80-FM-25
(Supersedes 76-FM-87, Rev. 1)

JSC-16523
(Supersedes JSC-11786)

MCC Level C
Formulation Requirements

Shuttle TAEM Guidance and Flight Control (STS-1 Baseline)

(NASA-TM-81100) MCC LEVEL C FORMULATION REQUIREMENTS. SHUTTLE TAEM GUIDANCE AND FLIGHT CONTROL, STS-1 BASELINE (NASA) 85 p
HC A06/MF A01 CSCL 22B
N80-28408
Unclas
G3/16 25340

Mission Planning and Analysis Division

June 1980

NASA
National Aeronautics and
Space Administration

Lyndon B. Johnson Space Center
Houston, Texas

80FM25

80-FM-25
(Supersedes 76-FM-87, Rev.1)

JSC-16523
Supersedes JSC-11786)

SHUTTLE PROGRAM

MCC LEVEL C FORMULATION REQUIREMENTS

SHUTTLE TAEM GUIDANCE AND FLIGHT CONTROL
STS-1 BASELINE

FM4

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

Approved:
Claude A. Graves, Jr., 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

June 1980

CONTENTS

TABLES

FIGURES

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. The logic presented is the required configuration for the first Shuttle orbital flight (STS-1). TAEM requirements for flights subsequent to STS-1 are presented in the optional TAEM targeting document (IN 80-FM-29). This processor will be used for preentry evaluation of the entrythrough landing maneuvers, and will include a simplified 3-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.

The TAEM guidance is simulated in detail. The rotational dynamics simulation is a simplified model that 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, the logic executes an S-turn to dissipate additional energy.

The previously available capability for selection of an alternate heading alinement cylinder nearer the runway (minimum entry point) has been deleted and replaced by an option to force the selection of the left-hand heading alinement cylinder. This option is implemented by adding MED input capability for the existing minimum entry point flag (MEP) and by redefining its function as follows:

MEP = 0, allow automatic selection by guidance of the nearest heading alinement cylinder (same side of runway centerline as the Shuttle position).

MEP = 1, force guidance to target for the left-hand heading alinement cylinder independent of Shuttle position.

Note that this option does not provide selection of either heading alinement cylinder. For trajectories approaching from the left of runway centerline, both options will result in a left-hand heading alinement turn. The left-hand HAC flag (MEP) will be initialized to zero in the mission constants table and changed by MED input if required.

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 below mach 0.9. 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 which is proportional to the Orbiter heading deviation from the tangent point on the nearest heading alinement cylinder. 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 combinations 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 (TGSBC)
Roll command function (TGPHIC)

Functions and routines 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) is then executed. The TAEM phase transition and MEP function (TGTRAN) is then called. The guidance commands are then generated by calling the normal acceleration command (TGNZC), the speedbrake command (TGSBC), 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:

IRESET = 0

ISR = RFTC/DTG

DNZCF = 0.0

DSBI = 0.0

QBLL = MXQBWT WEIGHT

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 X runway coordinate of the heading alinement cylinder center, XHAC, and the prefinal initiation range value (SHPLYK) as a function of surface wind conditions and weight.

The operations performed are:

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

XALI = XA(IGI) + HALI(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 thousand pounds.

Capability to select the minimum entry point (XALI) has been deleted, so XHAC is always set equal to XFTC.

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 thereshold. 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 alinement cylinder. The acquisition turn is defined as the turn that will aline the Orbiter heading tangent to the heading alinement cylinder. 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 cylinder. The distance in the heading alinement turn is computed from the turn angle in the heading alinement phase and the alinement cylinder radius. The distance d_T is a fixed distance from runway threshold of the alinement cylinder.

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), 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 Δ_ψ is the difference between the Orbiter heading, ψ, and the heading, ψ_T, to the heading alinement circle tangency point. The vector from the Orbiter to the center of the heading alinement cylinder is computed as

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

$$Y_{CIR} = YSGN\ R_{TURN} - Y$$

The distance to the center of the heading alinement circle is

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

and the straight line distance to the tangency point is

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

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

The heading, ψ_C, to the center of the heading alinement circle is given by

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

The heading to the tangency point is then

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

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 Δ_ψ, 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

$$d_{AC} = R_{AC} \left|\Delta\psi\right|$$

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

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

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

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

GROUNDTRACK PREDICTOR SUBPROGRAM

If IPHASE = 3,

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

and then exit GTP. Otherwise, if IPHASE < 2, YSGN = SIGH (Y) (YSGN = $\pm$ 1). The left hand HAC select flag is then tested, if MEP = 1, YSGN = -1.

Now execute equation set 1:

1.1 XCIR = XHAC -X

1.2 YCIR + YSGN RTURN - Y

1.3 RCIR = SQRT ($\text{XCIR}^2 + \text{YCIR}^2$)

If IPHASE = 2, execute equation set 2:

2.1 PSHA = ARCTAN2 (XCIR, YSGN YCIR) RTD

2.2 If PSHA < 0., PSHA = PSHA + 360

2.3 RPRED = RCIR PSHA DTR - XHAC

and then exit GTP. Otherwise (IPHASE = 0 or 1), compute the straight line distance to the HAC tangency point by

IF: RCIR > RTURN

THEN: RTAN = SQRT ($\text{RCIR}^2 - \text{RTURN}^2$)

ELSE: RTAN = 0

The heading error (from a heading tangent to the HAC) is then computed by equation set 3:

3.1 PSC = ARCTAN2 (YCIR, XCIR)

3.2 PST = (PSC - YSGN ARCTAN2(RTURN, RTAN)) RTD

3.3 PST = RES180 (PST)

3.4 DPSAC = RES180 (PST-PSD)

The acquisition turn radius and arc length are then computed in equation set 4:

4.1 PHAVG = PHAVGC - PHAVGS MACH

4.2 PHAVG = MIDVAL (PHAVG, PHAVGLL, PHAVGUL)

4.3 RTAC = VH V/(G TAN (PHAVG DTR))

4.4 ARCAC = RTAC ABS (DPSAC) DTR

The range from the end of the acquisition turn to the HAC tangency point, RC, is next computed by equation set 5:

5.1 A = RTAC (1.-COS(DPSAC DTR))

5.2 B = RTAN - RTAC ABS (SIN(DPSAC DTR))

5.3 RC = SQRT ($A^2 + B^2$)

The turn angle around the HAC is then defined by

PSHA = ABS (PST)

If (XCIR < 0 and ABS(Y) < 2. RTURN) or
(YSGN SIGN(Y) < 0.) , PSHA = 360 - PSHA

The acquisition or S-turn predicted groundtrack is then computed by equation set 6:

6.1 ARCHA = RTURN PSHA DTR

6.2 RPRED = ARCAC + RC + ARCHA - XHAC

GTP is then exited.

4.1.5 TAEM Reference and Dynamic Pressure 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 which 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 altitude reference (H ref) is a cubic curve which is tangent to the autoland steep glide slope at RPRED = -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.

TGCOMP SUBPROGRAM

Upon entering TGCOMP, the following computations are made:

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

1.1 DRPRED = RPRED + XALI

1.2 EOW = H + V V/(2G)

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

2.1 EN = EN_C1(IGS, IEL) + (DRPRED - RN1(IGS))EN_C2(IGS, IEL)

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

4.1 HREF = PBHC(IGS) + PBGC(IGS) (DRPRED - PBRC(IGS))

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

4.2 HREF = HALI(IGS) - TGGS(IGS) DRPRED

4.3 If (DRPRED > 0.) HREF = HREF + DRPRED DRPRED (CUBIC_C3(IGS) + DRPRED CUBIC_C4(IGS))

The dynamic pressure reference (QBREF), 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, QBRLL(IGS), QBRUL(IGS))

5.2 If (DRPRED > PBRCQ(IGS)), QBREF = MIDVAL (QBRLL(IGS) + QBC1(IGS) (DRPRED - PBRCQ(IGS)), QBRLL(IGS), QBRML(IGS))

5.3 HERROR = HREF - H

5.4 If (DRPRED > PBRC(IGS)), DHDRRF = - PBGC(IGS)

5.5 If (DRPRED ≤ PBRC(IGS)), DHDRRF = - MIDVAL (-TGGS(IGS) + DRPRED (2 CUBIC_C3(IGS) + 3 CUBIC_C4(IGS) DRPRED), PBGC(IGS), -TGGS(IGS))

5.6 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  QBD = CDEQD*QBD + CQDG*QBARD

6.4  QBERR = QBREF - QBARF

6.5  EAS_CMD = 17.1865*SQRT(QBREF)
```

4.1.6 Phase Transition and MEP 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 an S-turn (IPHASE = 0) situation. The S-turn initiation and termination is based on an energy error.

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). The direction in which the S-turn is made depends on which side of the runway centerline the Orbiter is located and on the Orbiter heading. The logic for selection of the turn direction is

YSGN < 0, $-135 < \psi < 0$	$S = -1$
YSGN < 0, $-135 > \psi$ or $\psi > 0$	$S = +1$
YSGN > 0, $135 > \psi > 0$	$S = +1$
YSGN > 0, $135 < \psi$ or $\psi < 0$	$S = -1$

$S = -1$ left bank

$S = +1$ right bank

Termination of the S-turn occurs when the current energy falls below the reference energy versus range line (EN).

Transition to the heading alinement phase from the acquisition phase is tested whenever RCIR < P2TRNC1 RTURN where RCIR is the Orbiter's radial distance from the HAC enter and RTURN is the HAC radius. If this test is passed, transition to the heading alinement phase will occur if either of the following tests is passed:

a. RT < RTBIAS

b. RCIR < P2TRNC2 RTURN

where

RT = (XDOT XCIR + YDOT YCIR)/V

Transition to the prefinal phase (IPHASE = 3) occurs whenever the predicted range is less than the prefinal initiation range value (SHPLYK).

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_ERROR and (|Y| < Y_ERROR) and

(|GAMMA - GAMSGS(IGS)| < GAMMA_ERROR) and

(|QBERR| < QB_ERROR1) and

(H > H_REF1)

or

(|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, 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 < SHPLYK). 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 RPRED < SHPLYK 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 reference energy, then execute equation set 2:

2.1 IPHASE = 1

2.2 PHILIM = PHILM1

For IPHASE = 1 (acquisition)

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 - RMINST(IGS)) EDRS(IGS)

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

$\mathrm{EOW} > \mathrm{ES}$ and $\mathrm{DRPRED} > \mathrm{RMINST(IGS)}$

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

3.2.1 $\mathrm{IPHASE} = 0$

3.2.2 $\mathrm{PHILIM} = \mathrm{PHILMO}$

3.2.3 $\mathrm{S} = \mathrm{YSGN}$

3.2.4 $\mathrm{SPSI} = \mathrm{S}\ \mathrm{PSD}$

3.2.5 If $\mathrm{SPSI} < 0$ or $\mathrm{SPSI} > 135$, then $\mathrm{S} = -\mathrm{S}$

3.2.6 If $\mathrm{MEP} = 1$, then $\mathrm{S} = 1$.

Next, the energy over weight value used in the MEP test is calculated as shown in equation 3.3:

3.3 $\mathrm{EMEP} = \mathrm{EMEP\_C1(IGS,\ IEL)} + (\mathrm{DRPRED} - \mathrm{RN1(IGS)})\mathrm{EMEP\_C2(IGS,\ IEL)}$

The test for transition to the heading alinement phase is made. If $\mathrm{RCIR} < \mathrm{P2TRNC1}\ \mathrm{RTURN}$, the parameter RT is calculated as shown in equation 3.4:

3.4 $\mathrm{RT} = (\mathrm{XDOT}\ \mathrm{XCIR} + \mathrm{YDOT}\ \mathrm{YCIR})/\mathrm{V}$

and a test is made for passing either of the following conditions:

3.4A $\mathrm{RT} < \mathrm{RTBIAS}$

3.4B $\mathrm{RCIR} < \mathrm{P2TRNC2}\ \mathrm{RTURN}$

If either 3.4A or 3.4B is satisfied, equation set 3.5 is executed.

3.5.1 $\mathrm{IPHASE} = 2$

3.5.2 $\mathrm{PHILIM} = \mathrm{PHILM2}$

For $\mathrm{IPHASE} = 2$ (heading alinement):

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 dynamic pressure and TAEM phase considerations. During an S-turn, a correction term, GCONT, to the basic turn coordination term computed in the FCS is added to NZC. A filter limiting the rate of change of the unlimited command (DNZC) is implemented to smooth the command for all TAEM phases except prefinal.

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)

1.5 QBMNNZ = QBLL/AMAX1 (COSPHI, CPMIN)

1.6 QBMXNZ = QBMX1

1.7 IF MACH > QBM1 then

QBMXNZ = MIDVAL (QBMX2

+ QBMXS(MACH - QBM2), QBMX2, QBMX3)

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 filtered and limited normal acceleration command (NZC) is computed using equation set 3:

3.1 EMAX = EN + EDELNZ(IGS) MIDVAL ((DRPRED

- RN1 (IGS))/DEL_R_EMAX(IGS), EDELC1, EDELC2)

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 DNZCD = (DNZC - DNZCF) CQG

3.6 DNZCF = MIDVAL (DNZCF + DNZCD DTG, EOWNZLL, EOWNZUL)

3.7 NZC = MIDVAL (DNZCF, QBNZLL, QBNZUL)

If IPHASE ≠ 3, a test for the S-turn phase is made (IPHASE = 0); and if true, the turn coordination correction term (GCONT) is computed and added to NZC using equation set 4:

4.1 GCONT = (1. - QBARF/QBMXNZ) TAS/((TAS + VCO) SECTH COSPHI)

4.2 NZC = NZC - GCONT

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 (TGSBC)

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 computed speedbrake command equals or exceeds either the upper or lower speedbrake command limit.

If MACH is greater than DSBCM, the speedbrake command is set to a constant value of DSBSUP and TGSBC is exited. Otherwise, the TAEM phase is tested and if IPHASE = 0, equation set 1 is executed.

1.1 DSBC = DSBLIM

Otherwise (IPHASE ≠ 0), equation set 2 is executed.

2.1 DSBE = GSBE QBERR

2.2 If DSBC < DSBLIM and DSBC > 0.0, then

DSBI = MIDVAL (DSBI + GSBI QBERR DTG,-DSBIL, DSBIL)

2.3 DSBC = DSBNOM - DSBE - DSBI

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

3.1 DSBC_AT = MIDVAL (DSBC, SBMIN, DSBLIM)

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 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 cylinder. 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 fixed for supersonic flight and is equal to the phase dependent roll command limit, PHILIM, for subsonic flight.

Upon entering TGPHIC, a test on MACH is made for setting the roll command limit, PHILIMIT. For MACH > PHIM, PHILIMIT = PHILMSUP, and for MACH ≤ PHIM, PHILIMIT = 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 PHIC = S PHILIMIT

2. IPHASE = 1 (acquisition):

    2.1 PHIC = GPHI DPSAC

3. IPHASE = 2 (heading alinement):

    3.1 RDOT = (XCIR XDOT + YCIR YDOT)/RCIR

