MSC - 04217
REVISION B

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

PROJECT/SPACE SHUTTLE

SPACE SHUTTLE GUIDANCE, NAVIGATION AND CONTROL DESIGN EQUATIONS

VOLUME IV

DEORBIT AND ATMOSPHERIC OPERATIONS

REVISED
DECEMBER 1, 1971

(NASA-TM-X-67709) SPACE SHUTTLE GUIDANCE, NAVIGATION AND CONTROL DESIGN EQUATIONS. VOLUME 4: DEORBIT AND ATMOSPHERIC OPERATIONS K.J. Cox (NASA) 1 Dec. 1971 94 p
CSCL 17G G3/31
N72-23912
Unclas
26897

SYSTEMS ANALYSIS BRANCH
GUIDANCE AND CONTROL DIVISION
MANNED SPACECRAFT CENTER
HOUSTON,TEXAS

MSC - 04217
REVISION B

NASA SPACE SHUTTLE PROGRAM WORKING PAPER

SPACE SHUTTLE GUIDANCE, NAVIGATION AND CONTROL DESIGN EQUATIONS

VOLUME IV
DEORBIT AND ATMOSPHERIC OPERATIONS

(Revised
December 1, 1971

Vol 1 - N71 [illegible]
Vol 2 - N71-3[illegible]
Vol [illegible]

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
MANNED SPACECRAFT CENTER
HOUSTON, TEXAS

Vol. 3 - N [illegible]
Vol 4 - N71- [illegible]

Prepared by
Systems Analysis Branch
Guidance and Control Division

K. J. Cox

K. J. Cox, Chief
Systems Analysis Branch

Authorized for
Distribution
for Maxime A. Faget
Director of Engineering and Development

PRECEDING PAGE BLANK NOT FILMED

FOREWORD

This second publication of the Space Shuttle GN&C Design Equation Document contains baseline equations for approximately fifty percent of the GN&C computation requirements as specified in the GN&C S/W Functional Requirements Document (MSC-03690 Rev. B). This document supercedes the original MSC-04217 and the subsequently published revision. Additions or corrections to this document since its original publication are indicated in the Table of Contents by asterisks in the margin.

It is planned to republish this document in a new revision in approximately four months time. At that time it is anticipated that equations will be available for virtually all requirements. The new revision will be issued with format changes intended to stress interdependency of related submittals and to eliminate duplication to the greatest degree practicable.

This issue has been modified to reflect the shuttle-structure and avionics-configuration changes which have occurred subsequent to the first issue. A significant change is that orbiter control of the booster has been added as a requirement. Decentralization of the computations and allocation to subsystems is the current trend with the MARK I & MARK II shuttle configurations. The computation requirements for shuttle vehicles and missions may be much less than those allowed for in this document. However, since the configurations are very fluid at this state in the shuttle development, the approach adopted in this document is to include as complete a set of design equations as possible to cover reasonable possibilities. Therefore, subsets of equations may be extracted from this document to form specifications for specific vehicles, computers and missions.

The GN&C Design Equations document is the result of the efforts of many people from NASA and support contractors. The list is too long to credit all contributors; however, contractors which made direct contributions to the document are as follows:

a. TRW Systems Group, Inc., Houston Operations
b. MIT/Charles Stark Draper Laboratory
c. Lockheed Electronics Co., Inc., Houston Aerospace Systems Division
d. The Boeing Co., Houston, Texas

The equations are reviewed by the GN&C Formulation and Implementation Panel and their comments included on submittal forms where appropriate. The names of equation submitters are included on the submittal sheet in each section. Comments on the submittals should be referred to the individual submitter or to the responsible NASA engineer. General comments on the document or proposed submittals should be referred to the System Analysis Branch, Guidance and Control Division.

TABLE OF CONTENTS

TABLE OF CONTENTS (CONTINUED)

VOLUME III - ORBITAL OPERATIONS

TABLE OF CONTENTS (CONTINUED)

VOLUME IV - DEORBIT AND ATMOSPHERIC OPERATIONS

TABLE OF CONTENTS (CONTINUED)

VOLUME V - FLOW DIAGRAMS

VOLUME VI - CONSTANTS AND KEYBOARD ACCESSIBLE PARAMETERS

1. PURPOSE

The purpose of this document is to specify the equations necessary to perform the guidance, navigation and control onboard computation functions for the space shuttle orbiter vehicle. This equations document will provide as comprehensive a set of equations as possible from which modules may be chosen to develop Part I Specifications for particular vehicles, computers and missions. This document is expected to be the source of any equations used to develop software for hardware/software feasibility testing, for ground-based simulations or flight test demonstrations.

2. SCOPE

This document defines a baseline set of equations which fulfill the computation requirements for guidance, navigation and control of the space shuttle orbiter vehicle. All shuttle mission phases are covered from Prelaunch through Landing/Rollout. The spacecraft flight mode and the aircraft flight mode are addressed. Equations are included for the Mark I systems and Mark II systems through the all-up shuttle configuation. Control of the booster during launch is covered. The baseline equations may be implemented in a single GN&C computer or may be distributed among several subsystem computers, depending upon the outcome of centralization/decentralization deliberations currently in progress.

3. APPLICABILITY

This document is applicable to the guidance, navigation and control (GN&C) computation functions for the space shuttle orbiter vehicle. It specifies a set of baseline design equations which may be used for the shuttle program software specification and hardware sizing. It defines the baseline equations for MSC G&CD hardware/software simulation.

9. DESCRIPTIONS OF EQUATIONS

The detailed equations for the GN&C functions are defined in this section. The organization of this section is tentative and will be modified so as to present the equations as they are designed in as clear a fashion as possible. As an introduction to each major subsection (usually a mission phase), the general GN&C software functions to be implemented will be identified and, where appropriate, a conceptual discussion and top level flow of the computations, inputs and outputs will be included in order to understand and summarize what is to be covered. This should be an order of magnitude less detailed than the flow diagrams of the equations which come later.

A GN&C Equation Submittal sheet will introduce each of the GN&C equation submittals and summarize the GN&C functions, and identify the source and NASA contact for each.

The detailed data to be presented for each GN&C function within each of the major subsections (usually a mission phase) is summarized below. Although items 6 through 10 are to be referenced only in the equations document, they are required submittals before the equations can be approved and finalized for flight software development.

1. Functional Requirements

   The specific functional requirements (from the GN&C Software Functional Requirements Document) which are satisfied by the equations should be identified.

2. Functional Diagram

   A brief functional explanation and description of the overall concept and approach. A functional block diagram should be used where clarity is enhanced. Inputs, outputs, and interfaces will be providdd.

3. Equations and Flows

   Detailed equations and a descriptive text which guides the reader through the flows of Section 10 should be provided. The minimum frequency of the computations shall be specified and rationale given or referenced.

4. Coordinate System

The coordinate systems used shall be defined.

5. Constants/Variables Summary

Constants and variables shall be summarized in tabular form with the following information:

a. Variables/constants symbols and definitions
b. Units
c. Allowable quantization
d. Range of values

6. FORTRAN Coding

The FORTRAN coding of the function for verification using the Space Shuttle Flight Simulation (SSFS) will be referenced.

7. Simulation

The SSFS specifications, description and user's guide used to verify each GN&C function will be referenced.

8. Testing

Test plans and test results will be referenced.

9. Derivation

The mathematical derivation of the equations including all mathematical assumptions shall be referenced.

10. Assumptions

The following will be referenced:

a. Avionics baseline system assumed
b. Reference missions assumed
c. Vehicle mass properties assumed
d. Propulsion models assumed
e. Environment models assumed
f. Error models assumed

The major subsections of this section are identified and partially expanded in the following.

9.12 DEORBIT AND ENTRY

The Deorbit and Entry phase of the Shuttle mission begins with the Shuttle performing the Deorbit Targeting for the Deorbit (or Deorbit phasing, if required) burn and terminates at the beginning of transition to low angle of attack. The primary landing point and the required orbit maneuvers have been determined from pre-phase planning computations. The GN&C Software functions to be performed during this phase are:

1. Deorbit Targeting will be performed in the coast period prior to the deorbit burn(s). Targeting outputs (for a specified landing site input) will include parameters as time(s) of ignition, initial attitude for the burn(s), burn time, total ΔV for each burn, IMU realignment schedule, time of reentry, transition time, cross and downrange required, jet engines ignition time, and time of touchdown.

2. Orbital navigation using ground beacon or star/horizon measurements as required to update Shuttle State prior to thrusting maneuvers and aerodynamic entry augmented by available external measurements.

3. Powered Flight guidance resulting in commands to autopilot (attitude), thrust vector controls, or thrust throttle commands to achieve desired conditions (state and attitude) for transition and limit entry g-loading and heating.

4. Autopilot computations to compute reaction control system commands to achieve the proper attitude.

Figure 1 displays a functional flow diagram of the Deorbit and Entry Software Functions.

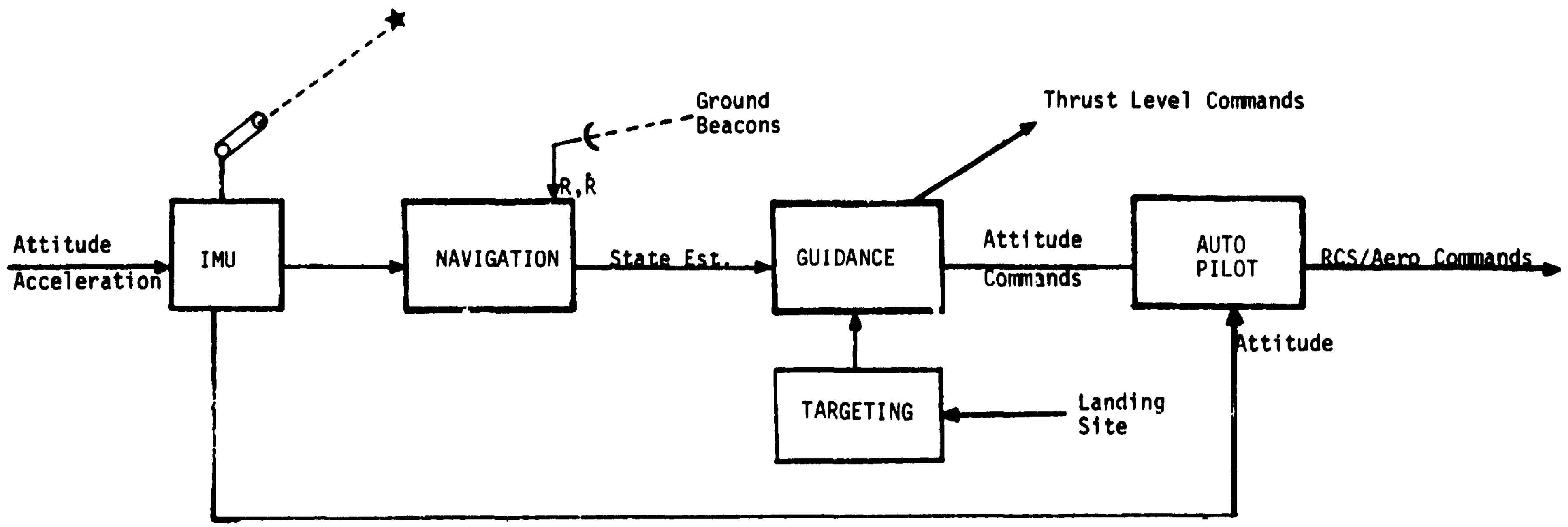


FIGURE 1. FUNCTIONAL DIAGRAM OF GN&C SOFTWARE FUNCTIONS DURING DEORBIT AND ENTRY

9.12.1 Debrbit Targeting

SPACE SHUTTLE

GN&C SOFTWARE EQUATION SUBMITTAL

Software Equation Section Deorbit Targeting Submittal No. 28

Function To determine the deorbit manuever as a function of entry range and flight path angle.

Module No. OG-5 Function No. 1,2,4,5 (MSC-03690 Rev. A)

Submitted by: Brand & Brennan (name) Co. MIT No. 14

Date: 24 August 1971

NASA Contact C. Lively Organization EG2

Approved by Panel III K.J. Cox (chairman) Date 9/24/71

Summary Description: This routine has been designed to step through successive solutions (valid opportunities to fulfill entry conditions withiinconstraints), allowing the crew to make selection based on entry crossrange, time-to ignition, required velocity change, l.s. lighting conditions, urgency of return, etc.

Shuttle Configuration: (Vehicle, Aero Data, Sensor, Et Cetera)

This software is independent of shuttle configuration. Its utility requires sigificant cross-range capability.

Comments:

(Design Status)

(Verification Status)

Panel Comments:

9.12.1 Deorbit Targeting (contd)

1. INTRODUCTION

The large entry crossrange capability of the shuttle permits deorbit to a specified landing site to be accomplished with a single maneuver. Since the required velocity change is smallest when no plane change is made, the equations presented here are designed to target the Powered Flight Guidance Routines (Reference 3) for an in-plane maneuver. The ignition time for this maneuver is selected to satisfy entry interface and landing site constraints with minimum fuel expenditure.

If the shuttle had no crossrange capability, then an in-plane deorbit maneuver to a specified landing site could only occur when that landing site, which rotates with the earth, intersects the orbital plane of the vehicle. Assuming the landing site latitude is less than the orbital inclination angle, and neglecting the effects of precession, the landing site will intersect the orbital plane twice every twenty-four hours. However, the time difference between these two intersections is in general not twelve hours. In the case when the landing site latitude is equal to the orbital inclination there will be only one intersection every twenty-four hours.

Since the shuttle has a high crossrange capability, deorbit does not require intersection of the landing site vector and the orbital plane. It is possible whenever the angle between the landing site vector and the orbital plane is less than approximately 20 deg. In general, there will be two sets of opportunities every twenty-four hours. Within each set, there may be several deorbit opportunities occurring on consecutive orbits with varying crossrange requirements. When the latitude of the landing site approaches the inclination of the orbit, these two sets merge to become one. It should be noted, in addition, that if the landing site latitude is greater than the orbital inclination, the landing site may still fall within the crossrange capability of the vehicle. With these facts in mind, this routine has been designed to continue stepping through successive solutions, allowing the crew to select a particular deorbit opportunity based upon entry crossrange, time-to-ignition, required velocity change, landing site lighting conditions, urgency of the return, etc.

The desired entry range and flight path angle will be considered inputs to this routine, since available data relating to footprint size and shape, entry heating at various ranges, and optimal entry flight path angle are only preliminary. In future revisions, consideration should be given to computing the optimum values of these quantities for the particular situation.

NOMENCLATURE

a	Semimajor axis
a_1	Alarm code—failure in Δv minimization loop
a_2	Alarm code—failure in Precision Required Velocity Determination Routine
a_F	Semimajor axis of Fischer Ellipsoid
a_T	Estimated magnitude of the thrust acceleration
b_F	Semiminor axis of Fischer Ellipsoid
d	Number of columns of navigation filter weighting matrix (set to 0 in this routine since the matrix is not required)
d_{ACR}	Maximum acceptable crossrange distance of Orbiter
d_{CR}	Estimated entry crossrange distance
d_{DR}	Entry downrange distance
f	Magnitude of the engine thrust
f_{ACS}	Magnitude of the attitude control system translational thrust
f_{OMS}	Magnitude of the orbital maneuvering system engine thrust
h_{EI}	Entry interface altitude (400, 000 ft)

9.12.1 Deorbit Targeting (cont'd)

$\underline{i}$ Unit vector formed by the cross product of the angular momentum and the landing site vectors

$\underline{i}_{EI}$ Unit vector in the direction of $\underline{r}_{EI}$

$\underline{i}'_{EI}$ First estimate of $\underline{i}_{EI}$

$i'_{EI,z}$ Z-component of the unit vector $\underline{i}'_{EI}$ (z-axis assumed North)

$\underline{i}_h$ Unit vector in the direction of the angular momentum

$\underline{i}_{LSP}$ Unit vector in the direction of the landing site projection into the orbital plane

$\underline{i}_N$ Unit normal to the trajectory plane (in the direction of the angular momentum at ignition)

k_γ Sensitivity coefficient used to compute adjustment to time-of-arrival at entry interface

m Estimated vehicle mass

n Iteration counter

n_{max} Iteration limit

n_{rev} Integral number of complete revolutions to be made in the transfer (set to zero in this routine)

p_D Semilatus rectum of deorbit trajectory

p_γ Secant squared of the desired entry flight path angle

$p_{PF\gamma}$ Secant squared of the offset entry angle used by the Powered Flight Guidance Routine

$\underline{r}_0$ Precision position vector

9.12.1 __Deorbit Targeting__ (cont'd)

$\underline{r}_2''$ Entry interface position from Precision Required Velocity Determination Routine

$\underline{r}_D$ Position of the impulsive deorbit maneuver

$\underline{r}_{EI}$ Entry interface position

$\underline{r}_{ig}$ Position vector at ignition

$\underline{r}_{LS}$ Estimated landing site position at the time of landing

$\underline{r}_{PFT}$ Powered flight offset target vector

s_{eng} Engine select switch

s_{fail} Switch set to indicate non-convergence of Precision Required Velocity Determination Routine

s_{FP} Switch set equal to one after the first pass through step one

s_{pert} Switch indicating which perturbations are to be included in the Precision State and Filter Weighting Matrix Extrapolation Routine (See Reference 5)

s_{proj} Switch set when the target vector must be projected into the plane defined by $\underline{i}_N$

t_0 Precision state vector time

t_1 Time of impulsive deorbit maneuver

t_2 Time-of-arrival at entry interface

t_3 Estimated time at which in-orbit position vector is coincident with the landing site projection into the orbital plane

9.10.1 Deorbit Targeting (cont'd)

t_{ETL}	Desired earliest time-of-landing
t_{ig}	Ignition time
t_L	Estimated time-of-landing
t_{LTL}	Desired latest time-of-landing
$\underline{v}_0$	Precision velocity vector
$\underline{v}''_2$	Entry interface velocity from Precision Required Velocity Determination Routine
$\underline{v}_D$	Pre-impulse velocity
$\underline{v}_{EI}$	Entry interface velocity
$\underline{v}_{ig}$	Ignition velocity vector
$\underline{v}_{PFT}$	Velocity associated with the powered flight offset target vector
v_{RD}	Post-impulse radial component of velocity
$\underline{v}_{req}$	Required velocity
$\underline{v}'_{req}$	Required velocity on the coasting trajectory
v_{HD}	Post-impulse horizontal component of velocity
δt_{01}	Adjustment to Δt_{01}
Δd_{ACR}	Increment added to the acceptable crossrange, used in rough crossrange check
Δp_γ	Difference between the predicted and desired p_γ

9.12.1 Deorbit Targeting (cont'd)

Δr_{proj}	Out-of-plane target miss due to the projection of the target vector
Δt_{01}	Transfer time $(t_1 - t_0)$
Δt_{12}	Transfer time $(t_2 - t_1)$
Δt_{23}	Transfer time $(t_3 - t_2)$
Δt_B	Estimated duration of the powered maneuver
Δt_{DE}	Time-of-flight difference between (1) the interval from deorbit through entry to landing, and (2) the time spent in orbit over the same total central angle
Δt_{IP}	Time-of-flight required to transfer through the central angle θ_{IP}
$\Delta \underline{v}$	Required velocity change
Δv_P	Previous value of $\lvert \Delta \underline{v} \rvert$
$\Delta \theta$	Increment in in-plane angle θ_D
$\Delta \theta_0$	Initial increment in in-plane angle θ_D
$\epsilon_{p\gamma}$	Convergence criterion on Δp_γ
ϵ_θ	Convergence criterion on angle $\Delta \theta$
γ_1	Post-impulse flight path angle
γ_{EI}	Desired entry flight path angle measured from the horizontal
λ_{LS}	Landing site longitude
θ	In-plane angle between precision state vector and entry interface

9.12.1 Deorbit Targeting (cont'd)

θ_{01} In-plane angle between precision state vector and deorbit position

θ_D In-plane angle over which search is made to find minimum deorbit Δv

θ_E In-plane central angle traversed during entry

θ_{IP} Angle between precision state vector and the projection of the landing site into the orbital plane

θ_P Previous value of θ_D

μ Gravitational parameter of the earth (product of the earth's mass and universal gravitation constant)

ϕ_{LS} Landing site latitude

τ Orbital period

2. FUNCTIONAL FLOW DIAGRAM

A functional flow diagram presenting the basic approach to the deorbit targeting problem can be found in Figure 3. In addition to the state vector, the primary inputs to the routine are the landing site location (latitude and longitude), the entry downrange distance, the entry angle (at 400,000 ft) and the earliest desired time of landing. Since the high crossrange capability may make deorbit possible on two or more consecutive orbits, after each solution the crew has the option to recycle the program to determine the next possible deorbit opportunity. To give the crew the flexibility to evaluate solutions in the future without stepping through all earlier opportunities, the earliest desired time-of-landing is included as an input. However, the vehicle is assumed to be in coasting flight until the deorbit maneuver, and therefore the effects of any maneuvers prior to deorbit are not accounted for.

After the vehicle state vector is extrapolated forward to the earliest desired time-of-landing, the solution process is initiated. This consists of three major steps. During the first step the vehicle state is further advanced until the landing site, which rotates with the earth, lies sufficiently near the orbital plane so that it is within the crossrange (or out-of-plane) capability of the entry phase. During the next step an iterative process is used to select the ignition time for this deorbit opportunity which requires the smallest velocity change, thus minimizing the fuel expenditure. Since the first two steps involve several conic approximations to minimize the computer time used, the third step fine tunes the solution by generating a precision trajectory which satisfies the constraint on the desired entry angle while accounting for gravitational perturbations and the non-impulsive nature of the deorbit maneuver. After completion of this step the results are displayed to the crew. They may then elect to accept the solution, recycle the routine to solve for the next deorbit opportunity, or exit. If they accept the solution, a few minor computations are required to initialize the Powered Flight Guidance Routines for a modified Lambert aimpoint maneuver.

To aid the reader in understanding the functional flow diagram, each of the three major steps in the solution process is discussed in more detail below.

2.1 Determination of the Next Deorbit Opportunity (Step 1)

To determine the next possible deorbit opportunity, it is necessary to calculate the inertial location of the landing site (which rotates with the earth) at the time-of-landing. Then the angle between the orbital plane and the landing site can be used to estimate the crossrange required during entry. To accomplish this, an estimate of the time-of-flight difference Δt_{DE} between (1) the interval from deorbit through entry to landing, and (2) the time spent in orbit over the same total central angle is used. Analysis has shown that a constant is probably adequate to represent this difference since more precise calculations in the following step will compensate for any error.

Upon completion of the initialization process, the state vector is extrapolated forward to the earliest desired time-of-landing. Then the inertial location of the landing site at the present state vector time, biased by the time difference Δt_{DE}, is computed. This landing site vector is projected into the orbital plane, allowing the in-plane central angle θ_{IP} between the vehicle position and the projection of the landing site to be determined.

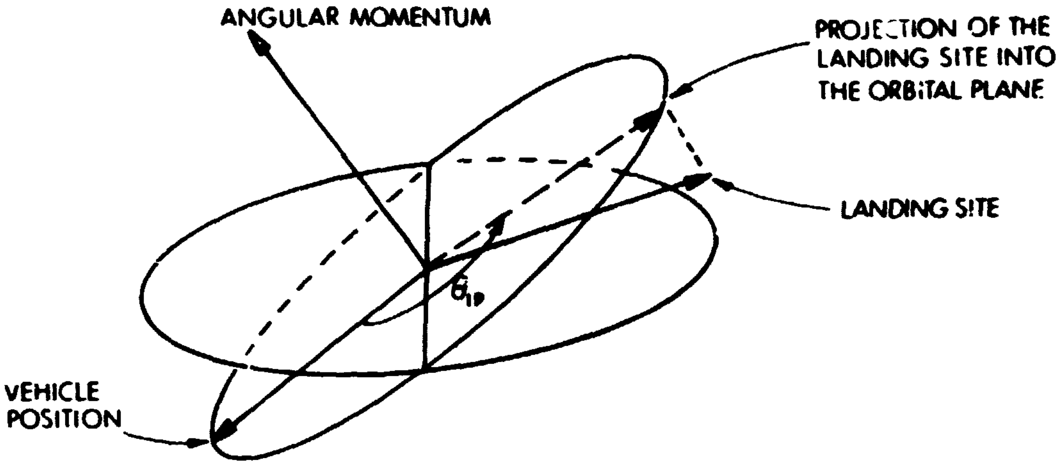


Figure 1. Out-of-plane Geometry

The conic routines can now be used to determine the time-of-flight Δt_{IP} required to coast in orbit through the central angle θ_{IP}. If the state was then propagated through this central angle, its position vector would be aligned with the previously determined projection of the landing site vector. Unfortunately, the landing site will move slightly due to earth rotation while the vehicle transfers through the central angle. Therefore, the inertial location of the landing site must be recomputed, accounting for the time difference Δt_{DE} explained previously. Thus, an iterative process is required to precisely determine the location of the landing site at the expected time-of-landing. During the first pass through the deorbit targeting routine, the previously described steps are repeated once to insure convergence. However, on subsequent passes no iteration is required, since the initial guess achieved by extrapolating the state vector one orbit beyond the previous solution guarantees a small value for the time-of-flight correction Δt_{IP}.

Assuming the deorbit maneuver is in-plane, the angle between the orbital plane and the landing site location at the estimated time-of-landing can be used to measure the crossrange required during the entry phase. If the crossrange is within the capability of the vehicle, the solution process continues on to the next step. If not, the vehicle state is extrapolated forward one revolution to the next potential deorbit opportunity and the process of estimating the crossrange is repeated.

It should be noted that the process used to determine the crossrange requirement is only approximate, and therefore a small increment is added to the tolerance used in the crossrange check to allow for this. A small number of cases which pass this check will actually lie outside the vehicle crossrange capability, however, a more precise check later will screen these out.

2.2 Ignition Time Selection (Step 2)

During this step in the solution process, an ignition time is selected which minimizes the impulsive velocity change required. For these computations the projection of the landing site into the orbital plane is assumed to be the real landing site. Then, based upon the desired entry downrange distance, a target position at entry interface which also lies in the orbital plane can be defined. This target position is set 400,000 ft above the Fischer ellipsoid.

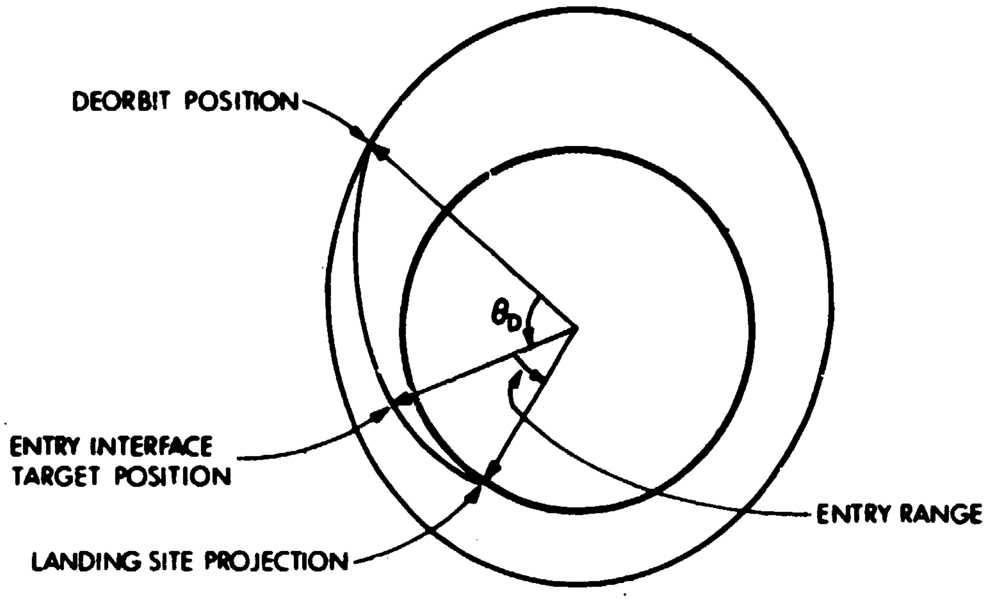


Figure 2. In-plane Geometry

Using this entry interface target, and the desired entry flight path angle, a search is made on the central angle θ_D traversed between the deorbit maneuver and entry interface to locate the position and time of the minimum Δv maneuver.

Then the time-of-flight required for the deorbit and entry phases can be accurately determined. Using this time-of-flight, an accurate calculation of the inertial location of the landing site at the time-of-landing can be made, and the entry interface target can also be updated. Tc preserve the central angle of the deorbit phase, the impulsive maneuver time is adjusted. Then the ignition time is biased from the impulsive time by half the expected length of the maneuver and the state vector is extrapolated to this time.

Since the location of the landing site at the time-of-landing is now known accurately, the angle between the c bital plane and the landing site is recomputed to precisely measure the entry crossrange required. Then a precision check is made, and any solution exceeding the crossrange capability is rejected, thus returning the routine to step one to search for the next opportunity.

2. 3 Precision Solution (Step 3)

During this step a precision integrated trajectory from deorbit to entry interface is generated which accounts for both the finite length of the thrusting maneuver and the effects of gravitational perturbations. Since the time-of-flight from deorbit to entry interface is known, the Precision Required Velocity Determination Routine can be used to generate this trajectory. However, the effects of conic approximations in the previous steps and the finite length of the maneuver can cause significant error in the reentry angle. Therefore, the resulting entry angle is checked and if it is in error, a slight modification is made in the time-of-flight from the deorbit maneuver to entry interface to adjust the entry angle. Then the precision trajectory is recomputed. After satisfying the flight path angle constraint, pertinent data relating to the maneuver can be displayed to the crew or transferred to the Mission Planning Module.

9.12.1 Deorbit Targeting (cont'd)

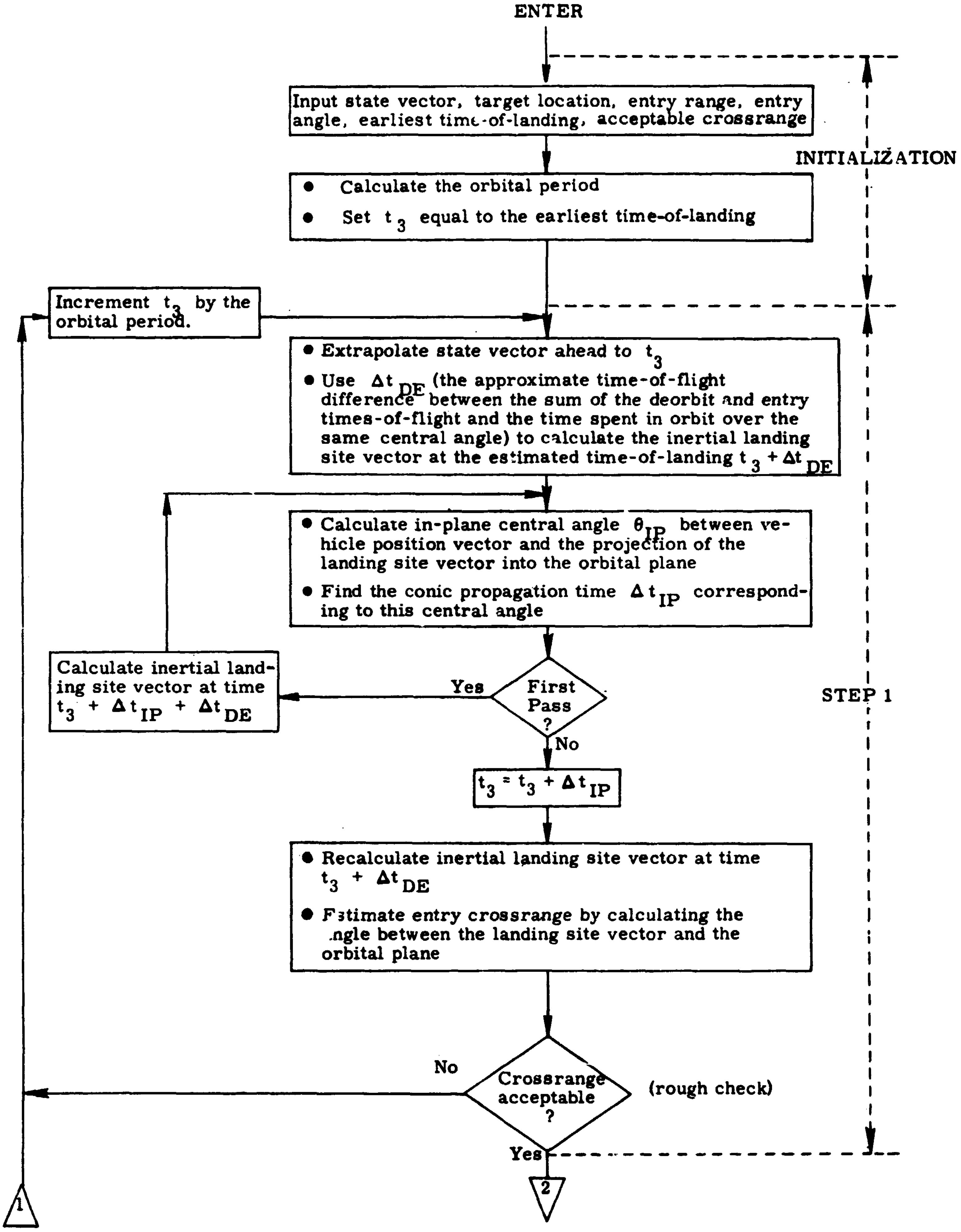
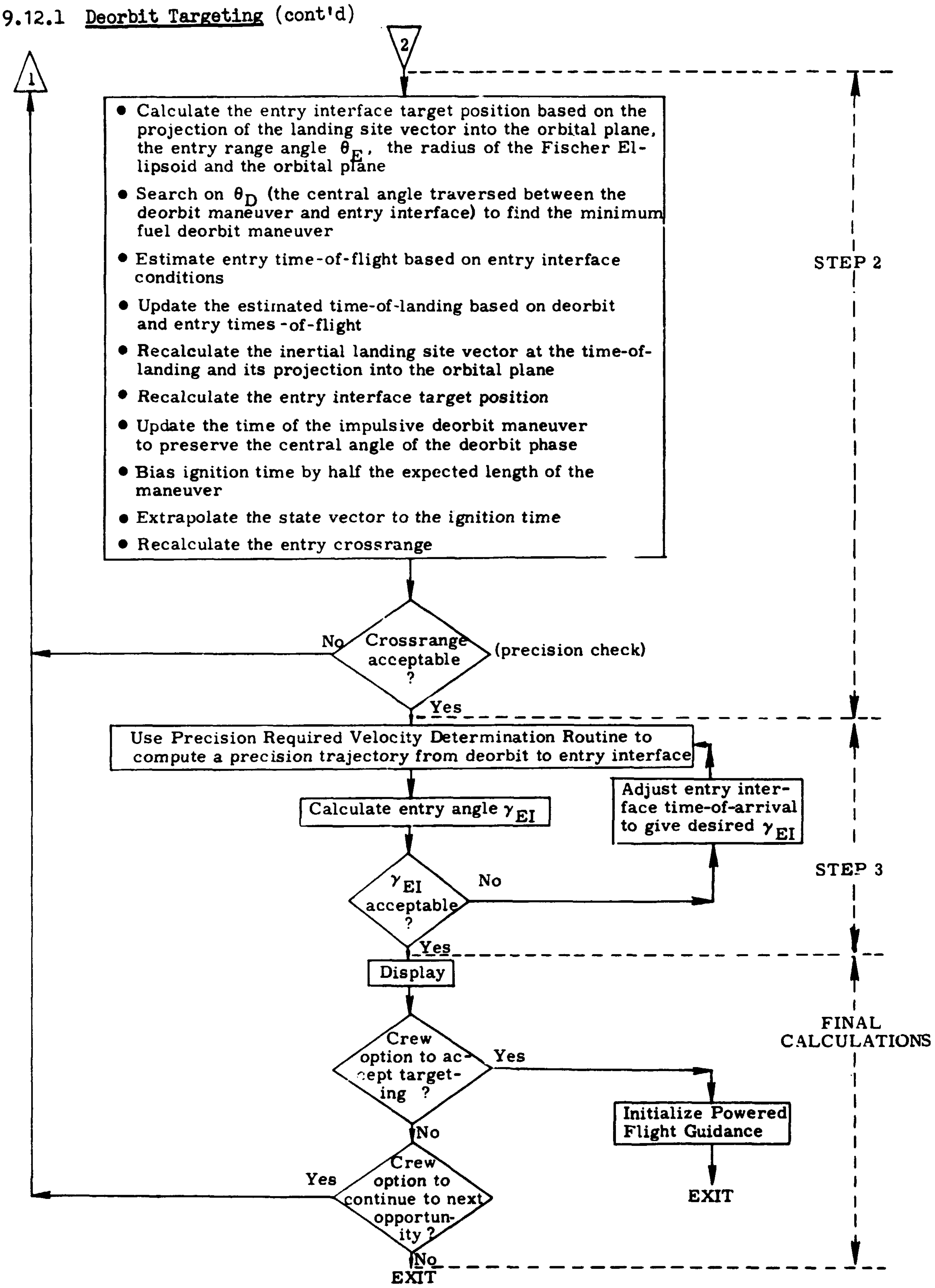

ENTER
Input state vector, target location, entry range, entry angle, earliest timc-of-landing, acceptable crossrange
INITIALIZATION
● Calculate the orbital period
● Set t_3 equal to the earliest time-of-landing
Increment t_3 by the orbital period.
● Extrapolate state vector ahead to t_3
● Use Δt_{DE} (the approximate time-of-flight difference between the sum of the deorbit and entry times-of-flight and the time spent in orbit over the same central angle) to calculate the inertial landing site vector at the estimated time-of-landing $t_3 + \Delta t_{DE}$
● Calculate in-plane central angle θ_{IP} between vehicle position vector and the projection of the landing site vector into the orbital plane
● Find the conic propagation time Δt_{IP} corresponding to this central angle
Calculate inertial landing site vector at time $t_3 + \Delta t_{IP} + \Delta t_{DE}$
Yes
First Pass ?
No
STEP 1
$t_3 = t_3 + \Delta t_{IP}$
● Recalculate inertial landing site vector at time $t_3 + \Delta t_{DE}$
● Fstimate entry crossrange by calculating the .ngle between the landing site vector and the orbital plane
No
Crossrange acceptable ?
(rough check)
Yes
2
1

Figure 3a. Functional Flow Diagram

9.12.1 Deorbit Targeting (cont'd)



2
1
• Calculate the entry interface target position based on the projection of the landing site vector into the orbital plane, the entry range angle θ_E, the radius of the Fischer Ellipsoid and the orbital plane
• Search on θ_D (the central angle traversed between the deorbit maneuver and entry interface) to find the minimum fuel deorbit maneuver
• Estimate entry time-of-flight based on entry interface conditions
• Update the estimated time-of-landing based on deorbit and entry times-of-flight
• Recalculate the inertial landing site vector at the time-of-landing and its projection into the orbital plane
• Recalculate the entry interface target position
• Update the time of the impulsive deorbit maneuver to preserve the central angle of the deorbit phase
• Bias ignition time by half the expected length of the maneuver
• Extrapolate the state vector to the ignition time
• Recalculate the entry crossrange
STEP 2
No
Crossrange acceptable ?
(precision check)
Yes
Use Precision Required Velocity Determination Routine to compute a precision trajectory from deorbit to entry interface
Calculate entry angle γ_{EI}
Adjust entry interface time-of-arrival to give desired γ_{EI}
γ_{EI} acceptable ?
No
Yes
STEP 3
Display
Crew option to accept targeting ?
Yes
Initialize Powered Flight Guidance
EXIT
FINAL CALCULATIONS
No
Yes
Crew option to continue to next opportunity ?
No
EXIT

Figure 3b. Functional Flow Diagram

3. INPUT AND OUTPUT VARIABLES

Input Variables

t_0	Time associated with the vehicle state vector
$\underline{r}_0$, $\underline{v}_0$	State vector
t_{ETL}	Earliest desired time-of-landing
t_{LTL}	Latest desired time-of-landing
m	Current estimated vehicle mass
s_{eng}	Engine select switch
ϕ_{LS}	Landing site latitude
λ_{LS}	Landing site longitude
γ_{EI}	Desired inertial entry angle
d_{DR}	Desired entry downrange distance
d_{ACR}	Maximum acceptable crossrange distance

Output Variables

t_{ig}	Ignition time
t_2	Time of arrival at entry interface and time associated with offset target
$\underline{r}_{PFT}$	Offset target vector
$p_{PF\gamma}$	Parameter defining the desired conic entry angle
n_{rev}	Integral number of complete 360° revolutions (n_{rev} = 0 for deorbit)
$\underline{i}_N$	Unit normal to transfer plane in direction of angular momentum vector
s_{proj}	Switch indicating whether the initial and target vectors are to be projected into the plane defined by the unit normal $\underline{i}_N$

4. DESCRIPTION OF EQUATIONS

To minimize the size of the Deorbit Targeting Routine, extensive use is made of other routines. Therefore, this routine consists primarily of simple equations, logical operations, and calls to other routines. Since most of the complicated equations requiring detailed explanation are contained in the description of the other routines, this section will be limited to a list of items not covered in the text describing the functional flow diagram. These items will be listed in their order of occurence, and are intended to supplement the detailed flow diagram in subsection 5.

4.1 Selection of Perturbing Acceleration during Precision State Extrapolation

During the first step in the solution process, which may require long term state vector extrapolation, it is desirable to maximize accuracy by including all significant perturbing accelerations in the extrapolation process. Therefore, the switch s_{pert}, which controls the selection of perturbing accelerations in the Precision State Extrapolation Routine, is set to 2. During the later portion of the routine, referred to as step three, the switch is reset to 1, thus limiting the disturbing acceleration to the J_2 term, the second harmonic of the earth's gravitational potential function. Since extrapolation during step three is limited to the interval from the deorbit maneuver to entry interface, the effects of smaller perturbing accelerations are not significant. In addition, extrapolation over this interval lies within an iterative loop, and thus may be repeated several times. The simplified model can therefore significantly reduce the running time of this step.

4.2 Selection of θ_{IP} Quadrant

During the discussion of the functional flow diagram, it was mentioned that successive solutions to the deorbit problem (when successive solutions exist) are about one revolution apart. To find succeeding solutions to the problem, the state vector is extrapolated forward one revolution and then the in-plane central angle θ_{IP} between the state vector and the projection of the landing site into the orbital plane is computed. Analysis has shown that for some selections of orbital inclination and landing site, the correction to the assumption of one revolution may be as large as 29^o. A lower limit on θ_{IP} of -30^o was chosen, thus allowing a small margin from the empirically determined limit of -29^o. The upper limit on θ_{IP} is $+330^o$. Large positive values for θ_{IP} only occur in situations where no solution existed on the previous revolution.

To determine θ_{IP}, the following equation is used,

$$\theta_{IP} = \cos^{-1}\left[\text{unit}\,(\underline{r}_0)\cdot\underline{i}'_{LSP}\right]\,\text{sign}\left[(\underline{r}_0 \times \underline{i}'_{LSP})\cdot\underline{i}_h\right]$$

where

$\underline{r}_0$ = vehicle position vector

$\underline{i}'_{LSP}$ = unit vector in the direction of the landing site projection

$\underline{i}_h$ = unit angular momentum vector

This places θ_{IP} between -180° and +180° and therefore an additional test, shown in Figure 4b, is made to force θ_{IP} between -30° and +330°.

In order to make the first entry into step one compatible with subsequent entries, the state vector is initially extrapolated forward beyond the earliest desired time-of-landing t_{ETL} by one-twelfth of the orbital period, to the time t_3, where

$$t_3 = t_{ETL} + \tau/12$$

One-twelfth of the period is nearly equivalent to a central angle of 30° for typical (near circular) orbits, and hence makes the first entry into step one compatible with later entries.

4.3 Effect of Approximate Entry and Deorbit Times-of-Flight on Entry Crossrange Calculation

During the first step in the solution process, an estimate of the time of landing is necessary to compute the inertial location of the landing site and the associated entry crossrange. Since the parameters of the deorbit trajectory have not been computed, the deorbit and entry times-of-flight are not known. To estimate the landing time, a constant Δt_{DE} is used to approximately represent the difference between the sum of the deorbit and entry times-of-flight and the time spent in orbit over the same total central angle. Preliminary analysis has shown that if an average value is selected for this time difference, the maximum error will be about 6 minutes. This analysis, described in Reference 7, did not include · ariations in entry time-of-flight for the particular entry range, but further analysis is ex ected to show this effect is small.

