambda & \cos \phi_D \\ -\sin \lambda & \cos \lambda & 0 \\ -\cos \phi_D \cos \lambda & -\cos \phi_D \sin \lambda & -\sin \phi_D \end{bmatrix}$$

The topodetic velocity related parameters can now be obtained.

$$\overline{\text{VT}} = (\text{CET}) \overline{\text{VE}}$$

$$\text{HDOT} = -\text{VT}_z$$

$$\text{VH} = \sqrt{\text{VT}_x^2 + \text{VT}_y^2}$$

$$\overline{\text{V}} = |\text{VT}|$$

$$\text{GAMMA} = \text{ASIN}(\text{HDOT}/\text{V}) \cdot \text{RTD}$$

$$\text{H} = h_D - h_{RW}$$

$$\text{X} = \text{RRW}_x$$

$$\text{Y} = \text{RRW}_y$$

$$\text{XDOT} = \text{VRW}_x$$

$$\text{YDOT} = \text{VRW}_y$$

$$\text{PSD} = \text{ATAN2}(\text{VRW}_y, \text{VRW}_x) \cdot \text{RTD}$$

- b. Attitude parameters - The pitch and roll angles are obtained from the body to topodetic transformation matrix.

$$(\text{CBT}) = (\text{CET}) (\text{CIE}) (\text{CBI})$$

$$\text{THETA} = -\text{ASIN}(\text{CBT}_{(3,1)}) \cdot \text{RTD}$$

$$\text{PHI} = \text{ATAN2}(\text{CBT}_{(3,2)}, \text{CBT}_{(3,3)}) \cdot \text{RTD}$$

$$\text{COSPFI} = \text{COS}(\text{PHI}/\text{RTD})$$

$$\text{SECTH} = 1./\text{COS}(\text{THETA}/\text{RTD})$$

The Shuttle flight control sensors include body axis accelerometers and rate gyros that provide translational acceleration and rotational rate measurements to the digital autopilot. For this simulation, these will be assumed to be perfect systems providing true quantities. Since body rates are calculated in TDAP, they will be available as measured values for the following pass. The body axis accelerometer measurements are obtained from the true aerodynamic forces.

$$\text{NY} = \text{FAB}_y/\text{WEIGHT}$$

$$\text{NZ} = -\text{FAB}_z/\text{WEIGHT}$$

- c. Air data parameters - The Shuttle avionics includes an air data subsystem that derives air relative free stream parameters (Mach number, true airspeed, dynamic pressure, and angle of attack) from the pilot-static probe pressure measurements. Because of thermal constraints, the air data probes cannot be deployed until approximately Mach 3.5. Calibration accuracies are expected to be poor in the supersonic and transonic region due to large uncertainties in local flow effects. Air data parameters, therefore, will not be supplied to the TAEM guidance and flight control until a predetermined Mach number is given (MACHAD), which will be mission dependent. This Mach number will be approximately 0.75 for the OFT-1 mission and subsequently increased as improved calibration data becomes available. The parameters supplied by the air data system are required continuously by the TAEM guidance and flight control and, therefore, must be obtained elsewhere when acceptance of air data are undesirable. In the flight software, this is done by deriving the required quantities from the navigated state vector and attitude data, assuming calm wind conditions and standard atmosphere. The inaccuracies that result in the presence of winds and nonstandard atmosphere have an effect on guidance performance and will be included in the MCC TAEM simulation by assuming perfect navigation and air data systems, with the changeover occurring at the proper time. This changeover should be irreversible so that air data will be used continuously after its first acceptance.

For the region prior to acceptance of air data ($\text{MACH} > \text{MACHAD}$), the velocity of sound is assumed constant (1000 fps) and standard air density is approximated by

$$\rho_s = 0.00413579 e^{(-h_D/20600.)} \quad (h_D > 35000.)$$

$$\rho_s = 0.00237689 e^{(-h_D/30550.)} \quad (h_D \leq 35000.)$$

To calculate angle of attack in this region, the Earth-relative velocity vector must be obtained in body coordinates.

$$\overline{VBE} = (CBT)^T \overline{VT}$$

The required quantities for the no-air-data region can now be defined.

$$TAS = |\overline{VBE}|$$

$$ALPHA = ATAN (VBE_z/VBE_x) \cdot RTD$$

$$MACH = TAS/1000.$$

$$QBAR = 0.5\rho_s TAS^2$$

After the acceptance of air data ($MACH < MACHAD$), the corresponding true air relative quantities are to be sent to the TAEM guidance and flight control.

$$\overline{VBW} = (CBT)^T (\overline{VT} - \overline{VW})$$

$$TAS = |\overline{VBW}|$$

$$ALPHA = ATAN (VBW_z/VBW_x) \cdot RTD$$

$$MACH = TAS/VS$$

$$QBAR = 0.5\rho_A TAS^2$$

SYMBOLS

(CBI)	body to inertial coordinate transformation matrix
(CBT)	body to topodetic coordinate transformation matrix
(CER)	Greenwich to runway coordinate transformation matrix
(CET)	Greenwich to topodetic coordinate transformation matrix
(CIE)	inertial to Greenwich coordinate transformation matrix
e	ellipticity (flattening) of Earth ellipsoid model
\overline{FAB}	total aerodynamic force vector (body coordinate)
h_D	geodetic altitude of vehicle center of mass
h_{RW}	geodetic altitude of runway threshold
MACHAD	true Mach number to initiate acceptance of air data
\overline{RE}	position vector of vehicle center of mass (Greenwich coordinate)
REQ	equatorial radius of Earth ellipsoid model
\overline{RI}	position vector of vehicle center of mass (inertial coordinate)
\overline{RRE}	position vector of runway threshold (Greenwich coordinate)
\overline{RRW}	position vector of vehicle center of mass (runway coordinate)
RTD	radians to degrees conversion constant
\overline{VBE}	vehicle Earth-relative velocity vector (body coordinate)
\overline{VBW}	vehicle air-relative velocity vector (body coordinate)
\overline{VE}	velocity vector of vehicle center of mass (Greenwich coordinate)
\overline{VI}	velocity vector of vehicle center of mass (inertial coordinate)
\overline{VRW}	velocity vector of vehicle center of mass (runway coordinate)
VS	true atmospheric velocity of sound
\overline{VT}	velocity vector of vehicle center of mass (topodetic coordinate)
\overline{VW}	Earth-relative velocity vector of air mass (topodetic coordinate)
λ	longitude of vehicle center of mass

λ_{RW} longitude of runway threshold

ρ_A true atmospheric air density

ρ_S standard air density estimate

ϕ_D geodetic latitude of vehicle center of mass

ϕ_{RW} geodetic latitude of runway threshold

ψ_{RW} runway azimuth with respect to true north

$\underline{\omega_e}$ Earth angular rate vector (inertial coordinate)