```
        3.2  RERRC = MIDVAL (GR(RCIR - RTURN), -RERRLM, RERRLM)

        3.3  PHIC = YSGN (PHIP2C + RERRC + GRDOT RDOT)

    4.  IPHASE = 3 (prefinal):

        4.1  YERRC = MIDVAL (-GY Y, -YERRLM, YERRLM)

        4.2  PHIC = YERRC - GYDOT YDOT

        4.3  If ISR > 0 then

             4.3.1  DPHI = (PHIC - PHIO)/ISR

             4.3.2  ISR = ISR - 1

             4.3.3  PHIC = PHIO + DPHI

             4.3.4  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  PHIC_AT = MIDVAL (PHIC, -PHILIMIT, PHILIMIT)
```

4.1.10 Mid Value (MIDVAL)

The MIDVAL function applies upper and lower limits to a variable.

a. Detailed requirements.- The call to function MIDVAL by the statement CALL MIDVAL (X, XLL, XUL) commands the limiting operations:

```
IF (X > XUL), THEN X = XUL

IF (X < XLL), THEN X = XLL
```

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

```
ANG = RES180 (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 (.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 functions 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 postive 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 (.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 3-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 6-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 6-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 which 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 which 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 which 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 subfunctions 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/DT2 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-0001C, Dec. 15, 1978.
Part C - Flight Control - Entry, SD 76-SH-0007, Nov. 26, 1976.

TABLE I.- TAEM GUIDANCE INPUTS

Description	Type	Source	Unit	Internal name
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 WRT RW centerline	F	GCOMP	deg	PSD
MACH number	F	GCOMP	--	MACH
Dynamic pressure	F	GCOMP	psf	QBAR
Cosine of body roll Euler angle	F	GCOMP	--	COSPHI
Secant of body pitch Euler angle	F	GCOMP	--	SECTH
Orbiter mass	F	AERO	lbm	WEIGHT
True airspeed	F	GCOMP	fps	TAS
y-component of velocity in runway coordinates	F	GCOMP	fps	YDOT

[a] Guidance and flight control input parameter calculations (appendix).

TABLE I.- Concluded

Description	Type	Source	Unit	Internal name
Topodetic Earth relative flight-path angle	F	GCOMP	deg	GAMMA
Flag indicating glide slope desired	I	MED[b]	--	GI_CHANGE
Left-hand HAC select flag	I	MED[c]	--	MEP

[b] MED input based on surface wind conditions (initialized in mission constants table to zero).

[c] MED input equal to 1 to force selection of the left-hand heading alinement cylinder (initialized in mission constants table to zero).

TABLE II.- TAEM GUIDANCE OUTPUTS

Description	Type	Unit	Internal name	Destination
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	DELRNG	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

[a] TAEM digital autopilot.
[b] Displays.
[c] This flag will terminate the MCC simulation.

TABLE III.- TAEM GUIDANCE CONSTANTS

Symbol	Description	Value	Unit
CDEQD	Gain in calculation of QBD	.68113130	--
CPMIN	Minimum value CosPHI	.707	--
CQDG	Gain used in calculation of QBD	.31886860	--
CQG	Gain used in calculation of DNZCD and QBARD	.5583958	--
CUBIC_C3(1)[a]	Quadratic term coefficient in HREF CURVE FIT	-.3641168E-6	ft/ft^2
CUBIC_C3(2)	Quadratic term coefficient in HREF CURVE FIT	-.3641168E-6	ft/ft^2
CUBIC_C4(1)	Cubic term coefficient in HREF CURVE FIT	-.9481026E-13	ft/ft^3
CUBIC_C4(2)	Cubic term coefficient in HREF CURVE FIT	-.9481026E-13	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	54000.	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	-0.75	g
DNZUC1	Phase 0, 1, and 2 NZC upper limit	0.5	g

---

[a] Array dimension 2.

TABLE III.- Continued

Symbol	Description	Value	Unit
DNZUC2	Phase 3 NZC upper limit	1.5	g
DSBCM	Mach at which speedbrake modulation begins	0.9	--
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
DSHPLY	Delta range value in SHPLYK	4000.	ft
DTG	TAEM guidance cycle interval	0.96	sec
DTR	Degrees-to-radian conversion factor	0.0174533	rad/deg
EDELC1	Constant used in determination of EMAX	1.	ND
EDELC2	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	4000.	ft
EDRS(1)	Slope for S-turn energy line	.6089492	ft^2/ft
EDRS(2)	Slope for S-turn energy line	.6089492	ft^2/ft
EMEP_C1(1,1)[b]	Y-intercept of MEP energy line	-371.3797	ft
EMEP_C1(2,1)	Y-intercept of MEP energy line	-371.3797	ft
EMEP_C1(1,2)	Y-intercept of MEP energy line	14604.87	ft

---

[b] 2 x 2 array given column wise.

TABLE III.- Continued

Symbol	Description	Value	Unit
EMEP_C1(2,2)	Y-intercept of MEP energy line	14604.87	ft
EMEP_C2(1,1)	Slope of MEP energy line	.4168274	ft^2/ft
EMEP_C2(2,1)	Slope of MEP energy line	.4168274	ft^2/ft
EMEP_C2(1,2)	Slope of MEP energy line	.2821187	ft^2/ft
EMEP_C2(2,2)	Y-intercept of nominal energy line	.2821187	ft^2/ft
EN_C1(1,1)	Y-intercept of nominal energy line	-254.2395	ft^2/ft
EN_C1(2,1)	Y-intercept of nominal energy line	-254.2395	ft^2/ft
EN_C1(1,2)	Y-intercept of nominal energy line	17645.34	ft^2/ft
EN_C1(2,2)	Y-intercept of nominal energy line	17645.34	ft^2/ft
EN_C2(1,1)	Slope of nominal energy line	.5500275	ft^2/ft
EN_C2(2,1)	Slope of nominal energy line	.5500275	ft^2/ft
EN_C2(1,2)	Slope of nominal energy line	.3890240	ft^2/ft
EN_C2(2,2)	Slope of nominal energy line	.3890240	ft^2/ft
ES1(1)	Y-intercept of S-turn energy line	90000.	ft
ES1(2)	Y-intercept of S-turn energy line	90000.	ft
EOW_SPT(1)	Range at which to change slope and intercept on the MEP and nominal energy line	117718.	ft
EOW_SPT(2)	Range at which to change slope and intercept on the MEP and nominal energy line	117718.	ft
G	Earth gravitational constant	32.174	ft/s^2
GAMMA_COEF 1	Flightpath error coefficient	0.0007	deg/ft
GAMMA_COEF2	Flightpath error coefficient	3.0	deg

TABLE III.- Continued

Symbol	Description	Value	Unit
GAMMA_ERROR	Flightpath error band	4.0	deg
GAMSGS(1)	A/L steep glideslope angle	-20.	deg
GAMSGS(2)	A/L steep glideslope angle	-20.	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	---
GEHDLL	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	0.02	deg/ft
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	0.05	deg/ft
GYDOT	Gain on YDOT in computing PFL roll angle command	0.6	deg/fps
H_ERROR	Altitude error bound	1000.	ft

TABLE III.- Continued

Symbol	Description	Value	Unit
H_REF1	Altitude reference	2000.	ft
H_REF2	Altitude reference	2000.	ft
HALI(1)	Altitude at A/L steep glideslope at MEP	10018.	ft
HALI(2)	Altitude at A/L steep glideslope at MEP	10018.	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	12018.	ft
MACHAD[a]	Mach no. to use air data	2.5	---
MXQBWT	Max L/D dynamic pressure for nominal weight	0.7368421E-3	psf/lb
PBGC(1)	Linear coefficient of range for HREF and lower limit for DHDRRF	.1125953	---
PBGC(2)	Linear coefficient of range for HREF and lower limit for DHDRRF	.1125953	---
PBHC(1)	Altitude reference for DRPRED = PBRC	84821.29	ft
PBHC(2)	Altitude reference for DRPRED = PBRC	84821.29	ft
PBRC(1)	Maximum range for cubic altitude reference	308109.5	ft
PBRC(2)	Maximum range for cubic altitude reference	308109.5	ft
PBRCQ(1)	Range breakpoint for QBREF	122000.	ft
PBRCQ(2)	Range breakpoint for QBREF	122000.	ft

---

[a] Used to GCOMP calculations (appendix).

TABLE III.- Continued

Symbol	Description	Value	Unit
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
PHILM0	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	1.	--
PHIP2C	Constant for computing phase 2 roll command	30.	deg
P2TRNC1	Constant used in Phase 2 transition logic	1.1	--
P2TRNC2	Constant used in Phase 2 transition logic	1.01	--
QB_ERROR1	Dynamic pressure error bound	-1.	psf
QB_ERROR2	Dynamic pressure error bound	24.	psf
QBARDL	Limit on QBARD	20.	psf/s
QBC1(1)	Slope of QBREF for DRPRED > PBRCQ	.5869565E-3	psf/ft
QBC1(2)	Slope of QBREF for DRPRED > PBRCQ	.5869565E-3	psf/ft
QBC2(1)	Slope of QBREF for DRPRED < PBRCQ	-.1031695E-2	psf/ft

TABLE III.- Continued

Symbol	Description	Value	Unit
QBC2(2)	Slope of QBREF for DRPRED < PBRCQ	-.1031695E-2	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
QBMXS	Slope of QBMXNZ with MACH > QBM2	20.	psf
QBMX1	Constant for computing QBMXNZ	340.	psf
QBMX2	Constant for computing QBMXNZ	220.	psf
QBMX3	Constant for computing QBMXNZ	250.	psf
QBM1	Mach breakpoint for computing QBMXNZ	1.0	--
QBM2	Mach breakpoint for computing QBMXNZ	1.0	--
QBRLL(1)	QBREF lower limit	153.	psf
QBRLL(2)	QBREF lower limit	153.	psf
QBRML(1)	QBREF middle limit	180.	psf
QBRML(2)	QBREF middle limit	180.	psf
QBRUL(1)	QBREF upper limit	265.43	psf
QBRUL(2)	QBREF upper limit	265.43	psf
RERRLM	Limit of RERRC	50.	deg
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	122204.6	ft

TABLE III.- Concluded

Symbol	Description	Value	Unit
RN1(1)	Range at which the gain on EDELNZ goes to ZERO	6542.886	ft
RN1(2)	Range at which the gain on EDELNZ goes to ZERO	6542.886	ft
RTBIAS	Maximum value for RT for HA initiation	3000.	ft
RTD	Radians-to-degrees conversion factor	57.29578	deg/rad
RTURN	HAC radius	20000.	ft
SBMIN	Minimum speedbrake command	5.	deg
TGGS(1)	Tan of steep glideslope for autoland	-.363978	--
TGGS(2)	Tan of steep glideslope for autoland	-.363978	--
VCO	Constant used in turn compensation	1.0E20	fps
WT_GS1	Maximum Shuttle Orbiter weight	250000.	lb
XA(1)	Steep glideslope intercept	-6500.	ft
XA(2)	Steep glideslope intercept	-5500.	ft
YERRLM	Limit on YERRC	120.	deg
Y_ERROR	Crossrange error bound	1000.	ft
Y_RANGE1	Crossrange coefficient	0.18	--
Y_RANGE2	Crossrange coefficient	800.	ft

TABLE IV.- TDAP INPUTS

Name	Description	Source	Unit
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
DELB[a]	Speedbrake deflection	AERO	deg
HMSB	Speedbrake hinge moment	AERO	in-lb
CBI[a]	Body to inertial attitude matrix	EDAP[c]	--
P[a]	Body roll rate	EDAP	deg/sec
R[a]	Body yaw rate	EDAP	deg/sec

[a] Required only for initialization.
[b] Guidance and flight control input parameter calculations.
[c] Entry guidance phase autopilot.

TABLE V.- TDAP OUTPUTS

Name	Description	Destination	Unit
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
DDOTRC	Speedbrake rate	MON	deg/sec

[a] Not presently used except for possible checkout monitoring.