During the first step in the solution, this error will affect the calculation of the inertial landing site vector and subsequent entry crossrange computation. This effect on the crossrange estimate will be largest for deorbit from a polar orbit, and result in a maximum error of less than 90 n.mi. To insure that potentially acceptable solutions are not rejected due to errors in the initial crossrange estimate, the rough check on crossrange during the first step uses a test criterion 90 n.mi. larger than the acceptable crossrange input to the routine. In step two, after the time-of-landing has been refined, the crossrange is recomputed and a precision check is made. Thus a few cases which pass the first test will be rejected later.

4.4 Velocity Change Minimization Method

Step two of the routine includes an iterative search to determine the location of the impulsive maneuver which minimizes the velocity change Δv. As shown in Figure 4d, this iteration uses θ_D, the central angle traversed between the impulsive maneuver and entry interface, as the independent variable. A very simple halving step iterator is used to search for the minimum. Although this does not converge quickly, it is safe and reliable. The more efficient technique of using a slope iteration was not selected because analysis has shown that inflection points exist in the relationship of Δv and θ_D. These inflection points would greatly complicate any iteration designed to determine the minimum by driving the slope to zero.

4.5 Required Velocity Equations

The equations used in the previously described iterative loop to determine the required velocity can be found in Reference 2. These equations, shown in Figure 4d of the detailed flow diagram, use the initial vehicle position $\underline{r}_D$, the entry interface position $\underline{r}_{EI}$, and the desired entry angle γ_{EI} as follows. First the tangent of the initial (post-impulse) flight path angle γ_1 is computed by

$$\tan \gamma_1 = (1 - r_D/r_{EI}) \cot(\theta_D/2) - r_D/r_{EI} \tan(\gamma_{EI})$$

where θ_D is the central angle between $\underline{r}_D$ and $\underline{r}_{EI}$ and also the independent variable in the search. The semilatus rectum p_D of the deorbit trajectory can then be determined from

$$p_D = \frac{2 r_D (r_D/r_{EI} - 1)}{(r_D/r_{EI})^2 \, p_\gamma - (1 + \tan \gamma_1{}^2)}$$

The parameter p_γ, the secant squared of the desired entry angle, is computed once during initialization of the routine.

The horizontal and radial components of the required velocity are then obtained from

$$v_{HD} = \sqrt{\mu \, p_D} / r_D$$
$$v_{RD} = v_{HD} \tan \gamma_1$$

The required velocity is then formed and differenced with the premaneuver velocity to obtain the impulsive Δv.

$$\underline{v}_{req} = \underline{v}_{RD} \text{ unit } (\underline{r}_D) + v_{HD} \text{ unit } \left[(\underline{r}_D \times \underline{v}_D) \times \underline{r}_D\right]$$
$$\Delta \underline{v} = \underline{v}_{req} - \underline{v}_D$$

4.6 Entry Time-of-Flight Computation (TBD)

In Figure 4e of the detailed flow diagram, the time-of-flight Δt_{23} from entry interface to landing is shown as a function of entry velocity, flight path angle, and range. Functionalization of this time-of-flight will be included later when entry guidance analysis is complete.

4.7 In-Plane Effect of Approximate Deorbit and Entry Times-of-Flight

As discussed in subsection 4.3, the first estimate of the inertial location of the landing site is dependent upon an estimate of the time-of-landing. A constant time difference Δt_{DE}, used to estimate the landing time, may be in error by as much as 6 minutes. This led to a significant error in the crossrange estimate for a high inclination orbit. For orbits of lower inclination, where the movement of the landing site can be nearly parallel to the orbital plane, this same error can affect the definition of the entry interface location used in the Δv minimization iteration.

The entry interface location, computed early in step two, is based upon the projection of the landing site vector into the orbital plane and the desired entry range. After the minimization process is complete, the deorbit and entry times-of-flight can be accurately calculated. As shown in Figure 4e, another calculation of the inertial landing site position is made, thus removing the error due to the Δt_{DE} approximation. To maintain the desired entry range input to the routine, the entry interface position is recalculated. This new position will be, at most, 1.5° (equivalent to 6 minutes of earth rotation) from the entry interface used in the Δv minimization. To maintain the geometry of the deorbit phase, the time of the deorbit maneuver is adjusted accordingly so that the central angle from deorbit to entry interface is preserved. This adjustment in deorbit time δt_{01} is computed from the following equation

$$\delta t_{01} = \left[\left(\underline{i}'_{EI} \times \underline{i}_{EI}\right) \cdot \underline{i}_h\right] \frac{\tau}{2\pi}$$

where $\underline{i}'_{EI}$ is a unit vector in the direction of the entry interface position used during minimization, $\underline{i}_{EI}$ is the new value, $\underline{i}_h$ is a unit angular momentum vector, and $\tau/2\pi$ is the inverse of the mean orbital rate. The cross product of the unit vectors is nearly equivalent to the angle between them, and the dot product gives the proper sign. The mean orbital rate is used to calculate the deorbit time adjustment from the angular adjustment. Following this adjustment to the impulsive deorbit time, the ignition time for the maneuver is biased from the impulsive time by one-half the expected length of the maneuver, thus centering the finite thrust maneuver about the impulsive maneuver.

4.8 Compensation for Oblateness and Finite Maneuver Length

Step three of the solution process contains calculations which account for the finite length of the thrusting maneuver on the required velocity change, and compensate for the effects of the J_2 gravitational perturbation on the deorbit trajectory. The Precision Required Velocity Determination Routine is used to accomplish these objectives, and the reader should refer to Reference 1 for a description of the technique. That routine, however, is designed to maintain the terminal (entry interface) time-of-arrival, and this can cause changes in the entry angle. Preliminary analysis, described in Reference 7, has shown that the nominal entry flight path angle error resulting from the oblateness and finite maneuver length is about 0.2°, but can be as large as 0.6° in extreme cases. Therefore, to preserve the desired entry angle, the time-of-arrival at entry interface is adjusted slightly. Delaying the time-of-arrival tends to loft the trajectory and thus increase the entry angle. An earlier time-of-arrival will depress the trajectory and result in a shallower flight path angle.

To determine the time-of-arrival adjustment, the approximate sensitivity of changes in time-of-flight to changes in entry angle is used. Analysis has shown that this sensitivity varies by a factor of about 13, depending on the characteristics of the pre-maneuver trajectory. However, the sensitivity divided by the deorbit time-of-flight varies by a factor of less than 3. This variation is sufficiently small such that a constant can be used as the sensitivity coefficient for all cases.

To reduce the computations required to constrain entry angle, both here and in the Powered Flight Guidance Routines*, the secant squared of the entry angle p_γ is used rather than the actual angle. In particular, no inverse trigonometric function evaluations are required.

The sequence of calculations designed to reduce the entry angle error are shown in Figures 4f and 4g. First the error Δp_γ in the secant squared of the entry flight path angle is computed from the following equation:

$$\Delta p_\gamma = \frac{1}{1 - \left[\text{unit}\,(\underline{r}''_2) \cdot \text{unit}\,(\underline{v}''_2)\right]^2} - p_\gamma$$

where $\underline{r}''_2$ and $\underline{v}''_2$ are the terminal position and velocity determined by the Precision Required Velocity Determination Routine and p_γ is the desired value. If the error is too large, the entry interface time-of-arrival t_2 is adjusted as follows:

$$t_2 = t_2 - k_\gamma \, \Delta t_{12} \, \Delta p_\gamma$$

*The Powered Flight Guidance Routines, described in Reference 3, use the same basic technique described here to maintain entry angle in the event of off-nominal thrusting conditions.

where k_γ is the sensitivity coefficient described earlier and Δt_{12} is the time-of-flight from deorbit to entry interface. After adjusting the time-of-arrival, the Precision Required Velocity Determination Routine is recalled with the adjusted time-of-arrival and the results are checked.

4.9 Offset Entry Angle

In the process of computing a required velocity, the Precision Required Velocity Determination Routine computes an offset target for use during the powered flight. For the deorbit maneuver, the powered flight guidance also requires an offset entry angle. This offset entry angle, actually the secant squared of the angle, is computed from the following equation

$$p_{PF\gamma} = \frac{1}{1 - \left[\text{unit} (\underline{r}_{PFT}) \cdot \text{unit} (\underline{v}_{PFT}) \right]^2}$$

where $\underline{r}_{PFT}$ is the offset target for the powered flight guidance and $\underline{v}_{PFT}$ is the associated velocity.

[redacted]

5. DETAILED FLOW DIAGRAM

This section contains detailed flow diagrams of the Deorbit Targeting Routine.

Each input and output variable in the routine and subroutine call statements can be followed by a symbol in brackets. This symbol identifies the notation for the corresponding variable in the detailed description and flow diagrams of the called routine. When identical notation is used, the bracketed symbol is omitted.

9.12.1 Deorbit Targeting (cont'd)

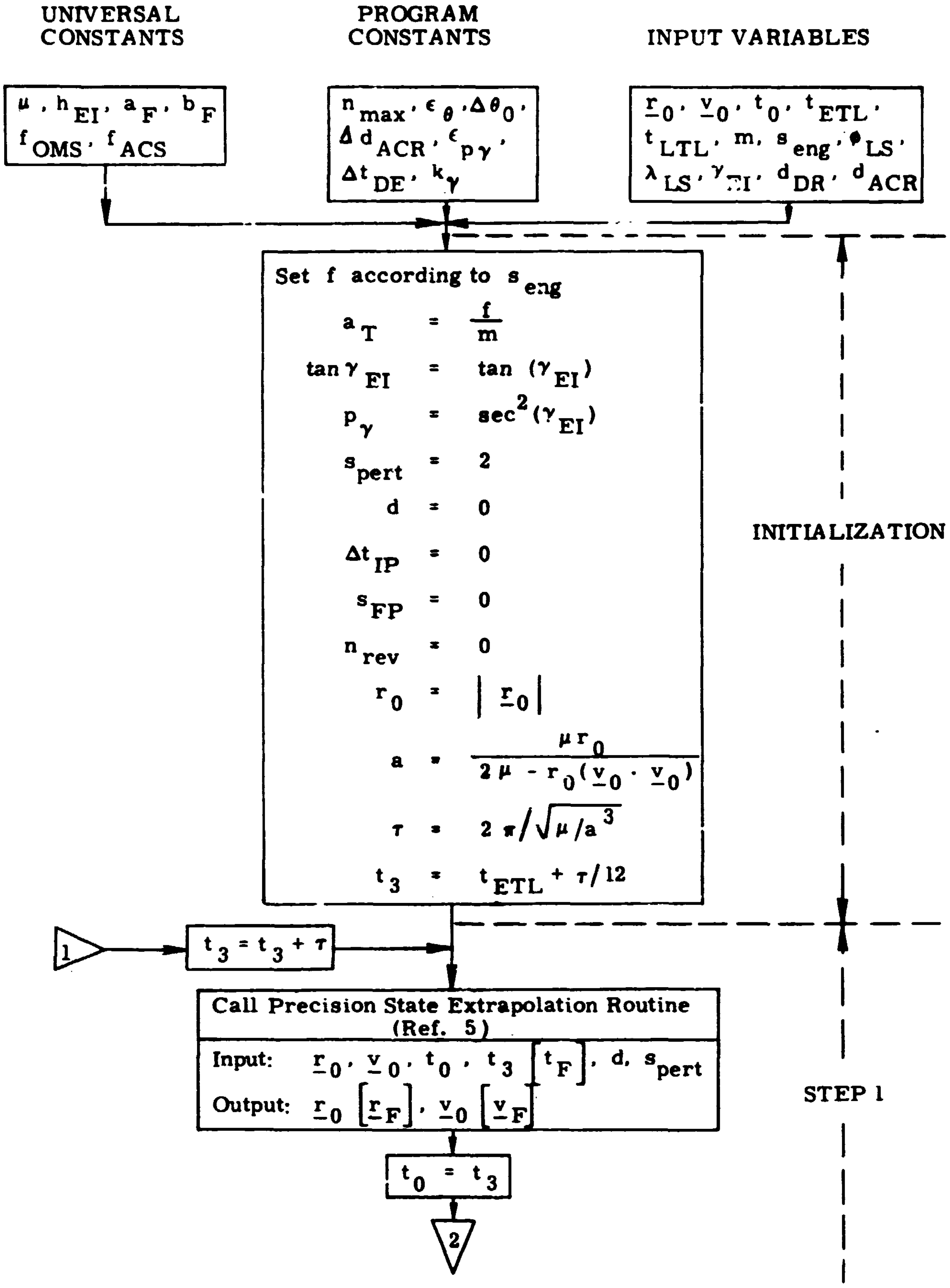

UNIVERSAL CONSTANTS
PROGRAM CONSTANTS
INPUT VARIABLES
μ, h_{EI}, a_F, b_F f_{OMS}, f_{ACS}
n_{max}, ϵ_θ, $\Delta\theta_0$, Δd_{ACR}, $\epsilon_{p\gamma}$, Δt_{DE}, k_γ
$\underline{r}_0$, $\underline{v}_0$, t_0, t_{ETL}, t_{LTL}, m, s_{eng}, ϕ_{LS}, λ_{LS}, γ_{EI}, d_{DR}, d_{ACR}
Set f according to s_{eng}
$a_T = \frac{f}{m}$
$\tan\gamma_{EI} = \tan(\gamma_{EI})$
$p_\gamma = \sec^2(\gamma_{EI})$
$s_{pert} = 2$
$d = 0$
$\Delta t_{IP} = 0$
$s_{FP} = 0$
$n_{rev} = 0$
$r_0 = |\underline{r}_0|$
$a = \frac{\mu r_0}{2\mu - r_0(\underline{v}_0 \cdot \underline{v}_0)}$
$\tau = 2\pi/\sqrt{\mu/a^3}$
$t_3 = t_{ETL} + \tau/12$
INITIALIZATION
1
$t_3 = t_3 + \tau$
Call Precision State Extrapolation Routine (Ref. 5)
Input: $\underline{r}_0$, $\underline{v}_0$, t_0, t_3 $[t_F]$, d, s_{pert}
Output: $\underline{r}_0$ $[\underline{r}_F]$, $\underline{v}_0$ $[\underline{v}_F]$
$t_0 = t_3$
2
STEP 1

Figure 4a. Detailed Flow Diagram

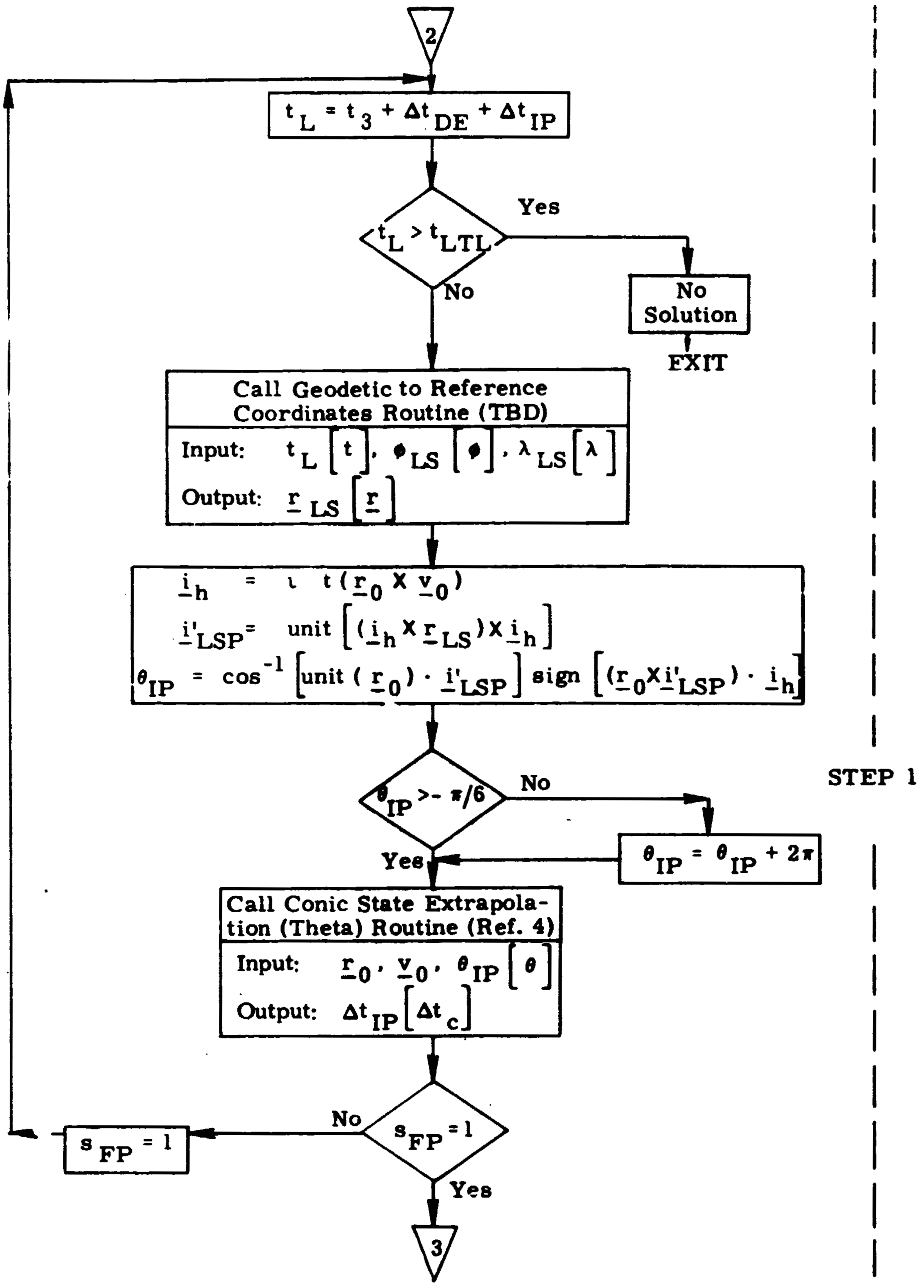

2
$t_L = t_3 + \Delta t_{DE} + \Delta t_{IP}$
$t_L > t_{LTL}$
Yes
No
No Solution
EXIT
Call Geodetic to Reference Coordinates Routine (TBD)
Input: $t_L \left[t\right], \phi_{LS} \left[\phi\right], \lambda_{LS} \left[\lambda\right]$
Output: $\underline{r}_{LS} \left[\underline{r}\right]$
$\underline{i}_h = \iota \; t(\underline{r}_0 \times \underline{v}_0)$
$\underline{i}'_{LSP} = \text{unit} \left[(\underline{i}_h \times \underline{r}_{LS}) \times \underline{i}_h\right]$
$\theta_{IP} = \cos^{-1} \left[\text{unit}(\underline{r}_0) \cdot \underline{i}'_{LSP}\right] \text{sign} \left[(\underline{r}_0 \times \underline{i}'_{LSP}) \cdot \underline{i}_h\right]$
$\theta_{IP} > -\pi/6$
No
$\theta_{IP} = \theta_{IP} + 2\pi$
Yes
Call Conic State Extrapolation (Theta) Routine (Ref. 4)
Input: $\underline{r}_0, \underline{v}_0, \theta_{IP} \left[\theta\right]$
Output: $\Delta t_{IP} \left[\Delta t_c\right]$
No
$s_{FP} = 1$
$s_{FP} = 1$
Yes
3
STEP 1

Figure 4b. Detailed Flow Diagram

9.12.1 Deorbit Targeting (cont'd)

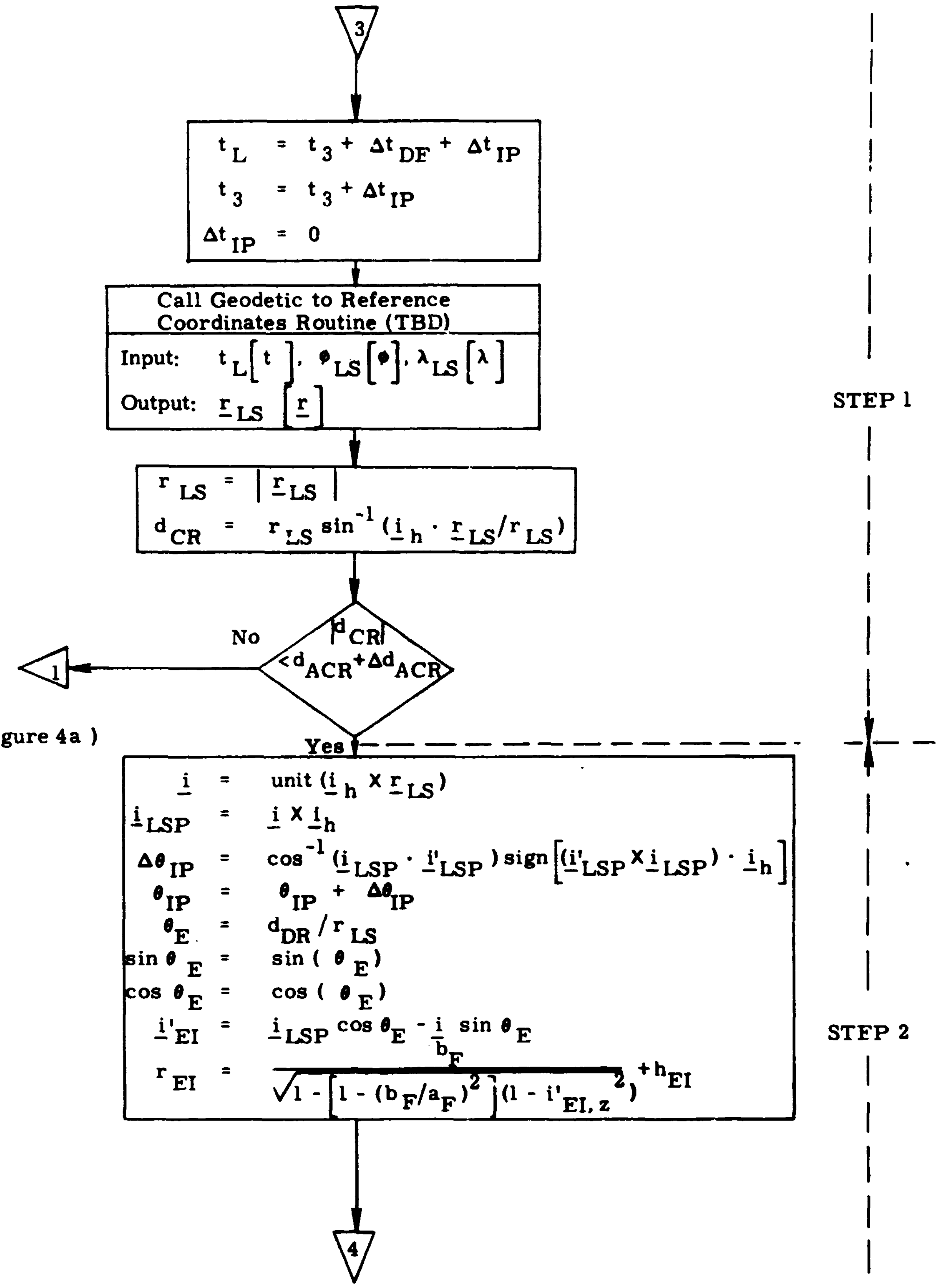
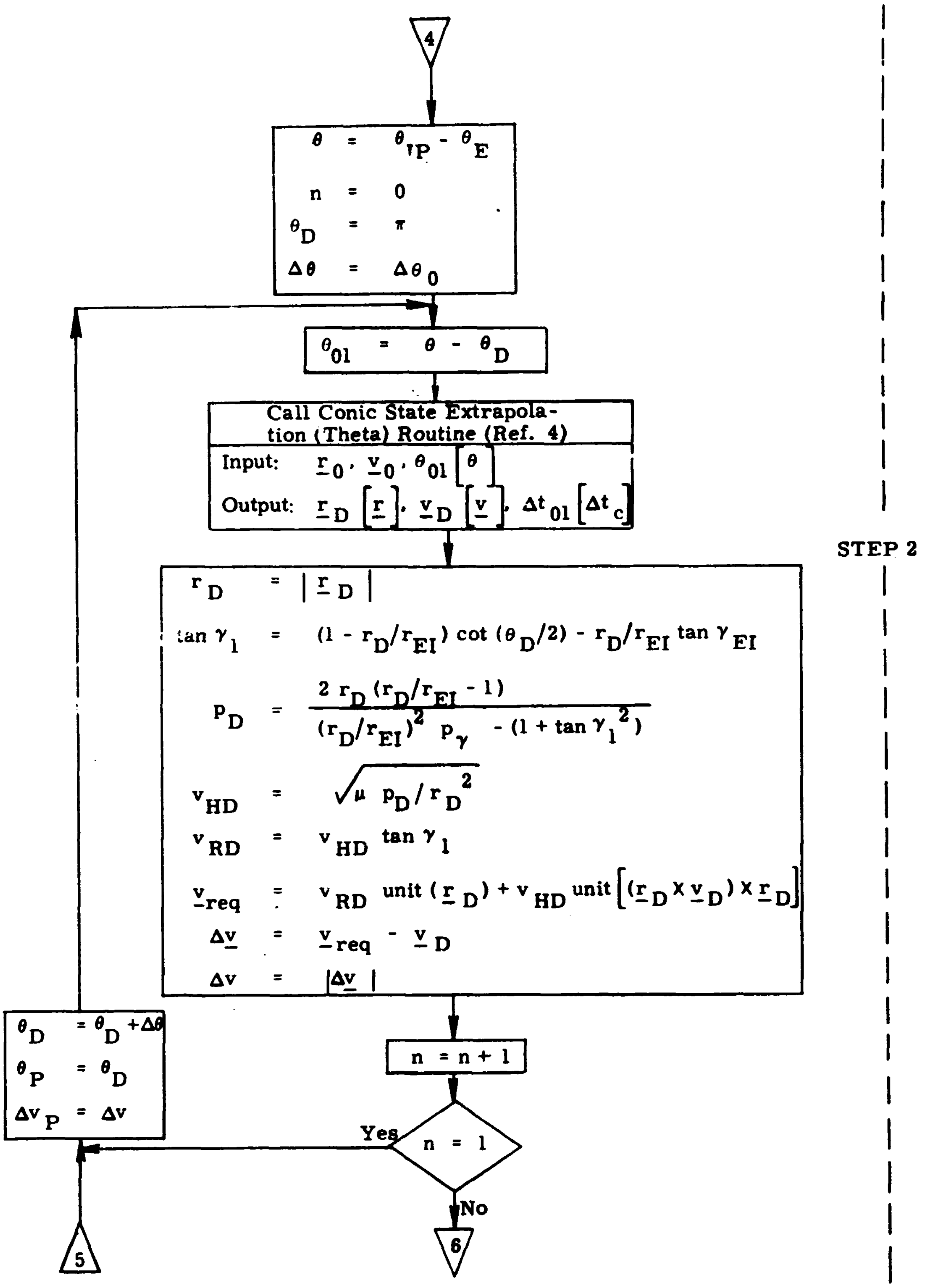

3
$t_L = t_3 + \Delta t_{DF} + \Delta t_{IP}$
$t_3 = t_3 + \Delta t_{IP}$
$\Delta t_{IP} = 0$
Call Geodetic to Reference Coordinates Routine (TBD)
Input: $t_L[t]$, $\phi_{LS}[\phi]$, $\lambda_{LS}[\lambda]$
Output: $\underline{r}_{LS}$ $[\underline{r}]$
$r_{LS} = |\underline{r}_{LS}|$
$d_{CR} = r_{LS} \sin^{-1}(\underline{i}_h \cdot \underline{r}_{LS}/r_{LS})$
$|d_{CR}| < d_{ACR} + \Delta d_{ACR}$
No
1
(Figure 4a)
Yes
$\underline{i} = \text{unit}(\underline{i}_h \times \underline{r}_{LS})$
$\underline{i}_{LSP} = \underline{i} \times \underline{i}_h$
$\Delta\theta_{IP} = \cos^{-1}(\underline{i}_{LSP} \cdot \underline{i}'_{LSP}) \text{ sign}\left[(\underline{i}'_{LSP} \times \underline{i}_{LSP}) \cdot \underline{i}_h\right]$
$\theta_{IP} = \theta_{IP} + \Delta\theta_{IP}$
$\theta_E = d_{DR} / r_{LS}$
$\sin\theta_E = \sin(\theta_E)$
$\cos\theta_E = \cos(\theta_E)$
$\underline{i}'_{EI} = \underline{i}_{LSP} \cos\theta_E - \underline{i} \sin\theta_E$
$r_{EI} = \frac{b_F}{\sqrt{1 - \left[1 - (b_F/a_F)^2\right](1 - i'^2_{EI,z})}} + h_{EI}$
4
STEP 1
STFP 2

Figure 4c. Detailed Flow Diagram

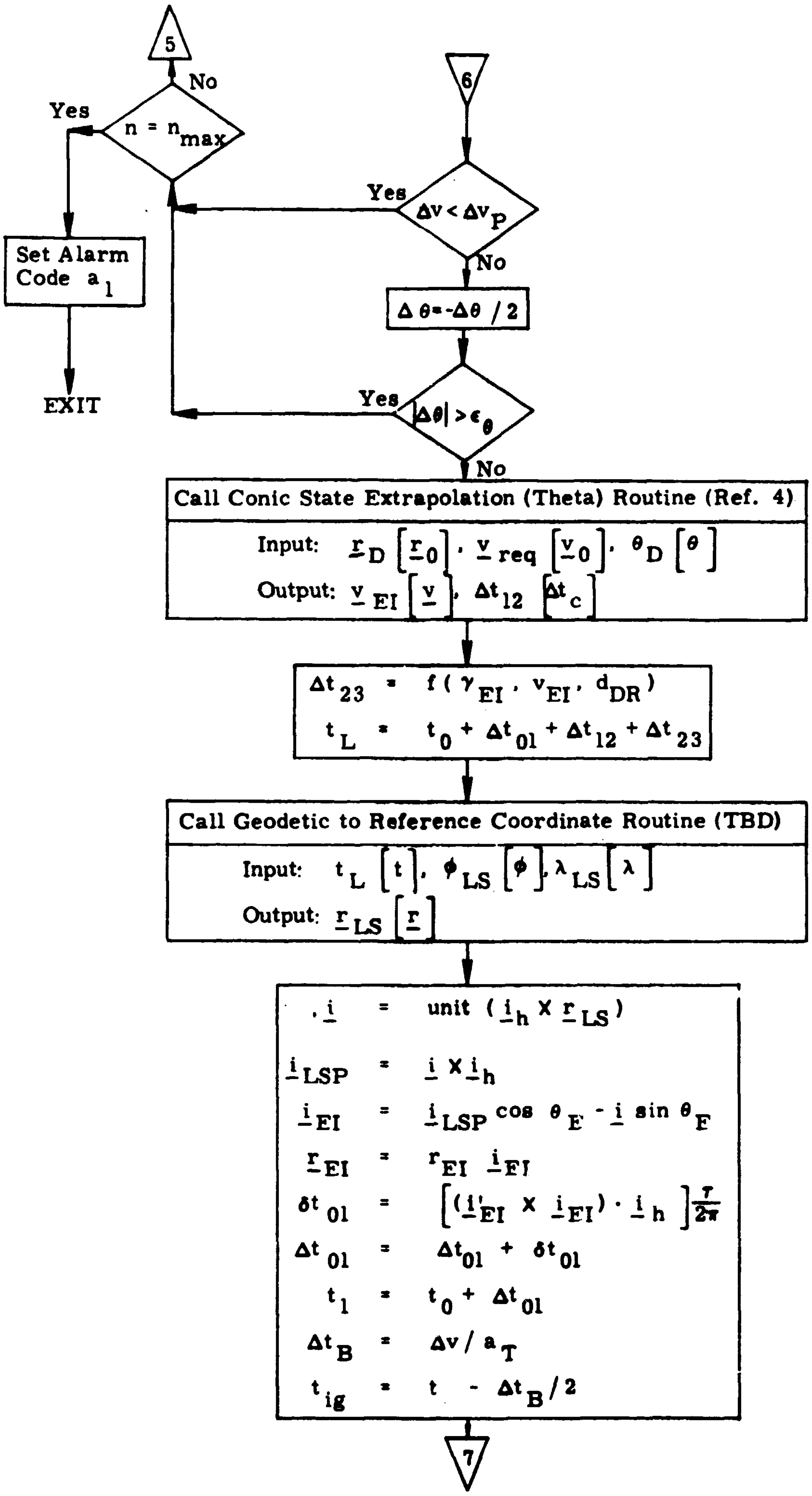
9.12.1 Deorbit Targeting (cont'd)



4
$\theta = \theta_{TP} - \theta_E$
$n = 0$
$\theta_D = \pi$
$\Delta\theta = \Delta\theta_0$
$\theta_{01} = \theta - \theta_D$
Call Conic State Extrapolation (Theta) Routine (Ref. 4)
Input: $\underline{r}_0, \underline{v}_0, \theta_{01} [\theta]$
Output: $\underline{r}_D [\underline{r}], \underline{v}_D [\underline{v}], \Delta t_{01} [\Delta t_c]$
$r_D = |\underline{r}_D|$
$\tan\gamma_1 = (1 - r_D/r_{EI}) \cot(\theta_D/2) - r_D/r_{EI} \tan\gamma_{EI}$
$p_D = \dfrac{2 r_D (r_D/r_{EI} - 1)}{(r_D/r_{EI})^2 p_\gamma - (1 + \tan\gamma_1^2)}$
$v_{HD} = \sqrt{\mu\, p_D / r_D^2}$
$v_{RD} = v_{HD} \tan\gamma_1$
$\underline{v}_{req} = v_{RD} \text{ unit}(\underline{r}_D) + v_{HD} \text{ unit}\left[(\underline{r}_D \times \underline{v}_D) \times \underline{r}_D\right]$
$\Delta\underline{v} = \underline{v}_{req} - \underline{v}_D$
$\Delta v = |\Delta\underline{v}|$
$n = n + 1$
$n = 1$
Yes
No
6
$\theta_D = \theta_D + \Delta\theta$
$\theta_P = \theta_D$
$\Delta v_P = \Delta v$
5
STEP 2

Figure 4d. Detailed Flow Diagram

9.12.1 <u>Deorbit Targeting</u> (cont'd)


5
No
Yes
$n = n_{max}$
Set Alarm Code a_1
EXIT
6
Yes
$\Delta v < \Delta v_P$
No
$\Delta\theta = -\Delta\theta / 2$
Yes
$|\Delta\theta| > \epsilon_\theta$
No
Call Conic State Extrapolation (Theta) Routine (Ref. 4)
Input: $\underline{r}_D \left[\underline{r}_0\right], \underline{v}_{req} \left[\underline{v}_0\right], \theta_D \left[\theta\right]$
Output: $\underline{v}_{EI} \left[\underline{v}\right], \Delta t_{12} \left[\Delta t_c\right]$
$\Delta t_{23} = f(\gamma_{EI}, v_{EI}, d_{DR})$
$t_L = t_0 + \Delta t_{01} + \Delta t_{12} + \Delta t_{23}$
Call Geodetic to Reference Coordinate Routine (TBD)
Input: $t_L \left[t\right], \phi_{LS} \left[\phi\right], \lambda_{LS} \left[\lambda\right]$
Output: $\underline{r}_{LS} \left[\underline{r}\right]$
$\underline{i} = \text{unit}(\underline{i}_h \times \underline{r}_{LS})$
$\underline{i}_{LSP} = \underline{i} \times \underline{i}_h$
$\underline{i}_{EI} = \underline{i}_{LSP} \cos\theta_E - \underline{i} \sin\theta_F$
$\underline{r}_{EI} = r_{EI}\, \underline{i}_{FI}$
$\delta t_{01} = \left[(\underline{i}'_{EI} \times \underline{i}_{EI}) \cdot \underline{i}_h\right] \frac{T}{2\pi}$
$\Delta t_{01} = \Delta t_{01} + \delta t_{01}$
$t_1 = t_0 + \Delta t_{01}$
$\Delta t_B = \Delta v / a_T$
$t_{ig} = t - \Delta t_B / 2$
7

STEP 2

Figure 4e. Detailed Flow Diagram

9.12.1 Deorbit Targeting (cont'd)

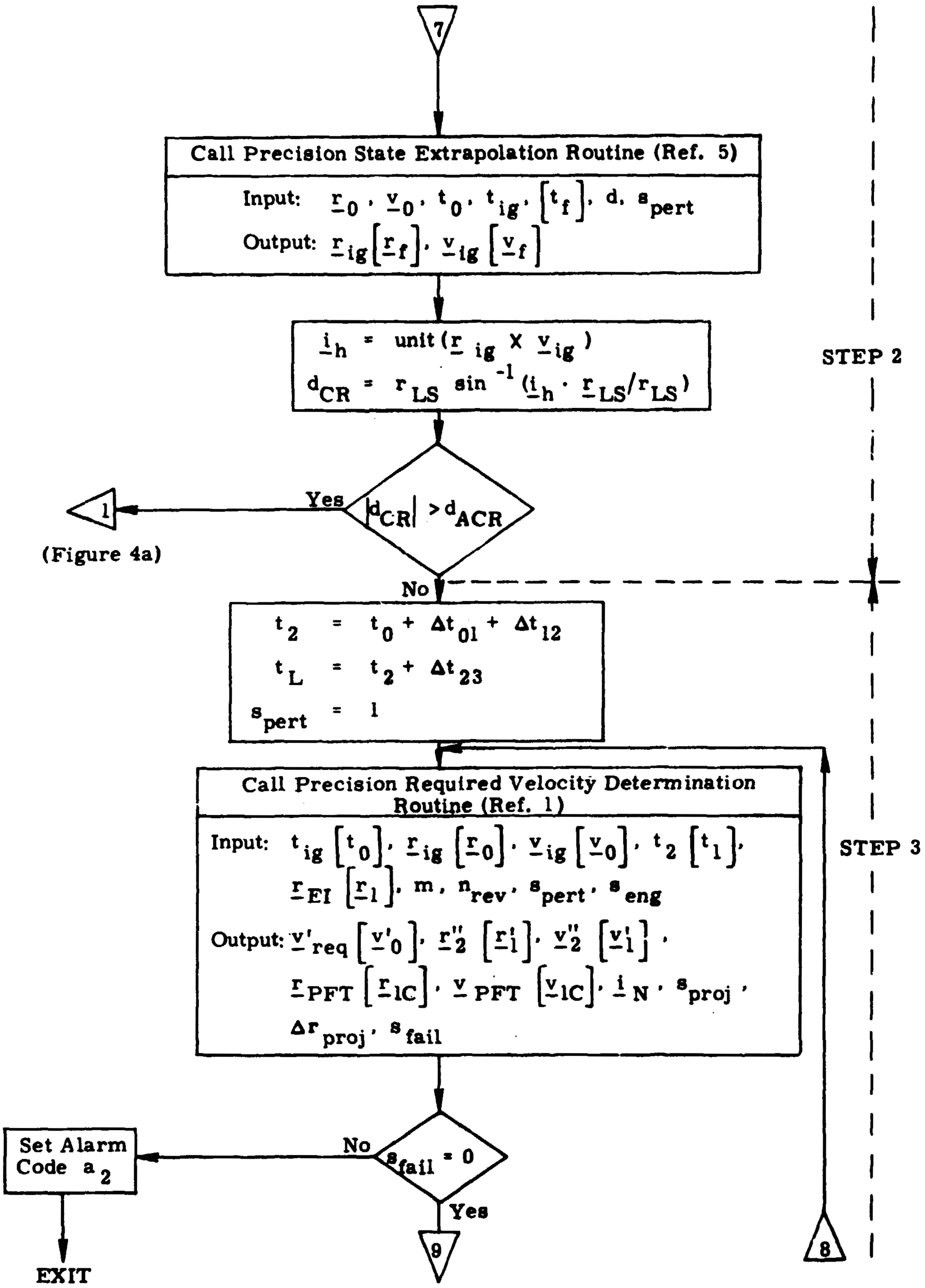

7
Call Precision State Extrapolation Routine (Ref. 5)
Input: $\underline{r}_0$, $\underline{v}_0$, t_0, t_{ig}, $[t_f]$, d, s_{pert}
Output: $\underline{r}_{ig}[\underline{r}_f]$, $\underline{v}_{ig}[\underline{v}_f]$
$\underline{i}_h = \text{unit}(\underline{r}_{ig} \times \underline{v}_{ig})$
$d_{CR} = r_{LS} \sin^{-1}(\underline{i}_h \cdot \underline{r}_{LS}/r_{LS})$
STEP 2
$|d_{CR}| > d_{ACR}$
Yes
1
(Figure 4a)
No
$t_2 = t_0 + \Delta t_{01} + \Delta t_{12}$
$t_L = t_2 + \Delta t_{23}$
$s_{pert} = 1$
Call Precision Required Velocity Determination Routine (Ref. 1)
Input: $t_{ig}[t_0]$, $\underline{r}_{ig}[\underline{r}_0]$, $\underline{v}_{ig}[\underline{v}_0]$, $t_2[t_1]$, $\underline{r}_{EI}[\underline{r}_1]$, m, n_{rev}, s_{pert}, s_{eng}
Output: $\underline{v}'_{req}[\underline{v}'_0]$, $\underline{r}''_2[\underline{r}'_1]$, $\underline{v}''_2[\underline{v}'_1]$, $\underline{r}_{PFT}[\underline{r}_{1C}]$, $\underline{v}_{PFT}[\underline{v}_{1C}]$, $\underline{i}_N$, s_{proj}, Δr_{proj}, s_{fail}
STEP 3
$s_{fail} = 0$
No
Yes
Set Alarm Code a_2
EXIT
9
8

Figure 4f. Detailed Flow Diagram

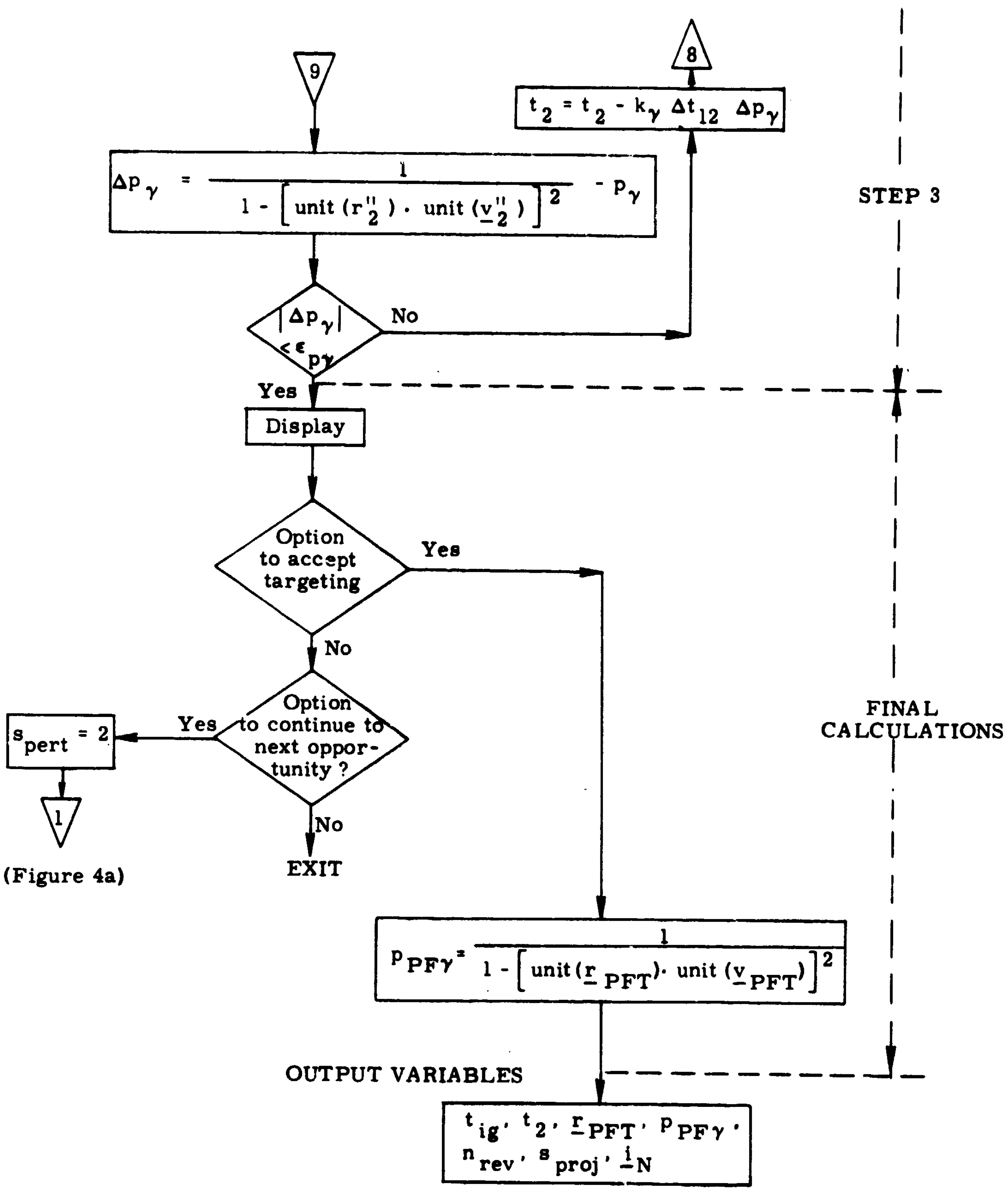

9
8
$t_2 = t_2 - k_\gamma \Delta t_{12} \Delta p_\gamma$
$\Delta p_\gamma = \frac{1}{1 - \left[\text{unit}(r''_2) \cdot \text{unit}(\underline{v}''_2)\right]^2} - p_\gamma$
STEP 3
$|\Delta p_\gamma| < \epsilon_{p\gamma}$
No
Yes
Display
Option to accept targeting
Yes
No
Option to continue to next opportunity ?
Yes
$s_{pert} = 2$
1
(Figure 4a)
No
EXIT
FINAL CALCULATIONS
$p_{PF\gamma} = \frac{1}{1 - \left[\text{unit}(\underline{r}_{PFT}) \cdot \text{unit}(\underline{v}_{PFT})\right]^2}$
OUTPUT VARIABLES
$t_{ig}, t_2, \underline{r}_{PFT}, p_{PF\gamma},$
$n_{rev}, s_{proj}, \underline{i}_N$

Figure 4g. Detailed Flow Diagram

6. SUPPLEMENTARY INFORMATION

As mentioned in Subsection 4.6, the equation necessary to determine entry time-of-flight is not specified. This must necessarily be postponed until further entry guidance design and analysis makes functionalization of this parameter possible.