TABLE VI.- TDAP CONSTANTS

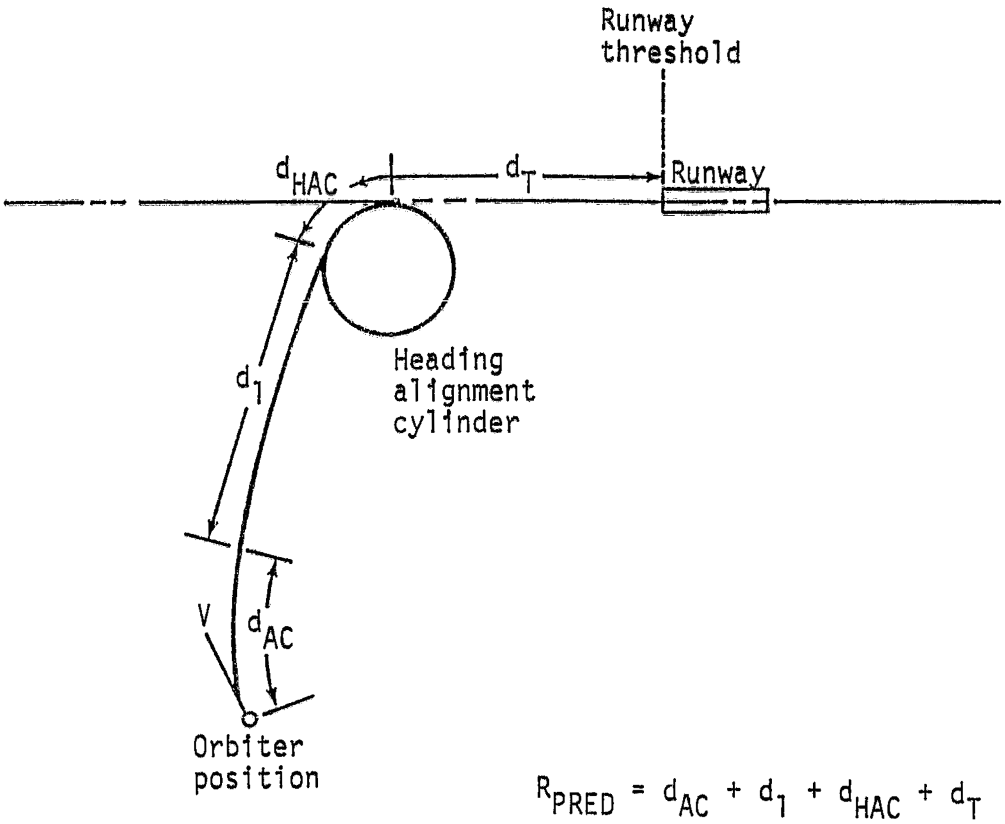
Name	Description	Value	Unit
AMNI	Y-intercept of minimum alpha profile	-16.	deg
AMNLL	Minimum alpha lower limit	-100.	deg
AMNS	Slope of minimum alpha profile	20.	deg
AMNUL	Minimum alpha upper limit	-100.	deg
AMXLL	Maximum alpha lower limit	100.	deg
AMXML	Maximum alpha middle limit	100.	deg
AMXM1	Maximum alpha profile first breakpoint	0.6	mach
AMXM2	Maximum alpha profile second breakpoint	0.	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	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.	--
GF12	Filter 1 second coefficient	1.	--
GF13	Filter 1 third coefficient	1.	--
GF21	Filter 2 first coefficient	0.2	--

TABLE VI.- Continued

Name	Description	Value	Unit
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	23.0	(deg/s)/g
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

TABLE VI.- Concluded

Name	Description	Value	Unit
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
SBMIN	Minimum speedbrake deflection	5.	deg
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



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

(Computed groundtrack range)

Figure 1.- TAEM guidance groundtrack predictor geometry.

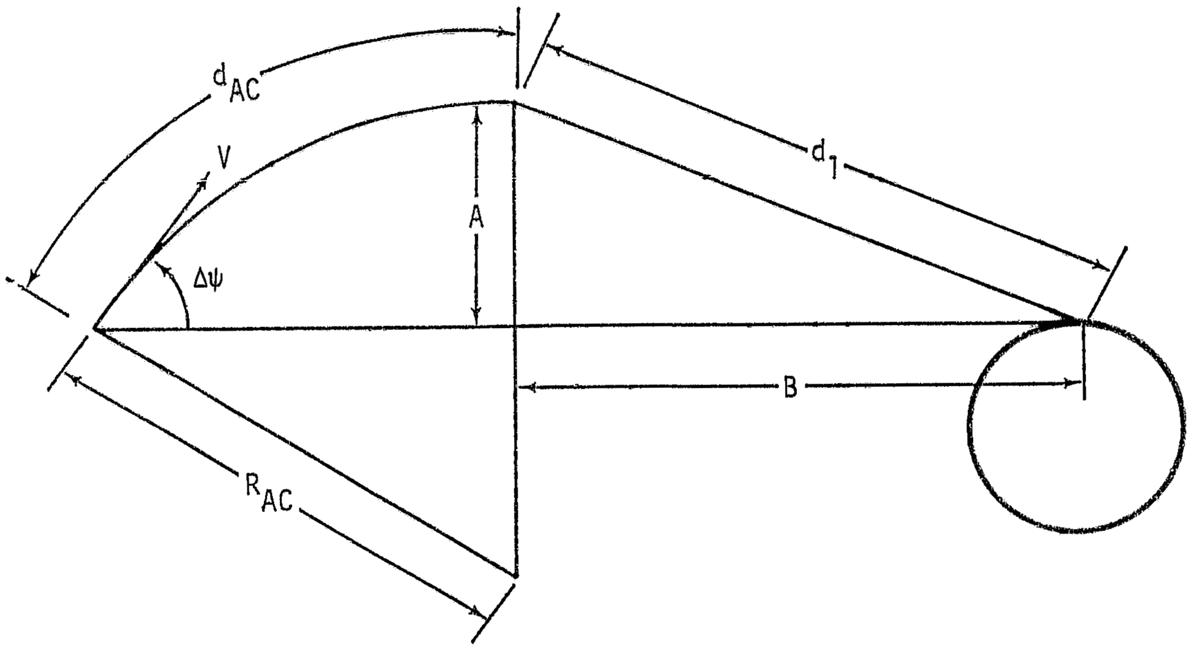


Figure 2.- GTP geometry for acquisition phase.

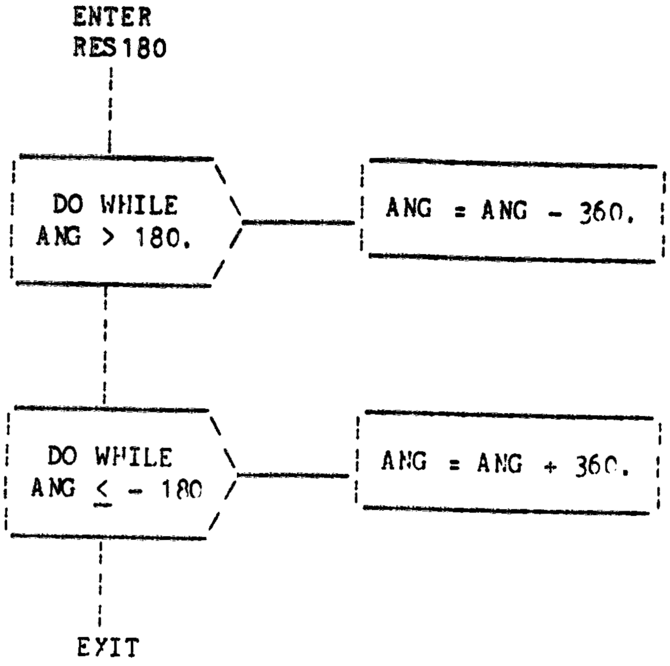


Figure 3.- Resolve to 180 degrees (RES180).

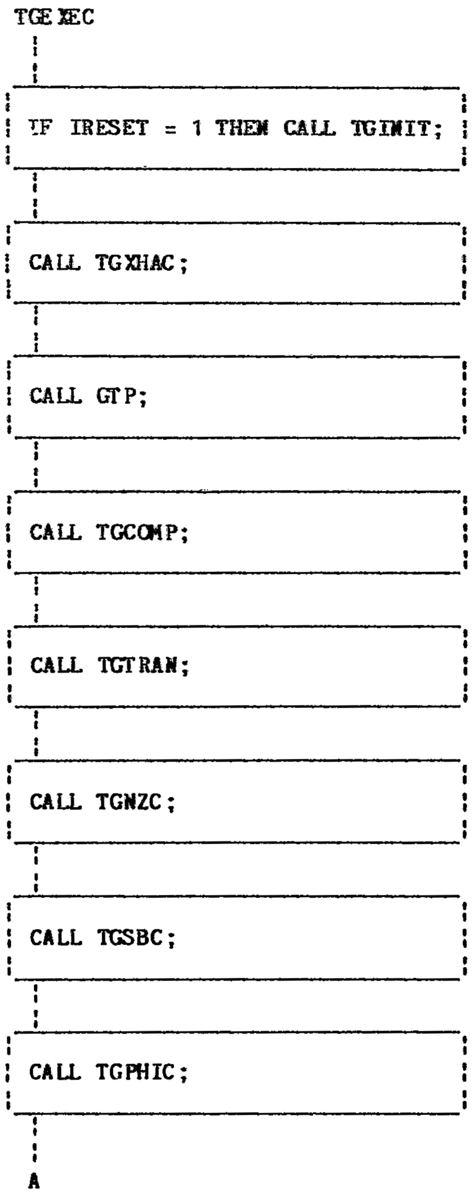
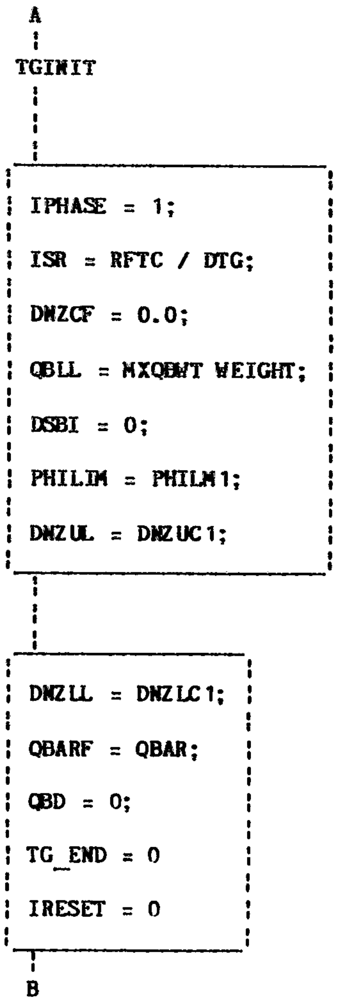


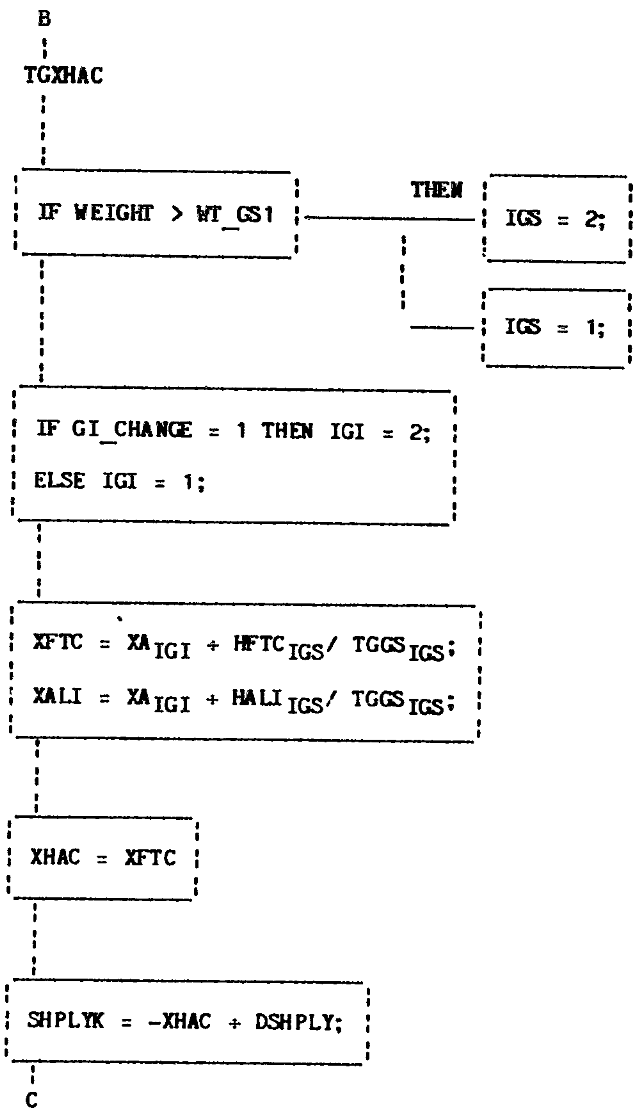
Figure 4.- TAEM guidance flow.



```
IPHASE = 1;
ISR = RFTC / DTG;
DNZCF = 0.0;
QBLL = MXQBWT WEIGHT;
DSBI = 0;
PHILIM = PHILM1;
DNZUL = DNZUC1;
```

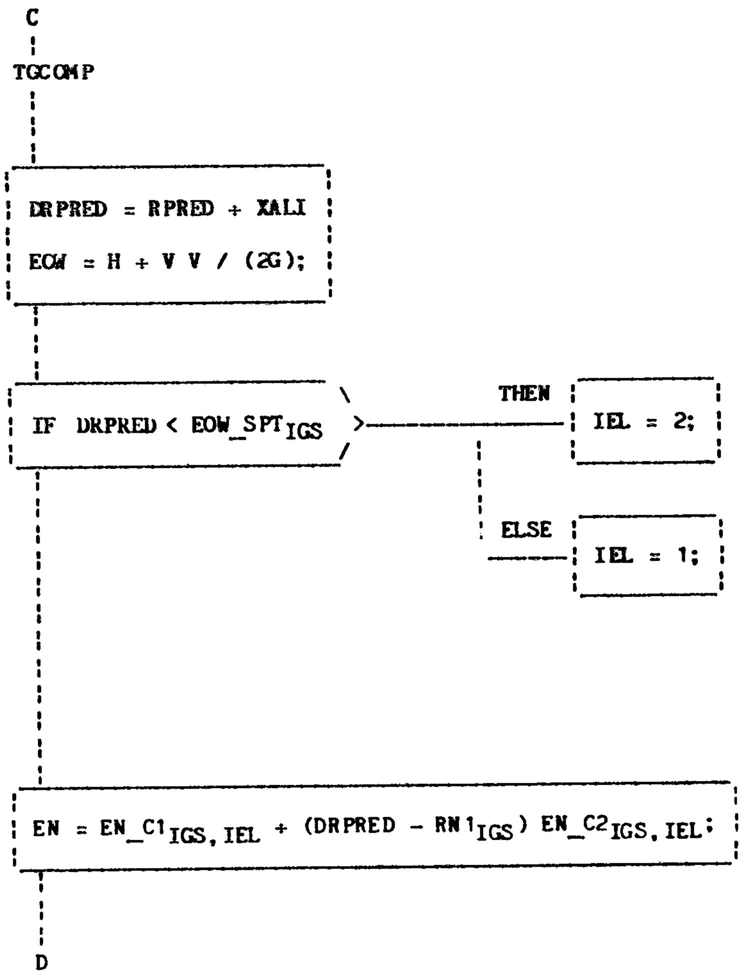
```
DNZLL = DNZLC1;
QBARF = QBAR;
QBD = 0;
TG_END = 0
IRESET = 0
```

Figure 4.- Continued.



```
IF WEIGHT > WT_GS1
    THEN IGS = 2;
    IGS = 1;
IF GI_CHANGE = 1 THEN IGI = 2;
ELSE IGI = 1;
XFTC = $XA_{IGI} + HFTC_{IGS} / TGGS_{IGS}$;
XALI = $XA_{IGI} + HALI_{IGS} / TGGS_{IGS}$;
XHAC = XFTC
SHPLYK = -XHAC + DSHPLY;
```

Figure 4.- Continued.



```
C
TGCOMP
DRPRED = RPRED + XALI
EOW = H + V V / (2G);
IF DRPRED < $EOW\_SPT_{IGS}$
    THEN
        IEL = 2;
    ELSE
        IEL = 1;
EN = $EN\_C1_{IGS, IEL}$ + (DRPRED - $RN1_{IGS}$) $EN\_C2_{IGS, IEL}$;
D
```

Figure 4.- Continued.

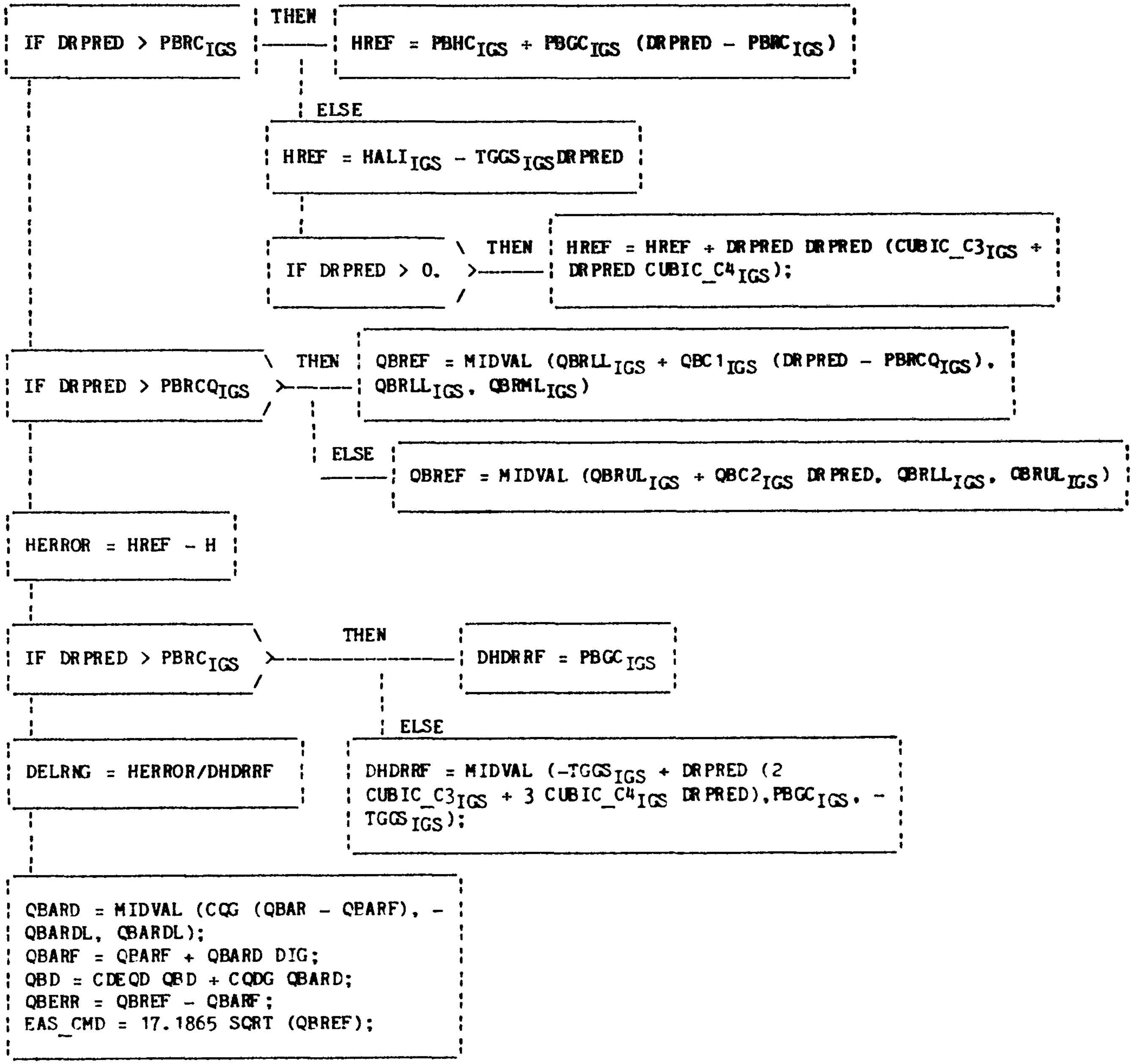
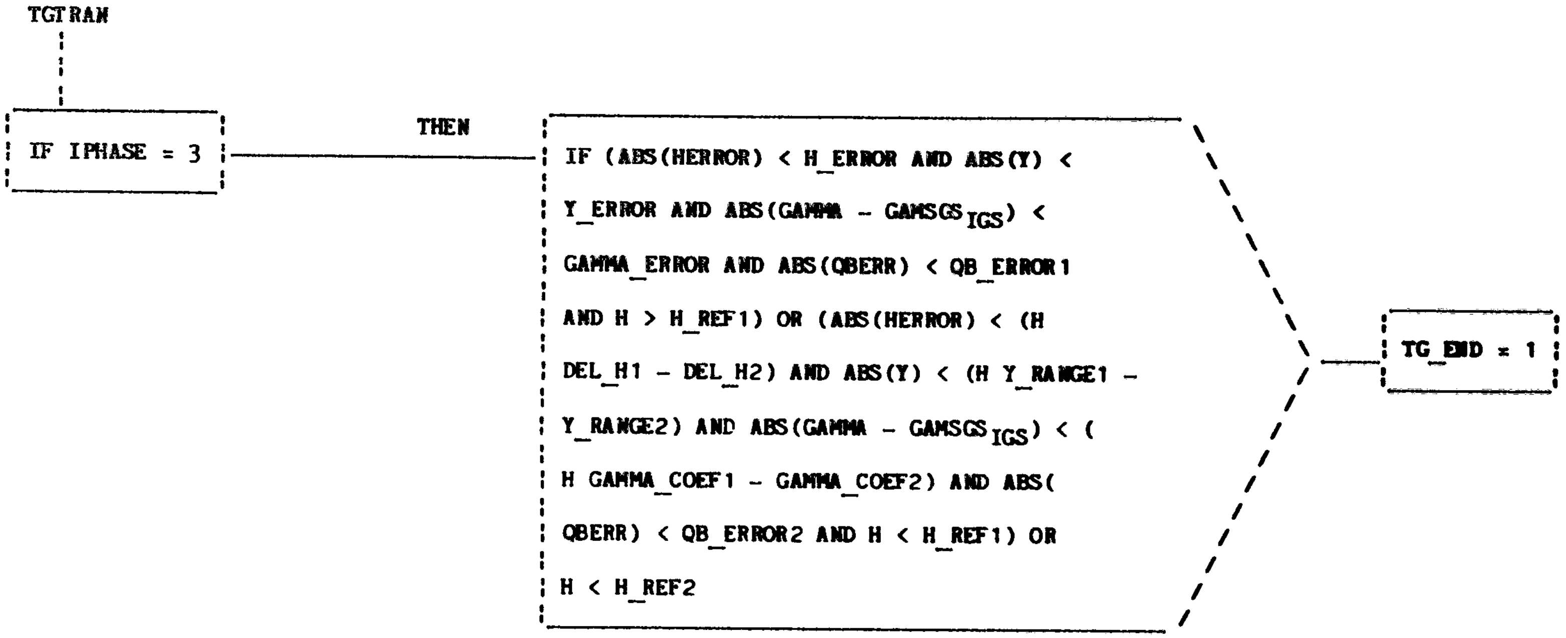


Figure 4.- Continued.

80FM25



```
IF (ABS(HERROR) < H_ERROR AND ABS(Y) <
Y_ERROR AND ABS(GAMMA - GAMSGS_IGS) <
GAMMA_ERROR AND ABS(QBERR) < QB_ERROR1
AND H > H_REF1) OR (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
```

Figure 4.- Continued.

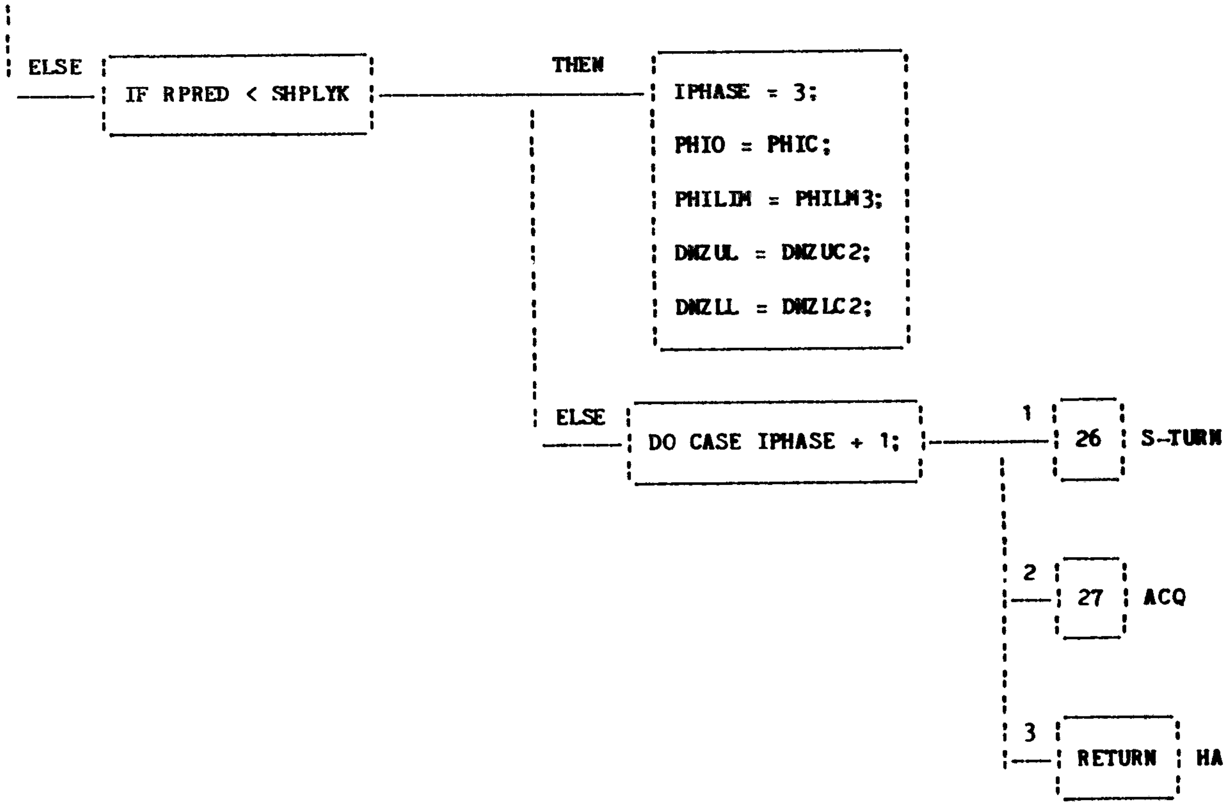


Figure 4.- Continued.

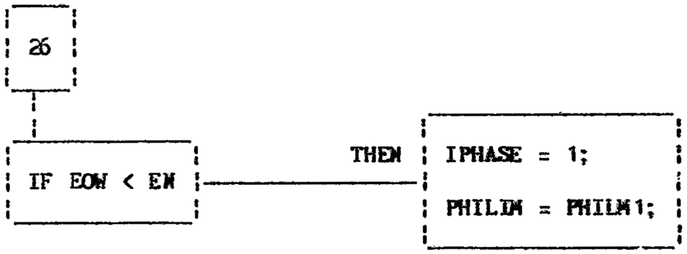


Figure 4.- Continued.