The effect of different azimuths on the entry phase has not be considered in this design. Since the crossrange capability of the vehicle is dependent on azimuth, and thus may be larger in one direction than the other, this should be considered in the acceptable crossrange criterion. In addition, varying entry azimuths also effect the required inertial flight path angle, and hence it may be more desirable to constrain the relative flight path angle at entry interface.

The deorbit targeting technique presented here requires long term state vector extrapolation to select a deorbit opportunity several revolutions later. The accuracy of this extrapolation is dependent upon accurate knowledge of the current state vector. Any errors due to imperfect navigation will tend to be magnified by the long term extrapolation. Fortunately, the out-of-plane component of error tends to oscillate, and thus can be expected to remain below about 1 n.mi. Consequently, the prediction of crossrange for a deorbit opportunity several revolutions later is not significantly affected by the expected state vector error. Mission planning functions, which are sensitive to crossrange requirements, can be carried out without being significantly affected by the navigation error.

The in-plane component of state vector error does increase during long term extrapolation, at a rate of about 1 n.mi. per revolution. Therefore the targeting should be repeated during the revolution prior to the deorbit maneuver, taking advantage of more precise knowledge of the vehicle state vector.

The deorbit targeting equations presented here are designed to target a single minimum fuel maneuver which satisfies entry interface and landing site constraints. Since deorbit is the final major maneuver, it may be logical to use all the remaining fuel to either effect a faster return or to reduce the entry crossrange requirement. This could be accomplished by (1) thrusting out-of-plane or (2) using multiple maneuvers to place the vehicle in a phasing orbit prior to landing. The first alternative could easily be added to the design presented here. The second alternative, using phasing orbits to adjust the timing and hence inertial landing site location, would require an additional logical structure to select the proper phasing orbit; however, targeting for the final deorbit maneuver could still be accomplished with the equations presented here.

9.12.1 Deorbit Targeting (cont'd)

REFERENCES

1. Brand, Timothy J., "Precision Required Velocity Determination", Space Shuttle GN & C Equation Document No. 13, Rev. 1, M. I. T. /DL, September 1971.

2. Marscher, William, "A Unified Method of Generating Conic Sections", R-479, M.I.T. /IL, February 1965.

3. Pu, Chung L., Higgins, John P., Brand, Timothy J., "Powered Flight Guidance", Space Shuttle GN & C Equation Document No. 11, Rev. 1, M. I. T. /DL, September 1971.

4. Robertson, William M., "Conic State Extrapolation", Space Shuttle GN & C Equation Document No. 3-71, M.I.T. /DL, February 1971.

5. Robertson, William M., "Precision State and Filter Weighting Matrxi Extrapolation", Space Shuttle GN & C Equation Document No. 4, Rev. 1, M. I. T. /DL, To be published.

6. Robertson, William M., "Conic Required Velocity Determination", Space Shuttle GN & C Equation Document No. 10, Rev. 1, M.I.T. /DL, September 1971.

7. Brennan, P. J., "Deorbit Targeting Design Analysis", 23A STS Memo No. 55-71, To be published.

9.12.2 Entry Navigation TBD

SPACE SHUTTLE

GN&C SOFTWARE EQUATION SUBMITTAL

Software Equation Section: Entry Guidance Submittal No. 39 *

Function: Provide guidance commands during entry

Module No. OG-5 Function No. 7,8,9 (MSC-03690 Rev. A)

Submitted by: W.H. Peters (MSC) J.K. Willoughby (LEC) - MSC - 051[illegible]3
(name)

Date: 10-21-71

NASA Contact: W. H. Peters Organization: EG2
(name)

Approved by Panel III K.J. Cox Date 10/21/71
(chairman)

Summary Description: Update to previously submitted fast time integration guidance method for orbiter entry. Incorporates numerous new features, the most notable of which is the ability to simultaneously control heating and both components of final range error.

Shuttle Configuration: (Vehicle, Aero Data, Sensor, Et Cetera)

Testing to date primarily on NR 161C orbiter but method not sensitive to vehicle configuration.

Comments:

(Design Status) Preliminary design complete

(Verification Status) Successfully followed TPS weight-optimum trajectory

Panel Comments:

* Previously baselined as submittal No. 2.

9.12.3 Entry Guidance

1. INTRODUCTION

The fast-time integration (FTI) guidance approach was originally developed for the MSC straight-wing orbiter reentry vehicle as a candidate guidance logic. The background study leading to the development of the original FTI subroutine was documented in reference 3. The details of the guidance algorithm as it applied to the single-control variable straight-wing orbiter appear in references 4 and 5. Extensions and significant modifications to the original routine have followed the decision to apply the method to the high-crossrange vehicle designs. These modifications include the addition of a second control variable with the resulting capability of controlling not only range but also any other parameter which depends only on the trajectory state variables. This report is intended to provide current documentation of the prediction, the sensitivity, and the guidance equations. Also included are a functional flow chart of the logic and a listing of the current FTI subroutine. Preliminary simulation results appear in reference 7 which is a collection of viewgraphs used for an oral presentation of the method. Additional simulation results will appear in future memoranda. A discussion of the relationship of the FTI guidance approach to onboard optimization and to parametric guidance methods is provided in reference 6.

The prediction and sensitivity equations are presented in Section 2. Section 3 discusses the control equations. The Appendix provides a functional flow chart of the guidance computational procedure and a listing of the current FTI subroutines. (The listing is supplied for definition of factors used, but not otherwise defined in the text.)

SYMBOL	DEFINITION OR DESCRIPTION
a	Break frequency in damping filter.
b	Control variable used to shift the angle of attack profile.
c	Weighting factor used in minimum effort control logic to scale relative magnitudes of $\Delta\alpha$ and $\Delta\phi$.
C_D	Drag coefficient.
C_L	Lift coefficient.
d	Constant in damping logic equal to e^{aT} .
D	Aerodynamic drag.
DET( )	Determinant of ( ).
g_0	Surface gravitational constant used in predictions.
$g^*(nT)$	Impulse response of the damping-loop transfer function.
h	Altitude.
i	Instantaneous orbit inclination.
K	Constant used to define the range (heating rate) error when minimum effort heating rate (range) control is desired. See Section 3.
K_0	Gain in altitude rate damping logic.
ℓ	Great circle crossrange distance.
L/D	Lift to drag ratio.
m	Vehicle mass.

SYMBOL	DEFINITION OR DESCRIPTION
P	Guidance sensitivity matrix defined by equation 3.1.
P_{ij}	Element from row i and column j of the P matrix.
$\dot{Q}$	Reference convective heating rate at the stagnation point of a unit sphere; $\dot{Q} = 17600\left(\frac{\rho}{\rho_0}\right)^{1/2}\left(\frac{v}{v_s}\right)^{3.15}$
R	Predicted downrange distance at the cutoff condition; i.e., $R = X_f$
r_E	Earth radius.
S	Reference area.
s	Laplace transform variable.
T	Guidance cycle time.
$\bar{u}$	Nondimensional horizontal velocity; $\bar{u} = \frac{v \cos \gamma}{v_s}$.
v	Total vehicle velocity relative to a rotating earth and atmosphere.
v_s	Circular satellite velocity at 400,000 feet altitude.
X	Great circle downrange distance.
α	Angle of attack.
α_p	Specified angle of attack profile used in guidance predictions and sensitivity calculations.

SYMBOL	DEFINITION OR DESCRIPTION
β	Reciprocal of atmospheric scale height in exponential atmosphere used in guidance predictions.
γ	Flight path angle.
γ^*	Sampled flight path angle.
$\delta\phi$	Damping correction to bank angle.
ϕ	Bank angle; vehicle rotation about the velocity vector.
ρ_0	Assumed sea-level density in exponential atmosphere model used in the guidance predictions.
μ	Instantaneous latitude.
ψ	Heading with respect to the original downrange direction.
ω_E	Earth spin rate.
$(\hat{\ })$	Nondimensional ( ).

2. PREDICTION AND SENSITIVITY EQUATIONS

The equations of motion which form the basis of the guidance predictions and the sensitivity calculations are the following.

$$\dot{v} = -\frac{1}{2m}\rho_0 e^{-\beta h} v^2 S C_D - \sin\gamma \left[g_0 \left(\frac{r_E}{r_E + h} \right)^2 - (r_E + h)\omega_E^2 \cos^2 i \right] \tag{2.1}$$

$$\dot{\gamma} = \frac{1}{2mv}\rho_0 e^{-\beta h} v^2 S C_L \cos\phi - \cos\gamma \left[\frac{g_0}{v} \left(\frac{r_E}{r_E + h} \right)^2 - \frac{(r_E + h)\omega_E^2 \cos^2 i}{v} - \frac{v}{(r_E + h)} \right] + 2\omega_E \cos i \tag{2.2}$$

$$\dot{h} = v \sin\gamma \tag{2.3}$$

$$\dot{x} = \frac{r_E}{r_E + h} v \cos\gamma \cos\psi \tag{2.4}$$

$$\dot{\psi} = \frac{1}{2m}\rho_0 e^{-\beta h} v S C_D (L/D \sin\phi) + 2\omega_E \sin i \sin\mu \tag{2.5}$$

$$\dot{\ell} = v \sin\psi \cos\gamma \tag{2.6}$$

Note that these equations include rotating earth effects but are based on an exponential atmosphere. Computationally the atmosphere consists of two separate exponential fits with a continuous junction at an altitude of 140,000 feet.

The equations actually integrated by the guidance are exactly equivalent to those above but have been transformed substantially to permit rapid and efficient integration. The algebra of these transformations is tedious but straightforward and therefore is not included here. However, the transformation procedure is itemized as follows:

1. Convert the dimensional equations to non-dimensional equations using the vehicle mass, the circular satellite velocity at 400,000 feet altitude and the circular orbit radius at 400,000 feet altitude as non-dimensionalizing factors.

2. Change the independent variable from time to non-dimensional horizontal velocity $\bar{u}$ by dividing equations (2.1) through (2.6) by equation (2.1) and multiplying the resulting equations by

$$\frac{d\hat{v}}{d\bar{u}} = \frac{1}{\cos\hat{\gamma}\left[1 - \hat{v}\tan\gamma\,\frac{d\hat{\gamma}}{d\hat{v}}\right]} , \qquad (2.7)$$

i.e.,

$$\frac{d(\;)}{d\bar{u}} = \frac{\frac{d}{dt}(\;)}{d\hat{v}/dt}\,\frac{d\hat{v}}{d\bar{u}} . \qquad (2.8)$$

3. Factor the resulting equations extensively so that quantities appearing frequently in the equations are evaluated only once for each derivative evaluation. Although this factorization renders the equations almost unrecognizable in the guidance routine, it is perhaps the most important coding detail to permit rapid integration.

In the guidance subroutine, the equations of motion appear as follows:

$$D(1) = (T56)[(T27)(LOD) - (T5)(T26) + 2(T2)] \tag{2.9}$$

$$D(2) = (T56)[(VN)(T10)] \tag{2.10}$$

$$D(3) = (T56)[(T25)(T5)(T99)] \tag{2.11}$$

$$\begin{aligned} D(4) = {} & (T56)\Big[\big((T27)/(T7)\big)\ \mathrm{SIN}(PPHI)(LOD) \\ & + 2(OMEGA)\ \mathrm{SIN}\ (ANGLEI)\ \mathrm{SIN}\ (XMU)\Big] \end{aligned} \tag{2.12}$$

$$D(5) = (T56)[(VN)(T8)(T5)]\ . \tag{2.13}$$

The factored "T" quantities are defined by the code itself which appear in the Appendix of this report.

The perturbation equations integrated by the guidance result from taking partial derivatives of equations (2.1) to (2.4) with respect to the quantities (L/D cos ϕ) and C_D . These quantities are used instead of α and ϕ directly to avoid algebraic sign reversals in the sensitivities as the bank angle goes through zero degrees. The sensitivities to (L/D cos ϕ) and to C_D are related to the sensitivities to α and ϕ by simple chain rule differentiation; i.e.,

$$\frac{\partial(\ )}{\partial\alpha} = \frac{\partial(\ )}{\partial(L/D\cos\phi)}\,\frac{\partial(L/D\cos\phi)}{\partial\alpha} = \frac{\partial(\ )}{\partial(L/D\cos\phi)}\cos\phi$$

$$\cdot\left[\frac{1}{C_D}\frac{\partial C_L}{\partial\alpha} - \frac{C_L}{C_D^2}\frac{\partial C_D}{\partial\alpha}\right] \qquad (2.14)$$

$$\frac{\partial(\ )}{\partial\phi} = \frac{\partial(\ )}{\partial(L/D\cos\phi)}\,\frac{\partial(L/D\cos\phi)}{\partial\phi} = \frac{\partial(\ )}{\partial(L/D\cos\phi)}\left(-\frac{L}{D}\sin\phi\right) \qquad (2.15)$$

In the guidance subroutine, the perturbation equations for (L/D cos ϕ) take the form:

D(6) = (A11)(Y(6)) + (A12)(Y(7)) + PFLOD1 (2.16)

D(7) + (A21)(Y(6)) + (A22)(Y(7)) + PFLOD2 (2.17)

D(8) = (A31)(Y(6)) + (A32)(Y(7)) + PFLOD3 . (2.18)

The perturbation equations for C_D appear as

D(9) = (A11)(Y(9)) + (A12)(Y(10)) + PFCD1 (2.19)

D(10) = (A21)(Y(9)) + (A22)(Y(10)) + PFCD2 (2.20)

D(11) = (A31)(Y(9)) + (A32)(Y(10)) + PFCD3 . (2.21)

The values of the aerodynamic coefficients C_L and C_D and their derivatives with respect to α (C_{L_α} and C_{D_α}) are included in the prediction and sensitivity integrations using cubic spline approximations (reference 2). Interpolation using splines is very nearly as efficient as using linear interpolation and provides accurate derivative information at no cost. It was found that very simple polynomial approximations for angle of attack dependence were too inaccurate to produce a net gain in guidance efficiency. Mach number effects are included using linear interpolation.

The crossrange equations (2.12) and (2.13) are integrated by the guidance with the bank angle profile multiplied by -1 to predict the crossrange capability that remains if the bank angle were reversed at the current time. The crossrange logic commands reversals when the reversed-bank crossrange equations predict termination inside a shrinking interval about the target. As a result of using $-\phi$ in equations (2.12) and (2.18), the value of ψ from the crossrange integration is not fed into the downrange equation. Instead, the value of ψ used in equation (2.11) is obtained as follows:

$$\dot{\psi} \approx \frac{D}{mv} (L/D \sin \phi) \qquad (2.22)$$

$$\dot{v} \approx -D/m \qquad (2.23)$$

$$\therefore \frac{d\psi}{dv} \approx -1/v \, (L/D \sin \phi) \qquad (2.24)$$

Assuming that L/D sin ϕ is constant over each increment of velocity corresponding tc a single integration step

$$\psi_{i+i} = \psi_i - (L/D \sin \phi) \, \ell n\left(V_{i+1}/V_i\right) . \qquad (2.25)$$

This simple integration provides a ψ value of sufficient accuracy to use in the downrange equation (2.11).

The value of the latitude used in equation (2.5) is calculated as in Figure 1.

A predictor-corrector integration method is used in the current FTI guidance routine to perform the numerical integration. Four steps of a fourth-order Runge-Kutta method are used to start the integration. The predictor is the fourth-order Adams Bashforth formula and the corrector is the fourth-order Adams-Moulton formula. Stepsizes are fixed at 50 feet per second. The integration code appears as an integral part of the guidance routine to avoid calls to external and overgeneralized integration subroutines.

To date, no attempt has been made to strip down either the equations of motion or the integration. The synthesis philosophy has been to build from formulae that are unquestionably accurate enough and to do so with as much efficiency as possible. Complete trajectory integrations are currently being performed in approximately 2 seconds of UNIVAC 1108 execution time. It is probable that an investigation of the following items would lead to a substantial reduction in computation time:

1. Simplification of the prediction and/or the sensitivity equations based on an order of magnitude examination of the terms.

2. Increase of the integration stepsize.

3. Alteration of the order of the integration formulae. For example, with current stepsizes, fourth-order formulae are probably not necessary. Elimination of the corrector alone would almost halve the integration time. Other investigators have recently recommended the use of higher-order integrators with very large stepsizes to improve efficiency. These suggestions should be investigated. It is anticipated that by using state-of-the-art integration methods the integration efficiency could be improved by at least a factor of two.

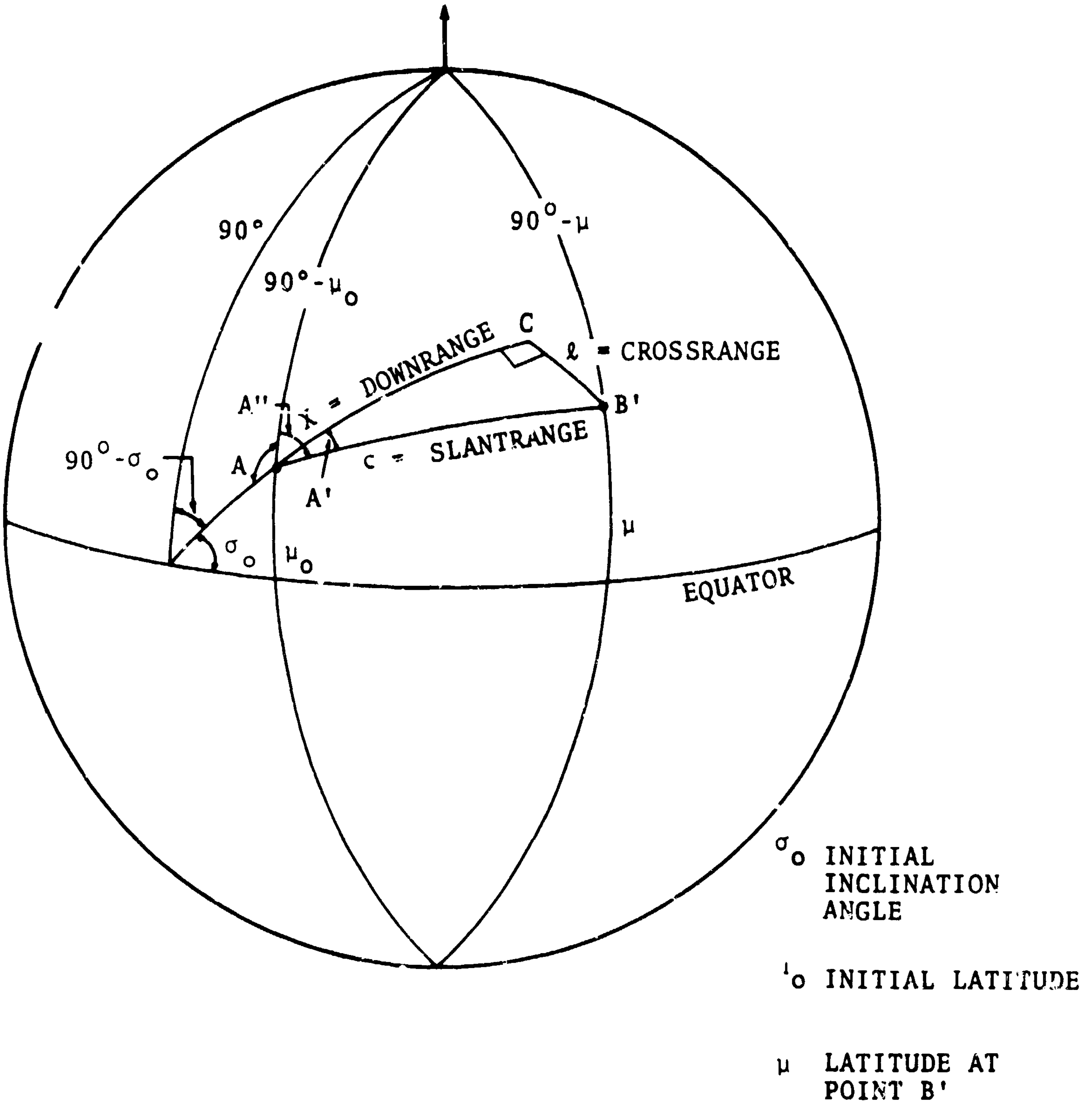


Figure 1. – Computation of latitude at point B'.

(1 of 2)

$$\frac{\sin A}{\sin 90^{\circ}} = \frac{\sin(90^{\circ}-\sigma_o)}{\sin(90^{\circ}-\mu_o)} = \frac{\cos \sigma_o}{\cos \mu_o} = \sin A$$

$$\cos c = \cos X \cos \ell$$

$$\sin A = \frac{\sin \ell}{\sin c}$$

$$A'' = (180^{\circ}-A) + A'$$

$$\begin{aligned}\cos(90^{\circ}-\mu) = \sin\mu &= \cos c \cos(90^{\circ}-\mu_o) \\ &\quad + \sin c \sin(90^{\circ}-\mu_o)\cos A'' \\ &= \cos c \sin\mu_o + \sin c \cos\mu_o \cos A''\end{aligned}$$

Figure 1. - Computation of latitude at point B'.

(2 of 2)

C.2

3. GUIDANCE EQUATIONS

As currently coded, the FTI subroutine issues angle of attack and bank angle commands based on the reference unit sphere convective heating rate at a future point and the downrange distance traveled when the vehicle reaches a specified cutoff altitude. It is apparent from reference 1 that approximate temperature control can be maintained by specifying an appropriate reference sphere heating rate profile for the nose panel of the orbiter. The guidance algorithm attempts to make the predicted reference heating rate at a future point on the trajectory equal to a desired value obtained from a specified profile. The prediction point is determined by subtracting a fixed velocity decrement from the current velocity. For example, suppose the specified velocity decrement is 400 feet per second. At a guidance computation which begins at a velocity of 23,000 feet per second, the heating rate at 22,600 feet per second is predicted. Also calculated are the sensitivities of the heating rate at 22,600 feet per second to the current control values. Similarly, when the vehicle is traveling at 22,400 feet per second the predicted heating rate at 22,000 feet per second is being controlled. Simultaneously, the predicted range at the cutoff altitude is being calculated along with its sensitivities to the current control values.

The guidance computes the elements of the sensitivity matrix P , where

$$P = \begin{bmatrix} \frac{\partial R}{\partial \alpha} & \frac{\partial R}{\partial \phi} \\ \frac{\partial \dot{Q}}{\partial \alpha} & \frac{\partial \dot{Q}}{\partial \phi} \end{bmatrix} \tag{3.1}$$

The sensitivities of R are evaluated at the cutoff condition and the sensitivities of $\dot{Q}$ are evaluated at the prediction time appropriate for the current guidance call. Linear analysis leads to

$$\begin{bmatrix} \Delta R \\ \Delta \dot{Q} \end{bmatrix} = P \begin{bmatrix} \Delta \alpha \\ \Delta \phi \end{bmatrix} \tag{3.2}$$

where ΔR and $\Delta \dot{Q}$ are determined by differencing the predicted and desired values of R and $\dot{Q}$, respectively. The control commands are obtained by solving this simple linear system simultaneously; i.e.,

$$\begin{bmatrix} \Delta \alpha \\ \Delta \phi \end{bmatrix} = P^{-1} \begin{bmatrix} \Delta R \\ \Delta \dot{Q} \end{bmatrix} \tag{3.3}$$

To first order, the above control law finds the control pair which eliminates the range error and the heating rate error simultaneously.

It is easily seen from equations (2.14) and (2.15) that P is singular when $\phi = 0$ or $\phi = 90°$. In the subroutine, the magnitude of the determinant of P is calculated and if necessary limited to a value greater than a preset minimum. In addition, if the determinant of P gets small causing large control changes to result from 3.3, these changes are limited so that $|\Delta\alpha| \leq 3°$ and $|\Delta\phi| \leq 10°$. This allows the controller to make the next prediction with a value of ϕ for which P is non-singular.

Since it is necessary to provide for upper and lower limits on both angle of attack and bank angle, special control saturation logic is used. Each forward integration is performed with control limiting logic. For example, if the controller has shifted the angle of attack profile so that in some segment of the trajectory α exceeds its maximum allowable value, the trajectory prediction and the sensitivity calculation are made using the maximum allowable value in that segment of the trajectory. Thus, all predictions and sensitivity calculations are based on perturbing the control only in regions where perturbations can be tolerated. This represents another major advantage of onboard sensitivity calculation. Other perturbation guidance schemes, such as the adjoint variable methods, cannot easily adapt to control saturation and therefore provide inaccurate sensitivities in those regions. If a control is saturated and if the change given by (3.3) does not cause the control to leave the control boundary, the FTI control law recomputes the other control variable change based on the proper single-control variable sensitivity. An example would help clarify this simple saturation procedure. Suppose $\alpha = \alpha_{max}$ and $\Delta\alpha$ computed from (3.3) is positive. The commanded change in ϕ is then recomputed as either

$$\Delta\phi = \frac{\Delta\dot{Q}}{\partial\dot{Q}/\partial\phi} = \frac{\Delta\dot{Q}}{P_{22}} \quad \text{or} \quad \Delta\phi = \frac{\Delta R}{\partial R/\partial\phi} = \frac{\Delta R}{P_{12}} \tag{3.4}$$

depending on the control segment in which the guidance is operating. Similarly, suppose ϕ is saturated and $\Delta\phi$ computed from (3.3) does not bring ϕ off its limit. The $\Delta\alpha$ command issued by the guidance is then recomputed as

$$\Delta\alpha = \frac{\Delta\dot{Q}}{\partial\dot{Q}/\partial\alpha} = \frac{\Delta\dot{Q}}{P_{21}} \quad \text{or} \quad \Delta\alpha = \frac{\Delta R}{\partial R/\partial\alpha} = \frac{\Delta R}{P_{11}} \tag{3.5}$$

Again the equation chosen depends on the current control segment. Note that the recomputation of the commands during regions of control saturation requires only an additional division since the single-control gains are elements of the sensitivity matrix P which is already computed.

The control segments referred to in the previous paragraph are defined as follows:

Segment 1 – From guidance startup at .05 g's deceleration to a fixed velocity point. The controls in this segment are calculated to eliminate the $\dot{Q}$ error with minimum control effort.

Segment 2 – From the end of Segment 1 to the point on the trajectory where the predicted maximum heating rate first becomes less than a threshold value. The controls in this segment are calculated to eliminate the $\dot{Q}$ error and the R error simultaneously. If a control saturates in this segment, range control is dropped temporarily and the $\dot{Q}$ error is used to compute the other control.

Segment 3 - From the end of Segment 2 to cutoff altitude. The controls in this segment are calculated to eliminate the R error with minimum control effort.

The minimum effort criterion is used in regions where the control of one parameter takes high priority over the other. In Segment 1, the objective of getting onto a desirable temperature profile takes priority over range considerations. Therefore, the quantity ΔR is computed as a scalar constant times the required $\Delta\dot{Q}$ and the constant is determined to minimize the control change. For example, let the measure of control change be given by

$$J = \Delta\alpha^2 + c\Delta\phi^2 \qquad (3.6)$$

where c is an arbitrary weighting constant. Assuming that range control is to be temporarily ignored, let

$$\Delta R = K\Delta\dot{Q} \qquad (3.7)$$

Then

$$\begin{bmatrix}\Delta\alpha\\ \Delta\phi\end{bmatrix} = P^{-1}\begin{bmatrix}K\Delta\dot{Q}\\ \Delta\dot{Q}\end{bmatrix} \qquad (3.8)$$

or

$$\Delta\alpha = \frac{1}{\text{DET } P}[P_{22}\,\Delta\dot{Q} - P_{12}K\,\Delta\dot{Q}] \qquad (3.9)$$

$$\Delta\phi = \frac{1}{\text{DET } P}[-P_{21}\,\Delta\dot{Q} + P_{11}K\,\Delta\dot{Q}] \qquad (3.10)$$

where DET P is the determinant of P .

Substituting these expressions for $\Delta\alpha$ and $\Delta\phi$ into J and setting $\partial J/\partial K = 0$ results in

$$K = \frac{P_{12}P_{22} + cP_{21}P_{11}}{P_{12}^2 + cP_{11}^2} \qquad (3.11)$$

It is easily shown that this value of K represents a relative minimum of J . A similar result follows if $\Delta\dot{Q}$ is specified as a constant times ΔR.

It has been found in simulation results that the minimum effort criterion to specify the "uncontrolled" parameter results in excellent targeting in Segment 3 where heating problems are no longer of consequence. Although extensive simulation verification has not yet resolved the proposition, it appears that Segment 1 may not be necessary. The advantage of Segment 2 cannot be overemphasized. In that segment, a desired heating rate profile (and therefore an approximate panel #1 temperature profile) can be tracked while the guidance is simultaneously solving the range problem.

Typically, Segment 2 covers from 5 to 10 minutes of trajectory time during which heating considerations are important. Neglect of direct range control for this length of time and at the high-energy end of the trajectory causes a considerable loss in targeting ability. Simultaneous control of range and heating rate in Segment 2 is overridden only in regions where the control saturates. These regions can be minimized or eliminated by choosing realistic control profiles for the prediction integrations. The result is that Segment 2 provides both path-type and terminal-type control and appears to increase substantially the guidance footprint areas over that obtained with a temperature-only control segment followed by a range-only control segment.

One of the principal advantages of integrated predictions lies in the fact that completely arbitrary control profiles can be assumed in the prediction. Currently the program uses a constant bank angle profile and a variable angle of attack profile of the form

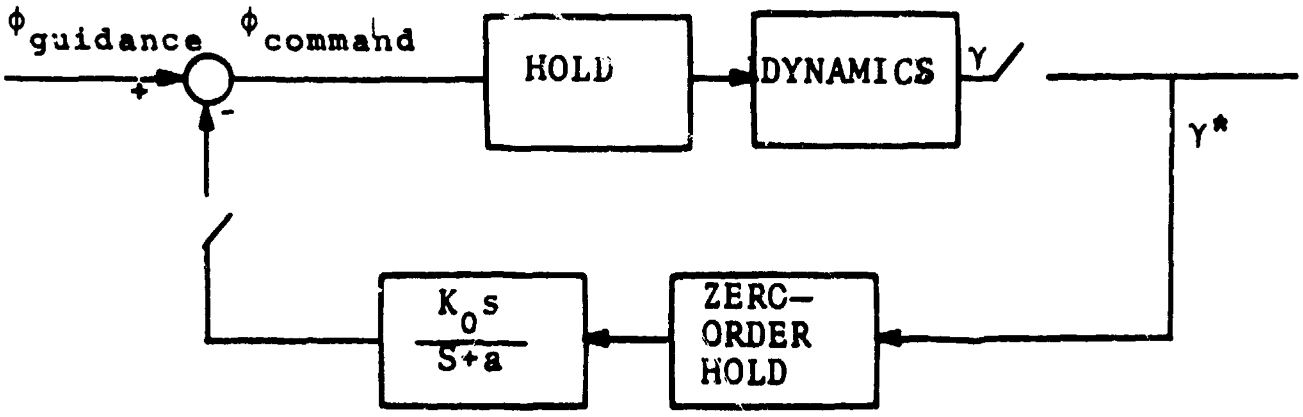
$$\alpha = b + \alpha_p(\bar{u}) \qquad (3.12)$$

where

$$\begin{aligned} \alpha_p(\bar{u}) &= 50^\circ \qquad \bar{u} \geq 0.8 \\ &= 50^\circ - 20^\circ \sin\left(\frac{\pi}{1.2}\,(0.8 - \bar{u})\right) \qquad 0.2 \leq \bar{u} \leq 0.8 \\ &= 30^\circ \qquad \bar{u} \leq 0.2 \qquad (3.13) \end{aligned}$$

The guidance adjusts the constant b to increase or decrease the entire profile by the amount necessary to null the guidance errors. The α profile choice was made to allow the guidance to stabilize and still permit the pitchover maneuver necessary to leave the vehicle at a small angle of attack at guidance cutoff. The best choice of prediction profiles has not been studied to date but it is anticipated that such a study could produce improvements such as enlarged target areas, more nearly optimal trajectories, and reduced sensitivities to navigation errors. The possibility of using several parameters to specify the prediction control profiles and of using the guidance to adjust these parameters in flight is discussed briefly in references 3 and 6.

In the FTI guidance routine, both the prediction bank angle and the commanded bank angle are modified, if necessary, to provide trajectory damping. It has been found that at lower angles of attack the delta-wing configurations have a tendency to show altitude oscillations that are undesirable. These oscillations can be overcome by using a simple flight path angle feedback. Since low frequency flight path angle changes must be permitted, a simple high-pass filter in the feedback loop is suggested. The bank angle command is used for damping. The feedback loop has the structure



The transfer function from the sampled flight path angle γ^* to the bank angle correction $\delta\phi$ must include the dynamics of the hold and is therefore given by

$$G(Z) = \frac{\delta\phi}{\gamma^*} = \frac{Z - 1}{Z} \mathcal{Z} \left[\frac{1}{s} \frac{K_0 s}{S + a}\right] \qquad (3.14)$$

$$= K_0 \frac{Z - 1}{Z - d} \qquad (3.15)$$

where $d = e^{-aT}$; T is the sample period, and Z represents the Z-transform. The inverse of G(Z) is

$$g(nT) = \begin{cases} K_0 & \text{for } n = 0 \\ K_0 \, d^{n-1}(d - 1) & \text{for } n \geq 1 \end{cases} \qquad (3.16)$$

Using the convolution summation, the correction $\delta\phi(nT)$ is obtained as

$$\delta\phi(nT) = \sum_{n=0}^{\infty} \gamma^*[(m - n)T]g^*(nT) \qquad (3.17)$$

The choice of an appropriate value for the filter break frequency, as based on the frequency of the undesirable altitude dynamics, results in a rapidly converging series. The guidance uses four terms of the series with approximately a 3% error due to truncation; i.e.,

$$\delta\phi(t) = \gamma^*(t)K_0 + \gamma^*(t - \Delta t)g^*(t - 2\,\Delta t)$$
$$+ \gamma^*(t - 2\,\Delta t)g^*(t - \Delta t) + \gamma^*(t - 3\,\Delta t)g^*(t) \; . \qquad (3.18)$$

This simple damping logic using three backpoints from previous guidance passes has proven very effective in smoothing the altitude-velocity profiles. Since the damping is incorporated into the prediction and sensitivity calculations as well as the commands, no degradation of the targeting accuracy is produced.

REFERENCES

(1) Curry, Donald M.: Local TPS Weight Estimation Methods for Space Shuttle Orbiter Entry Trajectory Optimization Study: NASA MSC Memo ES32, June 21, 1971.

(2) Vickery, S. A.: "Subroutine ARODYN with Spline Cubic Interpolation," LEC Tech Memo 67542G/118201, July 1, 1971.

(3) Willoughby, J. K. and Rao, T. C.: A Study of Alternative Guidance Concepts for Shuttle Reentry, LEC Tech Report 67542C/024502, September 1970.

(4) Willoughby, J. K.: "Detailed Documentation of Fast-Time Prediction Guidance Software," LEC Tech Memo 67542B/036303, December 28, 1970.

(5) Willoughby, J. K.: "FTP-4D Reentry Guidance," LEC Tech Memo 67542B/036201, December 30, 1970.

(6) Willoughby, J. K.: "In-Flight Optimization using FTI Guidance," LEC Tech Memo 67542B/122803, August 16, 1971.

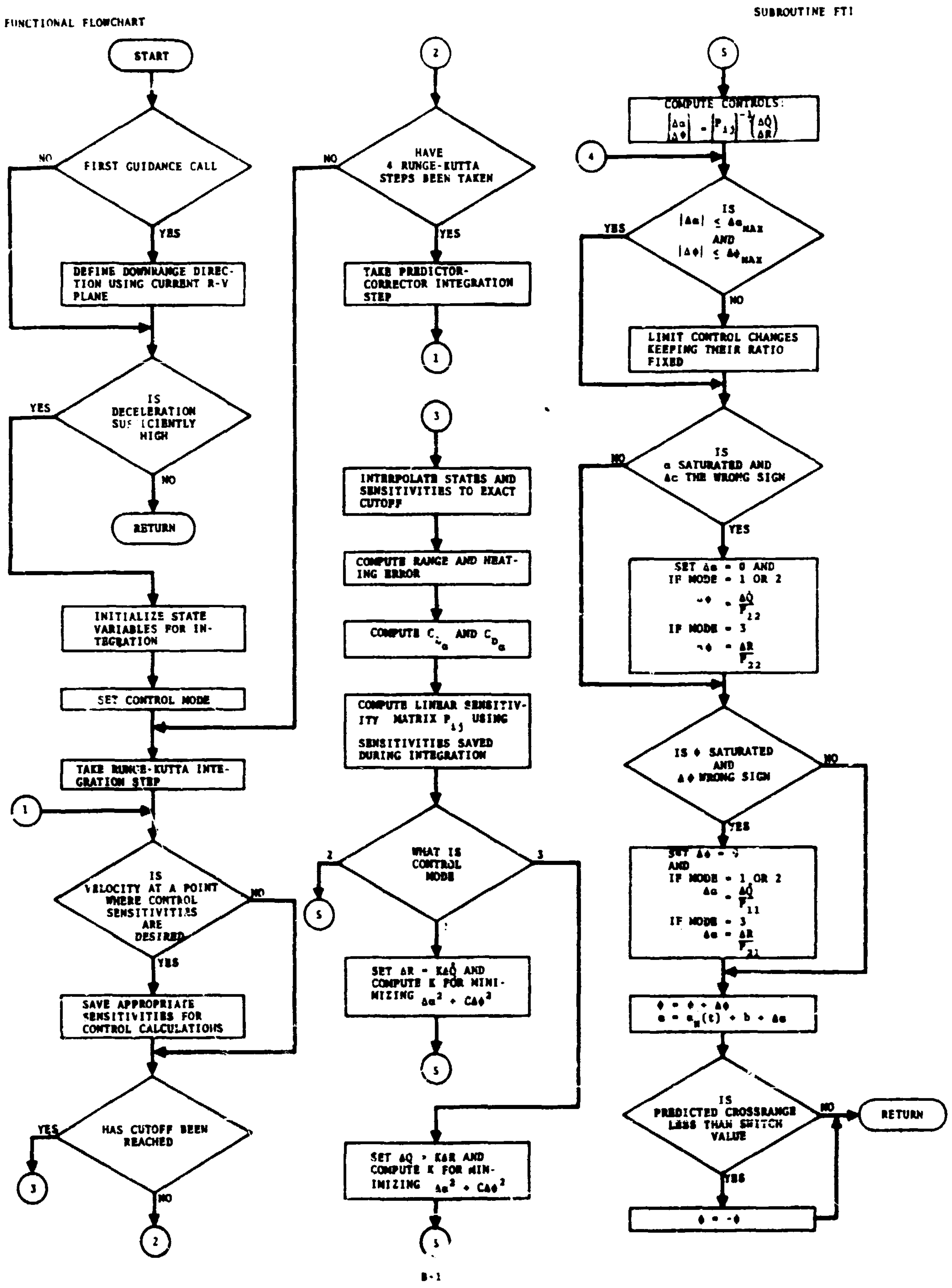
(7) Willoughby, J. K.: "2-Control Variable FTI Guidance Viewgraphs From Oral Documentation," LEC Tech. Memo. 67542B/122804, August 26, 1971.

APPENDIX

FUNCTIONAL FLOWCHART AND

LISTINGS OF FTI GUIDANCE SOFTWARE

9.12.3 Entry Guidance (cont'd)


FUNCTIONAL FLOWCHART
SUBROUTINE FTI
START
FIRST GUIDANCE CALL
NO
YES
DEFINE DOWNRANGE DIRECTION USING CURRENT R-V PLANE
IS DECELERATION SUFFICIENTLY HIGH
YES
NO
RETURN
INITIALIZE STATE VARIABLES FOR INTEGRATION
SET CONTROL MODE
TAKE RUNGE-KUTTA INTEGRATION STEP
1
IS VELOCITY AT A POINT WHERE CONTROL SENSITIVITIES ARE DESIRED
NO
YES
SAVE APPROPRIATE SENSITIVITIES FOR CONTROL CALCULATIONS
HAS CUTOFF BEEN REACHED
YES
NO
3
2
2
HAVE 4 RUNGE-KUTTA STEPS BEEN TAKEN
NO
YES
TAKE PREDICTOR-CORRECTOR INTEGRATION STEP
1
3
INTERPOLATE STATES AND SENSITIVITIES TO EXACT CUTOFF
COMPUTE RANGE AND HEATING ERROR
COMPUTE C_{L_α} AND C_{D_α}
COMPUTE LINEAR SENSITIVITY MATRIX P_{ij} USING SENSITIVITIES SAVED DURING INTEGRATION
WHAT IS CONTROL MODE
2
3
1
5
SET $\Delta R = K\Delta\dot{Q}$ AND COMPUTE K FOR MINIMIZING $\Delta\alpha^2 + C\Delta\phi^2$
5
SET $\Delta Q = K\Delta R$ AND COMPUTE K FOR MINIMIZING $\Delta\alpha^2 + C\Delta\phi^2$
5
B-1
5
COMPUTE CONTROLS: $\begin{vmatrix}\Delta\alpha\\ \Delta\phi\end{vmatrix} = \left|P_{ij}\right|^{-1}\begin{pmatrix}\Delta\dot{Q}\\ \Delta R\end{pmatrix}$
4
IS $|\Delta\alpha| \leq \Delta\alpha_{MAX}$ AND $|\Delta\phi| \leq \Delta\phi_{MAX}$
YES
NO
LIMIT CONTROL CHANGES KEEPING THEIR RATIO FIXED
IS α SATURATED AND Δc THE WRONG SIGN
NO
YES
SET $\Delta\alpha = 0$ AND IF MODE = 1 OR 2 $\Delta\phi = \frac{\Delta\dot{Q}}{P_{22}}$ IF MODE = 3 $\Delta\phi = \frac{\Delta R}{P_{22}}$
IS ϕ SATURATED AND $\Delta\phi$ WRONG SIGN
NO
YES
SET $\Delta\phi = 0$ AND IF MODE = 1 OR 2 $\Delta\alpha = \frac{\Delta\dot{Q}}{P_{11}}$ IF MODE = 3 $\Delta\alpha = \frac{\Delta R}{P_{21}}$
$\phi = \phi + \Delta\phi$
$\alpha = \alpha_N(t) + b + \Delta\alpha$
IS PREDICTED CROSSRANGE LESS THAN SWITCH VALUE
NO
RETURN
YES
$\phi = -\phi$

```
      SUBROUTINE FTP
      INTEGER STPCNT
      INTEGER DERIV , ENDPT , GUIDE
       REAL K1(12),K2(12),K3(12),K4(12)
      REAL MO , LOD
       REAL MSAV
       DIMENSION D(12),Y(12),YHAT(12),BKPT1(12)
       DIMENSION BKDRV0(12),BKDRV1(12),BKDRV2(12),BKDRV3(12),BKDRV4(12)
       DIMENSION YSAV(12)
       DIMENSION SA(11),SS(11)
       DIMENSION SPLSND(30),VSND(5)
      DIMENSION VATI(3), RIV(3), DDRLDT(3)
      COMMON/ENVCON/A( 600)
      COMMON/ACTVEC/ACT(162)
      COMMON/ATMOSC/AT(88)
      COMMON/LECCOM/XLEC(1000)
      COMMON/FSCOM/F(1000)
      COMMON/ENTCST/ENT(10)
      EQUIVALENCE (GAMMA  , Y(1))
      EQUIVALENCE (HN     , Y(2))
      EQUIVALENCE (PSI    , Y(4))
      EQUIVALENCE (CL          , ENT(  1))
      EQUIVALENCE (CD          , ENT(  2))
      EQUIVALENCE (S           , ENT(  3))
      EQUIVALENCE (W           , ENT(  4))
      EQUIVALENCE (LOD         , ENT(  5))
      EQUIVALENCE (VATI(1)     , AT( 9))
      EQUIVALENCE (RIV(1)      , ACT(86))
      EQUIVALENCE (DDRLDT(1)   , ACT(160))
      EQUIVALENCE (ALTUDE      , AT( 4))
      EQUIVALENCE (TIMEX       , XLEC( 1))
      EQUIVALENCE (PT19        , F(92))
      EQUIVALENCE (ACCVN       , F(93))
      EQUIVALENCE (PODOT       , F(94))
      EQUIVALENCE (ODTVN       , F(95))
      EQUIVALENCE (TN          , F(96))
      DATA IPASS/0/
       DATA DLTV/400./
      DATA (SA(I),I=1,11)
     1 / 0. , 24000.,50000.,80000.,155000.,184000.,196000.,254000.,
     2  285000., 312000., 350000./
      DATA (SS(I),I=1,11)
     1 /1116.,1020.,965.,965.,1080.,1080.,1064.,904.,868.,902.,1053./
       DATA (SPLSND(I),I=1,30)
     1 /-96.,    -15.3307,  +15.3307,
     2  -87.3917,  +35.9846,  -3.59286,
     3  -30.2321,  +33.5583,  -3.32624,
     4   67.2646,  147.372,  -99.6370,
     5   24.3981,  -22.6567,  -1.74136,
     6  -10.8163,  -4.77388,  -.409804,
     7  -104.369,  -140.244,  34.6121,
     8  -70.0276,   32.4503,   1.57732,
     9  - 0.344120, 28.2059,   6.13825,
     *  104.927,   92.3459,  -46.1730/
       DATA NSND/11/
      DATA MEOS /11/
      ASSIGN 350 TO DERIV
      ASSIGN 300 TO ENCPT
      ASSIGN 400 TO GUIDE
C**********************************************************************
C
C                        INITIALIZATION FOR INTEGRATION
C
C FIND NORMAL TO CURRENT R-V PLANE
C
      CALL CROSS(RIV,VATI,F(43))
      IF (IPASS .NE. 0) GO TO 20
C
C ERECT UNIT VECTOR THROUGH NORTH POLE
C
      F(46) = 0.
      F(47) = 0.
      F(48) = 1.
C
C COMPUTE INITIAL INCLINATION
C
      SIGO  = ACOS( DOT(F(43),F(46))/ABVAL(F(43)))
      MODE  = 0
       BIAS=0.
       ALBIAS = 0.
C COMPUTE INITIAL LATITUDE
      XMUO = 1.5707963 - ACOS( DOT(F(54),F(46)) / (ABVAL(F(54))))
       BIGA= ASIN( COS(SIGO)/COS(XMUO) )
      PHISGN = 1.
      LFLAG  = 0
      IPASS  = 1
C VDG AND SWTMAG ARE CROSSRANGE CONTROL CONSTANTS.
      SWTMAG = 1000.
      VDR    = 1.0
C C1 AND C2 ARE INTEGRATION CONSTANTS.
      C1     = 1./720.
      C2     = 1./1440.
       RE= 20906428.
C RO, VO, AND MO ARE THE NON-DIMENSIONALIZING CONSTANTS.
       RO=2.1326428E+07
       VO= 2.569006E+04
       MO= 268009.024/32.174
C TRHONO IS THE TRUE SEA-LEVEL DENSITY NONDIMENSIONALIZED.
      TRHONO = (.07647*RO**3) / (32.174*MO)
C ALPMIN IS AN ADJUSTABLE MINIMUM VALUE FOR ANGLE OF ATTACK.
C   THE PROGRAM ADJUSTS THIS VALUE BASED ON THE PREDICTED MAX HEATING
C   RATE. THIS IS AN INITIALIZING VALUE.
       ALPMIN = 45./57.3
       PHIOLD= F( 3)
C**********************************************************************
C
C DEFINE DESIRED RANGES BETWEEN THESE STATEMENTS
C
       DESRNG = 7000.   *6076.097/RO
       DSRDCR = 1200. * 6076.097/RO
C
C DEFINE DESIRED RANGES BETWEEN THESE STATEMENTS
C
C**********************************************************************
```

```
C PXR IS THE CROSSRNG VALUE FROM THE PREVIOUS CALL.
C PRNG IS THE DOWNRNG VALUE FROM THE PREVIOUS CALL.
       PXR=0.
       PRNG=0.
C
C
       SN=  S    /RO**2
       IDVPTH = 20
      OMEGAN = 7.29211508E-05 * RO / VO
      REN    = RE /RO
C                    REGULAR PASS LOGIC
   20 CONTINUE
C                    TWO PIECE EXPONENTIAL ATMOSPHERE DATA
      BETA   = 1./26200.
      RHONO  = .0010845 * RO**3 / MO
      ATMOSH = 140000.
      BETA2  = 1. / 21680.
      RHONO2 = .003346 * RO**3 / MO
       PHI=PHIHLD
      AOATK  =  F(88)
C
C COMPUTE CURRENT INCLINATION
      IFLAG  = 0
      SIG    = ACOS( DOT(F(43),F(46)) / ABVAL(F(43)))
C PT19 IS THE VALUE OF DV/DT ON THE PREVIOUS CALL NONDIMENSIONALIZED.
      PT19   = 0.
      PHISAV =  F(8)
C                    INITIALIZATION KEY
C               SXRA  = SIN( CROSSRANGE ANGLE )
C               GAMMA = CURRENT FLIGHT PATH ANGLE
C               ALT   = CURRENT ALTITUDE (NON-DIMENSIONAL)
C               XRA   = CROSSRANGE ANGLE
C               XR    = CROSSRANGE (NON-DIMENSIONAL)
C               SDRA  = SIN( DOWNRANGE ANGLE )
C               DRA   = DOWNRANGE ANGLE
C               RNG   = DOWNRANGE (NON-DIMENSIONAL)
C               HDA   = CROSS RANGE DEVIATION ANGLE
C
      SRA = F(26)
      HDA = ABS(F(25))
      SXRA= COS(1.5707963-SRA) * COS(1.5707963-HDA)
      XRA = ASIN(SXRA)
      SDRA= TAN(XRA) * TAN(1.5707963-HDA)
      DRA = ASIN(SDRA)
       IF(SRA.GE.1.5707963)DRA=3.1415926-DRA
      RNG = RE * DRA / RO
      XR  = RE * XRA / RO * SGN(F(25))
      CALL UNIT(RIV,F(57))
      CALL UNIT(VATI,F(60))
      Y(1)= ASIN( DOT(F(57),F(60)))
      Y(2)= ALTUDE * 3.28083 / RO
      Y(3)= RNG
      Y(5)= XR
       PY4=Y(4)
       Y(4)=ATAN( (XR-PXR)/(RNG-PRNG) )
       IF( RNG .LE. PRNG ) Y(4) = (3.1415926-ATAN(ABS((XR-PXR)/(RNG-PRNG
     1  ))))*SGN(XR)
```

```
       PXR=XR
       PRNG=RNG
      Y(6)= 0.
      Y(7)= 0.
      Y(8)= 0.
       Y(9)=0.
       Y(10)=0.
       Y(11)=0.
       P03=GAMMA
       P02=GAMMA
       P01=GAMMA
       APRXHD = Y(4)
       RLHD = APRXHD
       T99 = COS( APRXHD)
      VN  = ABVAL(VATI) * 3.28083 / VO
C DESQD IS THE DESIRED HEATING RATE
       DESQD=41.  +  700.*(0.980-VN)
       IF( DESQD .GT. 65. ) DESQD = 65.
       IF( VN .LT. .7) DESQD = 110.*VN - 12.
       IF( VN .GT.  .465) MODE = 1
       IF( VN .LE. .465) MODE = 2
      CUTOFF = 100000.
       X  = VN*COS(Y(1))
      RN  = REN + HN
      GN  = 32.174 * (REN/RN)**2 * RO / VO**2
      TN  = TIMEX * VO / RO
C
C LOAD VARIABLES FOR PRINT AND PLOT
C
      F(63) = TIMEX
      F(64) = RNG * RO / 6076.097
      F(65) = XR * RO / 6076.097
      F(66) = VN
      F(67) = ALTUDE * 3.28083
      IF (F(67) .GT. ATMOSH) GO TO 30
      BETA  = BETA2
      RHONO = RHONO2
   30 CONTINUE
      F(68) = ABVAL(VATI) * 3.28083
      F(69) = ABVAL(DDRLDT) * 3.28083 / 32.174
      F(70) = Y(1) * 57.295795
      F(71) = Y(4) * 57.295795
      F(72) =  F(9) * 57.295795
      F(73) =  F(8) * 57.295795
      F(74) = BIAS
      F(75) = PHISGN
      IF (F(69) .LT. 0.03) GO TO 460
      IF (Y(2)*RO .LT. CUTOFF) GO TO 480
      STPCNT = 0
C PQDOT IS THE VALUE OF THE MAX. HEATING RATE ON THE PREVIOUS CALL
       IF( PQDOT .GT. DESQD +15.) ALPMIN = ALPMIN + 2./57.3
       IF( PQDOT .LT. DESQD+5.)ALPMIN = ALPMIN - 5./57.3
       IF( ALPMIN .LT. 20./57.3) ALPMIN = 20./57.3
       IF(ALPMIN .GT. 40./57.3) ALPMIN=40./57.3
       PQDOT=0.
C*********************************************************************
C
```

```
C                    RUNGE KUTTA LOOP
C
C**************************************************************************
   50 J = 1
C THE PURPOSE OF IDRVCT IS TO PROVIDE A FLAG TO PROHIBIT THE CALCULATION
C   OF SOME QUANTITIES IN THE DERIVATIVE SECTION ON ALL EVALUATIONS
C   EXCEPT THE FIRST
        IDRVCT = 0
       GO TO DERIV
   60 DO 65 JKW=1,NEQS
   65 K1(JKW) = D(JKW)
        IF( STPCNT .GE. 1) GO TO 67
       QDOT  = 17600. * SQRT(T3*RHONO/TRHONO) * VN**3.15
       WRITE(6,5000) QDOT
 5000 FORMAT(1H ,13HCURRENT QDOT=,E20.6)
       F(89) = QDOT
   67 CONTINUE
       SSGN   = T19/ABS(T19)
        H=50./VO*SSGN
       X        = X + 0.5*H
       DO 70 JKW=1,NEQS
   70 Y(JKW) = Y(JKW) + 0.5*H*K1(JKW)
       J        = 2
       GO TO DERIV
   80 DO 85 JKW=1,NEQS
       K2(JKW)= D(JKW)
   85 Y(JKW) = Y(JKW) - 0.5*H*K1(JKW) + 0.5*H*K2(JKW)
       J        = 3
       GO TO DERIV
   90 DO 95 JKW=1,NEQS
       K3(JKW)= D(JKW)
   95 Y(JKW) = Y(JKW) - 0.5*H*K2(JKW) + H*K3(JKW)
       X        = X + 0.5*H
       J        = 4
       GO TO DERIV
  100 DO 105 JKW=1,NEQS
       K4(JKW)= D(JKW)
  105 Y(JKW) = Y(JKW) - H*K3(JKW) + H*.16666666*(K1(JKW)+2.*K2(JKW)+2.*
     1             K3(JKW)+K4(JKW))
        VN= X/COS(GAMMA)
        V = VN*VO
C APRXHD IS THE HEADING USED TO MODIFY THE DOWNRANGE PREDICTION
        APRXHD = APRXHD + LOD*T6*ALOG(   (V+50.)/V)
        IF(ABS( APRXHD*57.3).GT.85.)APRXHD=85./57.3*SGN(APRXHD)
        T99 = COS( APRXHD)
        IF( Y(2)*RO .GT. 350000.) GO TO 460
        IF( (F(66)  - VN) .GT.  DLTV/VO) GO TO 110
C CALCULATION OF SENSITIVITIES FOR PATH CONTROL,
       SENS1  = Y(6)
       SENS2  = Y(7)
       SENS3  = Y(8)
        SENS4= Y(9)
        SENS5= Y(10)
        SENS6= Y(11)
       QDOT   = 17600. * SQRT(T3*RHONO/TRHONO) * VN**3.15
        SENSQD= -17600.*(BETA/2.)*SQRT(T3*RHONO/TRHONO)*VN**3.15*RO
        SNSQD1=  SENSQD*SENS2
```

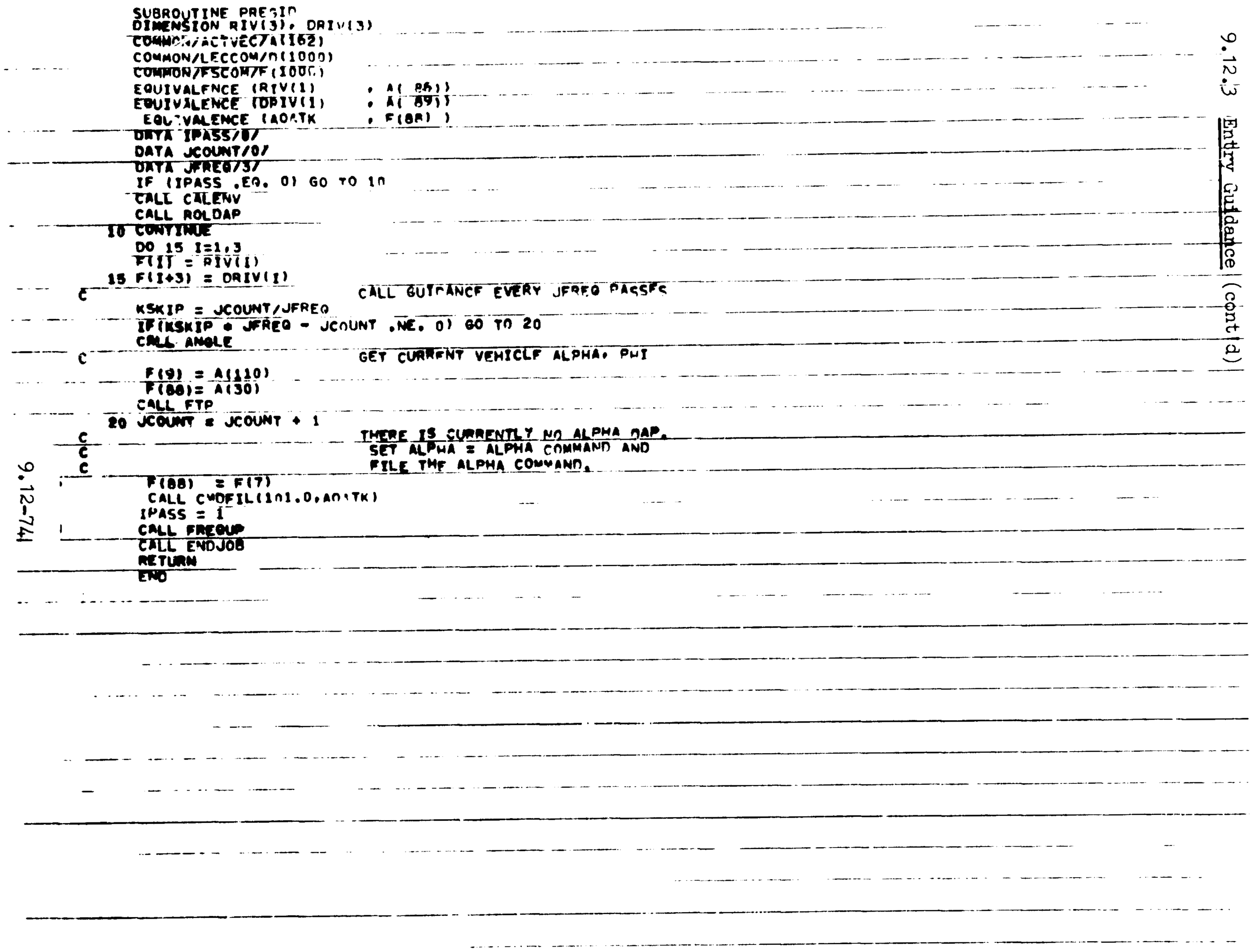
```
      SNSQD2=  SENSQD*SENSE
      SVQD   = QDOT
  110 CONTINUE
       P63=P62
       P62=P61
       P61=GAMMA
      ALT    = HN * RO
      IF ( ALT .LT. CUTOFF) GO TO ENDPT
      IF (ALT .GT. ATMOSH) GO TO 115
      BETA   = BETA2
      RHONO  = RHONO2
  115 CONTINUE
      TN     = TN + H/T19
      PALT   = ALT
      STPCNT = STPCNT + 1
      IF (T19 .GE. PT19) GO TO 120
      PT19   = T19
      ACCVN  = VN
  120 QDOT   = 17600. * SQRT(T3*RHONO/TRHONO) * VN**3.15
      IF (QDOT .LE. PQDOT) GO TO 125
      PQDOT  = QDOT
      QDTVN  = VN
  125 CONTINUE
      IF (STPCNT .EQ. 4) GO TO 150
      DO 140 JKW=1,NEQS
      BKPT1(JKW) = Y(JKW)
      BKDRV4(JKW)= BKDRV3(JKW)
      BKDRV3(JKW) = BKDRV2(JKW)
  140 BKDRV2(JKW) = K1(JKW)
      GO TO 50
  150 DO 155 JKW=1,NEQS
      BKPT1(JKW)  = Y(JKW)
  155 BKDRV1(JKW) = K1(JKW)
C***************************************************************
C
C                    PREDICTOR
C
C***************************************************************
  160 J     = 5
       IDRVCT = 0
      GO TO DERIV
  170 CONTINUE
      SSGN = T19 / ABS(T19)
       H=50./VO*SSGN
      DO 180 JKW=1,NEQS
      YHAT(JKW) = Y(JKW) + H*C1*(1901.*D(JKW)-2774.*BKDRV1(JKW)+2616.*
     1 BKDRV2(JKW)-1274.*BKDRV3(JKW)+251.*BKDRV4(JKW))
  180 BKDRV0(JKW) = D(JKW)
C***************************************************************
C
C                    CORRECTOR
C
C***************************************************************
      X = X + H
      IF (X .GT. 1.5) STOP
      DO 190 JKW=1,NEQS
      YSAV(JKW) = Y(JKW)
  190 Y(JKW)     = YHAT(JKW)
      J          = 6
      GO TO DERIV
  200 DO 205 JKW=1,NEQS
  205 Y(JKW) = YSAV(JKW) + H*C2*(475.*D(JKW) + 1427.*BKDRV0(JKW) - 798.*
     1 BKDRV1(JKW) + 482.*BKDRV2(JKW) - 173.*BKDRV3(JKW) + 27.*
     2 BKDRV4(JKW))
       VN= X/COS(GAMMA)
      V = VN * VO
       APRXHD = APRXHD + LOD*T6*ALOG(  (Y+50.)/V)
       IF(ABS( APRXHD*57.3).GT.50.)APRXHD=50./57.3*SGN(APRXHD)
       T99 = COS( APRXHD)
      IF (Y(2)*RO .GT. 350000.) GO TO 460
C
C*******************************************************************
C
       IF( (F(66)  - VN) .GT.  DLTV/VO) GO TO 210
C CALCULATION OF SENSITIVITIES FOR PATH CONTROL.
      SENS1   = Y(6)
      SENS2   = Y(7)
      SENS3   = Y(8)
       SENS4= Y(9)
       SENS5= Y(10)
       SENS6= Y(11)
      QDOT    = 17600. * SQRT(T3*RHONO/TRHONO) * VN**3.15
       SENSQD= -17600.*(BETA/2.)*SQRT(T3*RHONO/TRHONO)*VN**3.15*RO
      SVQD    = QDOT
       SNSQD1=  SENSQD*SENS2
       SNSQD2=  SENSQD*SENS5
  210 CONTINUE
      ALT     = HN * RO
      IF (T19 .GE. PT19) GO TO 220
      PT19    = T19
      ACCVN   = VN
  220 QDOT    = 17600. * SQRT(T3*RHONO/TRHONO) * VN**3.15
      IF (QDOT .LE. PQDOT) GO TO 230
      PQDOT   = QDOT
      QDTVN   = VN
  230 CONTINUE
       PG3=PG2
       PG2=PG1
       PG1=GAMMA
C
C CHECK FOR SIMULATION TERMINATION
C
      IF (ALT .LT. CUTOFF) GO TO ENDPT
      IF (ALT .GT. ATMOSH) GO TO 240
      BETA    = BETA2
      RHONO   = RHONO2
  240 CONTINUE
      TN      = TN + H/T19
C*******************************************************************
C
C UPDATE STORAGE
C
      DO 250 JKW=1,NEQS
      PALT    = ALT
      BKDRV4(JKW) = BKDRV3(JKW)
      BKDRV3(JKW) = BKDRV2(JKW)
      BKDRV2(JKW) = BKDRV1(JKW)
      HKDRV1(JKW) = BKDRV0(JKW)
  250 BKPT1(JKW)  = Y(JKW)
      STPCNT = STPCNT + 1
      GO TO 160
C*****************************************************************************
C
C                   INTERPOLATION FOR CUTOFF
C
C*****************************************************************************
  300 FRAC   = (CUTOFF - PALT) / (ALT - PALT)
      TN     = TN + FRAC*H/119
      DO 310 JKW=1,NEQS
  310 Y(JKW) = BKPT1(JKW) + FRAC * (Y(JKW)-BKPT1(JKW))
      WRITE(6,5007)SVQD, VN
 5007 FORMAT(1H ,5HSVQD=,E20.6,6HAT VN=,E20.6)
      F(90) = SVQD
      GO TO GUIDE
C*****************************************************************************
C
C                   EVALUATION OF DERIVATIVES
C
C*****************************************************************************
  350 CONTINUE
       IDRVCT = IDRVCT + 1
       IF(ABS(APRXHD).GT. 85./57.3) PHI = -PHI*SGN(APRXHD)
C CALCULATION OF SOUND SPEED, MACH NO., AND AERODYNAMIC COEFFICIENTS
       VN = X/COS(Y(1) )
       VSND(1) = Y(2) * RO
       CALL SPLN2(NSND,SA,SS,SPLSND,VSND)
       PMACH = VN*VO/VSND(2)
       IF( PMACH .LT. 2. ) PMACH = 2.
       PAOATK = 50./57.3 -20./57.3*SIN( 3.1415926/1.2*(0.8-VN))
       IF(  VN .GT.  0.8 )  PAOATK =  50./57.3
       IF(  VN .LT.  0.2 )  PAOATK =  30./57.3
       PAOATK = PAOATK + ALPIAS
       IF( PAOATK .GT. 60./57.3)  PAOATK = 60./57.3
       IF( PAOATK .LT. 20./57.3)  PAOATK = 20./57.3
       CALL AROOYN(PMACH,PAOATK)
C
C                   CALCULATION OF LATITUDE
C
       SMLC= ACOS(  COS(Y(3)*RO/RE)*COS(Y(5)*RO/RE) )
       APRM = ASIN(  SIN(Y(5)*RO/RE)/SIN(SMLC)   )
       A2P =  3.1415926 - ATGA + APRM
       XMU = ASIN(COS(SMLC)*SIN(XMUO)+ SIN(     SMLC)*COS(XMUO)*
     1   COS(A2P) )
C
C ILOFT IS A FLAG USED IN PULL-UP OVERRIDE LOGIC
C
       ILOFT= 0
      GO TO 365
  360 CONTINUE
  365 CONTINUE
C
```

```
C
      T1     = 0.5 * RHONO * CD * SN
      ANGLEI = SI5 - Y(4)
      T2     = OMEGAN * COS(ANGLEI)
      T20    = REN + HN
      RN     = T20
      T15    = BETA * RO
      GN     = 32.174 * (REN/RN)**2 * RO/VO**2
      T21    = T2 * T2
      T7     = COS( PHI )
      T8     = SIN( PSI )
       T9 = COS( PSI)
      T10    = SIN( GAMMA )
      T12    = VN * VN
      T5     = COS( GAMMA )
      T17    = GN / VN
      T22    = RN / VN
      T6     = SIN( PHI )
      T30    = GN - T20 * T21
      T3     = EXP(-(T15)*HN)
      T23    = 1. / T22
      T25    = REN * T23
      T26    = T17 - T22 * T21 - T23
      T27    = T1 * T3 * VN * T7
      T28    = T10 * T25
      T29    = T25 * T5
      T19    = -T1 * T3 * T12 - T10 * T30
      IF (T19 .GE. 0. .AND. HN*RO .GT. 280000.) GO TO 460
       HDDN= VN*(T27/(CD*T7)*(CL*T7-CD*T10) + (T25-T17) )
       DMPGN = 1.0/(LOD*ABS(SIN(PHI)) ) )
       PHIDMP= (0.3 *57.3)/(-LOD*SIN(PHI))*(-GAMMA +0.66*P61 + 0.224*
     1  P62  + 0.076*P63 )
       IF( ABS(PHIDMP ) .GT. 20./57.3 )PHIDMP=20./57.3*SGN(PHIDMP)
       IF( ABS(PHIDMP) .GE. ABS(PHI) )PHIDMP=0.8*ABS(PHI)*SGN(PHIDMP)
       IF( PAOATK .GT. 50./57.3) PHIDMP = 0.
       T7 = COS( PHI + PHIDMP*SGN(PHI) )
       T27= T1 * T3 * VN * T7
  369 CONTINUE
      IF (IFLAG .EQ. 1) GO TO 370
C
C CALCULATE CONSTANTS FOR FINITE ROLL-RATE CROSSRANGE PREDICTION
C
       PHIDLT = PHIDMP
      SVLOD  = LOD
       MSAV=PMACH
      ACSAVE = T19
      VNI    = VN
      TRAV   = ABS(PHI) + PHISGN*F(9)
      SWTCHL = -TRAV/(3./57.3) * ACSAVE * VO / RO
      IFLAG  = 1
  370 IF ((VNI-VN) .GE. SWTCHL) GO TO 380
      PPHI   =  F(9) - PHISGN*TRAV/SWTCHL*(VNI-VN)
      IF (SWTCHL .LE. 0.) PPHI=-PHI
      GO TO 390
C
C PPHI IS THE ROLL ANGLE USED IN THE CROSS RANGE PREDICTION
C
```

```
 380 PPHI   = -PHI
 390 CONTINUE
     T24    = 1. / T19
     T11    = T24 / T19
     T13    = -T11 * T5 * T30
     T14    = -T11 * (T1 * T3 * T15 * T12 + T10 * T21)
      T40 =   -T1/CD * T12 * T3
      T55= T27*LOD- T5*T26+2.*T2
      T56=    1./( T19*T5 - VN*T10*T55 )
      T85=  T56*T56
      T60=  T1*T3*T12
      T61=  T5 + LOD*T7*T10
      T86=  -T60*(-T10 + LOD*T7*T5)  -T23*VN*(T5*T5 - T10*T10)
    1      -2.*T2*VN*T5
      T87 =    T60*T15*T61 + T23*T23*T5*T10
      T81= (T17 - T22*T21 - T23)*T10
      T72 = VN*T10
      T73=  T25*T3 * T9
      T82 = -T27*LOD*T15 + T5/VN*T21-T23*T23/VN*T5
      T83 = VN*T5
      T84 = -T25*T10
      T90 = -T25/T20*T5
      A11= T56*T81-T85*T55*T86
      A12= T56*T82-T85*T55*T87
      A13=0.
      A21=  T56*T83-T85*T72*T86
      A22= -T85*T72*T87
      A23= 0.
      A31= T56*T84 -T85*T73*T86
      A32= T56*T90 -T85*T73*T87
      A33= 0.
      PFLOD1=  T56*T27 - T85*T55*(-T60*T10)
      PFLOD2=   -T85*T72*(-T60*T10)
      PFLOD3=   -T85*T73*(-T60*T10)
      PFCD1=   T56*T27*LOD -T85*T55*(-T60/CD*T61)
      PFCD2=   -T85*T72*(-T60/CD*T61)
      PFCD3=   -T85*T73*(-T60/CD*T61)
      A31 = A31*T99
      A32 = A32*T99
      A33 = A33*T99
      PFLOD3 = PFLOD3*T99
      PFCD3 = PFCD3*T99
      D(1) =  T56*( T27*LOD -T5*T26 +2.*T2 )
      D(2) =  T55*( VN*T10)
      D(3)=T56*T25*T5*T99
      D(4) =  T56*(T27/T7 *SIN(PPHI)*LOD + 2.*OMEGAN*SIN(ANGLEI)
    1   * SIN(XSII) )
      D(5)= T56*VN*T8*T5
      D(6) = A11*Y(6)+A12*Y(7)  + PFLOD1
      D(7) = A21*Y(6)+A22*Y(7)  + PFLOD2
      D(8) = A31*Y(6)+A32*Y(7)  + PFLOD3
      D(9) = A11*Y(9)+A12*Y(10) + PFCD1
      D(10)= A21*Y(9)+A22*Y(10) + PFCD2
      D(11)= A31*Y(9)+A32*Y(10) + PFCD3
     GO TO (60,90,90,100,170,200), J
C*********************************************************************
C
```

```
C                    COMPUTE SHIFT OF (L/D)COS(PHI)
C
C*******************************************************************
  400 CONTINUE
      DNRNG  = Y(3) * RO / 6076.097
      IF (DNRNG .LT. 0.) GO TO 460
      CRSRNG = Y(5) * RO / 6076.097
      VN     = ABVAL(VATI) * 3.28083 / VO
      DLTR   = DESRNG - Y(3)
      DLTQD  = DESQD  - SVQD
C EVALUATION OF THE PARTIALS OF THE AERO WITH RESPECT TO ALPHA
       CALL ARODYN(MSAV,  F(7))
       APRXCL = ENT(1)
       APRXCD = ENT(2)
       PCLA = ENT(  7)
       PCDA = ENT(  8)
       PLODA=  (APRXCD*    PC A          -(APRXCL*PCDA))/(APRXCD**2)
       SENSR  = Y(8)
      IF(ABS(DLTQD) .GT. 3.)  DLTQD=3.*SGN(DLTQD)
       IF(ABS(DLTR).GT.50.*6076.097/RO)DLTR=50.*6076.097/RO*SGN(DLTR)
       SCLCST=1.
       PAR11 =  SNSQD1 * COS(PHI) * PLODA  + SNSQD2 *  PCDA
       PAR12= SNSQD1*(-LOD*SIN(PHI))
      PAR11  = PAR11  * SCLCST
      PAR12  = PAR12  * SCLCST
       PAR21 =  SENSR * COS(PHI) * PLODA  +  Y(11) * PCDA
      PAR22  = SENSR  * (-LOD * SIN(PHI))
      DET    = PAR11  * PAR22 - PAR21 * PAR12
       CONST = 0.2
       IF( DLTR .LT. 0. .AND. MODE .EQ. 2) CONST=.005
       PROPOR =  (CONST*PAR11*PAR21 + PAR12*PAR22)/(PAR12**2+CONST*
     1   PAR11**2 )
       IF( VN .LT. .253 .AND. VN .GT. .272) GO TO 410
       IF( MODE.EQ.1)  DLTR= PROPOR* DLTQD
       IF(MODE.EQ.2) DLTQD=DLTR*(PAR12*PAR22+CONST*PAR11*PAR21)/(
     1  PAR22**2+CONST*PAR21**2)
  410 CONTINUE
C CHECK FOR SYSTEM SINGULARITY
       IF( ABS(DET)   .LE. 1.E-15 ) DET=1.E-15
      DALPHA = (1./DET) * (PAR22*DLTQD*SCLCST-PAR12*DLTR)
      DPHI   = (1./DET) * (-PAR21*DLTQD*SCLCST + PAR11*DLTR)
C LIMITING LOGIC ON ALPHA CHANGE
      IF( ABS(DALPHA) .LE. 3.1/57.3 ) GO TO 415
       PRP =    (3./57.3)/ ABS( DALPHA)
       DALPHA=PRP*DALPHA
       DPHI=PRP*DPHI
  415 CONTINUE
C LIMITING LOGIC ON PHI CHANGE
       IF( ABS(DPHI) .LE. 10.1/57.3 ) GO TO 420
       PRP= (10./57.3)/ABS(DPHI)
       DALPHA=PRP*DALPHA
       DPHI=PRP*DPHI
  420 CONTINUE
      IF( AOATK .GT.  58./57.3 .AND. DALPHA .GT. 0.)GO TO 422
      IF( AOATK .LT. ALPMIN+2./57.3 .AND. DALPHA .LT. 0.)GO TO 422
       IF(ABS(PHI).GT.85./57.3.AND. ABS(SGN(DPHI)-SGN(PHI)).LT. 1.E-05)
     1    GO TO 425
```

```
      IF(ABS(PHI).LT.10./57.3.AND. ABS(SGN(DPHI)-SGN(PHI)).GT. 1.E-05)
     1   GO TO 425
      GO TO 428
C PHI SATURATION LOGIC
  422 CONTINUE
       IF(MODE .EQ. 1 )  DPHI= DLTOD/PAR12
       IF(MODE .EQ. 2 )  DPHI= DLTR/ PAR22
       IF( ABS(DPHI) .GT. 20./57.3) DPHI=20./57.3*SGN(DPHI)
       IF( ABS(DPHI) .GT.  10./57.3)DPHI = 10./57.3*SGN(DPHI)
      GO TO 428
C ALPHA SATURATION LOGIC
  425 CONTINUE
       IF(MODE .EQ. 1)  DALPHA= DLTOD/PAR11
       IF(MODE .EQ. 2)  DALPHA= DLTR/PAR21
       IF(ABS(DALPHA).GT. 3./57.3) DALPHA = 3./57.3*SGN(DALPHA)
  428 CONTINUE
       PHI =  PHI + DPHI
       ALBIAS = ALBIAS + DALPHA
       TAOATK = 50./57.3 -20./57.3*SIN(3.1415926/1.2*(0.8-VN) )
       IF( VN .GT. 0.8)  TAOATK = 50./57.3
       IF( VN .LT. 0.2)  TAOATK = 30./57.3
       AOATK=  TAOATK + ALBIAS
       IF(AOATK  .GT. 60./57.3) AOATK = 60./57.3
       IF( AOATK  .LT.  ALPMIN ) AOATK = ALPMIN
       IF(ABS(PHI) .GT.  85./57.3)PHI= 85./57.3*SGN(PHI)
       IF( ALBIAS .GT. 60./57.3 - TAOATK ) ALBIAS= 60./57.3-TAOATK
       IF( ALBIAS .LT. ALPMIN - TAOATK )  ALBIAS =  ALPMIN -TAOATK
  430 CONTINUE
      F(7)    = AOATK
C*****************************************************************
C
C                 CROSSRANGE LOGIC FOR NON-ZERO DESIRED CROSSRANGES
C
C*****************************************************************
       Z= 378.3*VN**2 -105.*VN +26.7
      IF (VN .GT. VDB) GO TO 430
      UXR     = SGN(XR)
      UY5     = SGN(Y(5) - DSRDCR)
      IF(  ABS(UY5-PHISGN ) .LE. 1.E-10 )  GO TO 440
      IF (ABS(Y(5) - DSRDCR)*RO/6076.097 .GT. Z) GO TO 450
      IF (ABS(Y(5) - DSRDCR) .GT. SWTMAG) GO TO 450
      PHISGN = -PHISGN
       VDB= VN-.03
      SWTMAG = ABS(Y(5) - DSRDCR)
      IF (SWTMAG*RO/6076.097 .LE. 0.5) SWTMAG=0.5*6076.097/RO
      GO TO 450
  440 PHISGN = -UY5
       VDB=VN-.03
  450 CONTINUE
       IF(ABS(RLHD)  .GT. 85./57.3) PHISGN= -SGN(RLHD)
      PHI     = ABS(PHI) * PHISGN
       PHIHLD=PHI
       PHI = PHI + PHIDLY * SGN(PHI)
       IF( ABS(PHI) .LE. 10./57.3)PHI=10./57.3*SGN(PHI)
      F(8)    =  PHI
      F(76)   = Z
      F(77)   = Y(3)*RO/6076.097
      F(78)  = Y(5)*RO/6076.097
      GO TO 500
C*************************.****************.***.********************************
C
C
C*************************.****************.***.********************************
  460 CONTINUE
       IF( T19 .GT. 0.)WRITF(6, 2561)
       IF( T19 .LT. 0.)WRITF(6, 2562) T19
      IF( Z(69)  .LT.    .03) WRITE(6,2562) F(69)
 2562  FORMAT(1H ,32HGUIDANCE BYPASSED. ACC TOO SMALL,3X,4HACC=,F20.8)
       IF( Y(2)*RO .GT. 350000.) WRITF(6,2563)
 2563  FORMAT(1H ,31HGUIDANCE BYPASSED. ALT TOO HIGH,3X )
 2561  FORMAT(1H ,39HVELOCITY INCREASING. GUIDANCE BYPASSED.)
      GO TO 500
  480 FRAC2  = (CUTOFF - F(82)) / (F(67) - F(82))
      DO 490 J=1,9
  490 F(62+J) = F(78+J) + FRAC2 * (F(62+J)-F(78+J))
      LFLAG   = 1
C*************************.****************.***.********************************
C
C PRINT SECTION
C
C
  500 CALL FTPPRN
C
C*************************.****************.***.********************************
C
       IF (LFLAG .EQ. 1) CALL TERMIN
       DO 510 J=1,9
  510  F(78+J) = F(62+J)
       RETURN
       END
```



```
      SUBROUTINE PRESID
      DIMENSION RIV(3), DRIV(3)
      COMMON/ACTVEC/A(162)
      COMMON/LECCOM/D(1000)
      COMMON/FSCOM/F(1000)
      EQUIVALENCE (RIV(1)       , A( 86))
      EQUIVALENCE (DRIV(1)      , A( 89))
       EQUIVALENCE (AOATK       , F(88) )
      DATA IPASS/0/
      DATA JCOUNT/0/
      DATA JFREQ/3/
      IF (IPASS .EQ. 0) GO TO 10
      CALL CALENV
      CALL ROLDAP
   10 CONTINUE
      DO 15 I=1,3
      F(I) = RIV(I)
   15 F(I+3) = DRIV(I)
C                                    CALL GUIDANCE EVERY JFREQ PASSES
      KSKIP = JCOUNT/JFREQ
      IF(KSKIP * JFREQ - JCOUNT .NE. 0) GO TO 20
      CALL ANGLE
C                                    GET CURRENT VEHICLE ALPHA, PHI
       F(9) = A(110)
       F(88)= A(30)
      CALL FTP
   20 JCOUNT = JCOUNT + 1
C                                    THERE IS CURRENTLY NO ALPHA DAP.
C                                     SET ALPHA = ALPHA COMMAND AND
C                                     FILE THE ALPHA COMMAND.
       F(88)  = F(7)
       CALL CMDFIL(101.0,AOATK)
      IPASS = 1
      CALL FREQUP
      CALL ENDJOB
      RETURN
      END
      SUBROUTINE FTPPRN
      COMMON/FSCOM/F(1000)
      EQUIVALENCE (PT19         , F(92))
      EQUIVALENCE (ACCVN        , F(93))
      EQUIVALENCE (PQDOT        , F(94))
      EQUIVALENCE (QDTVN        , F(95))
      EQUIVALENCE (TN           , F(96))
      DATA RO   /2.1326428E+07/
      DATA VO   /2.569006E+04/
C******************************************,***,**********************************
C
C                     PRINT SECTION
C
C LOAD PREDICTED RANGES FOR PRINT AND PLOT
C
C******************************************,***,**********************************
  500 WRITE(6,5030) (F(J),J=63,78)
 5030 FORMAT(1H0,5X,4HTIME,6X,E15.8,7X,6HDWNRNG,4X,E15.8,7X,6HCRSRNG,4X,
     1 E15.8,7X,2HVN,8X,E15.8,/,6X,3HALT,7X,E15.8,7X,5HVRMAG,5X,F15.8,7X
     2 ,6HACCMAG,4X,E15.8,7X,6HRELOAM,4X,E15.8,/,6X,3HPSI,7X,E15.8,7X,
     3 4HROLL,6X,E15.8,7X,6HROLCMD,4X,E15.8,7X,4HBIAS,6X,F15.8,/,6X,
     4 6H[illegible]ISON,4X,E15.8,7X,5HSWTCH,5X,E15.8,7X,6HPREDDR,4X,E15.8,7X,
     5 6HPREDCR,4X,E15.8 )
      T       = TN * RO / VO
      WRITE(6,5040) T
 5040 FORMAT(1H ,23HESTIMATED FINAL TIME=  ,E13.6)
      AOATKD = F(7) * 57.3
      WRITE(6,5050) AOATKD
 5050 FORMAT(1H ,26HCOMMANDED ANGLE OF ATTACK=,F20.6)
      F(91) = AOATKD
      PT19    = PT19 * VO**2 / (RO*32.174)
      WRITE(6,5060) PT19, ACCVN, PQDOT, QDTVN
 5060 FORMAT(1H ,20HPREDICTED MAX ACC IS,F20.6,10X,18HOCCURING AT VN=
     1,E20.6,/,1X,21HPREDICTED MAX QDOT IS,E20.6,9X,18HOCCURING AT VN=
     2 ,E20.6,//)
      RETURN
      END
      SUBROUTINE ANGLE
      COMMON/LECCOM/D(1000)
      COMMON/ACTVEC/A(162)
      COMMON/ATMOSC/AT(80)
      COMMON/FSCOM/F(1000)
      EQUIVALENCE (TIME        , D(   1))
      DIMENSION RIV(3) , VATI(3)
      EQUIVALENCE (RIV(1)      , A( 86))
      EQUIVALENCE (VATI(1)     , AT( 8))
      EQUIVALENCE (SLTRNG      , F(  42))
      DATA IPASS/0/
      OMEGA = 7.29211508E-05
      IF (IPASS .EQ. 1) GO TO 10
C
C                 PUT INITIAL POSITION VECTOR IN FSCOM LOCATIONS 54-56.
C
      DO 5 I=1,3
    5 F(I+53) = RIV(I)
C                 COMPUTE NORMAL TO INITIAL R-V PLANE AND STORE IN
C                 FSCOM LOCATIONS 27,28, AND 29.
      CALL CROSS(F(54) ,VATI(1),F(27))
      IPASS = 1
      RETURN
C                 ROTATE INITIAL R AND NORMAL TO INITIAL R-V PLANE THRU
C                 OMEGA*TIME. STORE ROTATED INITIAL R IN FSCOM LOCATIONS
C                 30,31, AND 32.
C                 STORE ROTATED NORMAL TO INITIAL R-V PLANE IN FSCOM
C                 LOCATIONS 33,34, AND 35.
   10 F(30) = F(54) *COS(OMEGA*TIME) - F(55) *SIN(OMEGA*TIME)
      F(31) = F(54) *SIN(OMEGA*TIME) + F(55) *COS(OMEGA*TIME)
      F(32) = F(56)
      F(33) = F(27)*COS(OMEGA*TIME) - F(28)*SIN(OMEGA*TIME)
      F(34) = F(27)*SIN(OMEGA*TIME) + F(28)*COS(OMEGA*TIME)
      F(35) = F(29)
C                 COMPUTE NORMAL TO PLANE OF CURRENT R AND THE ROTATED
C                 INITIAL R AND STORE IN FSCOM LOCATIONS 36,37,38.
      CALL CROSS(F(30),RIV(1),F(36))
C                 CROSS ROTATATED INITIAL NORMAL WITH ROTATED INITIAL
C                 R, STORE IN FSCOM LOCATIONS 39,40,41.
      CALL CROSS(F(33),F(30),F(39))
C                 COMPUTE CROSSRANGE DEVIATION ANGLE AND STORE IN
C                 FSCOM 25.
      ARG = DOT(F(36),F(39)) / (ABVAL(F(36))*ABVAL(F(39)))
      IF (ABS(ARG) .GT. 1.) ARG=SGN(ARG)
      F(25) = ASIN(ARG)
C                 COMPUTE SLANTRANGE ANGLE, STORE IN FSCOM 26.
      ARG = DOT(RIV(1),F(30)) / (ABVAL(RIV(1))*ABVAL(F(30)))
      IF (ABS(ARG) .GT. 1.) ARG=SGN(ARG)
      F(26) = ACOS(ARG)
      ANGL = F(25) * 57.3
      SLTRNG = 20926428. * F(26) / 6076.097
      RETURN
      END
      SUBROUTINE ARODYN(FMACH,ALPHA)
      COMMON/ENTCST/E(10)
      EQUIVALENCE (E(  1),CL0),(E(  2),CD0),(E(  3),S),(E(  4),W)
     *            ,(E(  5),FLD),(E(  6),WCDA)
     *            ,(E(  7),DCL),(E(  8),DCD)
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      DIMENSION FM(7),A(15),CL(15,7),CD(15,7)
      DIMENSION CCL(42,7),CCD(42,7),V(5),D(2),T(42)
      DATA INIT/0/ NA/15/ NM/7/
C
C                               * * *   NAR 161C    * * *
C                               NORTH AMERICAN ROCKWELL
C                               DELTA WING ORBITER DATA
C                               MASS 8330 SLUGS - AREA 6650 SQ FT
C
C                               SPLINE INTERPOLATION ON ANGLE OF ATTACK
C                               LINEAR INTERPOLATION ON MACH NUMBER
C
      DATA S/6650./  W/268009.824/
      DATA FM/
     *   2.0,3.0,4.0,5.0,7.5,10.0,20.0/
      DATA A/
     *   0.0,5.0,7.5,10.0,12.5,15.0,20.0,25.0,30.0,35.0,40.0,
     *   45.0,50.0,55.0,60.0/
      DATA CL/
     1  -.0030,  .1200,  .1700,  .2365,  .2990,  .3640,  .4950,  .6250,
     1   .7510,  .8590,  .9370, 1.0120, 1.0520, 1.0620, 1.0430,
     2  -.0280,  .0680,  .1430,  .2040,  .2660,  .3280,  .4490,  .5600,
     2   .6660,  .7620,  .8460,  .9100,  .9430,  .9550,  .9290,
     3  -.0297,  .0616,  .1150,  .1720,  .2254,  .2850,  .3939,  .5106,
     3   .6810,  .7150,  .7920,  .8500,  .8910,  .8970,  .8740,
     4  -.0403,  .0460,  .0965,  .1450,  .2003,  .2550,  .3600,  .4700,
     4   .5690,  .6750,  .7690,  .8420,  .8820,  .8830,  .8550,
     5  -.0295,  .0277,  .0630,  .0998,  .1430,  .1936,  .2990,  .4120,
     5   .5260,  .6340,  .7360,  .8170,  .8610,  .8690,  .8400,
     6  -.0259,  .0270,  .0615,  .0968,  .1411,  .1895,  .2870,  .3930,
     6   .5120,  .6250,  .7290,  .8120,  .8620,  .8720,  .8420,
     7  -.0269,  .0220,  .0485,  .0809,  .1202,  .1586,  .2600,  .3760,
     7   .4970,  .6160,  .7230,  .8020,  .8520,  .8610,  .8360/
      DATA CD/
     1   .0667,  .0825,  .0902,  .1048,  .1240,  .1488,  .2155,  .3109,
     1   .4620,  .6271,  .8148, 1.0395, 1.2867, 1.5564, 1.8616,
     2   .0560,  .0640,  .0752,  .0913,  .1131,  .1415,  .2105,  .2982,
     2   .4167,  .5564,  .7292,  .9230, 1.1395, 1.3840, 1.6350,
     3   .0489,  .0536,  .0630,  .0793,  .0984,  .1254,  .1921,  .2792,
     3   .3892,  .5923,  .6806,  .8653, 1.0790, 1.3020, 1.5510,
     4   .0457,  .0473,  .0549,  .0662,  .0835,  .1068,  .1665,  .2514,
     4   .3597,  .4948,  .6611,  .8558, 1.0665, 1.2900, 1.5095,
     5   .0349,  .0366,  .0422,  .0507,  .0642,  .0833,  .1369,  .2192,
     5   .3286,  .4645,  .6341,  .8272, 1.0355, 1.2540, 1.4775,
     6   .0325,  .0355,  .0409,  .0490,  .0624,  .0801,  .1307,  .2082,
     6   .3194,  .4609,  .6273,  .8286, 1.0350, 1.2565, 1.4800,
     7   .0309,  .0333,  .0380,  .0454,  .0559,  .0705,  .1195,  .1999,
     7   .3119,  .4533,  .6213,  .8114, 1.0220, 1.2410, 1.4680/
C
      IF(INIT.NE.0) GO TO 100
C                               CONVERT ALPHA ARRAY TO RADIANS
```