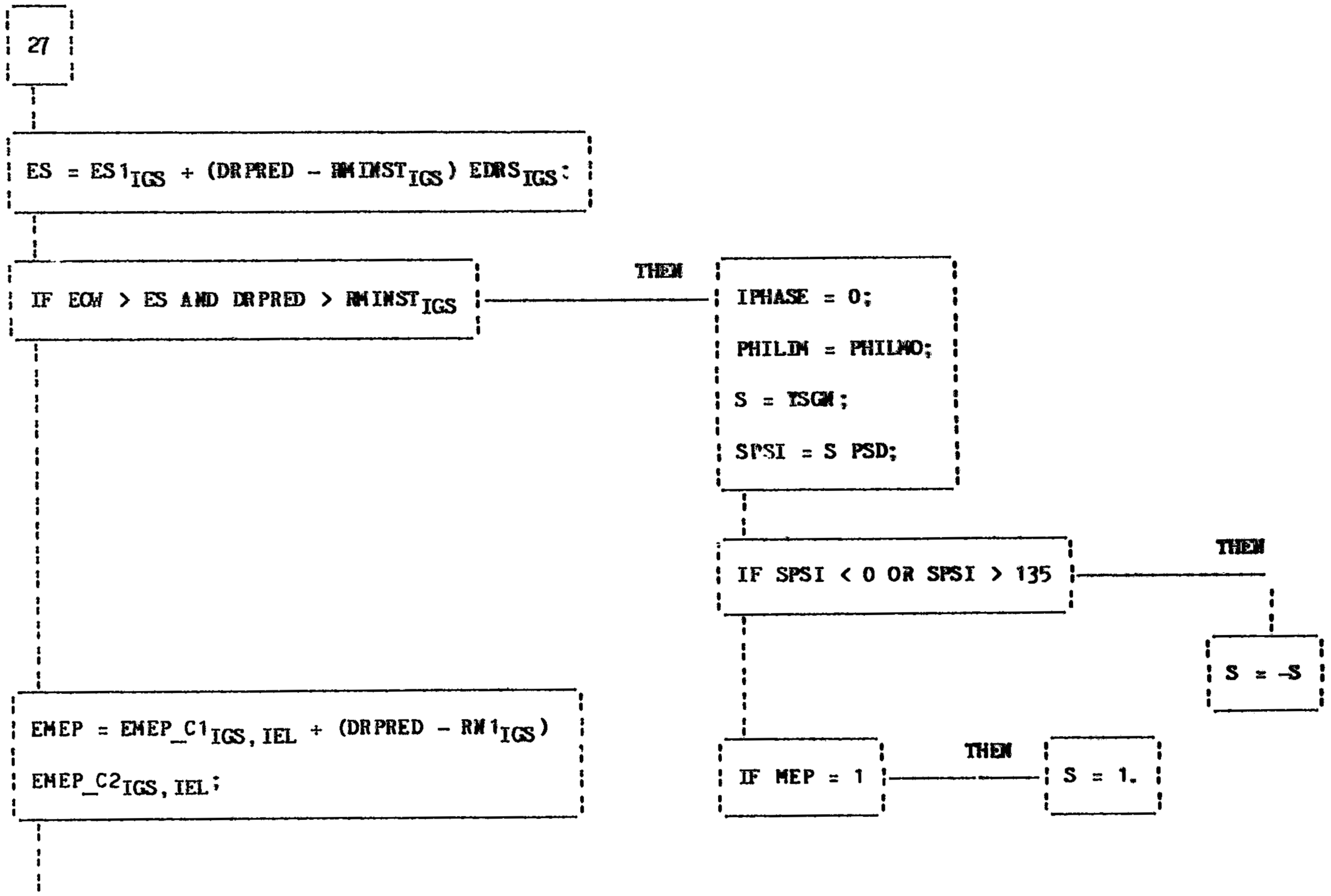


Figure 4.- Continued.

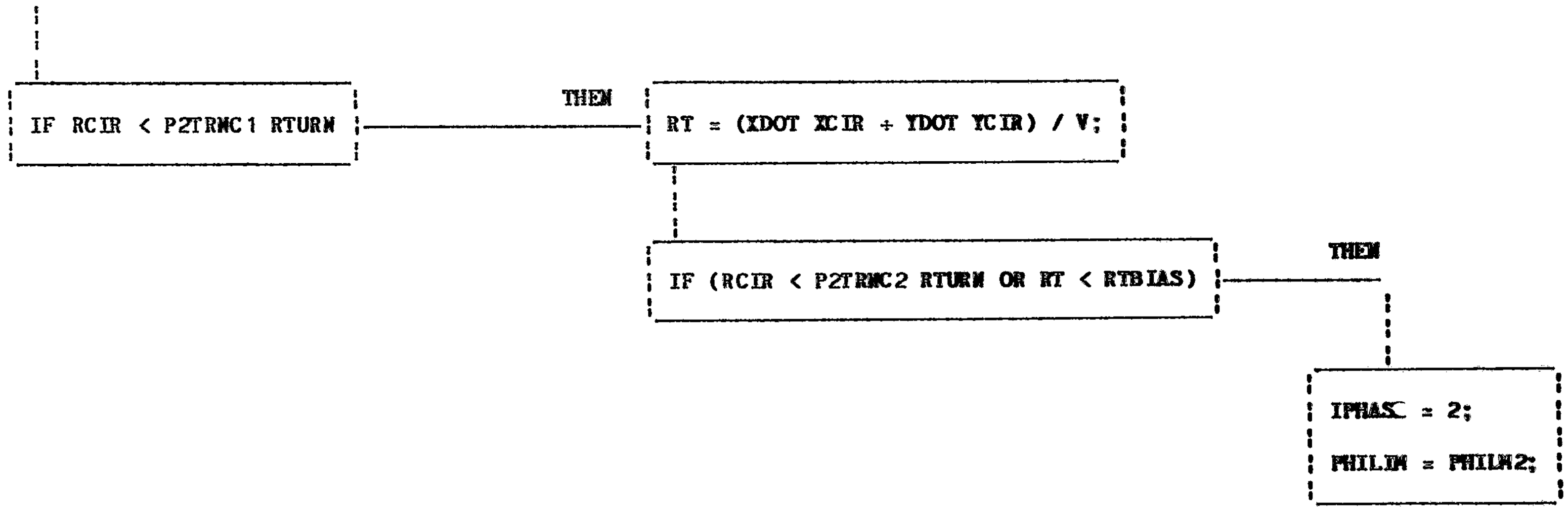
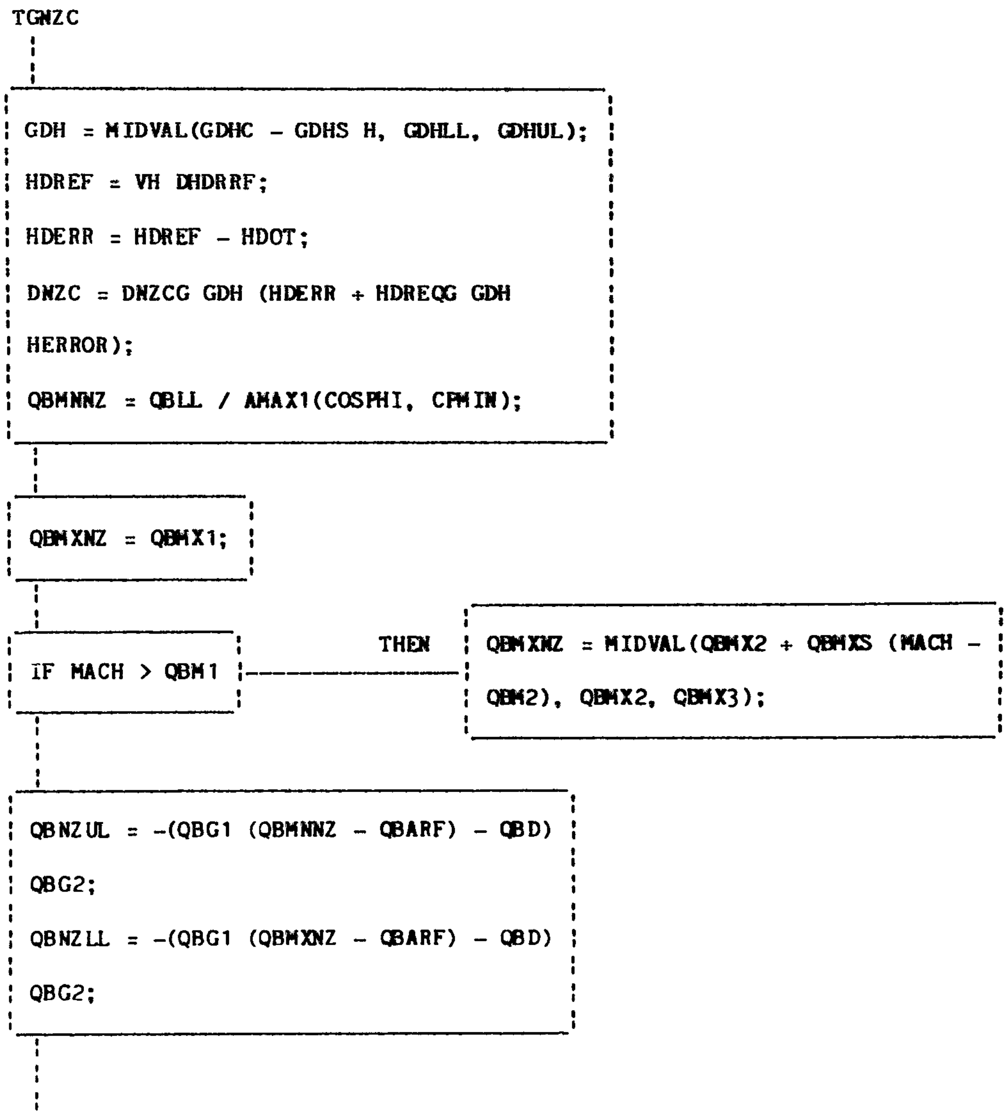


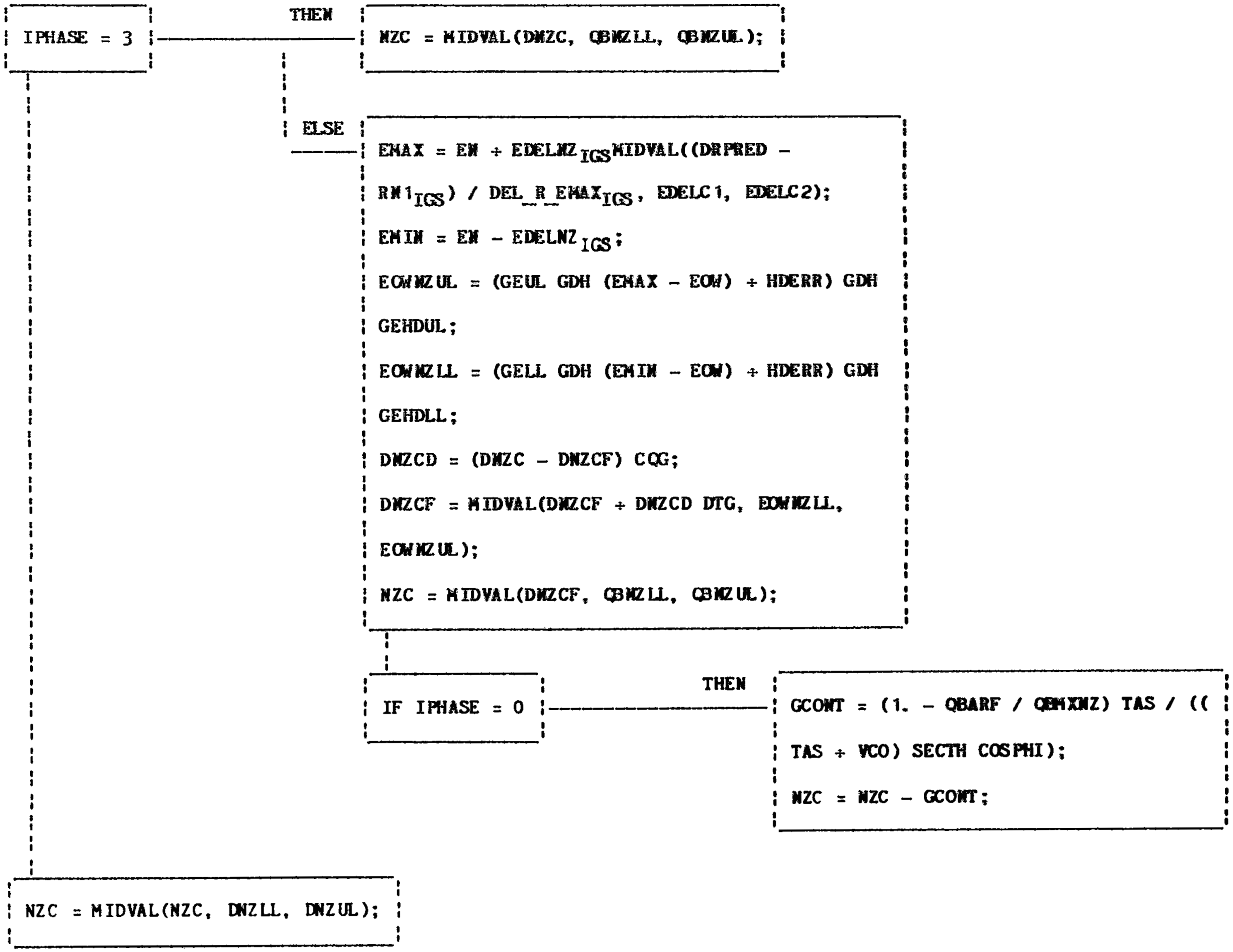
Figure 4.- Continued.



```
TGNZC

GDH = MIDVAL(GDHC - GDHS H, GDHLL, GDHUL);
HDREF = VH DHDRRF;
HDERR = HDREF - HDOT;
DNZC = DNZCG GDH (HDERR + HDREQG GDH
HERROR);
QBMNNZ = QBLL / AMAX1(COSPHI, CPMIN);

QBMXNZ = QBMX1;

IF MACH > QBM1    THEN    QBMXNZ = MIDVAL(QBMX2 + QBMXS (MACH -
                          QBM2), QBMX2, QBMX3);

QBNZUL = -(QBG1 (QBMNNZ - QBARF) - QBD)
QBG2;
QBNZLL = -(QBG1 (QBMXNZ - QBARF) - QBD)
QBG2;
```

Figure 4.- Continued.

80FM25



```
IPHASE = 3
    THEN
        NZC = MIDVAL(DNZC, QBNZLL, QBNZUL);
    ELSE
        EMAX = EN + EDELNZ$_{IGS}$MIDVAL((DRPRED -
        RN1$_{IGS}$) / DEL_R_EMAX$_{IGS}$, EDELC1, EDELC2);
        EMIN = EN - EDELNZ$_{IGS}$;
        EOWNZUL = (GEUL GDH (EMAX - EOW) + HDERR) GDH
        GEHDUL;
        EOWNZLL = (GELL GDH (EMIN - EOW) + HDERR) GDH
        GEHDLL;
        DNZCD = (DNZC - DNZCF) CQG;
        DNZCF = MIDVAL(DNZCF + DNZCD DTG, EOWNZLL,
        EOWNZUL);
        NZC = MIDVAL(DNZCF, QBNZLL, QBNZUL);
        IF IPHASE = 0
            THEN
                GCONT = (1. - QBARF / QBMXNZ) TAS / ((
                TAS + VCO) SECTH COSPHI);
                NZC = NZC - GCONT;
NZC = MIDVAL(NZC, DNZLL, DNZUL);
```

Figure 4.- Continued.