9.12-77

9.15.77

```
      DO 5 I=1,NA
    5 A(I) = A(I)*.017453292
C                                ON FIRST PASS, GENERATE THE
C                                 CUBIC COEFFICIENTS AND STORE
      DO 10 M=1,NM
C                                DEFINE 1ST DERIVATIVES AT END POINTS FOR CL
      D(1)    = (CL(2,M)-CL(1,M))/(A(2)-A(1))
      D(2)    = (CL(NA,M)-CL(NA-1,M))/(A(NA)-A(NA-1))
      J=1
C                                GET CL CUBIC COEFFICIENTS
      CALL SPLN1(NA,A,CL(1,M),J,D,CCL(1,M),T)
C                                DEFINE 1ST DERIVATIVES AT END POINTS FOR CD
      D(1)    = (CD(2,M)-CD(1,M))/(A(2)-A(1))
      D(2)    = (CD(NA,M)-CD(NA-1,M))/(A(NA)-A(NA-1))
      J=1
      CALL SPLN1(NA,A,CD(1,M),J,D,CCD(1,M),T)
   10 CONTINUE
C
      INIT=1
  100 CONTINUE
      MFLAG=0
C                                DETERMINE MACH INTERVAL
C                                IF MACH LESS THAN FM(1) USE FM(1) DATA
      IF(FMACH.GT.FM(1)) GO TO 110
      MNDX=1
      MFLAG=-1
      GO TO 200
C                                IF MACH MORE THAN FM(NM) USE FM(NM) DATA
  110 CONTINUE
      IF(FMACH.LT.FM(NM)) GO TO 120
      MNDX=NM
      MFLAG=1
      GO TO 200
C
  120 DO 130 I=2,NM
      IF(FMACH.LE.FM(I))GO TO 150
  130 CONTINUE
  150 MNDX=I
  200 CONTINUE
      V(1)    = ALPHA
C                                IF ALPHA IS BEYOND THE SCOPE OF THE DATA,
C                                USE THE 1ST OR LAST TABLE VALUE.
      IF(V(1).LT.A(1)) V(1)=A(1)
      IF(V(1).GT.A(NA)) V(1)=A(NA)
      CALL SPLN2(NA,A,CL(1,MNDX),CCL(1,MNDX),V)
      CL1     = V(2)
      DCL1    = V(3)
      CALL SPLN2(NA,A,CD(1,MNDX),CCD(1,MNDX),V)
      CD1     = V(2)
      DCD1    = V(3)
      IF(MFLAG) 210,220,210
  210 CL0     = CL1
      CD0     = CD1
      DCL     = DCL1
      DCD     = DCD1
      GO TO 300
  220 CONTINUE
```

```
      CALL SPLN2(NA,A,CL(1,MNDX-1),CCL(1,MNDX-1),V)
      CL2   = V(2)
      DCL2  = V(3)
      CALL SPLN2(NA,A,CD(1,MNDX-1),CCD(1,MNDX-1),V)
      CD2   = V(2)
      DCD2  = V(3)
C
      GRAD  = (FMACH-FM(MNDX))/(FM(MNDX-1)-FM(MNDX))
      CL0   = CL1 + GRAD*(CL2-CL1)
      CD0   = CD1 + GRAD*(CD2-CD1)
      DCL   = DCL1 + GRAD*(DCL2-DCL1)
      DCD   = DCD1 + GRAD*(DCD2-DCD1)
C
  300 CONTINUE
      FLD   = CL0/CD0
      WCDA  = W/(CD0*S)
      RETURN
      END
```

SPACE SHUTTLE

GN&C SOFTWARE EQUATION SUBMITTAL

Software Equation Section Entry Autopilot Submittal No. 33

Function Blended DFCS for Entry

Module No. 0C4 Function No. 1,2,3,4 (MSC 03690 Rev. B)

Submitted by: G. L. Zacharias Co. EG2

Date: October 21, 1971

NASA Contact: W. H. Peters Organization EG2

Approved by Panel III K.J. Cox (Chairman) Date 10/21/71

Summary Description: This submittal defines an entry DFCS which controls stability axis attitude using both the aerodynamic control surfaces and the ACPS jets. The design minimizes variations in airframe dynamics and uses a blended approach for aerodynamic/ACPS control.

Shuttle Configuration: (Vehicle, Aero Data, Sensor, Et Cetera)

NAR 161C (Rev. A)

Comments:

(Design Status)

(Verification Status)

Panel Comments:

9.12.4 A PRELIMINARY DESIGN FOR A BLENDED ENTRY DFCS

INTRODUCTION

This chapter presents a preliminary design for SSV attitude control during entry, using both the ACPS and ACS (aerodynamic control surface) torques in an integrated or "blended" approach. In addition to providing a means of minimizing ACPS fuel expenditures (by blending with the ACS torques), a prime objective of the design approach is to minimize variations in the closed-loop attitude dynamics due to variations in the open-loop airframe dynamics.

The material presented here is a summary of the design synthesis and gain computation documented in references 1 through 4. Current changes to the latest documented design (given in reference 1) have been implemented in this work.

This chapter is organized into three sections: Section 1 describes the trajectory parameters defining the entry operational environment of the NR 161C vehicle used for controller design; section 2 describes the longitudinal control channel; while section 3 describes the lateral control channel.

1. Trajectory Parameters

The trajectory parameters presented here were generated by the Statistical Analysis Section (EG2), using a point mass simulation with pitch and bank angle modulation to control heating, loads, and targeting. Given in Table I are the initial entry conditions for two NR 161C vehicle trajectories: A low cross-range, low down-range entry (CR: 370 mi: DR 4000 mi), and a high cross-range, high down-range entry CR 1150 mi: DR: 6490 mi), herein designated as trajectories A and B respectively. (Initial conditions are identical.)

Table I. - ENTRY TRAJECTORY INITIAL CONDITIONS

Parameter	Value
V_T	24,409 ft/sec
α_T	53 deg
γ_o	-0.9 deg
h	395,052 ft
ψ	90 deg
Lat	0 deg
Long	81.53 deg

Shown in figures 1 through 5 are the pertinent entry trajectory parameters, for both trajectories. Plotted against Mach number M in the first three figures are: Trim angle of attack α_T, dynamic pressure $\hat{q}$, and flight path angle γ_o . Plotted against time from entry interface t in the last two figures are: Mach number M and total relative velocity V_T . It should be noted that in addition to a large parametric variation along a particular trajectory, there is also a significant variation between trajectories.

2. Longitudinal Control

This section presents the control block diagrams and associated gains for longitudinal control, or equivalently, control of the vehicle's angle of attack α. The basic design procedure, described in reference 2, involves a separation of the ACS and ACPS control loop synthesis, with the addition of an interface logic to define the operational region in which the two control torque systems operate simultaneously. Further, the design of the ACS control loop is predicated on closed-loop dynamic insensitivity to variations in the airframe characteristics.

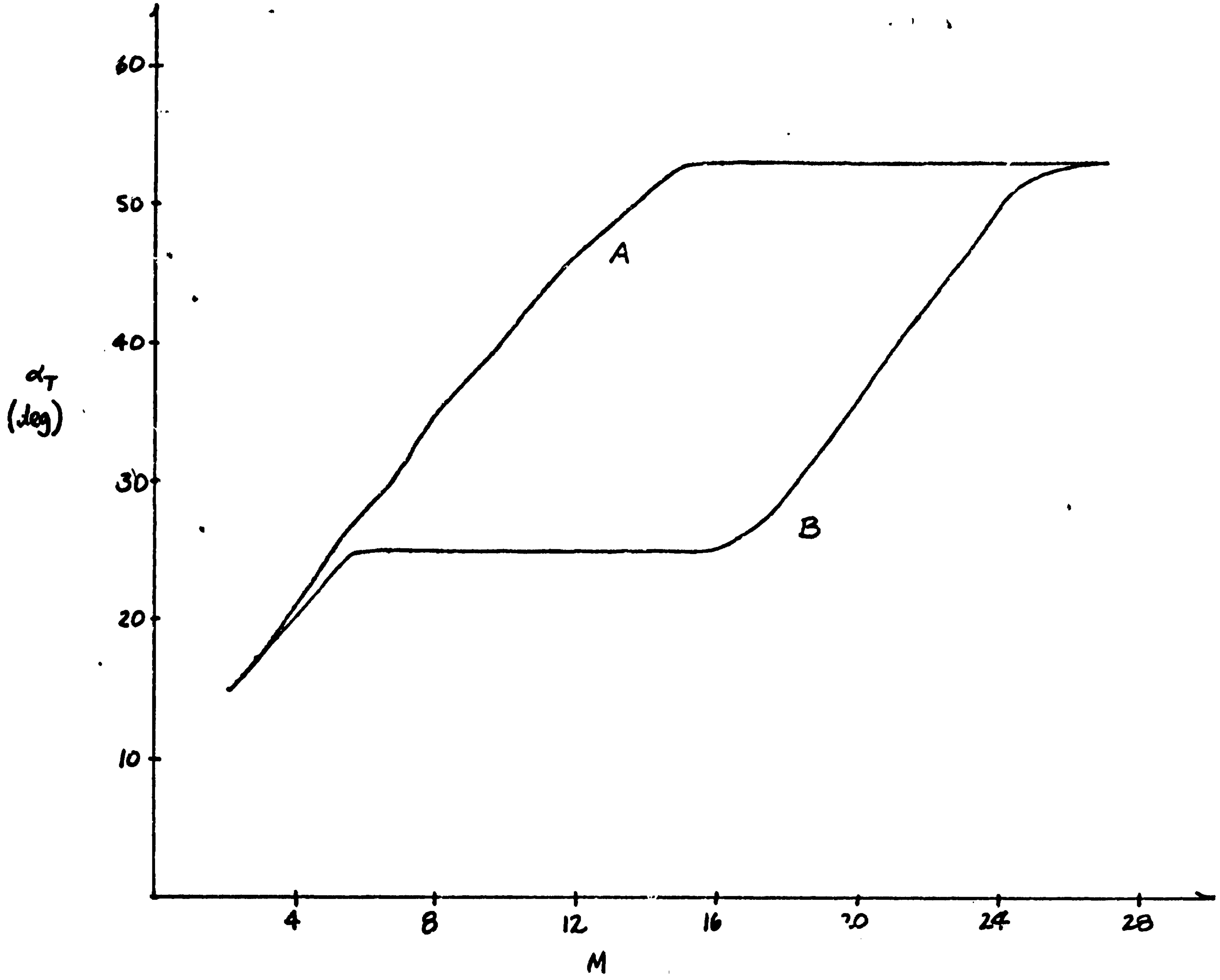


FIGURE 1.- TRIM ANGLE OF ATTACK

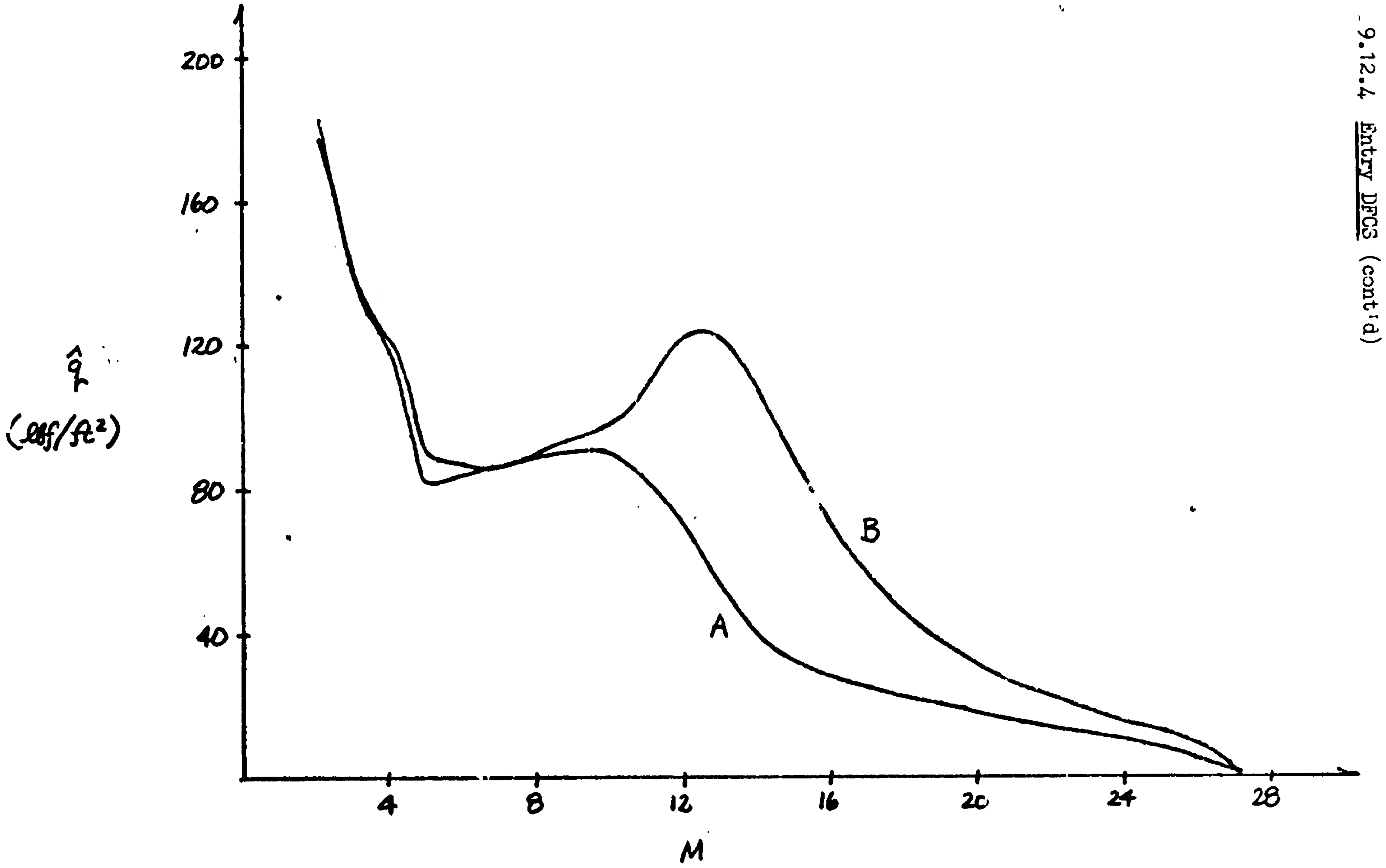


FIGURE 2.- DYNAMIC PRESSURE

9-12-85

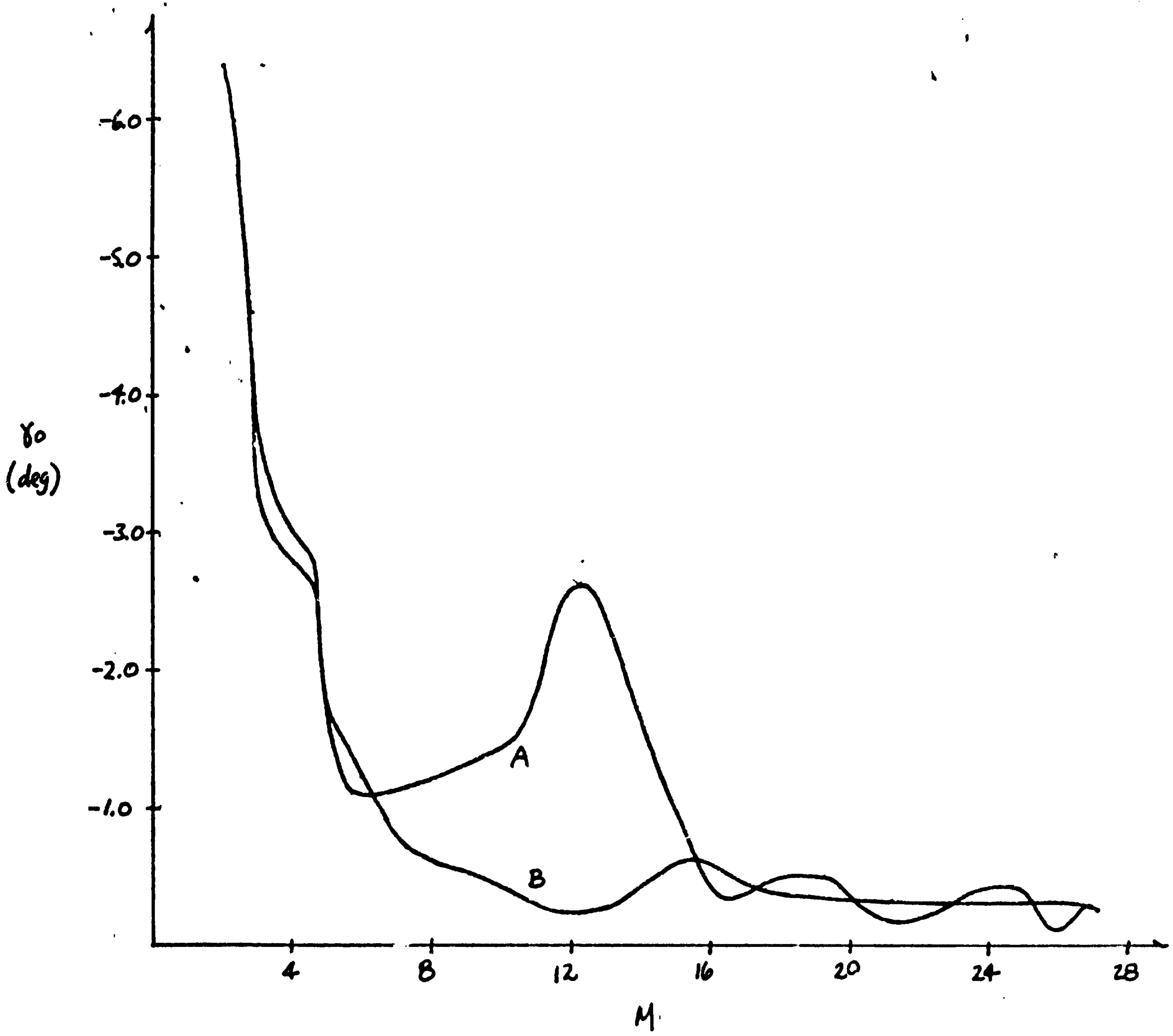


FIGURE 3.- FLIGHT PATH ANGLE

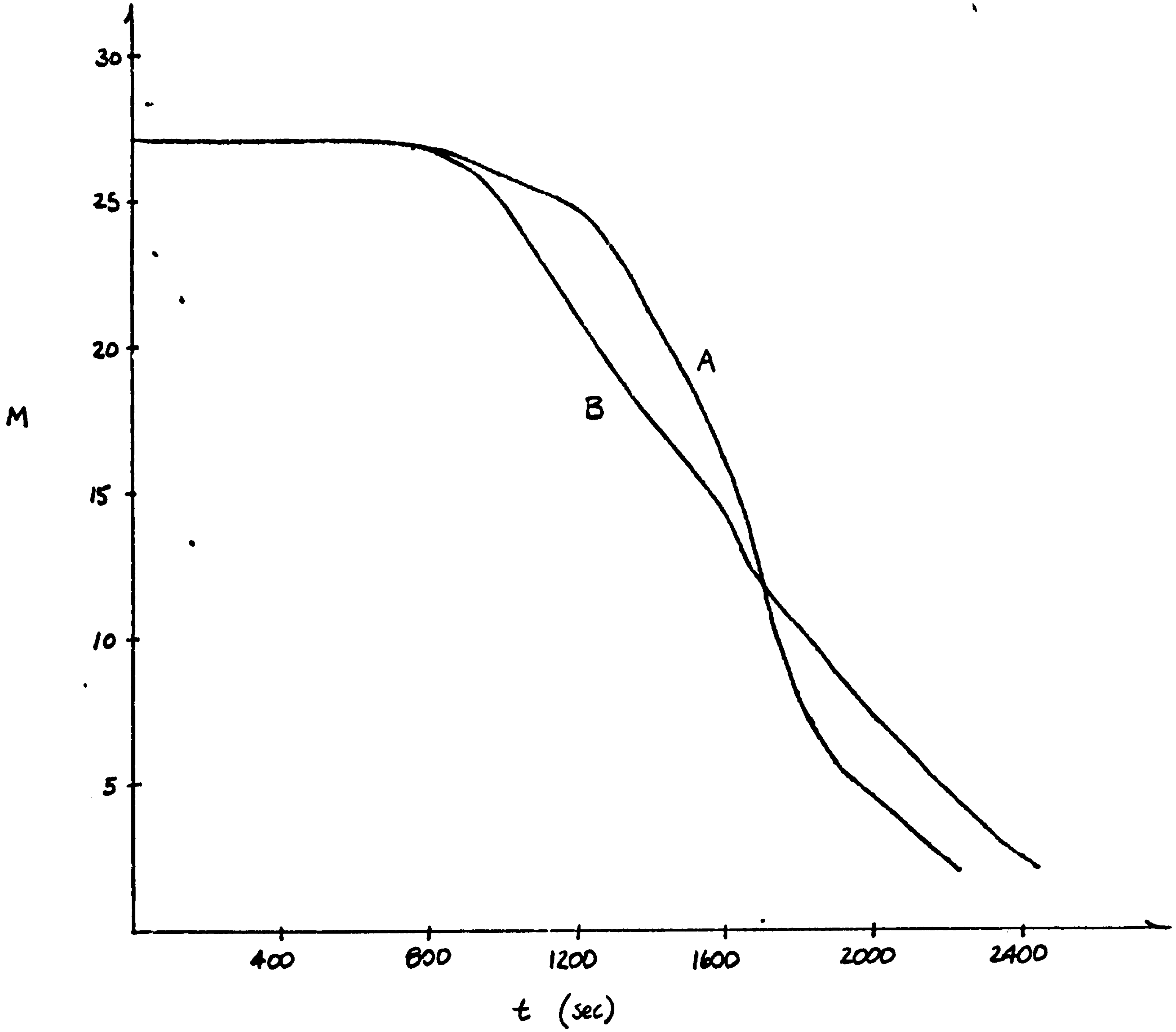


FIGURE 4.- MACH NUMBER

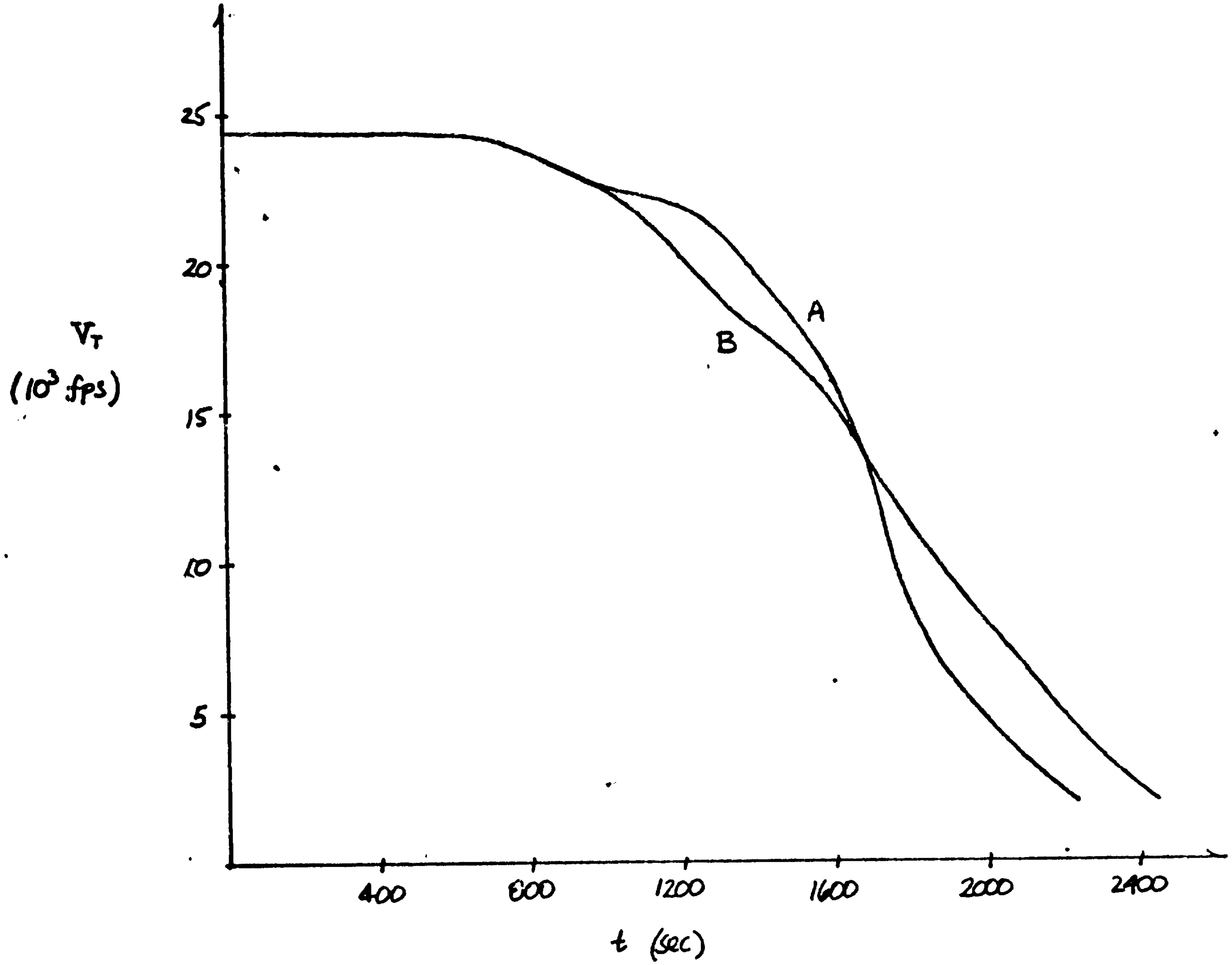


FIGURE 5.- RELATIVE VELOCITY

Section 2.1 and 2.2 present the block diagrams defining the aerodynamic surface (i.e., elevator) control logic and the ACPS (i.e., pitch jets) control logic, while Sections 2.3 and 2.4 present the fixed and scheduled gain parameters appropriate to these controllers. Finally Section 2.5 discuss the inputs, outputs and sample rates of the longitudinal control channel.

2.1 Elevator Control

Shown in figure 6 is the block diagram for elevator control, adaptedfrom reference 1. Basically, control consists of attack angle error and rate feedback to appropriately position the elevator actuator. Use is made of both fixed and trajectory dependent gains for transient pitch control, in parallel with a clamped integrator to assure a trim capability.

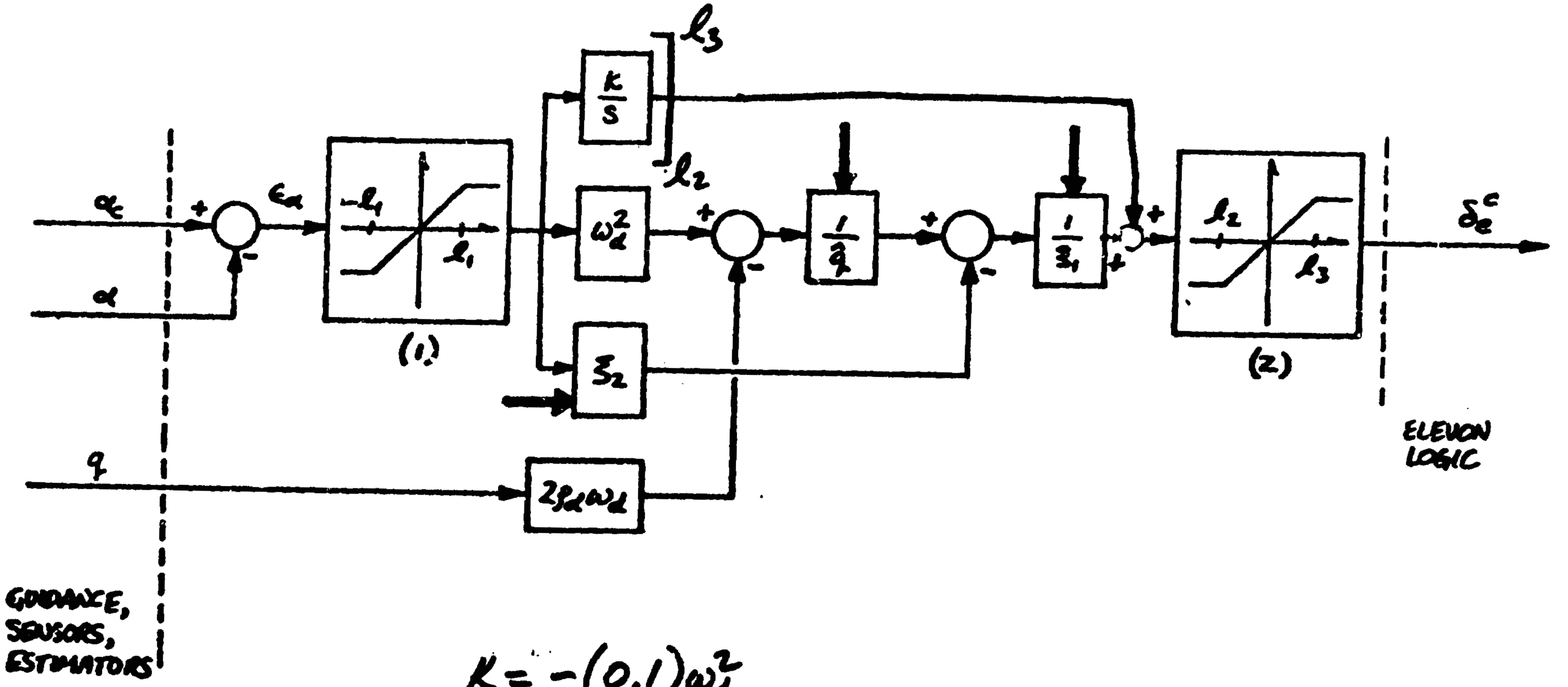
2.2 Pitch ACPS Control

Shown in figure 7 is the block diagram for pitch ACPS control, taken from reference 1. As with the previous logic, attitude control is achieved through the use of pitch rate damping, and the appropriate torque signal is sent to the jet select logic.

The interface logic between the ACS and ACPS subsystems is also shown in figure 7. The approach taken here is to inhibit firing of the pitch ACPS when there is "sufficient" pitch control acceleration available from the elevator. This determination of sufficiency is obtained from the commanded elevator deflection (δ_e^c) and hence, is a closed-loop index of elevator effectiveness.

2.3 Fixed Longitudinal Gains

Given below are values for the fixed longitudinal gain (and limit) parameters, which, from present simulation efforts, result in reasonable closed-loop vehicle response throughout the entry flight envelope.

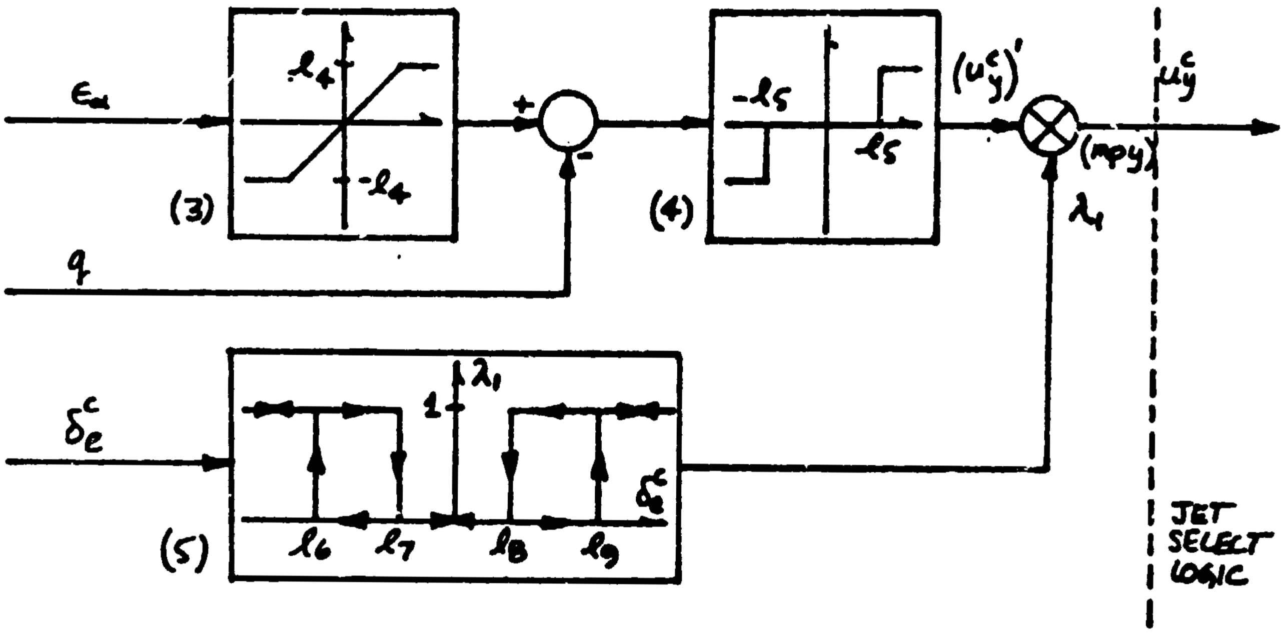


$$K = -(0.1)\omega_d^2$$

NON-LINEARITIES GIVEN BY : (1) UNITY SLOPE ; $\ell_1 = e_\alpha^{max}$

(2) UNITY SLOPE ; $\ell_2 = (\delta_e)_{min}$, $\ell_3 = (\delta_e)_{max}$

Figure 6 - Elevator Control Block Diagram



NON-LINEARITIES GIVEN BY : (3) SLOPE M_1 ; $\ell_4 = \dot{\alpha}_{max}$

(4) DEADBAND : $\ell_5 = M_1 \alpha_{db}$

OUTPUT : $0, \pm U_y$

(5) DEADBAND WITH HYSTERISIS :

$$M_1 = \frac{3U_y}{\dot{\alpha}_{MAX}}$$

$\ell_6 = f_1 \ell_2$
$\ell_7 = f_2 \ell_2$
$\ell_8 = f_2 \ell_3$
$\ell_9 = f_1 \ell_3$

$0 < f_2 < f_1 < 1$

OUTPUT : $\lambda_1 = 1$ or 0

Figure 7 Pitch ACPS Control Block Diagram

9.12.4 Entry DFCS (cont'd)

Table II: Fixed Longitudinal Gain

Paremeter	Values	Dimension
ω_d	0.7	sec^{-1}
ζ_d	0.6	—
ϵ_α^{max}	10	deg
l_2	-45	deg
l_3	+15	deg
$\dot{\alpha}_{max}$	5	deg/sec
α_{db}	0.25	deg
U_g	0.5	deg/sec^2
f_1	0.8	—
f_2	0.6	—

It should be noted that the above gains are applicable to the NR 161C vehicle.

2.4 Schedule Longitudinal Gains

As shown in figure 6, there are three variable gain parameters in the longitudinal control channel: ξ_1 , ξ_2, and $\hat{q}$. This section presents gain schedules for the first two parameters and a simple limiting logic for the third.

As discussed in reference 4, the gain parameters ξ_1 and ξ_2 may be plotted against Mach number and angle of attack, respectively. By comparing the values for two different entry trajectories (described in section 1), a piecewise linear approximation may be made for each parameter, so that trajectory independent gain scheduling is possible. The results are shown in figures 8 and 9, with ξ_1 a function of Mach number and ξ_2 a function of angle of attack.

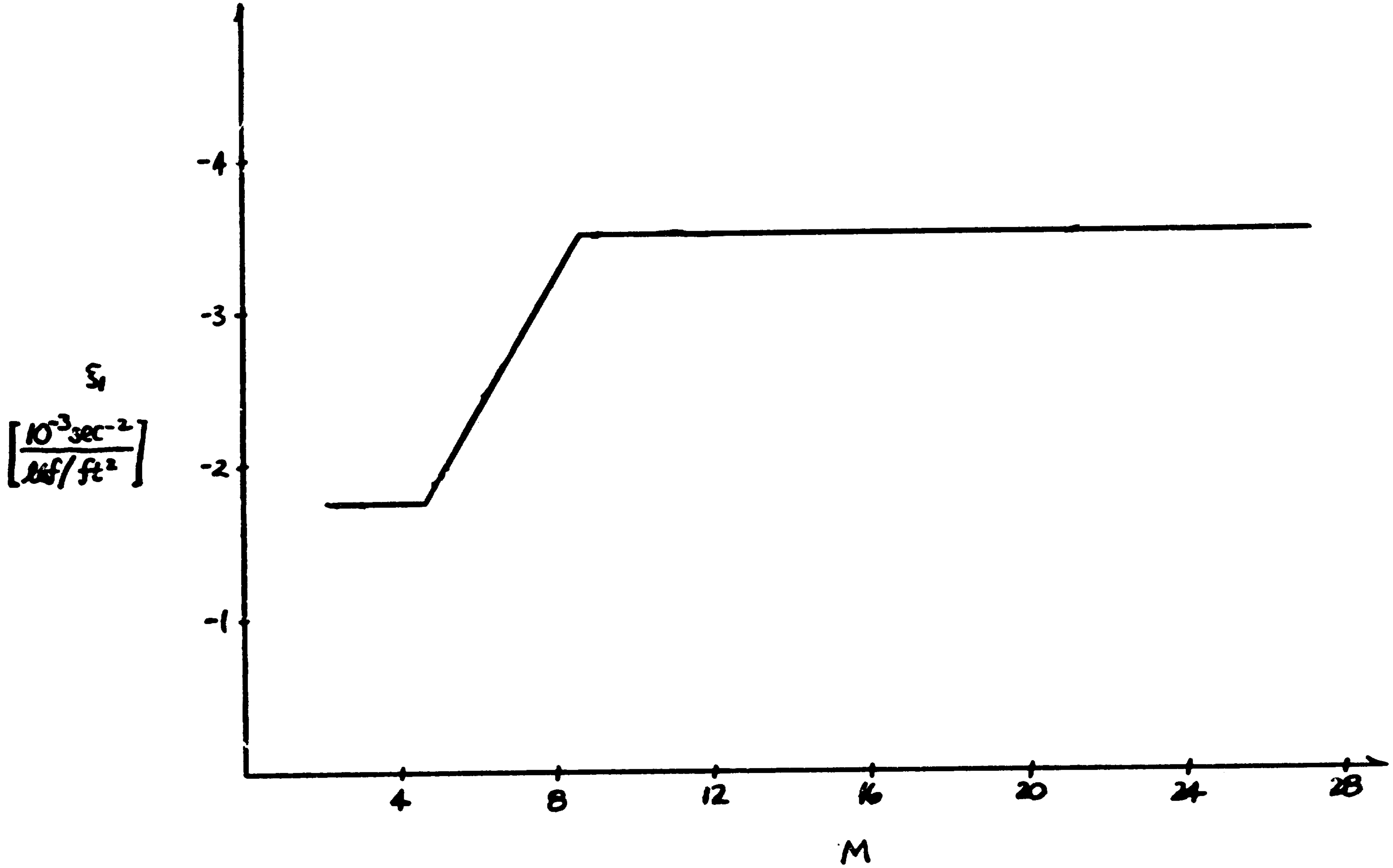


Figure 8 - Gain Schedule for ξ_1

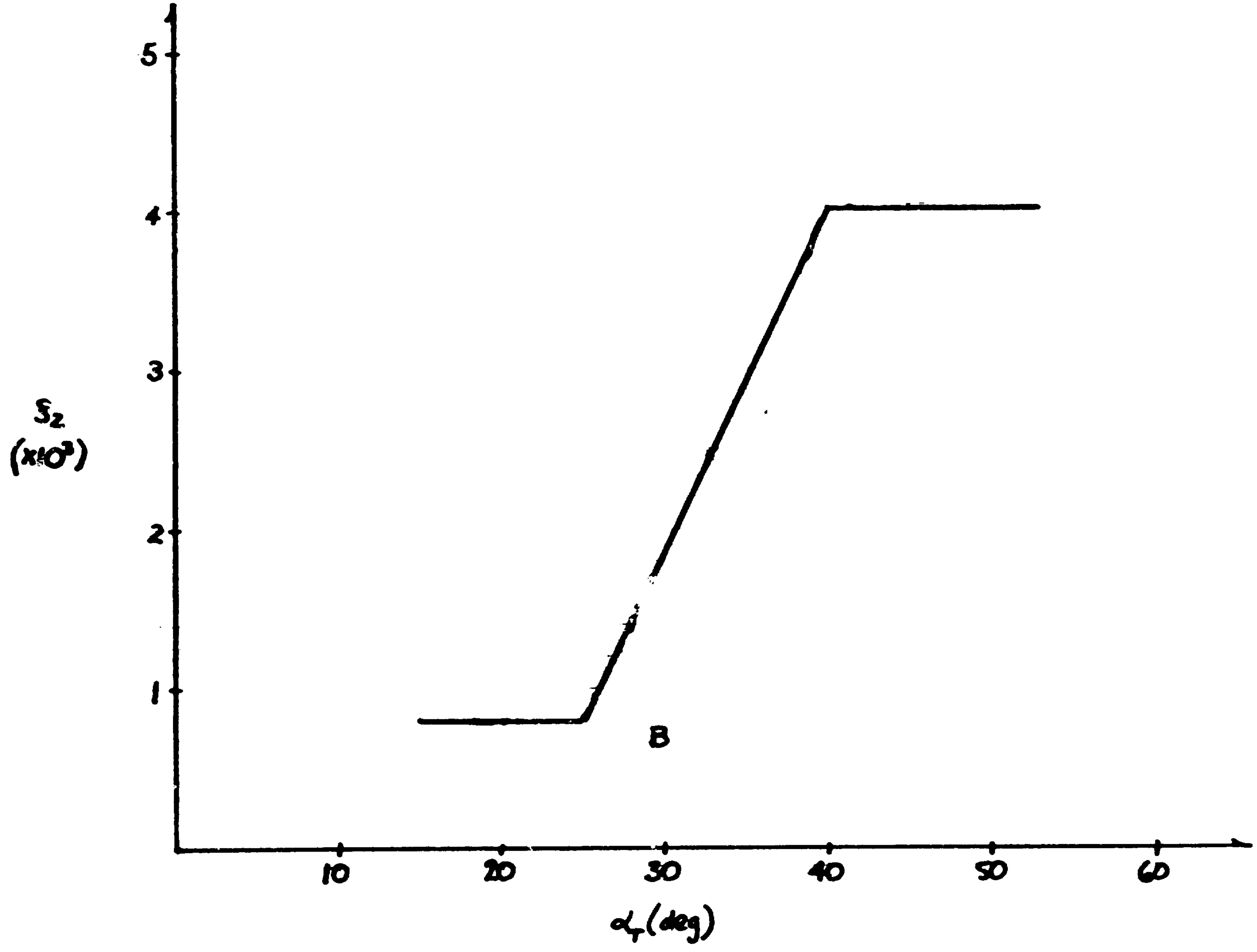
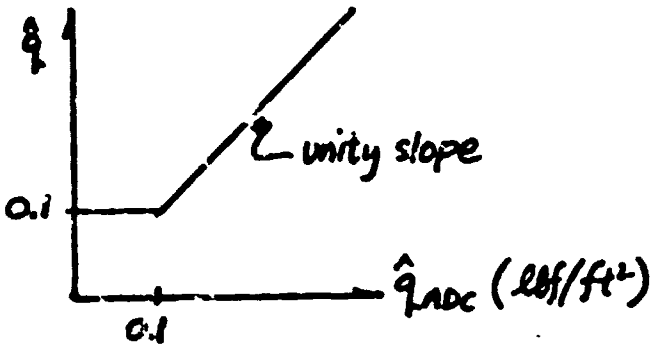


Figure 9.- Gain Schedule for ξ_2

In order to avoid computational overflow problems due to division by the dynamic pressure $\hat{q}$, a simple limiting logic is used to set a lower bound on this parameter, and is shown below:



Limiting Logic for Dynamic Pressure

In the above diagram $\hat{q}_{ADC}$ is the dynamic pressure obtained from the air data computer, which is lower limited to become the dynamic pressure $\hat{q}$ used in the controller logic of figure 6.

2.5 Longitudinal Channel, Inputs, Outputs, and Sample Rates

The inputs for the longitudinal controller fall into two categories: those pertaining to the inner "control" loop, and those pertaining to the outer "gain" loop. These two sets of inputs, and their corresponding origins, are tabulated below:

Table III: Longitudinal Inputs

	CONTROL LOOP			GAIN LOOP		
INPUT	α_c	α	q	M	$\hat{q}_{ADC}$	α_T
ORIGIN	guidance	IMU	rate gyro	ADC	ADC	ADC

The outputs for the longitudinal controller are commanded elevator deflection, δ_e^c, and commanded pitch ACPS acceleration, u_q^c. To obtain elevon commands, the elevator and aileron commands (the latter from the lateral logic) must be differenced in an elevon logic as shown in figure 10. In a similar manner, commanded ACPS accelerations must make use of an appropriate jet select logic, which will be the subject of a future memorandum.

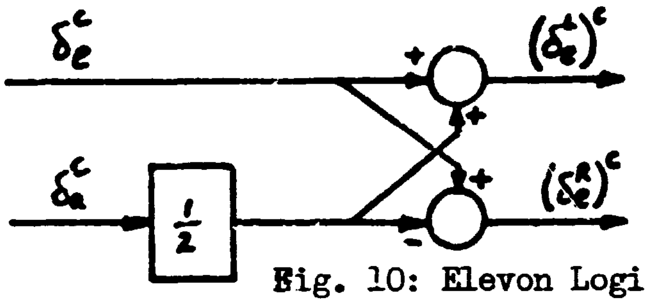


Fig. 10: Elevon Logic

There are two sample rates presently in use for DFCS simulation efforts: a high-frequency control loop rate of 8hz, and a low-frequency gain loop rate of 0.4hz. It should be noted that this implies gain updating every 20 control cycles, and may thus prove to be conservative.

3.0 Lateral Control

This section presents the control block diagrams and associated gains for lateral control, or, equivalently, control of the vehicles sideslip and bank angles. As with the longitudinal controller, the basic design procedure, described in reference 2, involves a separation of ACS and ACPS control, with the addition of an interface logic. Further, the ACS controller utilizes an analogous procedure of gain scheduling to minimize sensitivity to open-loop vehicle parameter changes.

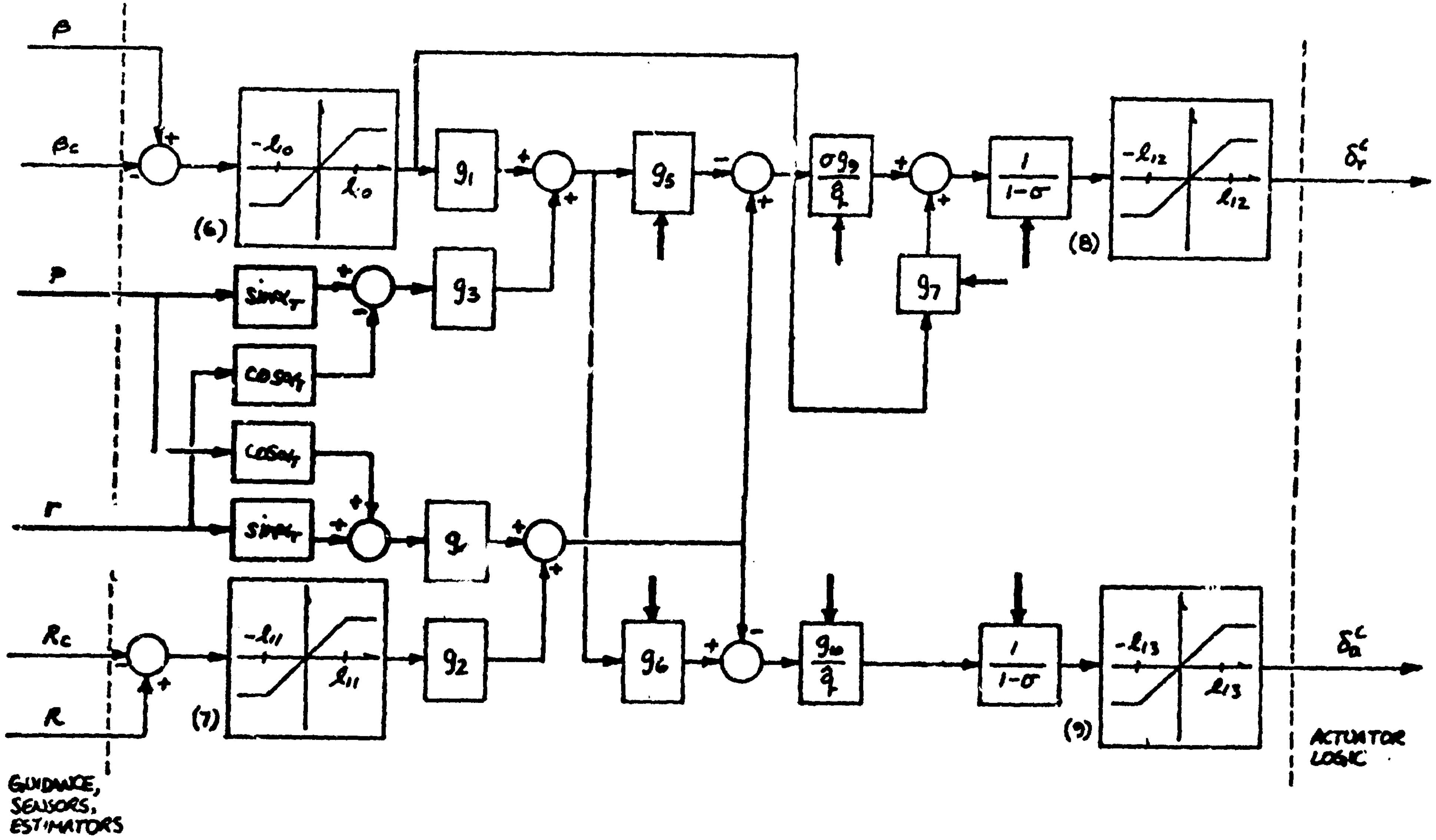
Sections 3.1 and 3.2 present the block diagrams defining the aerodynamic surface (i.e., rudder and aileron) control logic and the ACPS (i.e., yaw and roll jets.) control logic, while sections 3.3 and 3.4 present the fixed and scheduled gain parameters appropriate to these controllers. Finally section 3.5 discusses the inputs, outputs and sample rates of the lateral control channels.

3.1 Rudder and Aileron Control

Shown in figure 11 is the block diagram for rudder and aileron control, adapted from reference 1. Basically, control consists of sideslip and bank angle attitude and rate feedback to appropriately position the rudder and aileron actuators, with the gains chosen so as to minimize lateral coupling and simultaneously provide adequate dynamic response with sufficient damping. The synthesis procedure is described in reference 2; as can be seen, the design incorporates both fixed and scheduled gains for lateral control with the ACS.

3.2 Roll and Yaw ACPS Control

Shown in figure 12 is the block diagram for roll and yaw ACPS control, taken from reference 1. As with the previous logic, attitude control is achieved through the use of sideslip and bank angle



NON-LINEARITIES GIVEN BY : (6) UNITY SLOPE ; $\ell_{10} = \epsilon_\beta^{max}$

(7) UNITY SLOPE ; $\ell_{11} = \epsilon_R^{max}$

(8) UNITY SLOPE ; $\ell_{12} = (\delta_r)_{max}$

(9) UNITY SLOPE ; $\ell_{13} = (\delta_a)_{max}$

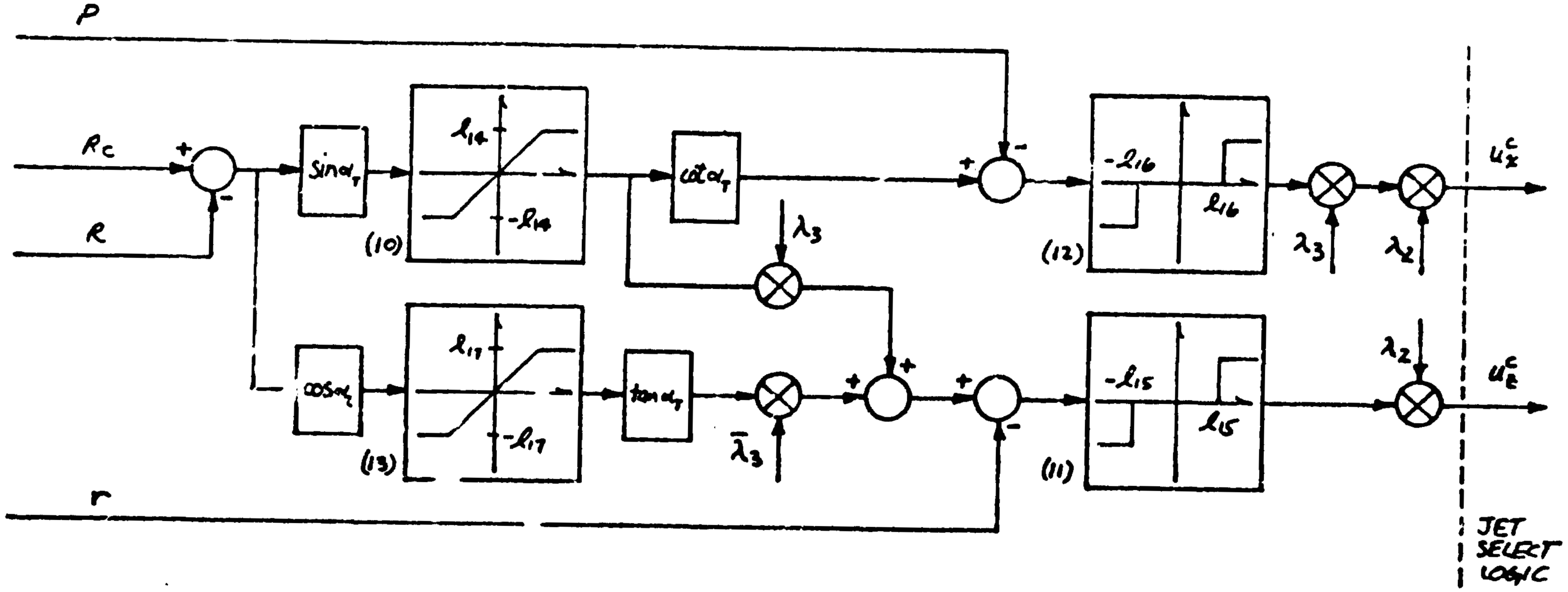
$g_1 = \omega_1^2$

$g_2 = \omega_2^2$

$g_3 = 2\zeta_1\omega_1$

$g_4 = 2\zeta_2\omega_2$

Figure 11 - Rudder and Aileron Control Block Diagram



NON-LINEARITIES GIVEN BY : (10) SLOPE μ_2 ; $l_{14} = \dot{R}^{max}\sin\alpha_T$

$$\mu_2 = \frac{3U_z}{\dot{R}^{max}\sin\alpha_T}$$

(11) DEADBAND : $l_{15} = \mu_2 R_{db}\sin\alpha_T$

OUTPUT : $0, \pm U_z$

$$\mu_3 = \mu_2 \tan\alpha_T$$

(12) DEADBAND : $l_{16} = l_{15}\cot\alpha_T$

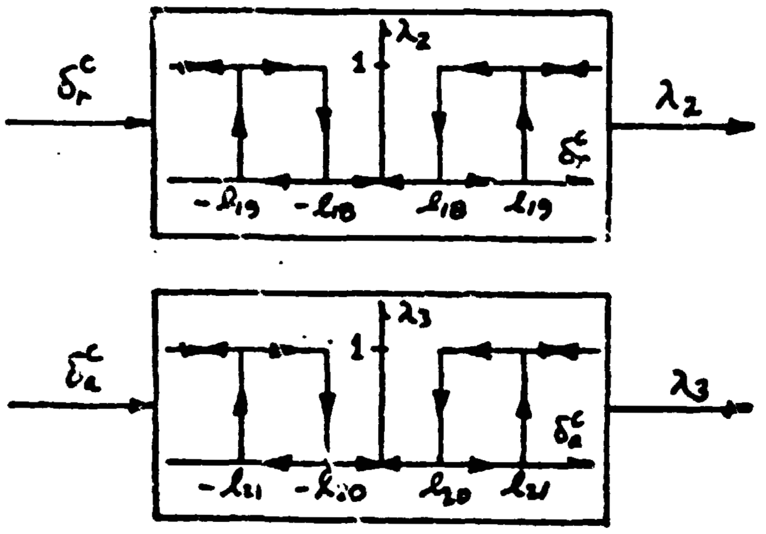
OUTFUT : $0, \pm U_x$

(13) SLOPE μ'_3 ; $l_{17} = \dot{R}^{max}\cos\alpha_T$

Figure 12 - Lateral ACPS Control Block Diagram

rate damping, and the appropriate torque signals are sent to the jet select logic.

As discussed in reference 2, three control modes are provided for in the lateral ACPS logic. Early in the entry, both aileron and rudder are ineffective, requiring that both the roll and yaw jets be used. Later in the trajectory, the ailerons become effective, requiring that only the yaw jets be used. Finally, when the rudder eventually becomes effective, no ACPS torques are necessary. This modal logic is summarized by the inhibit multiplications of figure 12 and the modal parameter definitions of figure 13. Note that, as with the longitudinal controller, surface effectiveness is measured by commanded deflection.



$$\ell_{18} = f_3 \ell_{12}$$
$$\ell_{19} = f_4 \ell_{12}$$
$$0 < f_3 < f_4 < 1$$

$$\ell_{20} = f_5 \ell_{13}$$
$$\ell_{21} = f_6 \ell_{13}$$
$$0 < f_5 < f_6 < 1$$

Figure 13 - Lateral Mode Parameters

3.3 Fixed Lateral Gains

Given in Table 4 are values for the fixed lateral gain (and limit) parameters, which, from present simulation efforts, result in reasonable vehicle response throughout the entry flight envelope.

Table 4: Fixed Lateral Gains

Parameter	Values	Dimension
ω_1	1	sec^{-1}
ω_2	1	sec^{-1}
ζ_1	0.7	—
ζ_2	0.7	—
ϵ_β^{max}	1	deg
ϵ_R^{max}	1	deg
ℓ_{12}	20	deg
ℓ_{13}	20	deg
$\dot{R}^{max}$	5	deg/sec
R_{db}	1	deg
U_x	1	deg/sec^2
U_z	1	deg/sec^2
f_3	0.6	—
f_4	0.8	—
f_5	0.6	—
f_6	0.[illegible]	—

As before, it should be noted that the above gains are applicable to the NR 161C vehicle.

3.4 Scheduled Lateral Gains

As shown in figure 11, there are sevon variable gain parameters in the lateral control channel: g_5 through g_{10} (excluding g_8), σ and $\hat{q}$. This section presents gain schedules for the first six; the dynamic pressure is limited in the same manner as dicussed in section 2.4.

As with the longitudinal gains, the lateral gains, obtained from reference 4, may be plotted against trim attack angle, and then fit in a piecewise linear sense. The trajectory independent gain schedules are shown in figures 14 through 19 with all gains functions of attack angle.

3.5 Lateral Channel Inputs, Outputs, and Sample Rates

As with the longitudinal controller, the inputs for the lateral controller fall into the two categories of control inputs and gain inputs. These inputs and their corresponding origins are tabulated below:

Table 5: Lateral Inputs

	CONTROL LOOP			GAIN LOOP	
INPUT	β_c, R_c	β, R	p, r	$\hat{q}$	α_T
ORIGIN	guidance	IMU	rate gyro	ADC	ADC

The outputs for the lateral controller are commanded rudder and aileron deflections, δ_r^c and δ_a^c , and commanded roll and yaw ACPS accelerations μ_x^c and μ_z^c . To obtain elevon commands the aileron command must be differenced with the elevator command as shown in figure 10. As with the longitudinal controller, the roll and yaw ACPS acceleration commands must make use of a jet select logic. The sample rates for the lateral controller are the same as for the longitudinal channel: a sampling frequency of 8 hz for the control loop and 0.4 hz for the gain loop.

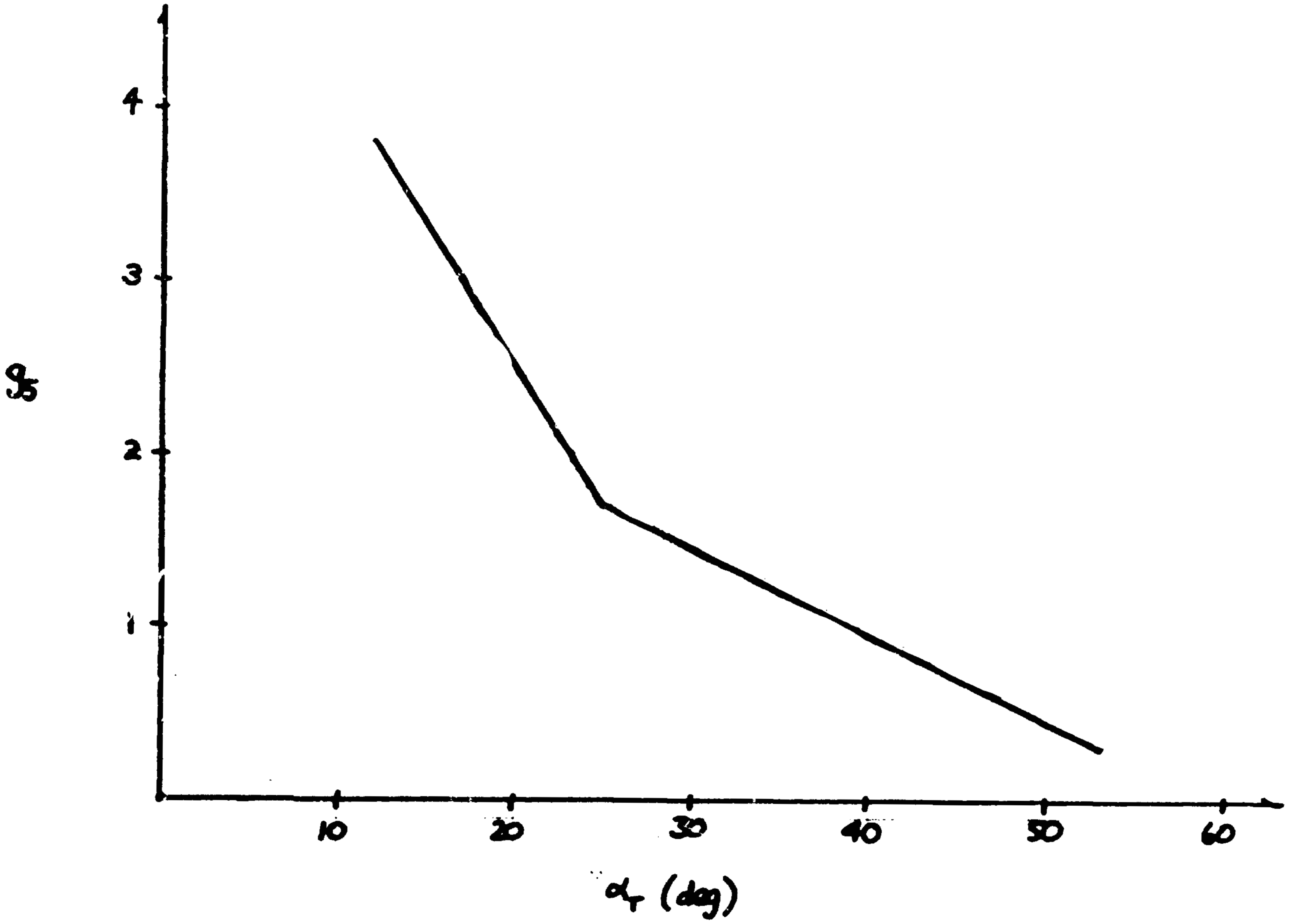


Figure 14 - Gain Schedule for g_5

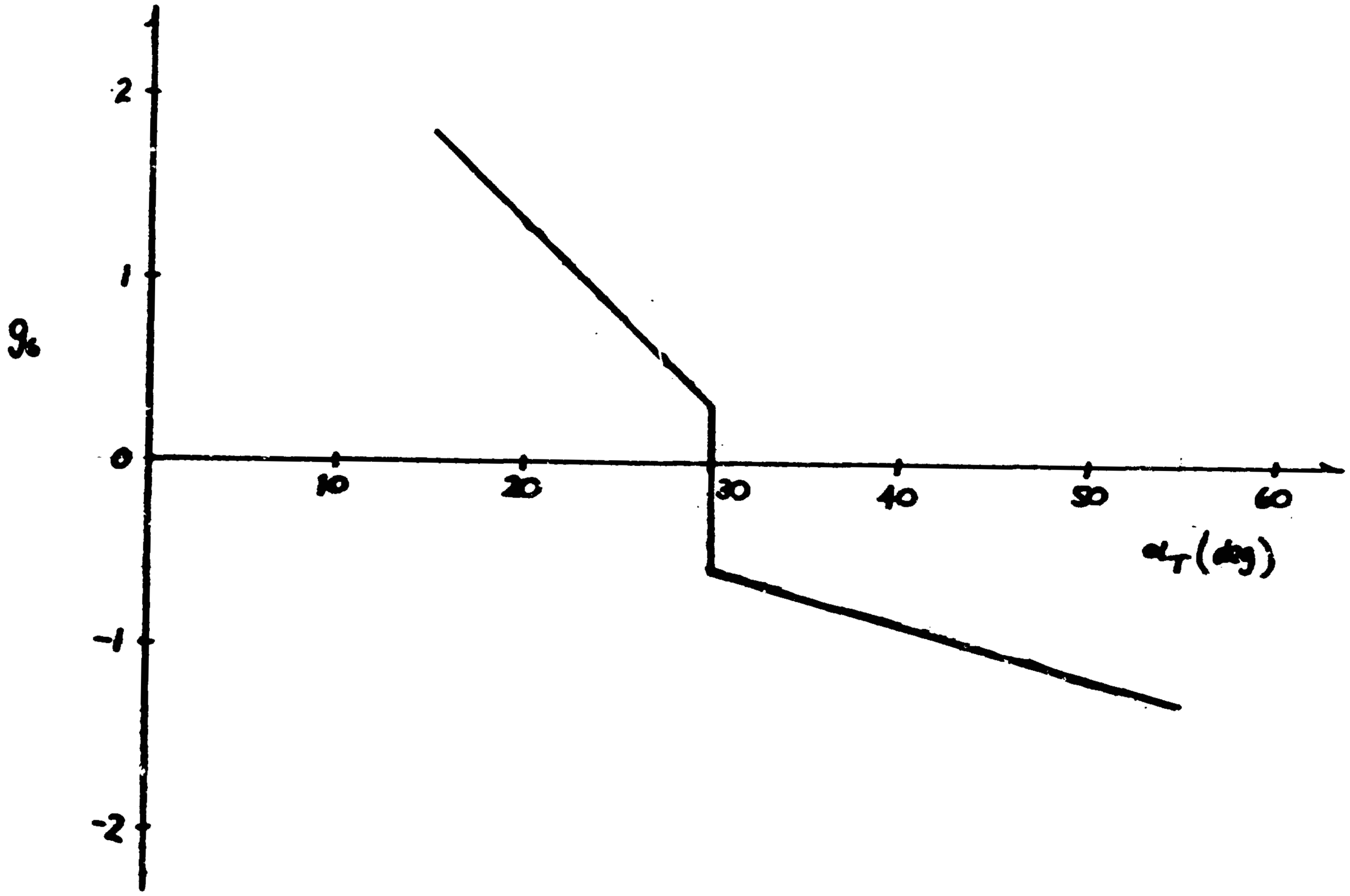


Figure 15 - Gain Schedule for g_6

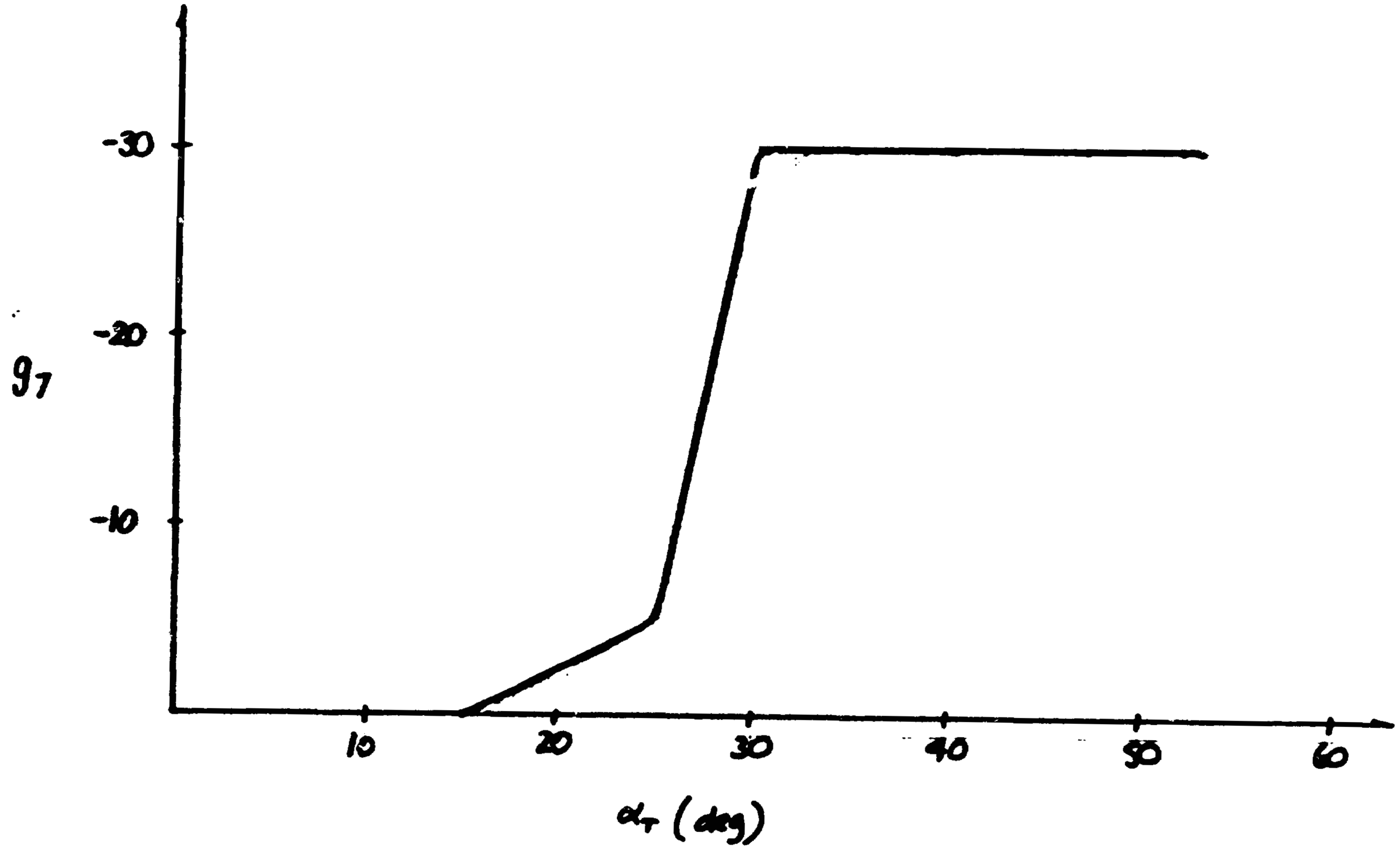


Figure 16 - Gain Schedule for g_7

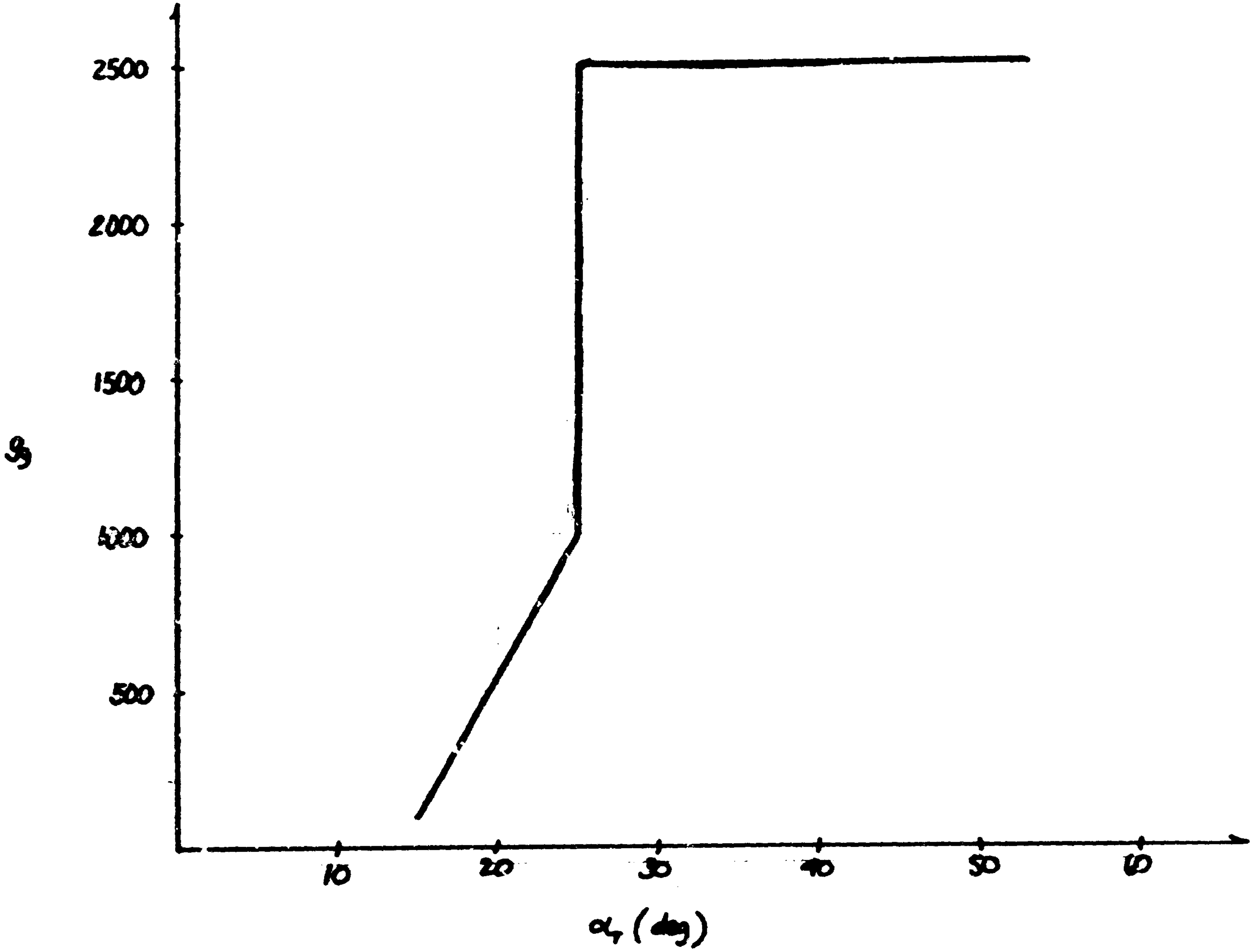


Figure 17 - Gain Schedule for g_9

C.3

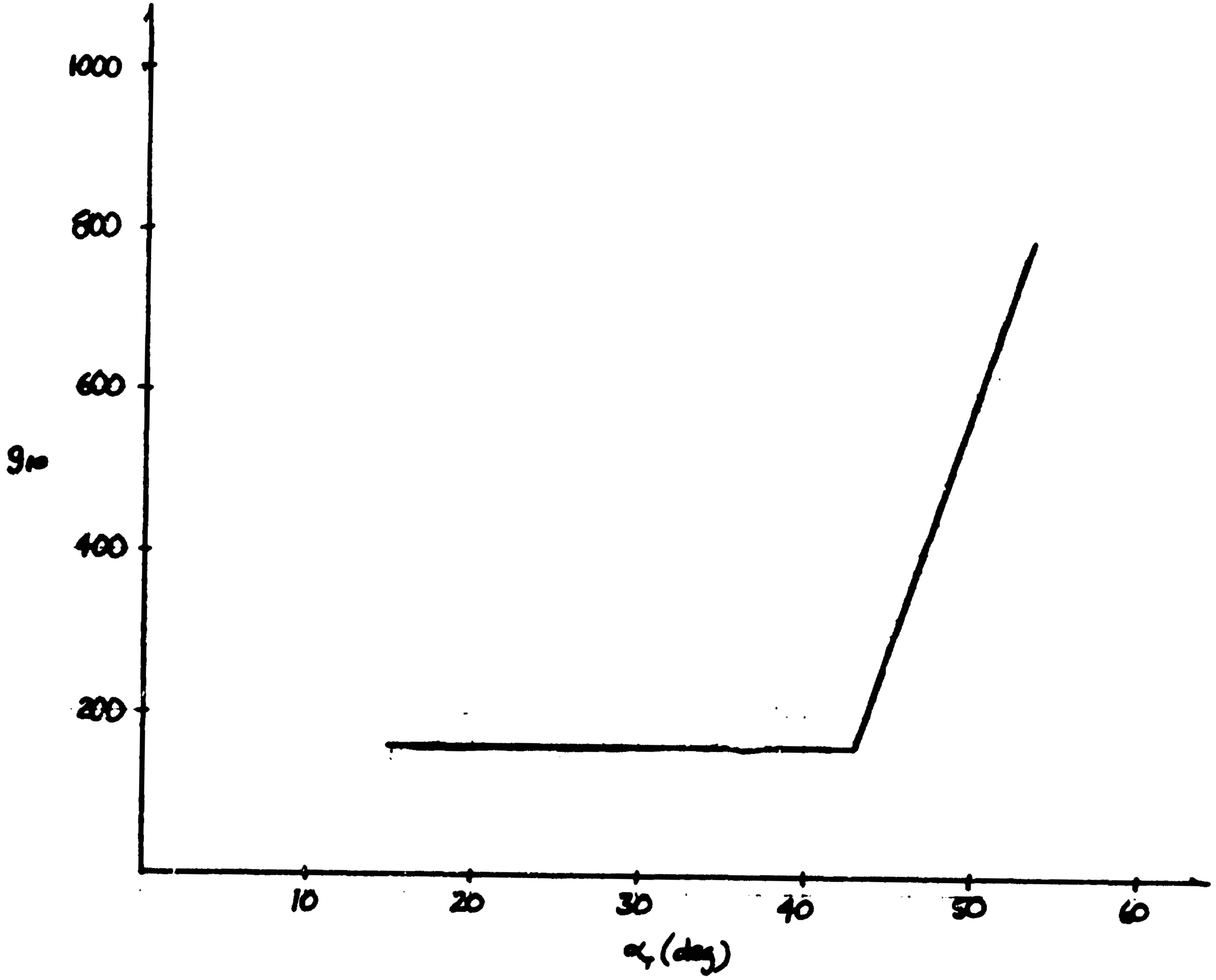


Figure 18.- Gein Schedule for g_{10}

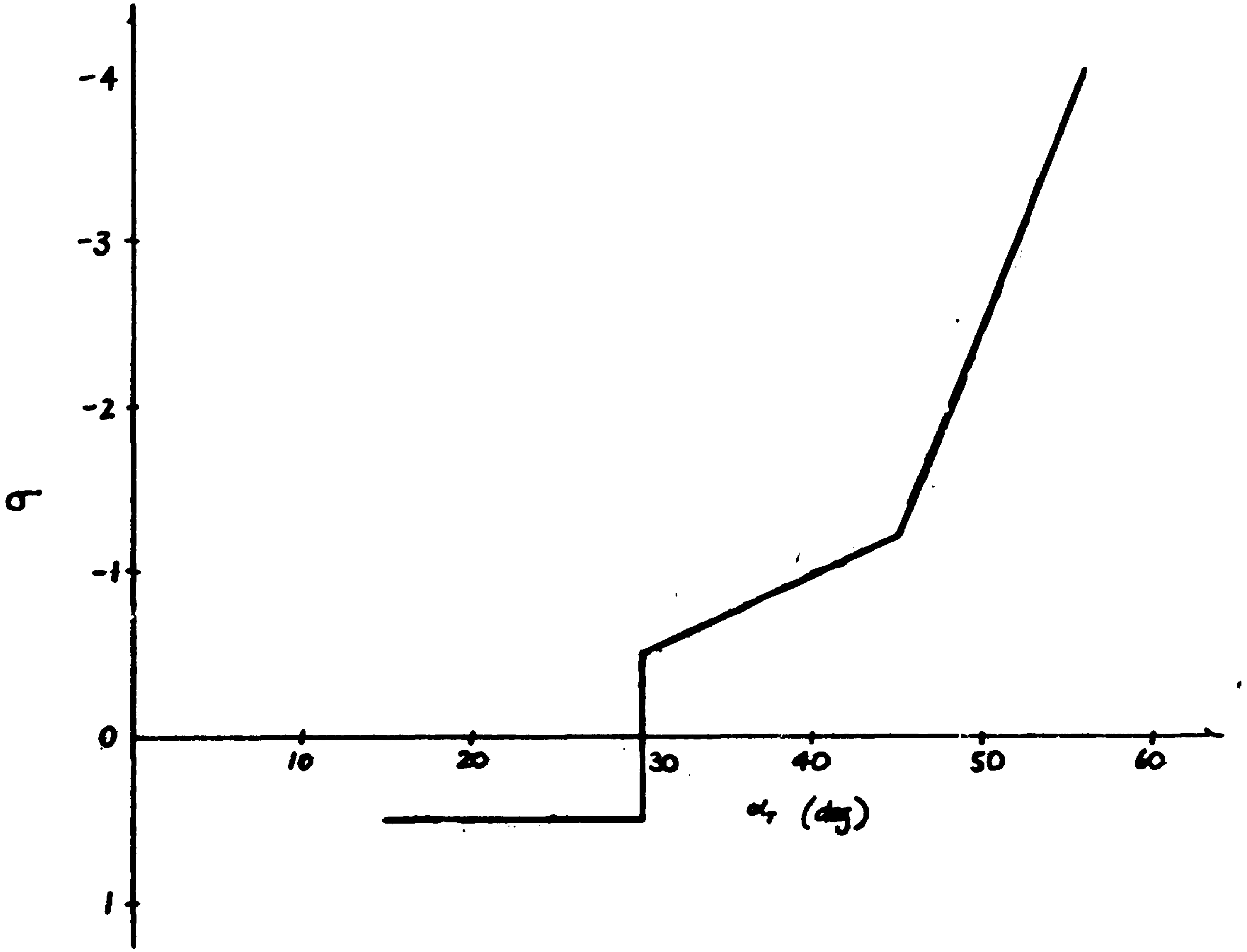


Figure 19 - Gain Schedule for σ

9.12.4 <u>Entry DFCS</u> (cont'd)

References:

1. MSC Memorandum EG2-71-218, "A Preliminary Design for a Blended Entry DFCS (Rev. 1)," November 10, 1971.
2. MSC Memorandum EG2-71-206, "A Preliminary Design for a Blended Entry DFCS," October 19, 1971.
3. MSC Memorandum EG2-71-207, "Vehicle Dynamic Parameters During Entry," October 19, 1971.
4. MSC Memorandum EG2-71-208, "Autopilot Gain Parameters for Entry," October 19, 1971.