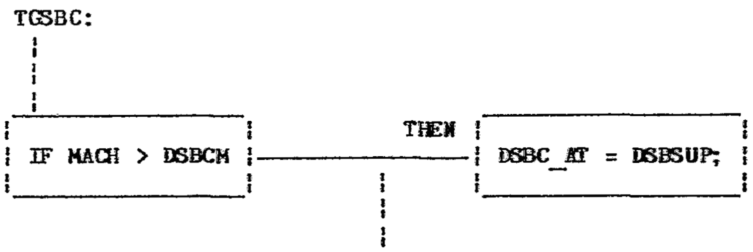
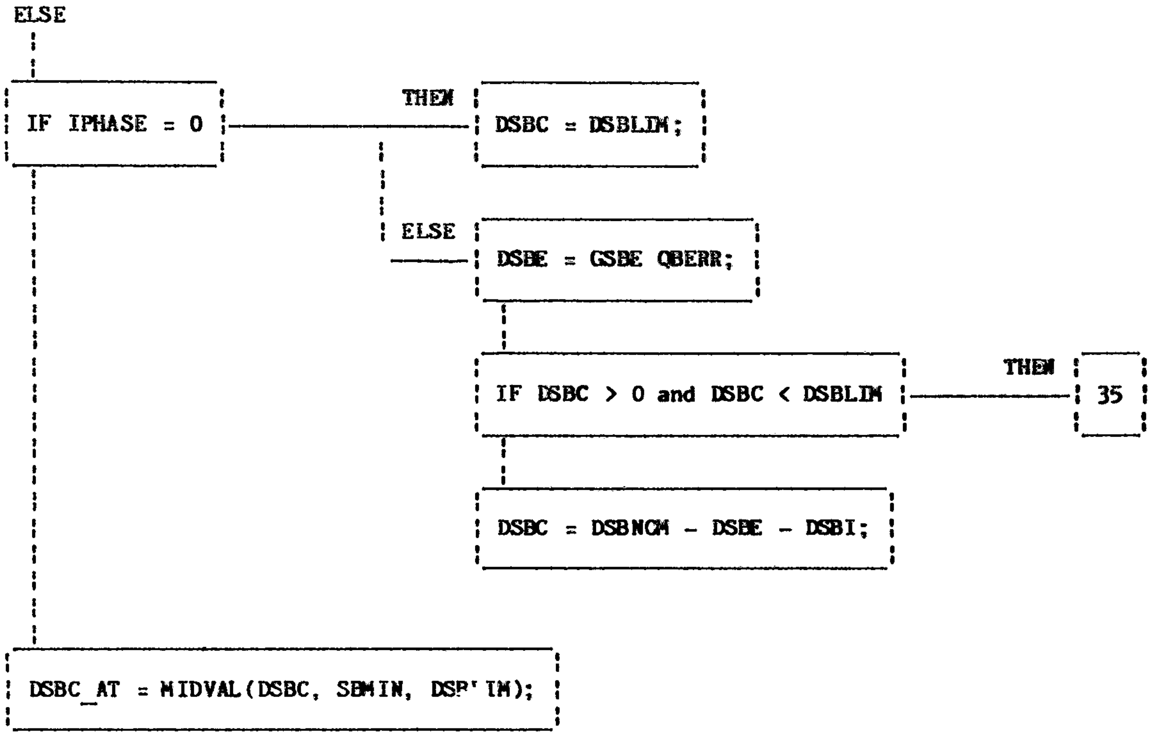
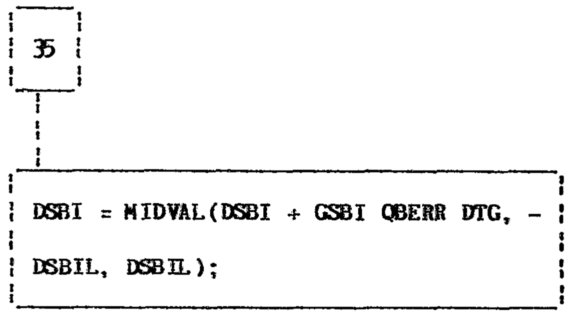


Figure 4.- Continued.



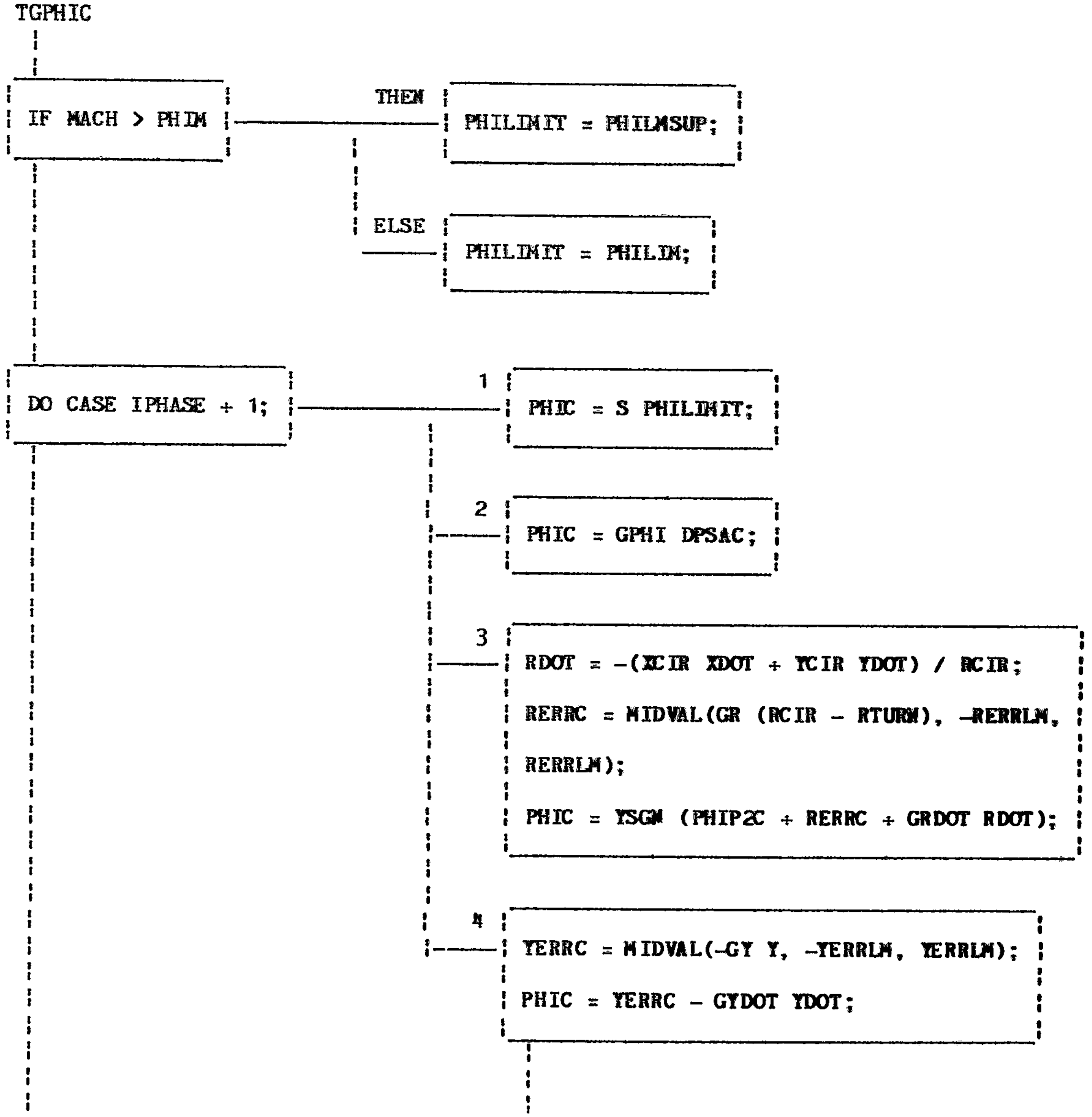
```
ELSE
IF IPHASE = 0
    THEN  DSBC = DSBLIM;
    ELSE  DSBE = GSBE QBERR;
          IF DSBC > 0 and DSBC < DSBLIM
              THEN  35
          DSBC = DSBNOM - DSBE - DSBI;
DSBC_AT = MIDVAL(DSBC, SBMIN, DSBLIM);
```

Figure 4.- Continued.



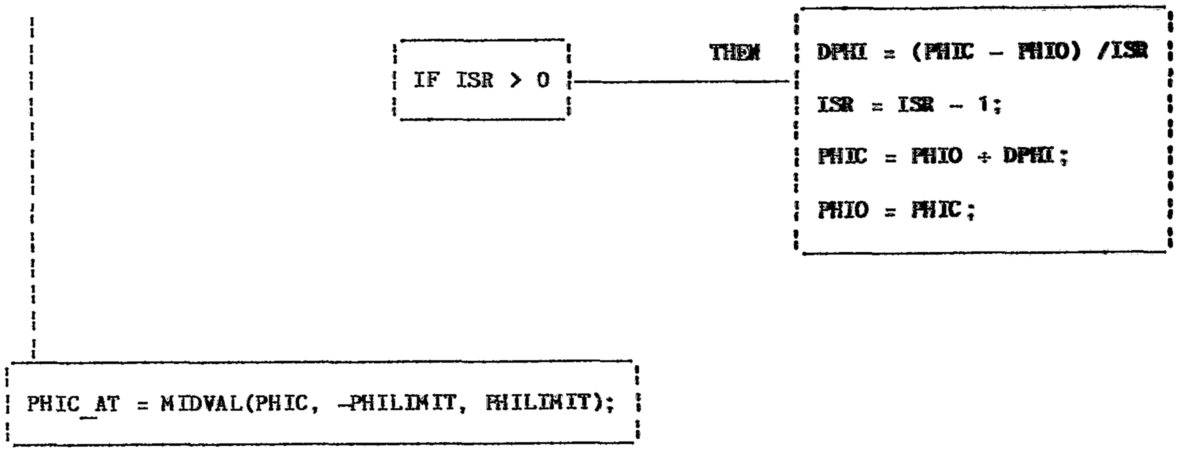
```
DSBI = MIDVAL(DSBI + GSBI QBERR DTG, -
DSBIL, DSBIL);
```

Figure 4.- Continued.



```
TGPHIC
IF MACH > PHIM
    THEN
        PHILIMIT = PHILMSUP;
    ELSE
        PHILIMIT = PHILIM;
DO CASE IPHASE + 1;
    1
        PHIC = S PHILIMIT;
    2
        PHIC = GPHI DPSAC;
    3
        RDOT = -(XCIR XDOT + YCIR YDOT) / RCIR;
        RERRC = MIDVAL(GR (RCIR - RTURN), -RERRLM,
        RERRLM);
        PHIC = YSGN (PHIP2C + RERRC + GRDOT RDOT);
    4
        YERRC = MIDVAL(-GY Y, -YERRLM, YERRLM);
        PHIC = YERRC - GYDOT YDOT;
```

Figure 4.- Continued.



```
IF ISR > 0
THEN
    DPHI = (PHIC - PHIO) /ISR
    ISR = ISR - 1;
    PHIC = PHIO + DPHI;
    PHIO = PHIC;
PHIC_AT = MIDVAL(PHIC, -PHILIMIT, PHILIMIT);
```

Figure 4.- Continued.