9.13 TRANSITION

The transition phase starts at the end of the entry phase just prior to the pitchhover from a high angle of attack to the cruise angle of attack. This period of pitchover constitutes the transition phase. The transition phase ends with the attainment of the nominal cruise angle of attack. Two distinct functions are performed during transition:

1. Transition from entry control which is basically RCS attitude control to cruising flight control, which is basically full aerodynamic control.
2. Transition from entyy angle of attack (high a) to cruising angle of attack (low a).

The transition phase is characterized by a transition in control modes and some form of flight path angle control to effect a smooth transition to a new equilibrium altitide. This suggests that some changes to the guidance logic will also occurr during this period.

In this Section is presented the first cut at a Unified Digital Autopilot. A basic structure is provided for the purpose of encourging commonized coding for all control systems. A specific control law is included for only the Transition phase. The approach is currently being extended to other mission phases, and the Transition autopilot is being upgraded as additional information becomes available. The autopilot structure presented here was intentionally made general enough to compass all anticipated requirements. As specific autopilot designs mature for the various mission phases, the requirements for certain aspects of the coding presented here may not materialize. For example, the bending state estimation is predicated on a requirement for active control of several bending modes, a requirement that may not materialize.

SPACE SHUTTLE

GN&C SOFTWARE EQUATION SUBMITTAL

Software Equation Section Transition Attitude Control Submittal No. 22

Function: Unified Digital Autopilot with specific reference to the Transition phase.

Module No. OC-4 Function No. 2 (MSC 03690 Rev. A)

Submitted by: R. F. Stengel Co. MIT No. 8-71

Date: 1 July 1971

NASA Contact: W. H. Peters Organization EG2

Approved by Panel III KJ. Cox (Chairman) Date 6/1/71

Summary Description: Basic structure is presented for a Unified Digital Autopilot. A preliminary treatment is provided for attitude control during the transition phase.

Shuttle Configuration: (Vehicle, Aero Data, Sensor, Et Cetera)

With the current generalized treatment, the software is basically independent of configuration.

Comments:

(Design Status)

(Verification Status)

Panel Comments:

9.13.1 Autopilot

1. INTRODUCTION

The objective of the Unified Digital Autopilot Program is to provide rotational and translational control of the space shuttle orbiter in all phases of flight, from launch ascent through orbit to entry and touchdown. The program provides a versatile autopilot structure while maintaining simplified communications with other programs, with sensors, and with control effectors by the use of an executive routine /functional subroutine format. The program reads all external variables at a single point, copying them into its dedicated storage, and controls its major support subroutines to be synchronous with the autopilot cycle. As a result, the autopilot program is largely independent of other programs in the guidance computer and is equally insensitive to the characteristics of the processor configuration (dedicated guidance computer vs. shared multi-processor).

The unified autopilot program makes provision for sampling rates which are integer multiples of a basic sampling rate, using counters to establish the synchronous cycles. Extended computations carried out with a low repetition rate must be provided with pre-determined break points, in order that the low rate - and high rate - calculations can be interleaved. This requires programmer control, but it has the advantage of precluding skipped or lost computation cycles (unless interrupting external programs of higher priority monopolize the computer's time).

The sequence of autopilot subroutines is arranged to minimize transportation lag, the time interval between receiving a measurement and effecting a control force. While this lag may be largely due to equipment external to the guidance computer, the time required for control computation can be significant. As a consequence, the state estimation computations are separated into two subroutines. State measurements are incorporated in the estimator or filter in the early portion of the autopilot cycle, but the remaining state propagation or filter "push-down" does not occur until the autopilot commands have been written in the control effector output channels.

9.13.1 Autopilot (cont'd)

In order to discuss the Unified Digital Autopilot Program beyond the level of the functional flow diagram, it is necessary to make specific reference to the autopilot for a particular mission phase. The transition from entry to cruising flight has been chosen for this purpose, as this phase makes use of all of the types of autopilot subroutines.

The prototype transition autopilot described here can be considered an upper limit on the control system computational requirements, as it includes adaptive bending mode stabilization, high order state estimators and controllers, optimal state estimator gain computations, aerodynamic and reaction jet control, inertial and aerodynamic parameter estimation, and sensor /effector failure detection. It is almost certain that simplifications can be found; however, it is equally likely that unexpected complexities will occur. It should be emphasized that this is not a final transition autopilot design.

9.13.1 Autopilot (cont'd)

NOMENCLATURE

$\underline{\underline{A}}$	Mode stabilization estimation matrix (diagonal)
a_{II}	Element of $\underline{\underline{A}}$
$\underline{\underline{B}}$	Mode stabilization estimation matrix (diagonal only for uncoupled elastic modes)
b_{II}	Element of $\underline{\underline{B}}$
$\underline{\underline{C}}$	Mode stabilization estimation matrix (diagonal)
C_l	Roll moment effectiveness ratio
C_m	Pitch moment effectiveness ratio
C_n	Yaw moment effectiveness ratio
c_{II}	Element of $\underline{\underline{C}}$
$\underline{\underline{D}}$	Mode stabilization estimation matrix (diagonal)
d_{II}	Element of $\underline{\underline{D}}$
$\underline{\underline{E}}$	Mode stabilization control matrix (diagonal only for uncoupled elastic modes)
$\underline{f}_{1,\ 2,\ 3}$	Rigid-body control vectors
H	Altitude
$\underline{\underline{I}}$	Identity matrix
$\underline{\underline{K}}_{1,\ 2,\ 3,\ 4}$	Rigid-body estimator gain matrices
k	Ratio of specific heats
$k_{1,\ 2}$	Dynamic pressure estimator gains
$\underline{\underline{M}}$	Rigid-body measurement transformation matrix (longitudinal axis)
M	Mach number
$\underline{\underline{N}}$	Rigid-body measurement transformation matrix (lateral-directional axes)
$\underline{\underline{P}}$	Measurement noise covariance matrix (longitudinal axis)

9.13.1 Autpilot (cont'd)

ΔP	Stagnation pressure minus static pressure
p	Roll rate (body axis)
$\underline{\underline{Q}}$	Measurement noise covariance matrix (lateral-directional axes)
q	Dynamic pressure
$\underline{\underline{R}}$	Covariance matrix of $\underline{x}$
R	Universal gas constant
r	Yaw rate (body axis)
$\underline{\underline{S}}$	Covariance matrix of $\underline{v}$
T	Mode stabilization sampling interval
$T\ (H)$	Air Temperature
U	Velocity component along x-body axis (earth-relative, positive forward)
$\underline{u}$	Mode stabilization measurement vector
$\underline{\underline{V}}$	Rigid-body disturbance input covariance matrix (lateral-directional axes)
V	Velocity component along y-body axis (earth-relative, positive right)
$\mathcal{V}$	Magnitude of earth-relative velocity vector
$\underline{v}$	Perturbation state estimate vector (lateral-directional axes)
$\underline{\underline{W}}$	Rigid-body disturbance input covariance matrix (longitudinal axis)
W	Velocity component along z -body axis (earth-relative, positive down)
$\underline{w}$	Perturbation state measurement vector (lateral-directional axes)
$\underline{x}$	Perturbation state estimate vector
$\underline{y}$	Mode stabilization estimate vector
$\overline{y}_j^2$	Variance estimate for $j^{\underline{th}}$ component of y
$\underline{z}$	Perturbation state measurement vector (longitudinal axis)

9.13.1 Autopilot (cont'd)

α	Angle of attack
β	Sideslip angle
$\underline{\delta}$	Control deflection vector
ζ	Bending mode estimator damping ratio
θ	Pitch angle
$\rho(H)$	Air density
ϕ	Roll angle
ψ	Yaw angle
Ω	Transformed bending mode natural frequency
ω	Bending mode natural frequency

Subscripts

a	Aileron
Control	Autopilot output quantity
DB	Deadband
DLC	Direct lift control
e	Elevator
Estimate	Quantity estimated by autopilot
Guidance	Input quantity from guidance program
I	Bending mode index
i	Current sample of subscripted quantity
Measurement	Measurement input to autopilot
RCS	Reaction control system
r	Rudder
SB	Speed brake
δ	Derivative with respect to control deflection

9.13.1 Autopilot (cont'd)

Special Notation

$\dot{(\;)}$	Derivative with respect to time
$\tilde{(\;)}$	Mode stabilization control output
$\hat{(\;)}$	State estimate before measurement update
$(\;)'$	De-tuned bending mode estimation quantity
$(\;)^T$	Transpose of matrix or vector
$\underline{(\;)}$	Vector
$\underline{\underline{(\;)}}$	Matrix

9.13.1 Autpoilot (cont'd)

2. FUNCTIONAL FLOW DIAGRAM

The sequence for a single autopilot cycle is illustrated in the functional flow diagram of Figure 1. The autopilot program is called on a periodic basis with a sampling frequency determined by the highest bandwidth control mode, which, in turn, is a function of the flight phase. Active bending mode stabilization will require the highest sampling rate; where this is not necessary, the sampling rate will be determined by rigid body control requirements. Initialization branches are asynchronous, occurring only when the flight control mode changes or when there is a computer restart. All other branches are synchronous with the autopilot sampling rate, although their sampling intervals may be integer multiples of the basic sampling interval.

The basic subroutines of the unified digital autopilot are the following:

a) Sequence and Input / Output Initialization Subroutine
This subroutine establishes the address of a list of subroutine addresses according to the flight mode, determines the basic sampling interval and the integer multiples for medium and slow sampling rates, and initializes all indices.

b) Read Subroutine
This subroutine copies all inputs to the autopilot into dedicated temporary storage. The read list and sampling rate are functions of the flight control mode.

c) Filter and Parameter Initialization Subroutine
This subroutine initializes state and parameter estimates either at predetermined values or at the appropriate values read by (b) as required by the restart or control mode change.

d) Bending Stabilization Subroutine
This subroutine performs the minimum computations necessary to stabilize the bending mode(s). It incorporates new measurements in the bending coordinate estimates, computes control commands, and, if bending parameter adaptation is performed, computes the averages required by the parameter adjustment law. Parameter adjustment, which may

be carried out at a lower rate, is performed in (h). In the present scheme, this subroutine does no rigid-body control.

e) Perturbation State Estimation Subroutine - Part I
Measurements and computed control outputs are incorporated in the first part of the estimation subroutine. Perturbations from the flight profile generated by the guidance program are estimated here for use in the control law which follows. Only those perturbation states which are necessary for control or for display to the crew are estimated. This may include coordinates of bending or sloshing modes which are not sufficiently decoupled from the rigid-body modes for stabilization in (d).

f) Rigid-body Control Subroutine
This subroutine uses the outputs of (e) to derive control effector commands, which are intended to null the error between the actual and desired states. For proportional control effectors, e.g., engine gimbals or aerodynamic control surfaces, the control law may simply consist of scaling and coordinate transformation. For RCS thrusters, phase plane switching logic can be used.

g) Perturbation State Estimation Subroutine - Part II
The calculations performed here prepare the estimator for the incorporation of measurements and control outputs on the next appropriate autopilot cycle. In the case of an estimator expressed as a constant-coefficient digital filter, this consists of "pushing-down" the filter variables, i.e., storing the $i^{\underline{th}}$ value in the $(i - 1)^{\underline{st}}$ location, etc. For a time-varying filter expressed as a state-space estimator, the state must be propagated to the next sampling instant, and revised gains must be computed (it is possible that the latter be done at a rate which is slower that the propagation sampling rate).

h) Bending Parameter Estimation Subroutine
Should uncertainty or time variation in the bending parameters be excessive. this subroutine will revise the parameter estimates, either by parameter scheduling or parameter tracking, as required. This subroutine can be executed at a slow rate.

i) Inertial Parameter Update Subroutine
This subroutine will alter mass, moment-of-inertia, and RCS specific control moment estimates at a slow rate as required for revision of estimator and control law constants.

j) Aerodynamic Parameter Update Subroutine
This subroutine will alter dynamic pressure, Mach number, state transition matrices, and aerodynamic specific control moments at a slow rate as required for revision of estimator and control law constants. The calculations made in this subroutine are dependent on the flight control mode.

k) Failure Detection Subroutine
This subroutine examines the variances in the state and parameter estimates of the previous subroutines for "reasonability", detecting failures in sensors or control effectors by comparing the computed results to thresholds for normal operation. The subroutine is an adjunct to the failure discretes issued outside the autopilot program; since it depends upon an averaging process, it can be executed at a slow rate.

Although not explicitly shown in Figure 1, the option to branch to "Closeout" is available at each branch point. This precludes wasted testing when, for example, a bending stabilization computation is the only calculation required on a given autopilot cycle. It is also implicit that lengthy subroutines contain break points, allowing internal branching on intermediate autopilot cycles. For example, an aerodynamic parameter update subroutine with sampling interval 50 times longer than the bending stabilization sampling interval can be entered at the higher rate, with computations proceeding from breakpoint to breakpoint on sequential passes.

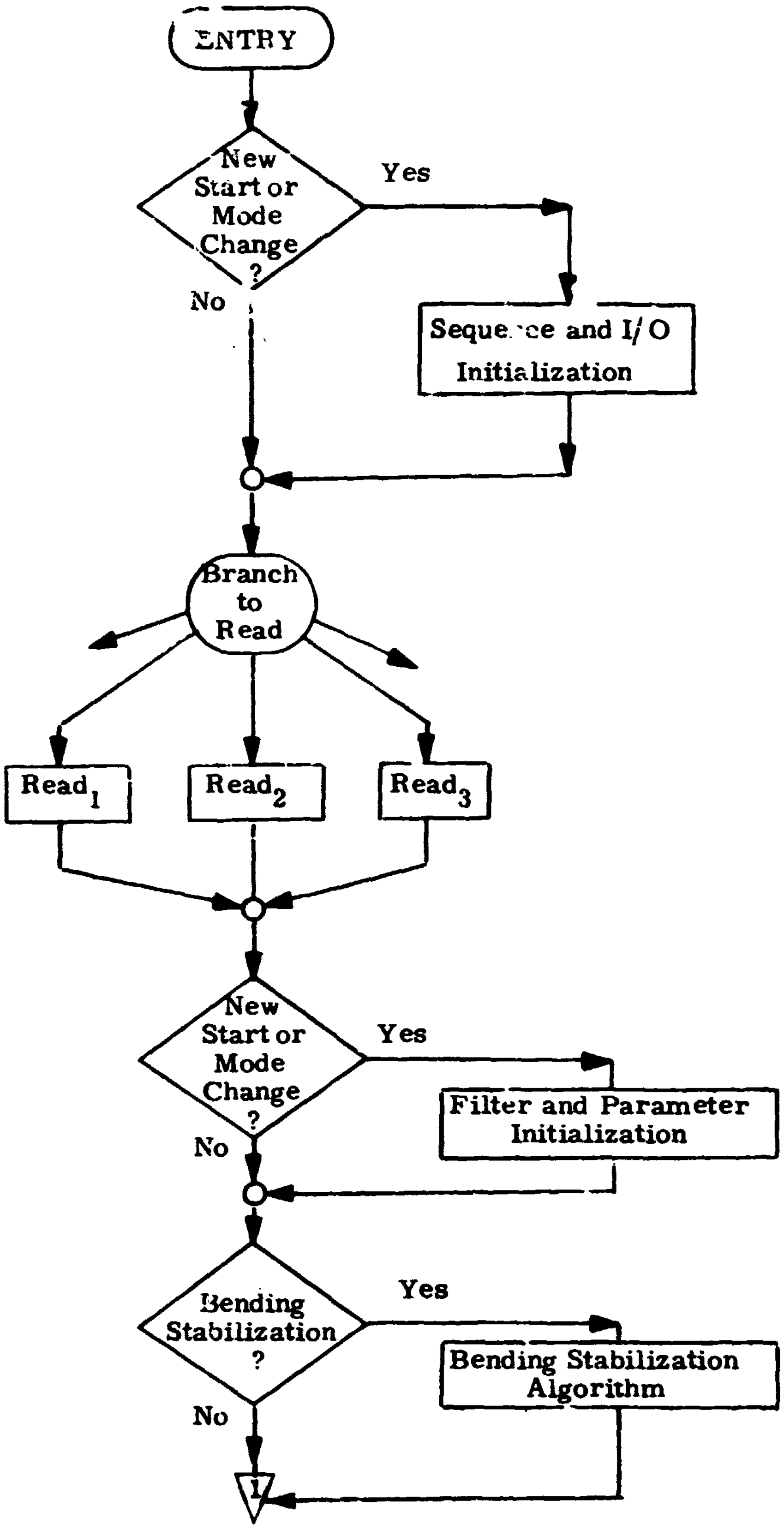

ENTRY
New Start or Mode Change ?
Yes
No
Sequence and I/O Initialization
Branch to Read
Read$_1$
Read$_2$
Read$_3$
New Start or Mode Change ?
Yes
No
Filter and Parameter Initialization
Bending Stabilization ?
Yes
No
Bending Stabilization Algorithm
1

Figure 1a FUNCTIONAL FLOW DIAGRAM

9.13.1 Autopilot (cont'd)

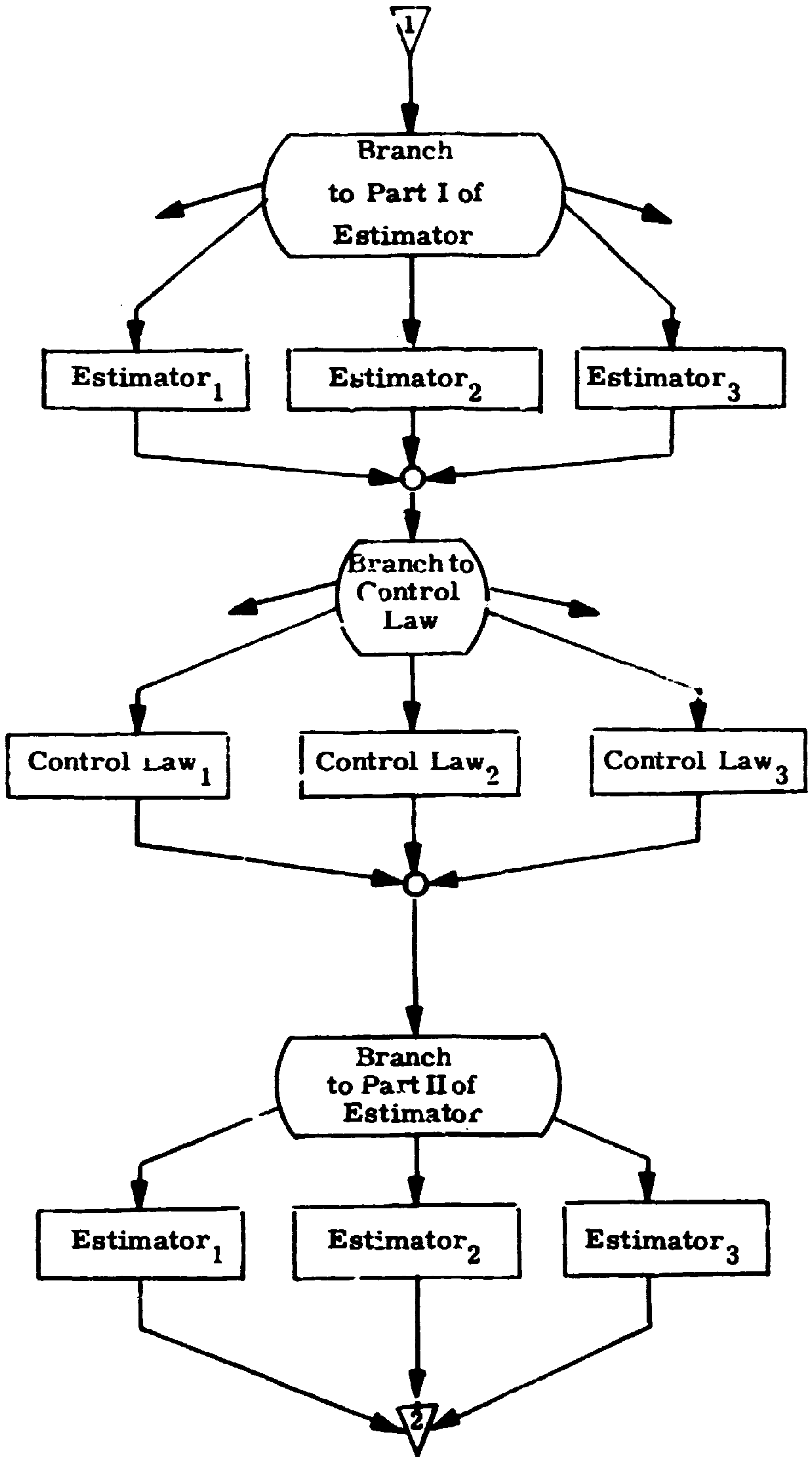


Figure 1b FUNCTIONAL FLOW DIAGRAM

9.13.1 Autopilot (cont'd)

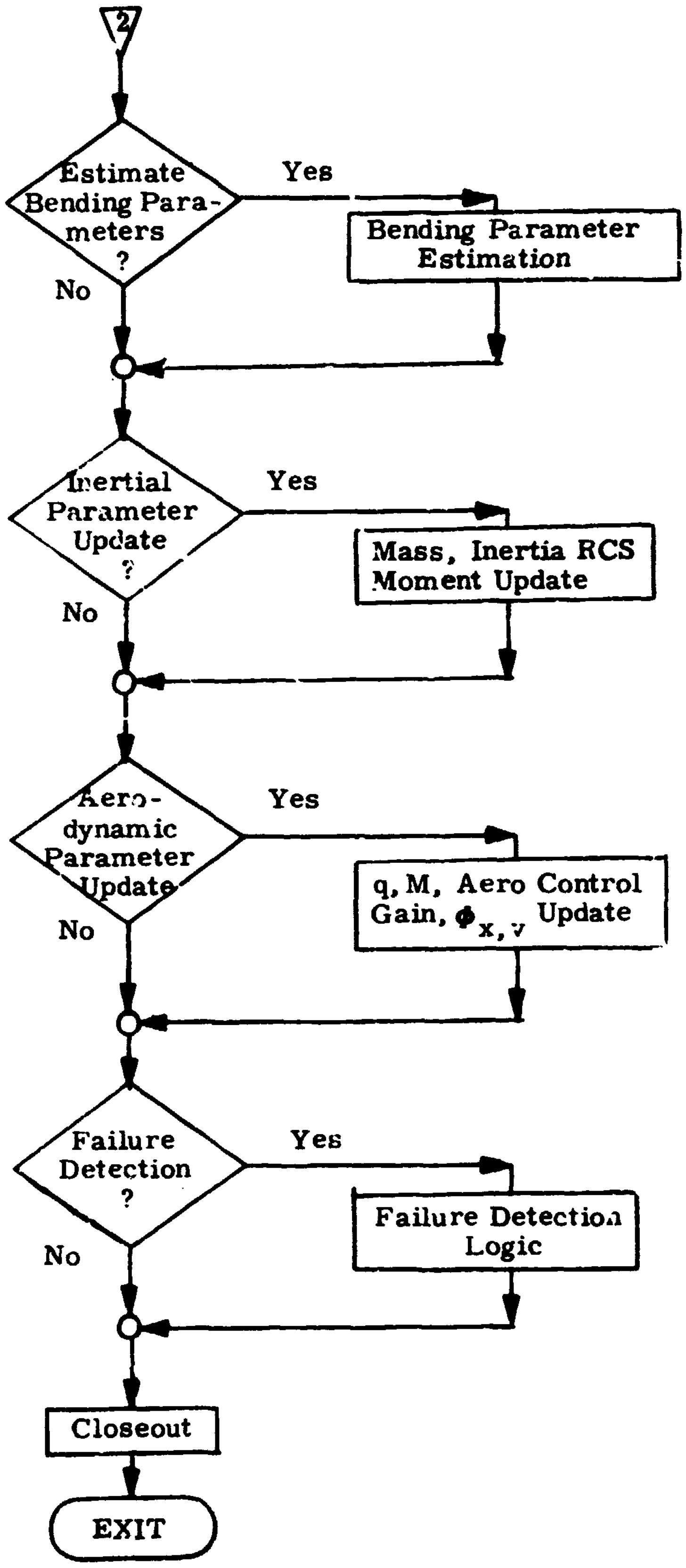

2
Estimate Bending Parameters ?
Yes
Bending Parameter Estimation
No
Inertial Parameter Update ?
Yes
Mass, Inertia RCS Moment Update
No
Aero-dynamic Parameter Update
Yes
q, M, Aero Control Gain, $\phi_{x,y}$ Update
No
Failure Detection ?
Yes
Failure Detection Logic
No
Closeout
EXIT

Figure 1c FUNCTIONAL FLOW DIAGRAM

9.13.1 Autopilot (cont'd)

3. PROGRAM INPUT-OUTPUT

The unified autopilot program will require inputs from inertial sensors, air data sensors, automatic guidance programs, manual controllers, failure monitor devices or programs, and the data management system's executive or control program. Control could be improved by inputs from control effectors, e.g., aerodynamic surface or engine gimbal deflection angles, and bending or slosh displacement sensors. Autopilot outputs will include control effector commands, crew display variables, and parameters of value to guidance and service programs.

The input-output lists are dependent on the flight control mode; the lists which follow are based upon the transition autopilot prototype. For this case, it is further assumed that angular and translational state vectors are transformed to earth-relative, body-axis components before being read by the autopilot, that control effector sensors and elastic displacement sensors are not used, and that the space shuttle vehicle is equipped with conventional aircraft control surfaces as well as a reaction control system. The subject of what types of sensors are used to measure the vehicles state is not addressed. It is assumed that the transition maneuver is unpowered (no thrust input) and automatic (no manual input). Direct lift and drag controls are accounted for in the estimator but are not specifically commanded by the autopilot prototype. Mode and failure discretes are not specifically identified in these lists, nor are crew display variables.

9.13.1 Autopilot (cont'd)

Input Variables

$H_{guidance}$	Desired altitude
$H_{measurement}$	Measured altitude
ΔP	Stagnation pressure minus static pressure
$p_{guidance}$	Desired roll rate
$p_{measurement}$	Measured roll rate
$r_{guidance}$	Desired yaw rate
$r_{measurement}$	Measured yaw rate
$T(H)$	Air Temperature
t	Time
t_{go}	Time-to-go
$U_{guidance}$	Desired velocity along x-axis
$U_{measurement}$	Measured velocity along x-axis
$V_{guidance}$	Desired velocity along y-axis
$V_{measurement}$	Measured velocity along y-axis
$W_{guidance}$	Desired velocity along z-axis
$W_{measurement}$	Measured velocity along z-axis
δ_a	Nominal aileron deflection
$\delta_{DLC\ guidance}$	Desired direct lift control setting
$\delta_{DLC\ Measurement}$	Measured direct lift control setting
$\delta_{e\ guidance}$	Nominal elevator deflection
$\delta_{r\ guidance}$	Nominal rudder deflection
$\delta_{SB\ guidance}$	Desired speed brake setting
$\delta_{SB\ measurement}$	Measured speed brake setting
$\theta_{guidance}$	Desired pitch attitude
$\theta_{measurement}$	Measured pitch attitude
$\dot{\theta}_{guidance}$	Desired pitch rate
$\dot{\theta}_{measurement}$	Measured pitch rate
$\rho(H)$	Air density

9.13.1 Autopilot (cont'd)

$\phi_{guidance}$	Desired roll attitude
$\phi_{measurement}$	Measured roll attitude
$\psi_{guidance}$	Desired yaw attitude
$\psi_{measurement}$	Measured yaw attitude

Output Variables

N_{jet}	RCS thruster identification (one per commanded thruster)
t_{on_x}	RCS thruster on-time - roll axis
t_{on_y}	RCS thruster on-time - pitch axis
t_{on_z}	RCS thruster on-time - yaw axis
δa	Aileron deflection
δe	Elevator deflection
δr	Rudder deflection

4. DESCRIPTION OF EQUATIONS

The equations of this section are applicable to the control of an elastic vehicle in atmospheric flight, with particular reference to the transition phase of the shuttle's return from orbit. The transition phase actually consists of 2 transitions: an angle-of attack transition and a control effector transition. The transition from high-to low-angle of attack is necessitated by the need to meet heating constraints early in the flight and the requirement for increased maneuverability toward the end of the flight. Similarly, reaction control thrusters provide attitude control during hypersonic flight, while aerodynamic control surfaces are used during the terminal portion of the flight. Each type of transition causes major changes in the dynamic characteristics of the system over a relatively short time interval.

The transition autopilot prototype includes the following elements:

a) Adaptive stabilization of 3 structural modes based upon a "classical" resonance filter,
b) Linear optimal estimation of the rigid-body perturbation state,
c) Gain-scheduled control commands to 3 aerodynamic surfaces,
d) Simplified phase-plane logic for backup control using the reaction control system,
e) Inertial and aerodynamic parameter update, and
f) Sensor and control effector failure detection.

4.1 Bending Mode Stabilization

It has not been determined that active bending stabilization will be necessary for the space shuttle; however, should this be the case, a high bandwidth mode stabilization control loop will be required. This infers not only that sensors and control effectors have high bandwidth capability, but that the digital compensation be executed at a high sampling rate. If rigid-body control and mode stabilization are combined in a single estimator-control law, the resulting number of computations may be excessive. As an alternative, it is suggested that the computations be partitioned, with the rigid and elastic control equations executed at different rates. A simple mode stabilization law is then executed at a high rate, while the more complex rigid-body control occurs at a slower rate which is consistent with the frequencies of rigid-body motion.

The mode stabilization technique is divided into 3 parts: identification of the bending frequencies, estimation of the bending mode components in the state measurement, and control to oppose the bending deflection. The present design neglects the transportation and dynamic lags of the control loop but could be modified as these quantities are identified. It also assumes that inertial angle measurements are the only inputs for bending mode control.

The continuous-time resonant filter,

$$H(s)=\frac{2\zeta\omega_f s}{s^2 + 2\,\zeta\,\omega_f\, s + \omega_f{}^2}\,, \tag{1}$$

where ζ is a damping ratio, and ω_f is a radian frequency, has a frequency response of 1 when $\omega = \omega_f$, while its frequency response at zero and infinite frequency is 0. The sharpness of the resonant peak is determined by ζ, and the filter's tuning is fixed by ω_f. A discrete-time realization of this filter is required for implementation in a digital system.

The details of digital resonant and notch filters can be found in Ref. 1, where it is shown that the bilinear transformation, $s = (z-1)/(z+1)$, can be used to find the discrete-time version of eq. 1. This transformation eliminates the "folding" problems of the z-transform by mapping the entire s-plane into a horizontal segment with width $\pm\,\pi/\,T$ rad/sec (where T is the sampling interval), i. e., it establishes the frequency relationship, $\Omega = \tan\,\omega T/2$, and provides the corresponding digital filter as well[2]. The recursive digital filter corresponding to eq. 1 is

$$y_i \;=\; a\,[b(u_i - u_{i-2}) + c\;y_{i-1} + d\;y_{i-2}]\;, \tag{2}$$

where

$$a \;=\; 1\,/\,(1 + \zeta\Omega_f + \Omega_f^2)$$

$$b \;=\; 2\zeta\Omega_f$$

$$c \;=\; 2\,(1 - \Omega_f^2)$$

$$d = -(1 - 2\zeta\Omega_f + \Omega_f^2)$$

Equation 2 has unity frequency response with zero phase lag when $\omega = \omega_f$ and has zero frequency response for $\omega = 0$ or π/T. Thus, equation 2 passes a sinusoid at frequency ω_f, without modification, while frequencies to either side are attenuated. If this resonant filter is tuned to the bending mode frequency, then y_i is a sampled estimate of the bending displacement, and a control command of the form,

$$\Delta\delta_i = e\,y_i, \tag{3}$$

is in-phase with the bending oscillation. By proper choice of the magnitude and sign of e, commanding the appropriate control effector with $\Delta\delta_i$ will oppose the oscillation. It will be recognized that this is a form of phase stabilization, and the lags mentioned earlier can not be neglected. This control law is proposed only for a case in which passive - or gain-stabilization is not sufficient. In the latter case, it would be sufficient to use equation 2 to form a subtractive notch filter[1]

$$u_i' = u_i - y_i, \tag{4}$$

to neglect explicit commands to the control effectors for bending stabilization, and to use u_i' in the rigid-body estimation and control.

The mechanism for the adaptation of the resonant filter is the tracking of a relative maximum in the power spectrum of the input signal. The variance of y_i forms an estimate of the power spectral density at $\omega = \omega_f$ (with a bandwidth proportional to $\zeta\omega_f$). A resonant filter tuned to $\omega_f' = \omega_f + \Delta\omega$ yields an estimate of the power spectral density at $\omega = \omega_f'$ in the form of the variance of y_i'. The greater variance is assumed to lie closer to the spectral peak, indicating the direction for adaptation. Upon reaching the relative maximum, the frequency estimate will limit cycle about the proper value, a minor deficiency which could be rectified at the expense of adding a third resonant filter. A recursive estimate for the variance of y_i can be found from

$$\overline{y_i^2} = \overline{y_{i-1}^2} + k\,(y_i^2 - \overline{y_{i-1}^2}) \tag{5}$$

where k is a constant between 0 and 1, and a similar equation can be applied to y_i'.

It may be necessary to stabilize more than one elastic mode, and each mode may be present in more than one axis. In the worst case, it will be necessary to transform the measurements into normal coordinates to perform the bending mode estimation on the transformed coordinates, and to control each mode with more than one control effector. Then vector analogs of eq. 2, 3, and 5 must be formed:

$$\underline{y}_i = \underline{A}[\underline{B}(\underline{u}_i - \underline{u}_{i-2}) + \underline{C}\,\underline{y}_{i-1} + \underline{D}\,\underline{y}_{i-2}] \tag{6}$$

$$\Delta\delta_i = \underline{E}\,\underline{y}_i \tag{7}$$

$$\overline{y_{i,j}^2} = \overline{y_{i-1,j}^2} + k\,(y_{i,j}^2 - \overline{y_{i-1,j}^2}),\quad j = 1,n \tag{8}$$

In all cases, $\underline{A}$, $\underline{C}$, and $\underline{D}$ are diagonal matrices with elements dependent on the damping and frequency of the normal modes. $\underline{B}$ and $\underline{E}$ are diagonal only if $\underline{y}$, $\underline{u}$, and $\Delta\underline{\delta}$ have identical dimensions and if the components of $\underline{u}$ are normal (in the bending modes). If this is not the case,

$$\underline{B} = \underline{B}_1\underline{B}_2 \tag{9}$$

$\underline{B}_1$ is diagonal and has elements $2\zeta_j\,\Omega_{f_j}$, and $\underline{B}_2$ transforms $\underline{u}$ to normal coordinates. $\underline{E}$ scales $\underline{y}$ for proper control of the normal modes and transforms the result to control effector coordinates. Equation 8 is actually n scalar equations, as the components of $\underline{y}$ are independent.

For the flow charts of Section 5, it is assumed that there are 3 elastic modes, 3 attitude measurements, and 3 control effectors.

4.2 Rigid-Body Perturbation State Estimation

For control during the transition, it will be necessary to have an estimate of the error between the desired state, as determined by the guidance program, and the actual state, as inferred by the state measurements. The error, or perturbation state, can be estimated using a linear recursive filter whose general form is

$$\hat{\underline{x}}_i = \underline{\underline{\Phi}}\,\underline{x}_{i-1} \tag{10}$$

$$\underline{x}_i = \hat{\underline{x}}_i + \underline{\underline{K}}_1 (\underline{z}_i - \underline{\underline{M}}\,\hat{\underline{x}}_i) + \underline{\underline{K}}_2 (\underline{\delta}_{i-1}) \tag{11}$$

Equation 10 propagates the perturbation state according to the homogeneous equations of a physical model using the state transition matrix, $\underline{\underline{\Phi}}$. The effects of model errors and forcing terms are added in eq. 11. Equations 10 and 11 provide a completely general formulation for linear estimation; it is the means of determining $\underline{\underline{\Phi}}$, $\underline{\underline{K}}_1$, and $\underline{\underline{K}}_2$ which defines the filter as optimal, near-optimal, or ad hoc. The Lunar Module rate estimator is an example of an ad hoc estimator which can be expressed as a special case of eq. 10 and 11[3].

For the transition phase an optimal estimator has been proposed, and the Section 5 flow charts make additional assumptions. The transition estimator is partitioned according to aircraft convention, i. e., into a longitudinal set and a lateral-directional set. The estimator for each set incorporates all the variables which are necessary to solve the linearized equations normally used for the study of stability and control. Keeping in mind that the autopilot is primarily responsible for attitude control, it may prove possible to eliminate from the autopilot estimator some or all of the variables normally associated with "guidance", thus reducing the dimensions of each of the two estimators; however, this should not be done without assurances that the integrity of guidance-control interaction is maintained.
It is assumed that the measurement variables have been transformed to the desired body-axis coordinates before being read by the

autopilot; hence, no additional transformations are necessary, and the $\underline{M}$ matrix of eq. 11 is an identity matrix with the dimension of the measurement, $\underline{z}$ (which need not be of the same dimension as $\underline{x}$).

The measurement vector of the longitudinal estimator is defined as

$$\underline{z} = \begin{bmatrix} U \\ W \\ H \\ \theta \\ \dot{\theta} \end{bmatrix}_{\text{Measurement}} - \begin{bmatrix} U \\ W \\ H \\ \theta \\ \dot{\theta} \end{bmatrix}_{\text{Guidance}} ; \qquad (12)$$

thus it includes axial velocity, normal velocity, and height errors as well as the conventional pitch and pitch rate errors. The control vector,

$$\underline{\delta}_1 = \begin{bmatrix} \delta_e \\ \delta_{DLC} \\ \delta_{SB} \\ \Delta \dot{\theta}_{RCS} \end{bmatrix}_{\text{Control}} - \begin{bmatrix} \delta_e \\ \delta_{DLC} \\ \delta_{SB} \\ 0 \end{bmatrix}_{\text{Guidance}} , \qquad (13)$$

includes direct lift control and speed brake setting as well as elevator setting and pitch thruster rate, with the assumption that the guidance provides estimates of what these quantities should be (with the exception of reaction control rates). No assumption is made with regard to the source of the "control" quantities; they may be derived from autopilot and guidance commands or from control effector measurements. The latter is clearly desirable if accurate measurements are available; however, there is precedent for using the former in the Apollo digital autopilots.

The estimated perturbation state, $\underline{x}$, is

$$\underline{x} = \begin{bmatrix} \Delta U \\ \Delta W \\ \Delta H \\ \Delta \theta \\ \Delta \dot{\theta} \end{bmatrix}_{\text{Estimate}} \tag{14}$$

The gains $\underline{\underline{K}}_1$ and $\underline{\underline{K}}_2$ will be described in a later paragraph. The angle of attack estimate is derived from the total linear velocity estimates (summing perturbation estimates with nominal values) using the conventional formula

$$\begin{aligned} \alpha &= \tan^{-1}\left(\frac{-W}{U}\right)_{\text{Estimate}} \\ &= \tan^{-1}\left(\frac{-W_{\text{Guidance}} - \Delta W_{\text{Estimate}}}{U_{\text{Guidance}} + \Delta U_{\text{Estimate}}}\right) \end{aligned} \tag{15}$$

Similarly, the lateral-directional estimator includes the measurement vector,

$$\underline{w} = \begin{bmatrix} v \\ \phi \\ r \\ \varphi \\ p \end{bmatrix}_{\text{Measurement}} - \begin{bmatrix} v \\ \phi \\ r \\ \varphi \\ p \end{bmatrix}_{\text{Guidance}} \tag{16}$$

the control vector,

$$\underline{\delta}_2 = \begin{bmatrix} \delta_a \\ \delta_r \\ \Delta p_{RCS} \\ \Delta r_{RCS} \end{bmatrix}_{\text{Control}} - \begin{bmatrix} \delta_a \\ \delta_r \\ \Delta p_{RCS} \\ \Delta r_{RCS} \end{bmatrix} \tag{17}$$

and the estimate vector

$$\underline{v} = \begin{bmatrix} \Delta v \\ \Delta \psi \\ \Delta r \\ \Delta \phi \\ \Delta p \end{bmatrix}_{\text{Estimate}} \tag{18}$$

The sideslip angle estimate again uses a standard formula:

$$\begin{aligned} \beta &= \sin^{-1}\left(\frac{v}{V_{\text{Total}}}\right)_{\text{Estimate}} \\ &= \sin^{-1}\left[\frac{(V_{\text{Guid}}+\Delta V_{\text{Est}})}{[(U_{\text{Guid}}+\Delta U_{\text{Est}})^2+(V_{\text{Guid}}+\Delta V_{\text{Est}})^2+(W_{\text{Guid}}+\Delta W_{\text{Est}})^2]^{1/2}}\right] \end{aligned} \tag{19}$$

The flow charts of Section 5 indicate that calculations of eq. 10 and 11 occur at different points in the autopilot sequence. In particular, eg. 11 occurs soon after the input is read, while propagation of the state (eq. 10) occurs after the control command has been issued. This is done to minimize transportation lag between the computer input and output. Since eq. 10 is solved for the <u>next</u> autopilot cycle, the indices are incremented by one, and the equation appears in the flow chart as

$$\hat{\underline{x}}_{i+1} = \underline{\underline{\Phi}}\,\underline{x}_i \tag{10a}$$

Although the longitudinal and lateral-directional estimators are each 5-dimensional, it is likely that many of the off-diagonal elements of $\underline{\underline{\Phi}}$ will be negligible, allowing a reduction in the number of additions and multiplications necessary to solve eq. 10 and 11.

Estimator gains for the transition autopilot prototype are computed in real-time and are the linear optimal gains associated with Kalman[4] and presented by Liebelt[5]. The gain calculation is based upon the estimation of state covariances, $\underline{\underline{R}}$, which in this case also depends upon *à priori* knowledge of the disturbance input and measurement noise covariances. The covariance propagation, covariance estimate, and gain computation equations are

$$\underline{\underline{R}}_i = (\underline{\underline{I}} - [\underline{\underline{K}}_1 \vdots \underline{\underline{K}}_2]_i \, \underline{\underline{M}}_A) \, \hat{\underline{\underline{R}}}_i, \tag{20}$$

$$\hat{\underline{\underline{R}}}_{i+1} = \underline{\underline{\Phi}} \, \underline{\underline{R}}_i \, \underline{\underline{\Phi}}^T + \underline{\underline{W}}_{i+1}, \tag{21}$$

$$[\underline{\underline{K}}_1 \vdots \underline{\underline{K}}_2]_{i+1} = \hat{\underline{\underline{R}}}_{i+1} \, \underline{\underline{M}}_A^T \, (\underline{\underline{P}}_i + \underline{\underline{M}}_A \, \hat{\underline{\underline{R}}}_{i+1} \, \underline{\underline{M}}_A^T 0^{-1} \tag{22}$$

Here the indices have been incremented by one, as the solutions apply to the next autopilot cycle.

It will be noted that the gain matrix has been partitioned. $\underline{\underline{K}}_1$ is the gain matrix which operates on the measurement residuals; $\underline{\underline{K}}_2$ operates on the control inputs, which are represented as estimated residuals in the gain computations[6]. The measurement tranformation matrix, $\underline{\underline{M}}$, is augmented as necessary to incorporate the control inputs (hence the subscript, A). The disturbance input covariance matrix, $\underline{\underline{W}}$, appears in eq. 21; in general, this term prevents the gain matrices from vanishing under the influence of the inverse of the measurement noise covariance, $\underline{\underline{P}}$, in eq. 22.