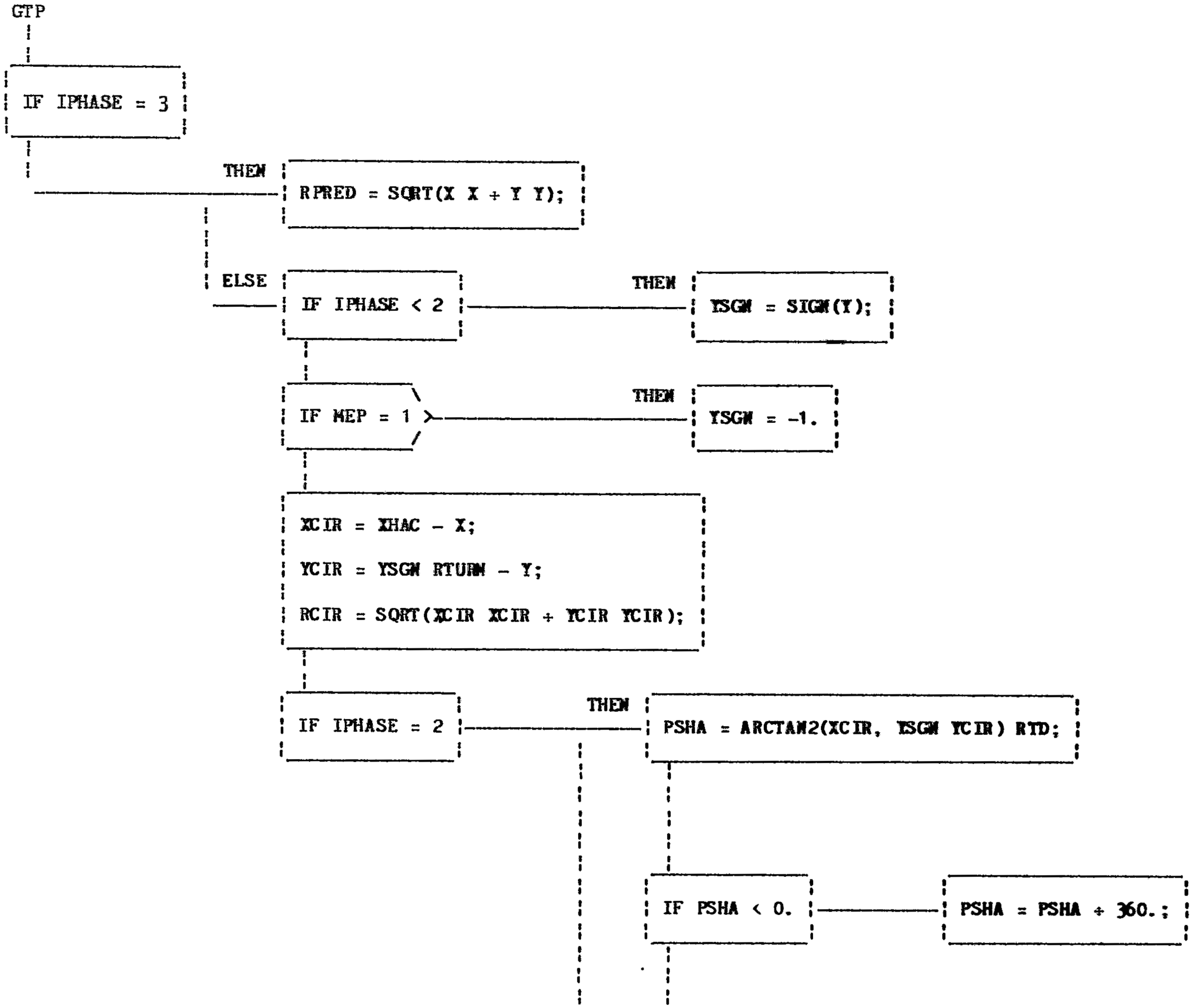


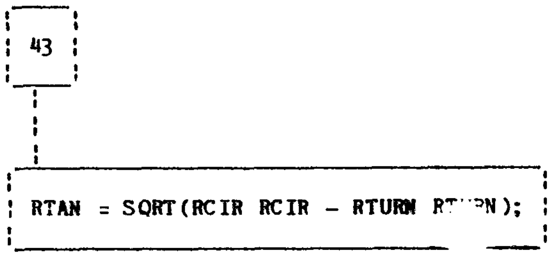
Figure 4.- Continued.

```
RPRED = RCIR PSHA DTR - XHAC;

ELSE
    IF RCIR > RTURN
        THEN  43
        ELSE  RTAN = 0.;

    PSC = ARCTAN2(YCIR, XCIR);
    PST = (PSC - YSGN ARCTAN2(RTURN, RTAN))
    RTD;
    PST = RES180(PST);
    DPSAC = RES180(PST - PSD);
    PHAVG = MIDVAL(PHAVGC - PHAVGS MACH,
    PHAVGLL, PHAVGUL);
    RTAC = V VH / (G TAN(PHAVG DTR));
    ARCAC = RTAC ABS(DPSAC) DTR;
    A = RTAC (1. - COS(DPSAC));
    B = RTAN - RTAC ABS(SIN(DPSAC));
    RC = SQRT(A A + B B);
    PSHA = ABS(PST);

    IF XCIR < 0. AND ABS(Y) < 2. RTURN
    OR (YSGN SIGN (Y) < 0.)
        THEN  44
```

Figure 4.- Continued.

```
ARCHA = RTURN PSHA DTR;

RPRED = ARCAC + RC + ARCHA - XHAC;
```

Figure 4.- Continued.



```
RTAN = SQRT(RCIR RCIR - RTURN RTURN);
```

Figure 4.- Continued.

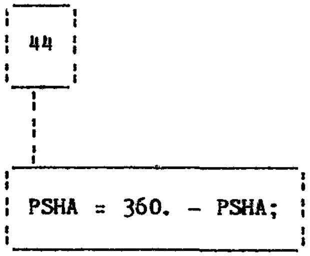
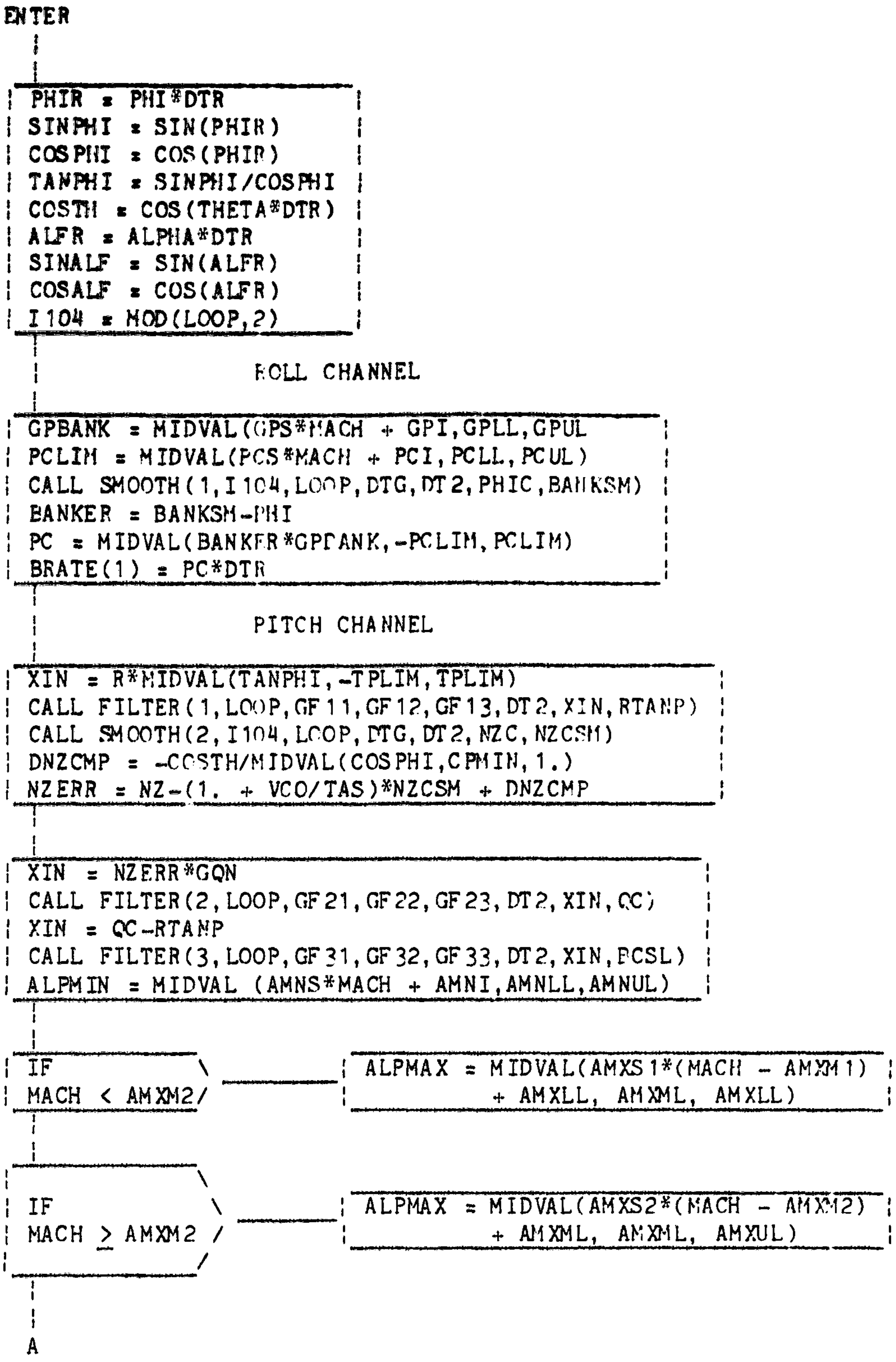


Figure 4.- Concluded.



```
ENTER
  |
  |
 ___________________________
| PHIR = PHI*DTR            |
| SINPHI = SIN(PHIR)        |
| COSPHI = COS(PHIR)        |
| TANPHI = SINPHI/COSPHI    |
| COSTH = COS(THETA*DTR)    |
| ALFR = ALPHA*DTR          |
| SINALF = SIN(ALFR)        |
| COSALF = COS(ALFR)        |
| I104 = MOD(LOOP,2)        |
 ---------------------------
  |
  |             ROLL CHANNEL
  |
 ______________________________________________
| GPBANK = MIDVAL(GPS*MACH + GPI,GPLL,GPUL     |
| PCLIM = MIDVAL(PCS*MACH + PCI,PCLL,PCUL)     |
| CALL SMOOTH(1,I104,LOOP,DTG,DT2,PHIC,BANKSM) |
| BANKER = BANKSM-PHI                          |
| PC = MIDVAL(BANKER*GPBANK,-PCLIM,PCLIM)      |
| BRATE(1) = PC*DTR                            |
 ----------------------------------------------
  |
  |             PITCH CHANNEL
  |
 _________________________________________________
| XIN = R*MIDVAL(TANPHI,-TPLIM,TPLIM)             |
| CALL FILTER(1,LOOP,GF11,GF12,GF13,DT2,XIN,RTANP)|
| CALL SMOOTH(2,I104,LOOP,DTG,DT2,NZC,NZCSM)      |
| DNZCMP = -COSTH/MIDVAL(COSPHI,CPMIN,1.)         |
| NZERR = NZ-(1. + VCO/TAS)*NZCSM + DNZCMP        |
 -------------------------------------------------
  |
  |
 ________________________________________________
| XIN = NZERR*GQN                                |
| CALL FILTER(2,LOOP,GF21,GF22,GF23,DT2,XIN,QC)  |
| XIN = QC-RTANP                                 |
| CALL FILTER(3,LOOP,GF31,GF32,GF33,DT2,XIN,PCSL)|
| ALPMIN = MIDVAL (AMNS*MACH + AMNI,AMNLL,AMNUL) |
 ------------------------------------------------
  |
  |
 ___________         ____________________________
| IF        \ ______| ALPMAX = MIDVAL(AMXS1*(MACH - AMXM1) |
| MACH < AMXM2/     |        + AMXLL, AMXML, AMXLL)        |
 -----------         ----------------------------
  |
  |
 ___________
|           \        ____________________________
| IF         \ _____| ALPMAX = MIDVAL(AMXS2*(MACH - AMXM2) |
| MACH ≥ AMXM2 /    |        + AMXML, AMXML, AMXUL)        |
|            /       ----------------------------
 ------------
  |
  |
  A
```

Figure 5.- TDAP detailed formulation.

```
  A
  |
 ____________________________________
| DQLOA  = GQA*(ALPHA-ALPHAX)        |
| DQHIA  = GQA*(ALPHA-ALPMIN)        |
| DQLON  = GQNL*(NZ-NZMAX)           |
| DQHIN  = GQNL*(NZ-NZMIN)           |
| DQLO  = AMAX1(DQLOA,DQLON)         |
| DQHI  = AMIN1(DQHIA,DQHIN)         |
 ------------------------------------
  |
 ________________
| IF             \              THEN | BCSL = MIDVAL(BCSL,DQLO,DQHI) |
| DQLO < DQHI    /  -----------------  -------------------------------
 ----------------
  |
| BRATE(2) = -BCSL*DTR |
  |
  |                      YAW CHANNEL
  |
 _________________________________________________________
| GRAY = MIDVAL(GYS*MACH + GYI,GYLL,GYUL)                 |
| DRPRM = COSTH*SINPHI*GRPHI/TAS                          |
| RSTAB = DRPRM*COSALF-P*SINALF                           |
| DPC = GBDOT*RSTAB                                       |
| CALL FILTER(4,LOOP,GF41,GF42,GF43,DT2,NY,NYF)           |
| DRRC = NYF*GRAY + DPC                                   |
| BRATE(3) = -DRRC*DTR                                    |
 ---------------------------------------------------------
  |
  |                      ATTITUDE INTEGRATOR
  |
 ____________________________________
| CALL EIGEN(BRATE, DT2, CBI, CBI)   |
| P = PC                             |
| Q = -BCSL                          |
| R = -DRRC                          |
 ------------------------------------
  |
  |                      SPEEDBRAKE CHANNEL
  |
 ______________________________
| HR = MIDVAL(HMSB/HMS,0.,1.)  |
| RMAX = SBRLC*SQRT(1.-HR)     |
| RMAX = AMIN1(SBRLO,RMAX)     |
| RMIN = -SBRLC                |
 ------------------------------
  |
 _______________
|               \
| IF             \            THEN | RMIN = -SBRS |
| DELB < SBSOFT  /  ---------------  --------------
|_______________/
  |
 ____________________________________
|DDOTRC = (DSBC-DELB)/DT2            |
|DDOTRC = MIDVAL(DDOTRC,RMIN,RMAX)   |
|DELB = DELB + DDOTRC*DT2            |
 ------------------------------------
  |
  B
```

Figure 5.- Continued.

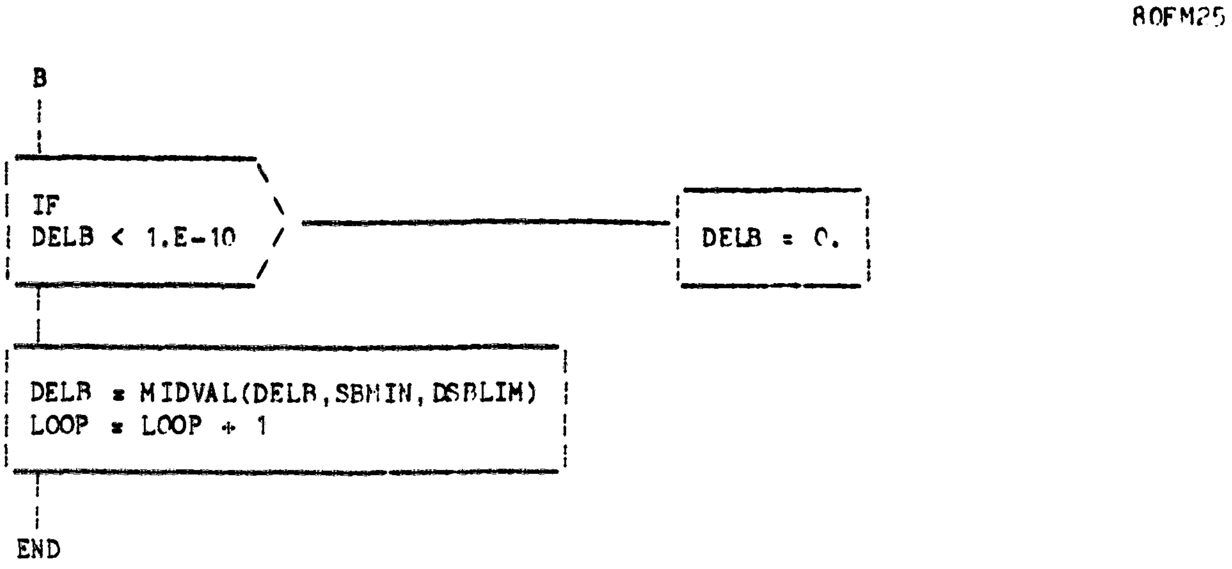


Figure 5.- Concluded.

80FM25

```
ENTER
  |
  |
INPUT:    NF,LOOP,C1,C2,A,DT2,XIN
OUTPUT:   XOUT
  |
  |
 _|___________                 ______________________________________________________________
|             \               |                                                              |
| IF           \    THEN      | GX1(NF) = C1                                                 |
| LOOP = 0     / -------------| GX2(NF) + (1.-EXP(-A*DT2))*C2/A-C1                            |
|_____________/               | GX3(NF) = -EXP(-A*DT2)                                       |
  |                           | XI(NF) = XIN                                                 |
  |                           | XO(NF) = (GX1(NF) + GX2(NF))/(1. + GX3(NF))*XIN              |
  |                           |______________________________________________________________|
  |
 _|______________________________________________________________
|                                                                |
| XOUT = GX1(NF)*XIN + GX2(NF)*XI(NF)-GX3(NF)*XO(NF)             |
| XI(NF) = XIN                                                   |
| XO(NF) = XOUT                                                  |
|________________________________________________________________|
  |
  |
RETURN
```

Figure 6.- FILTER detailed formulation.

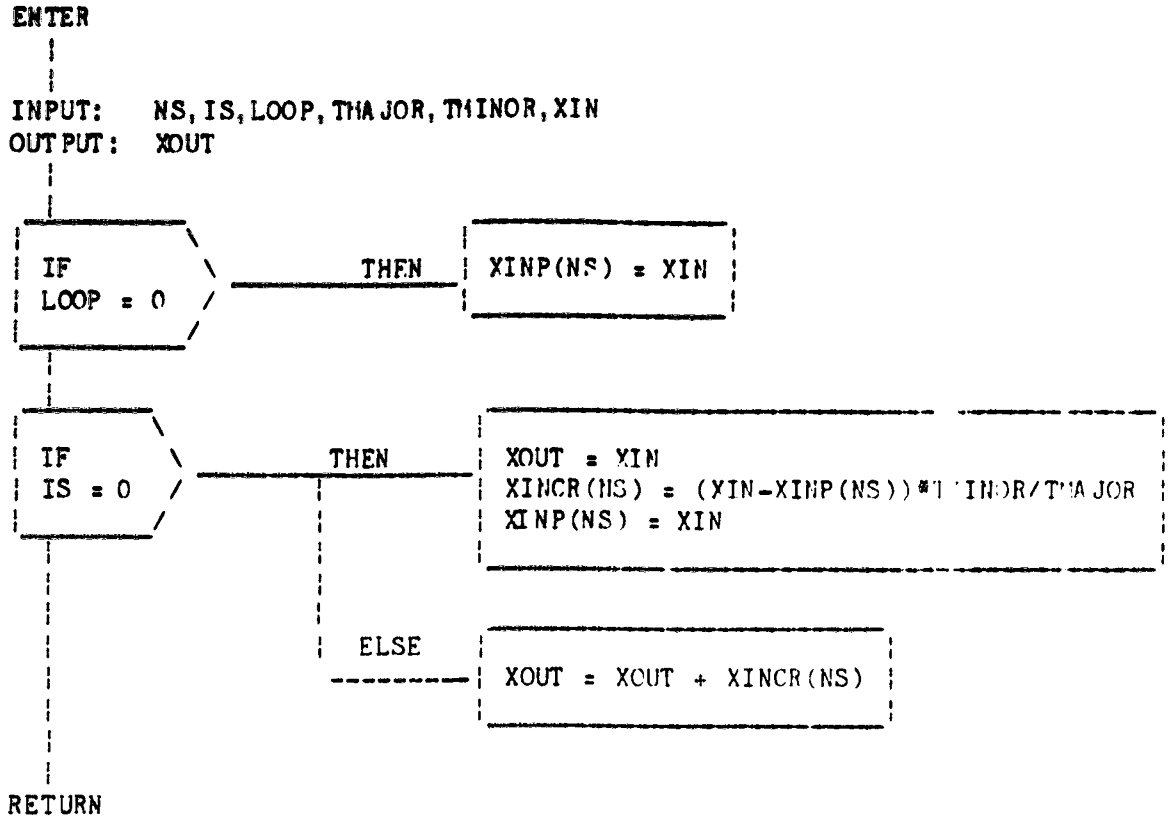
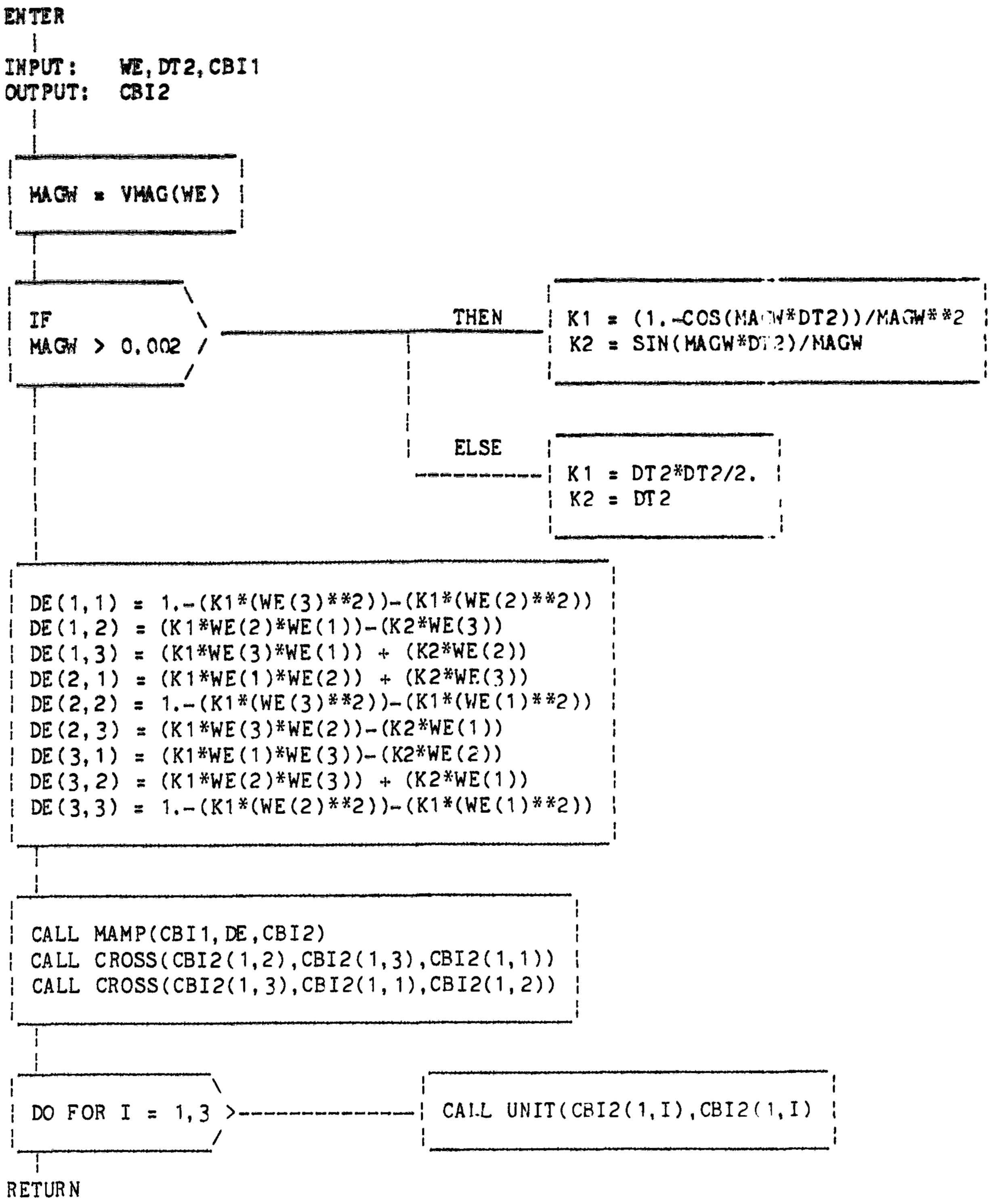


Figure 7.- SMOOTH detailed formulation.



```
ENTER
INPUT:   WE,DT2,CBI1
OUTPUT:  CBI2

MAGW = VMAG(WE)

IF MAGW > 0.002
  THEN
    K1 = (1.-COS(MAGW*DT2))/MAGW**2
    K2 = SIN(MAGW*DT2)/MAGW
  ELSE
    K1 = DT2*DT2/2.
    K2 = DT2

DE(1,1) = 1.-(K1*(WE(3)**2))-(K1*(WE(2)**2))
DE(1,2) = (K1*WE(2)*WE(1))-(K2*WE(3))
DE(1,3) = (K1*WE(3)*WE(1)) + (K2*WE(2))
DE(2,1) = (K1*WE(1)*WE(2)) + (K2*WE(3))
DE(2,2) = 1.-(K1*(WE(3)**2))-(K1*(WE(1)**2))
DE(2,3) = (K1*WE(3)*WE(2))-(K2*WE(1))
DE(3,1) = (K1*WE(1)*WE(3))-(K2*WE(2))
DE(3,2) = (K1*WE(2)*WE(3)) + (K2*WE(1))
DE(3,3) = 1.-(K1*(WE(2)**2))-(K1*(WE(1)**2))

CALL MAMP(CBI1,DE,CBI2)
CALL CROSS(CBI2(1,2),CBI2(1,3),CBI2(1,1))
CALL CROSS(CBI2(1,3),CBI2(1,1),CBI2(1,2))

DO FOR I = 1,3
    CALL UNIT(CBI2(1,I),CBI2(1,I)
RETURN
```

Figure 8.- EIGEN detailed formulation.

APPENDIX

GUIDANCE AND FLIGHT CONTROL INPUT PARAMETER CALCULATIONS

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.

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

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

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

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

where

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

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

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.

$$\overline{RE} = \left[CIE \right] \overline{RI}$$

$$\overline{VE} = \left[CIE \right] (\overline{VI} - \overline{\omega_e} \times \overline{RI})$$

The position and velocity vectors in runway coordinates are:

$$\overline{RRW} = \left[CER \right] (\overline{RE} - \overline{RRE})$$

$$\overline{VRW} = \left[CER \right] \overline{VE}$$

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

$$\phi_D = \mathrm{ATAN} \left[\frac{RE_z}{(RXY/A)} \right]$$

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

Longitude is easily obtained from

$$\lambda = \mathrm{ATAN2}(RE_y, RE_x)$$

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

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

The topodetic velocity related parameters can now be obtained.

$$\overline{VT} = \begin{bmatrix} CET \end{bmatrix} \overline{VE}$$

$$HDOT = -VT_z$$

$$VH = \sqrt{VT_x^2 + VT_y^2}$$

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

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

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

$$X + RRW_x$$

$$Y = RRW_y$$

$$XDOT = VRW_x$$

$$YDOT = VRW_y$$

$$PSD = ATAN2(VRW_y, VRW_x) \cdot RTD$$

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

$$CBT = \begin{bmatrix} CET \end{bmatrix} \begin{bmatrix} CIE \end{bmatrix} \begin{bmatrix} CBI \end{bmatrix}$$

$$THETA = -ASIN(CBT_{(3,1)}) \cdot RTD$$

$$PHI = ATAN2(CBT_{(3,2)}, CBT_{(3,3)})\cdot RTD$$

$$COSPHI = COS(PHI/RTD)$$

$$SECTH = 1./COS(THETA/RTD)$$

The Shuttle flight control sensors include body axis accelerometers and rate gyros which 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.**

$$NY = FAB_y/WEIGHT$$

$$NZ = -FAB_z/WEIGHT$$

c. Air data parameters - The Shuttle avionics includes an air data subsystem which 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. Air data parameters will be supplied to the TAEM guidance and flight control below a predetermined mach number (MACHAD) which will be mission dependent. This mach number will be 2.5 for the STS-1 mission. The parameters supplied by the air data system are required continuously by the TAEM guidance and fight control and, therefore, must be obtained elsewhere when acceptance of air data is 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 which 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 (MACH > MACHAD), the velocity of sound is assumed constant (1000 fps) and standard air density is approximated by

$$\rho_s = .00413579\ e^{(-h_D/20600.)} \qquad (h_D > 35000.)$$

$$\rho_s = .00237689\ e^{(-h_D/30550.)} \qquad (h_D \leq 35000.)$$

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

$$\overline{VBE} = \left[CBT\right]^T \overline{VT}$$

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

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

$$ALPHA = ATAN\ (VBE_Z/VBE_X) \cdot RTD$$

$$MACH = TAS/1000.$$

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

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

$$\overline{VBW} = \left[CBT\right]^T (\overline{VT} - \overline{VW})$$

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

$$ALPHA = ATAN\ (VBW_Z/VBW_X) \cdot RTD$$

$$MACH = TAS/VS$$

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

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
$\overline{\text{FAB}}$	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
$\overline{\text{RE}}$	position vector of vehicle center of mass (Greenwich coordinate)
REQ	equatorial radius of Earth ellopsoid model
$\overline{\text{RI}}$	position vector of vehicle center of mass (inertial coordinate)
$\overline{\text{RRE}}$	position vector of runway threshold (Greenwich coordinate)
$\overline{\text{RRW}}$	position vector of vehicle center of mass (runway coordinate)
RTD	radians to degrees conversion constant
$\overline{\text{VBE}}$	vehicle Earth relative velocity vector (body coordinate)
$\overline{\text{VBW}}$	vehicle air relative velocity vector (body coordinate)
$\overline{\text{VE}}$	velocity vector of vehicle center of mass (Greenwich coordinate)
$\overline{\text{VI}}$	velocity vector of vehicle center of mass (inertial coordinate)
$\overline{\text{VRW}}$	velocity vector of vehicle center of mass (runway coordinate)
VS	true atmospheric velocity of sound
$\overline{\text{VT}}$	velocity vector of vehicle center of mass (topodetic coordinate)
$\overline{\text{VW}}$	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
$\overline{\omega_e}$	Earth angular rate vector (inertial coordinate)

☆ U.S. GOVERNMENT PRINTING OFFICE 1980 — 671-089/1587