The above notation is used for the longitudinal estimator; in the lateral-directional estimator, $\underline{\underline{N}}$, $\underline{\underline{Q}}$, $\underline{\underline{S}}$, and $\underline{\underline{V}}$ are substituted for $\underline{\underline{M}}$, $\underline{\underline{P}}$, $\underline{\underline{R}}$, and $\underline{\underline{W}}$. The state transition matrices are different in each case, being subscripted by x and v accordingly.

Although it may not be apparent from the equations, this formulation of the optimal gain is well-suited to incorporating data at differing rates. As an example, let us examine the longitudinal estimator, assuming that $\dot{\theta}$ is not measured and that only reaction control torques are present. Further assume that θ is measured on each autopilot cycle, that U and W are measured every tenth cycle, and that H is measured every fiftieth cycle. The diagonal matrix is potentially fifth order; however, it is only fifth order when all measurements are taken simultaneously, i. e., on an autopilot cycle when U, W, H, and θ are measured simultaneously and when $\Delta\dot{\theta}_{RCS}$ in non-zero. When there is no control torque, $\underline{\underline{M}}$ has a maximum dimension of four, but this occurs, at most, every fiftieth cycle. It is easy to see that measurements can be staggered so that the dimension of M is never greater than 2 (assuming that the pitch attitutde measurement, θ, is incorporated on every cycle). This simplifies the estimator greatly, particularly as a result of the reduced dimension of the matrix inversion in eq. 22. The price paid for the reduced dimensionality on any single pass is the increased logic and storage required to account for the 5 measurement combinations (θ plus U or W or H or $\Delta\dot{\theta}_{RCS}$); however, the computation-time saving is considerable. This estimation scheme is demonstrated in Ref. 6.

4.3 Rigid-Body Aerodynamic Control

It is assumed for the transition autopilot prototype that the shuttle vehicle is equipped with three independent aerodynamic controls -- aileron, rudder, and elevator. In fact, each control surface generates secondary torques, and for the delta-wing configuration, the elevator and ailerons are likely to be combined in "elevons".

The postulated control law includes nominal aileron, elevator, and rudder commands obtained from the guidance program (where the commands may be stored, computed, or arbitrarily set to zero) plus perturbations derived from linear feed back control laws. The control gains may simply be stored functions of the time-to-go, t_{go}; however, the nonlinearity of control effect,[7] the wide range of angle of attack, dynamic pressure, and Mach number,[8, 9] and the variation in "nominal" conditions from one mission to another may impose a requirement for additional flexibility. This will be especially true if the guidance program is unable to provide nominal control settings.

It is proposed, therefore, that the control gains be comprised of two parts: 1) a set of feedback gains stored as a function of t_{go} and 2) a factor to correct control surface effectiveness as a function of the estimated flight condition and the actual control setting. The gains dependent on t_{go} can be obtained prior to the flight by minimizing quadratic costs on the state and control in the linearized model of perturbations along a nominal flight path.[10] The control effectiveness factors are stored tables or functions which are dependent on the deviation of flight parameters from their nominal values. The control surface commands then take the form

$$\Delta \delta_{e_i} = C_m(\alpha, M, \delta_e) \underline{f}_1^T \underline{x}_i \qquad (23)$$

$$\Delta \delta_{a_i} = C_\ell(\alpha, M, \delta_a) \underline{f}_2^T \underline{v}_i \qquad (24)$$

$$\Delta \delta_{r_i} = C_n(\alpha, M, \delta_r) \underline{f}_3^T \underline{v}_i \qquad (25)$$

and

$$\delta_{e_i} = (\delta_{e_{Guidance}} + \Delta \delta_e)_i \qquad (26)$$

$$\delta_{a_i} = (\delta_{a_{Guidance}} + \Delta \delta_a)_i \qquad (27)$$

$$\delta_{r_i} = (\delta_{r_{Guidance}} + \Delta \delta_r)_i \qquad (28)$$

Here, C_m, C_ℓ, and C_n are not the conventional non-dimensional aerodynamic coefficients; they are the control effectiveness factors. The vectors $\underline{f}_1$, $\underline{f}_2$ and $\underline{f}_3$ are the feedback gains which are stored as functions of t_{go}.

For the delta wing configuration, control moments about the three body axes will be supplied by the rudder* and a pair of elevons. The specific control moments due to rudder deflection (δ_r), left elevon deflection (δ_{e_L}), and right elevon

*Some configurations have dual vertical tails and, therefore, two rudders.

deflection (δ_{e_R}), will be

$$L = L(\delta_{e_L}, \delta_{e_R}, \delta_r) \tag{29}$$

$$M = M(\delta_{e_L}, \delta_{e_R}) \tag{30}$$

$$N = N(\delta_{e_L}, \delta_{e_R}, \delta_r) \tag{31}$$

Expanding to first order,

$$\begin{aligned} L = {} & L(\delta_{e_L}, \delta_{e_R}, \delta_r)_{\text{Guidance}} \\ & + (L_{\delta_{e_L}} \Delta\delta_{e_L} + L_{\delta_{e_R}} \Delta\delta_{e_R} + L_{\delta_r} \Delta\delta_r)_{\text{Control}} \end{aligned} \tag{32}$$

$$\begin{aligned} M = {} & M(\delta_{e_L}, \delta_{e_R})_{\text{Guidance}} \\ & + (M_{\delta_{e_L}} \Delta\delta_{e_L} + M_{\delta_{e_R}} \Delta\delta_{e_R})_{\text{Control}} \end{aligned} \tag{33}$$

$$\begin{aligned} N = {} & N(\delta_{e_L}, \delta_{e_R}, \delta_r)_{\text{Guidance}} \\ & + (N_{\delta_{e_L}} \Delta\delta_{e_L} + N_{\delta_{e_R}} \Delta\delta_{e_R} + N_{\delta_r} \Delta\delta_r)_{\text{Control}} \end{aligned} \tag{34}$$

Then the perturbation specific control moments due to feedback control can be expressed as

$$\begin{bmatrix} L_{\delta_{e_L}} & L_{\delta_{e_R}} & L_{\delta_r} \\ M_{\delta_{e_L}} & M_{\delta_{e_R}} & 0 \\ N_{\delta_{e_L}} & N_{\delta_{e_R}} & N_{\delta_r} \end{bmatrix} \begin{bmatrix} \Delta\delta_{e_L} \\ \Delta\delta_{e_R} \\ \Delta\delta_r \end{bmatrix}_i = \begin{bmatrix} \underline{f}_2^T \underline{v} \\ \underline{f}_1^T \underline{x} \\ \underline{f}_3^T \underline{v} \end{bmatrix}_i \tag{35a}$$

or

$$\begin{bmatrix} \Delta\delta_{e_L} \\ \Delta\delta_{e_R} \\ \Delta\delta_r \end{bmatrix}_i = \begin{bmatrix} L_{\delta_{e_L}} & L_{\delta_{e_R}} & L_{\delta_r} \\ M_{\delta_{e_L}} & M_{\delta_{e_R}} & 0 \\ N_{\delta_{e_L}} & N_{\delta_{e_R}} & N_{\delta_r} \end{bmatrix}_i^{-1} \begin{bmatrix} 0 & \underline{f}_2^T \\ \underline{f}_1^T & 0 \\ 0 & \underline{f}_3^T \end{bmatrix}_i \begin{bmatrix} \underline{x} \\ \underline{v} \end{bmatrix}_i \qquad \text{'35b)}$$

and

$$\delta_{e_{L_i}} = (\delta_{e_{L_{Guidance}}} + \Delta\delta_{e_L})_i \qquad (36)$$

$$\delta_{e_{R_i}} = (\delta_{e_{R_{Guidance}}} + \Delta\delta_{e_R})_i \qquad (?7)$$

$$\delta_{r_i} = (\delta_{r_{Guidance}} + \Delta\delta_r)_i \qquad (38)$$

The partials $L_{\delta_{e_L}}$, $L_{\delta_{e_R}}$, are equal only for identical mean deflections, as control effects are nonlinea", particularly at high angle of attack. This distinction is particularly important as a result of the coupling of longitudinal and lateral-directional control actions indicated in eq. 35b. The separation of control gains into time-dependent and flight condition-dependent components is retained in eq. 35b; however, it is clearly possible to evaluate the aerodynamic control partials for the nominal flight condition and con rol settings, making them functions of time as well.

Above a certain angle of attack, the rudder effectiveness of most orbiter configurations will be negligible; thus, the 3rd column of the partial matrix in eq. 35 is effectively zero. While the elevons (or ailerons) could be used to control yaw, there are only 2 control surfaces for 3 axes, and the control of each axis is no longer independent of the other axes. This coupling could be beneficial ("proverse") or detrimental ("adverse"); in general, it will be necessary to supplement yaw control at high angles of attack using the reaction control system.

4.4 Rigid Body Reaction Jet Control

The reaction control system used for attitude control during orbital flight and high-angle entry will serve as a backup control system during the transition (as well as a primary system for yaw during the early transition phase). Consequently, it is sufficient to provide a rate damping and attitude limiting control law with wide deadbands for pitch and roll (or sideslip), assuming that the aerodynamic control surfaces will exert precise control within the dead zones. A more precise parabolic-curve switching logic is retained for high angle of attack yaw control. An important feature of the reaction control logic is that it not oppose the aerodynamic control inadvertently. Unless the lags in one system or the other are excessive, opposition will be prevented by using the same attitude estimates for each set of control laws.

The control law for the pitch axis commands the appropriate control jets when $|\Delta\dot{\theta}| > \dot{\theta}_{DB}$ or when $|\Delta\theta| > \theta_{DB}$. At high angles of attack, sideslip angle and yaw attitude are very nearly normal-mode components of the lateral-directional motion.[11] Thus it is appropriate to choose these as control axes, firing roll and yaw jets simultaneously when sideslip errors exceed their deadbands and firing yaw jets alone when yaw limits are exceeded. Sideslip rate can be approximated by a combination of roll and yaw rate errors, allowing the sideslip rate damping to command jets on when $|\Delta\dot{\psi} \cos \alpha + \Delta\dot{\varphi} \sin \alpha| > \dot{\beta}_{DB}$. The sideslip limiting logic consists of commanding jets on when $|\beta| > \beta_{DB}$. Parabolic switch curves similar to the Lunar Module's TJETLAW[3] or to those presented in Ref. 12 would be used for yaw control.

Below some angle of attack (to be determined) the sideslip control law degenerates into a redundant yaw controller, and the rudder becomes effective as well. At this point, the sideslip control law should be replaced by a body-axis roll law, firing on $|\Delta\dot{\varphi}| > \dot{\varphi}_{DB}$ or $|\Delta\varphi| > \varphi_{DB}$. Since the rudder becomes the primary yaw control effector, simple $\dot{\psi}$ and ψ deadband logic then replaces the parabolic curves.

The selection and timing logic for the control thrusters is similar to the Lunar Module logic [3] and is not repeated here. The choice of selection logic is based on the number and positioning of control jets; should a highly redundant configuration be chosen, a linear programming approach could be considered as an alternative to the LM design.[13] Timing jet firings to the nearest millisecond, as in the LM-Alone autopilot (rather than to the resolution of the sampling interval, as in the LM's CSM-Docked autopilot), is essential 1) when the reaction control system is the primary system and 2) when the mode of motion most closely associated with a control axis is unstable (this can occur for both longitudinal and lateral-directional modes).[14, 15]

4.5 Inertial and Aerodynamic Parameter Update

Inertial and aerodynamic parameter updates are separated in the Section 5 flow charts because it will not always be necessary to update both at the same time. Although the inertial update is included in the transition autopilot prototype, this subroutine need be done only when a main engine is firing and substantial changes in the vehicle mass are occuring (reaction control propellant usage can be ignored). In this case, the mass estimate is decreased as a function of the average thrust during the last autopilot sampling interval,

$$\mathrm{Mass}_i - \mathrm{Mass}_{i-1} \quad \Delta\mathrm{Mass}_i\ (\mathrm{Thrust}) \tag{39}$$

while moments of inertia are functions of the mass,

$$I_{xx_i} = I_{xx}\ (\mathrm{Mass}_i) \tag{40}$$

$$I_{yy_i} = I_{yy}\ (\mathrm{Mass}_i) \tag{41}$$

$$I_{zz_i} = I_{zz}\ (\mathrm{Mass}_i)\ , \tag{42}$$

and reaction jet specific control moments ("jet accelerations") are functions of the moments of inertia. Both the moments of inertia and the specific control moments are required, as these data are necessary for aerodynamic control as well as reaction control. The need to vary the relationships between mass and inertia as functions of cargo bay loading must be determined. It is not anticipated that products of inertia will be estimated; however, cargo bay loading could have a significant effect on the "nose-up" product of inertia, J_{xz}.

The aerodynamic parameter subroutine will update dynamic pressure, q, Mach number, M, aerodynamic control gains, and the estimator's state transition matrices. The dynamic pressure can be derived from the difference between measured stagnation and static pressure; it also can be derived from the estimated total velocity V_{Est}, and the air density, $\rho(H)$. A "best estimate" can be formed by combining the two using parametric optimization. Assuming static pressure and air temperature are measured (or stored as a function of altitude),

$$\rho(H) = p(H)\, R\, T(H)\,, \tag{43}$$

where R is the universal gas constant, and

$$\begin{aligned} q &= k_1 \left[1/2\, \rho(H) V_{Est}^2\right] + k_2 \Delta P \\ &= k_1 q_1 + k_2 q_2 \end{aligned} \tag{44}$$

The weighting constants, k_1 and k_2, are determined by the known variances in q_1 and q_2:

$$k_1 = \sigma_{q_2}^2 / (\sigma_{q_1}^2 + \sigma_{q_2}^2) \tag{45}$$

$$k_2 = \sigma_{q_1}^2 / (\sigma_{q_1}^2 + \sigma_{q_2}^2) = 1 - k_1 \tag{46}$$

The Mach number is determined from the total velocity estimate and the measured air temperature:

$$M = V_{Est} / \left[kRT(H)\right]^{1/2}, \tag{47}$$

where k is the ratio of specific heats for air.

An alternative formulation for q_1 eliminates eq. 43 and uses eq. 47. It is

$$q_1 = \frac{k}{2} p(H)M^2 \tag{48}$$

The control effectiveness correction parameters of eq. 23-25 or the inverse matrix of partial derivatives in eq. 35b are next computed as functions of α, M, and control deflection. The optimal gain vectors $\underline{f}_1$, $\underline{f}_2$, and $\underline{f}_3$ are computed as functions of t_{go}. The product of control effectiveness corrections and the gain vectors is taken in the update subroutine in order to minimize computations in the more frequently computed aerodynamic control subroutine. Finally, the estimator's state transition matrices are updated as functions of α, q, M, Mass (or Inertia), and t_{go}, using a dual approach similar to that for control gain computation. As in the case of the control gains, the purpose behind the dual approach is to maximize reliability and flexibility by incorporating both expected matrix values with measured values in the state transition matrix update.

4.6 Failure Detection

This subroutine has not been examined in detail. The failure detection algorithms will monitor estimator residuals for abnormalities which would indicate failed sensors or control effectors. They will be tailored to the most probable failure modes of the monitored instruments.

9.13.1 Autopilot (cont'd)

5. Detailed Flow Diagrams

This section contains detailed flow diagrams for the Unified Digital Autopilot Program with specific reference to the transition flight control mode.

9.13.1 Autopilot (cont'd)

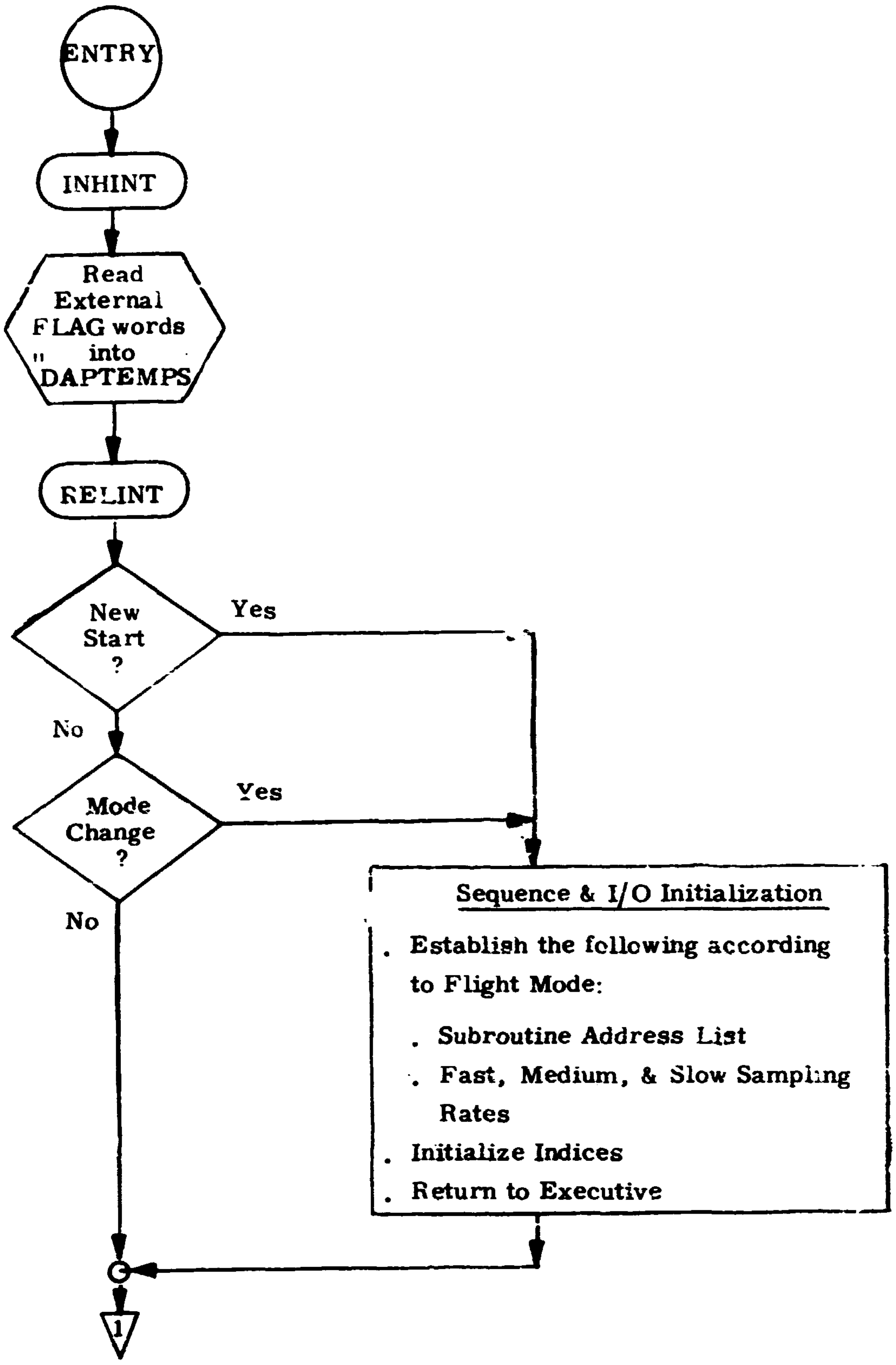

ENTRY
INHINT
Read External FLAG words into DAPTEMPS
RELINT
New Start ?
Yes
No
Mode Change ?
Yes
No
Sequence & I/O Initialization
. Establish the following according to Flight Mode:
. Subroutine Address List
. Fast, Medium, & Slow Sampling Rates
. Initialize Indices
. Return to Executive
1

Figure 2a DETAILED FLOW DIAGRAM

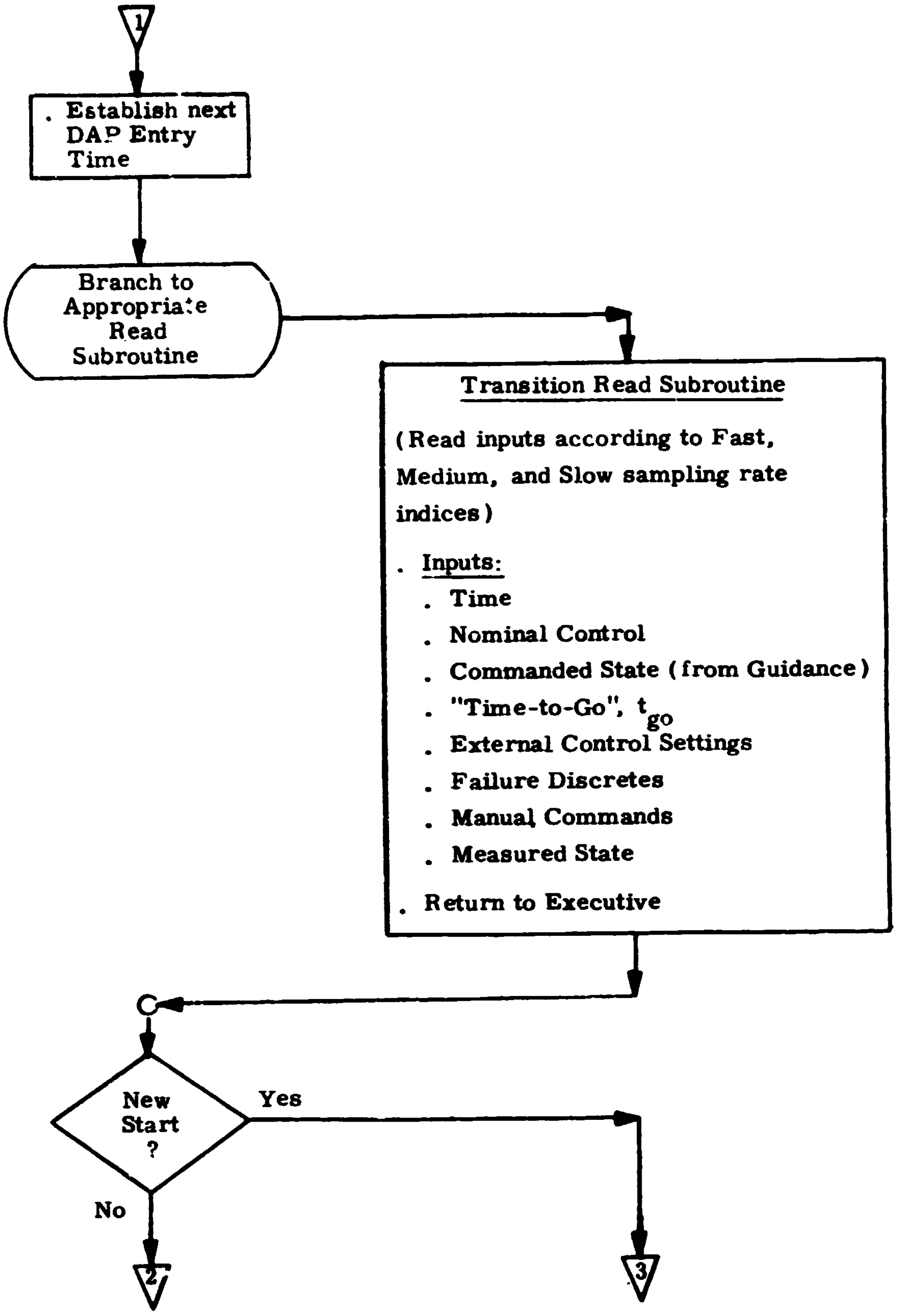

1
. Establish next DAP Entry Time
Branch to Appropriate Read Subroutine
Transition Read Subroutine
(Read inputs according to Fast, Medium, and Slow sampling rate indices)
. Inputs:
. Time
. Nominal Control
. Commanded State (from Guidance)
. "Time-to-Go", t_{go}
. External Control Settings
. Failure Discretes
. Manual Commands
. Measured State
. Return to Executive
C
New Start ?
Yes
No
2
3

Figure 2b DETAILED FLOW DIAGRAM

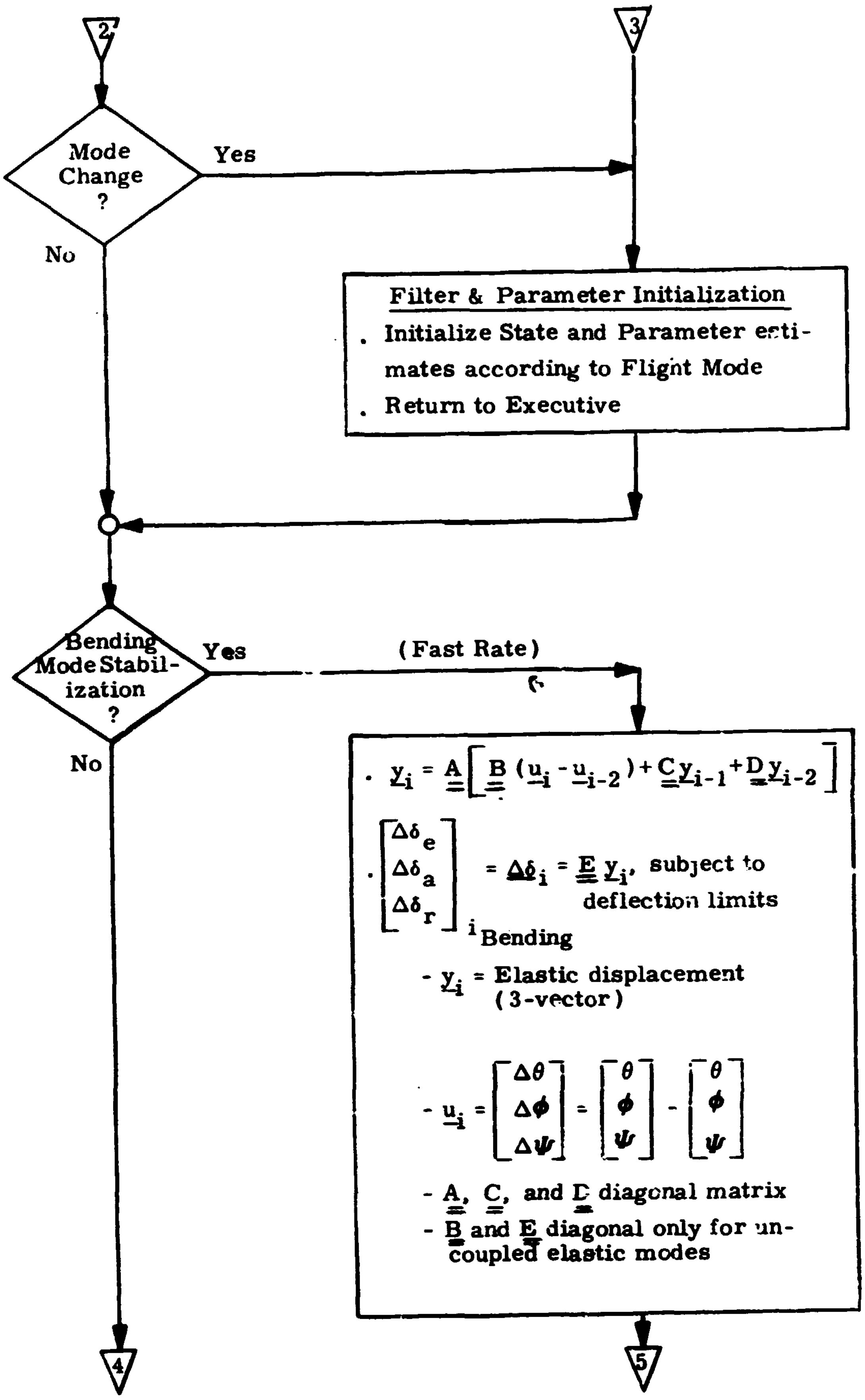

2
3
Mode Change ?
Yes
No
Filter & Parameter Initialization
. Initialize State and Parameter esti- mates according to Flight Mode
. Return to Executive
Bending Mode Stabil- ization ?
Yes
(Fast Rate)
No
• $\underline{y}_i = \underline{\underline{A}}\left[\underline{\underline{B}}(\underline{u}_i - \underline{u}_{i-2}) + \underline{\underline{C}}\underline{y}_{i-1} + \underline{\underline{D}}\underline{y}_{i-2}\right]$
• $\begin{bmatrix}\Delta\delta_e \\ \Delta\delta_a \\ \Delta\delta_r\end{bmatrix}_{i_{\text{Bending}}} = \underline{\Delta\delta}_i = \underline{\underline{E}}\,\underline{y}_i$, subject to deflection limits
- $\underline{y}_i$ = Elastic displacement (3-vector)
- $\underline{u}_i = \begin{bmatrix}\Delta\theta \\ \Delta\phi \\ \Delta\psi\end{bmatrix} = \begin{bmatrix}\theta \\ \phi \\ \psi\end{bmatrix} - \begin{bmatrix}\theta \\ \phi \\ \psi\end{bmatrix}$
- $\underline{\underline{A}}$, $\underline{\underline{C}}$, and $\underline{\underline{D}}$ diagonal matrix
- $\underline{\underline{B}}$ and $\underline{\underline{E}}$ diagonal only for un- coupled elastic modes
4
5

Figure 2c DETAILED FLOW DIAGRAM

9.13.1 <u>Autopilot</u> (cont'd)

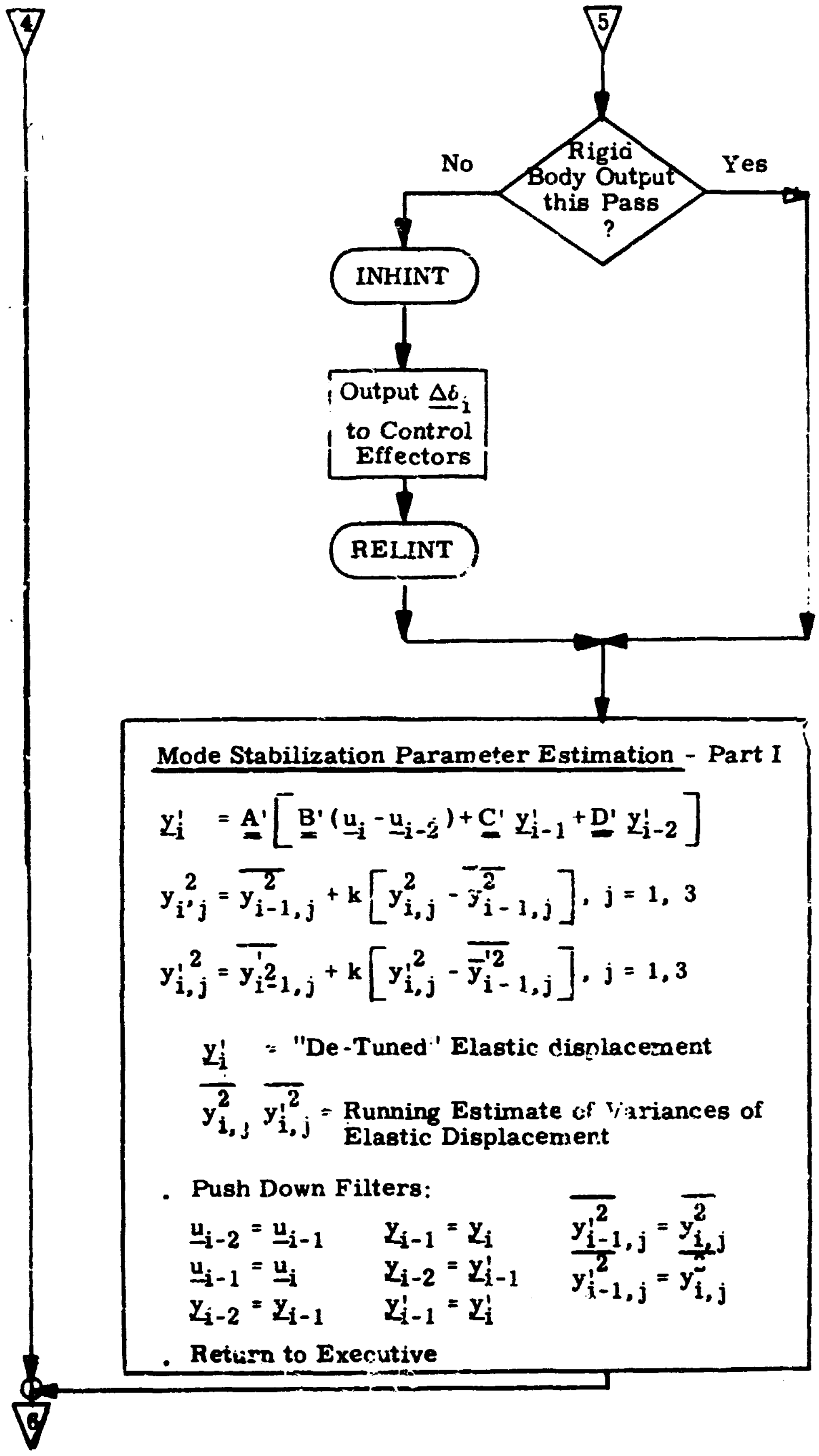
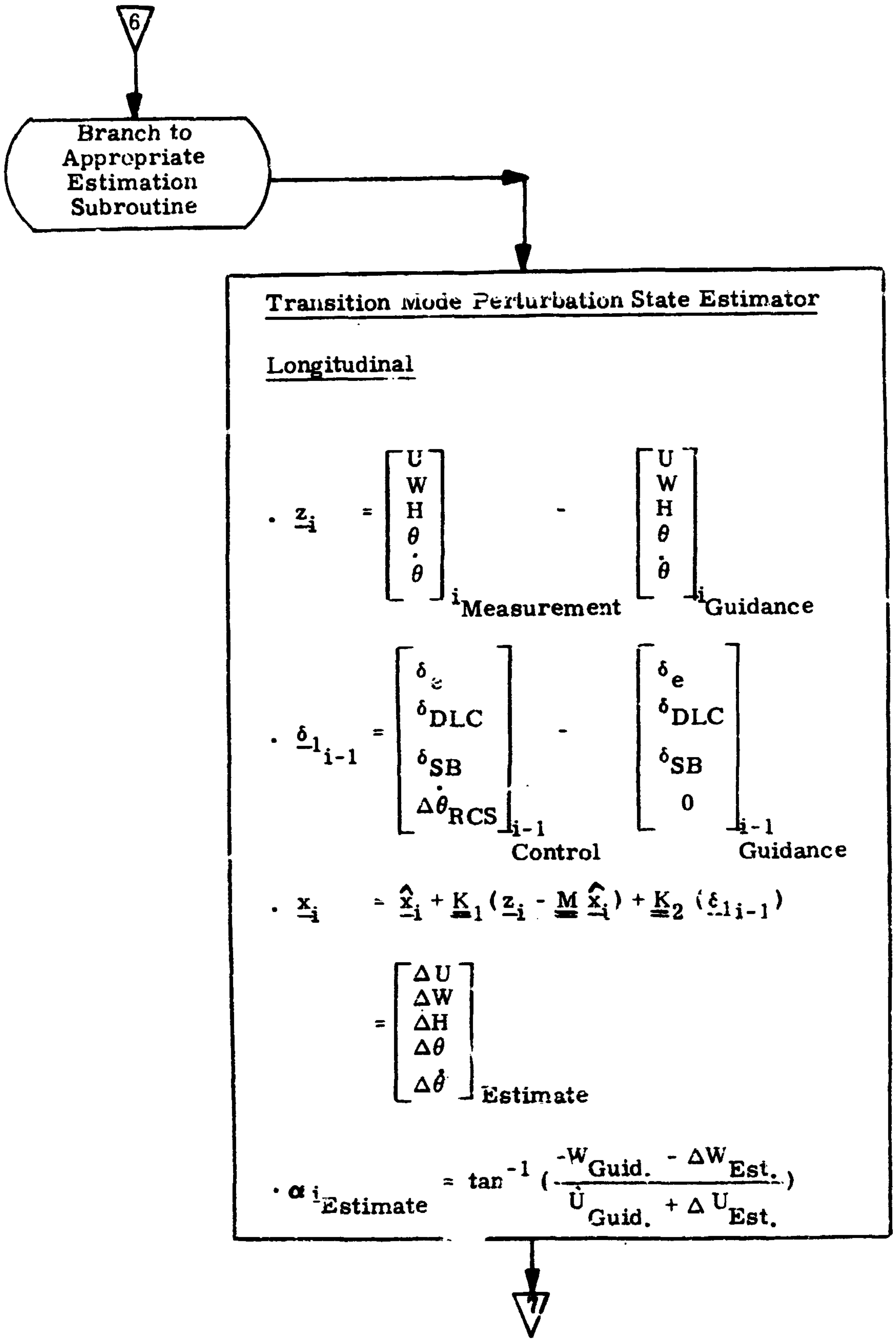

4
5
Rigid Body Output this Pass ?
No
Yes
INHINT
Output $\underline{\Delta\delta}_i$ to Control Effectors
RELINT
Mode Stabilization Parameter Estimation - Part I
$\underline{y}'_i = \underline{\underline{A}}'\left[\underline{\underline{B}}'(\underline{u}_i - \underline{u}_{i-2}) + \underline{\underline{C}}'\,\underline{y}'_{i-1} + \underline{\underline{D}}'\,\underline{y}'_{i-2}\right]$
$y^2_{i',j} = \overline{y^2_{i-1,j}} + k\left[y^2_{i,j} - \overline{y^2_{i-1,j}}\right]$, j = 1, 3
$y'^2_{i,j} = \overline{y'^2_{i-1,j}} + k\left[y'^2_{i,j} - \overline{y'^2_{i-1,j}}\right]$, j = 1, 3
$\underline{y}'_i$ = "De-Tuned" Elastic displacement
$\overline{y^2_{i,j}}$, $\overline{y'^2_{i,j}}$ = Running Estimate of Variances of Elastic Displacement
. Push Down Filters:
$\underline{u}_{i-2} = \underline{u}_{i-1}$
$\underline{u}_{i-1} = \underline{u}_i$
$\underline{y}_{i-2} = \underline{y}_{i-1}$
$\underline{y}_{i-1} = \underline{y}_i$
$\underline{y}_{i-2} = \underline{y}'_{i-1}$
$\underline{y}'_{i-1} = \underline{y}'_i$
$\overline{y'^2_{i-1,j}} = \overline{y^2_{i,j}}$
$\overline{y'^2_{i-1,j}} = y_{i,j}$
. Return to Executive
6

Figure 2d DETAILED FLOW DIAGRAM



Transition Mode Perturbation State Estimator

Longitudinal

$$\cdot\ \underline{z}_i = \begin{bmatrix} U \\ W \\ H \\ \theta \\ \dot{\theta} \end{bmatrix}_{i\,\text{Measurement}} - \begin{bmatrix} U \\ W \\ H \\ \theta \\ \dot{\theta} \end{bmatrix}_{i\,\text{Guidance}}$$

$$\cdot\ \underline{\delta}_{1_{i-1}} = \begin{bmatrix} \delta_e \\ \delta_{DLC} \\ \delta_{SB} \\ \Delta\dot{\theta}_{RCS} \end{bmatrix}_{i-1\ \text{Control}} - \begin{bmatrix} \delta_e \\ \delta_{DLC} \\ \delta_{SB} \\ 0 \end{bmatrix}_{i-1\ \text{Guidance}}$$

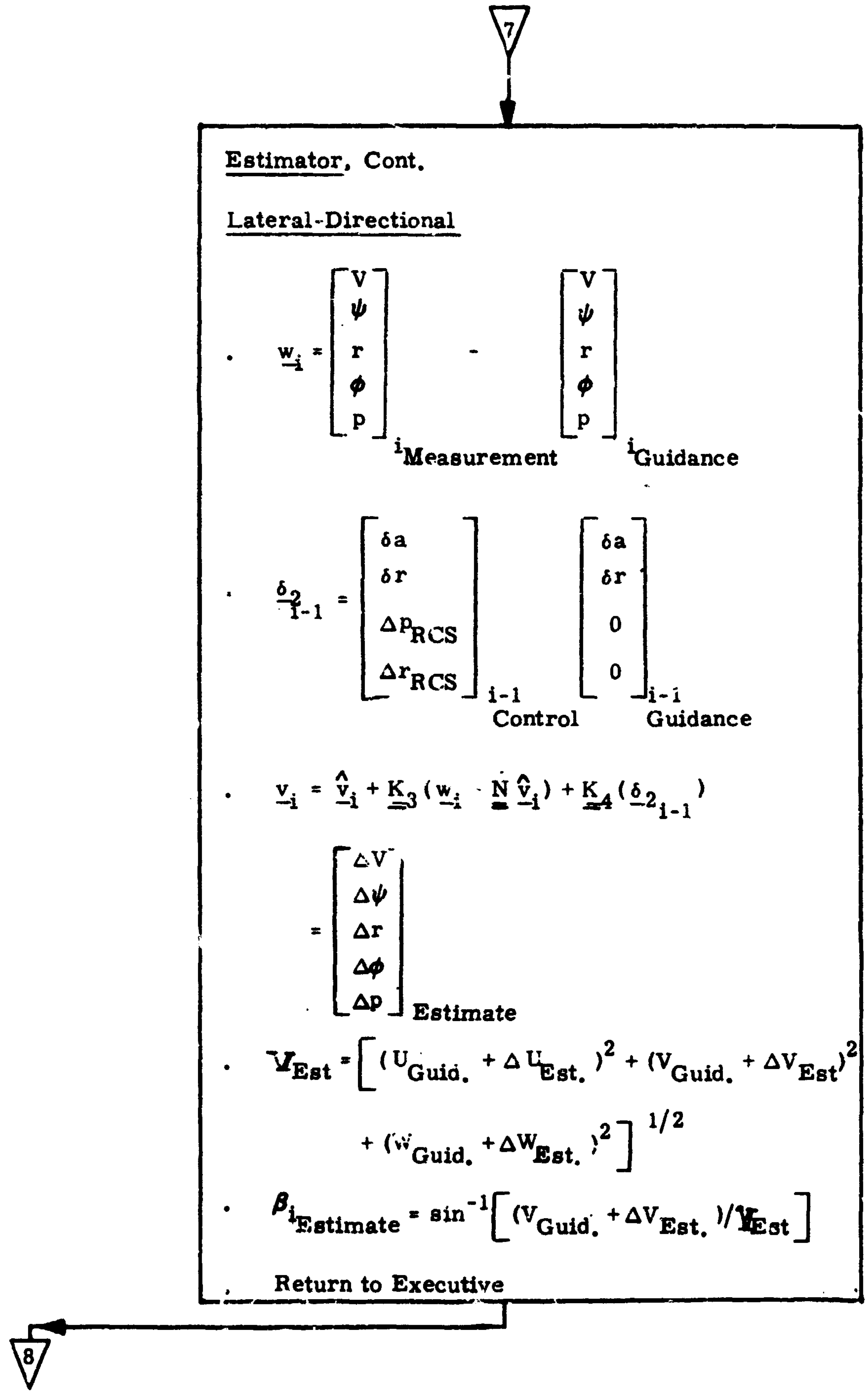
$$\cdot\ \underline{x}_i = \hat{\underline{x}}_i + \underline{\underline{K}}_1 (\underline{z}_i - \underline{\underline{M}}\, \hat{\underline{x}}_i) + \underline{\underline{K}}_2 (\underline{\delta}_{1_{i-1}})$$

$$= \begin{bmatrix} \Delta U \\ \Delta W \\ \Delta H \\ \Delta\theta \\ \Delta\dot{\theta} \end{bmatrix}_{\text{Estimate}}$$

$$\cdot\ \alpha_{i_{\text{Estimate}}} = \tan^{-1}\left(\frac{-W_{\text{Guid.}} - \Delta W_{\text{Est.}}}{U_{\text{Guid.}} + \Delta U_{\text{Est.}}}\right)$$

Figure 2e DETAILED FLOW DIAGRAM

9.13.1 Autopilot (cont'd)



Estimator, Cont.

Lateral-Directional

- $$\underline{w}_i = \begin{bmatrix} V \\ \psi \\ r \\ \phi \\ p \end{bmatrix}_{i\,\text{Measurement}} - \begin{bmatrix} V \\ \psi \\ r \\ \phi \\ p \end{bmatrix}_{i\,\text{Guidance}}$$

- $$\underline{\delta}_{2_{i-1}} = \begin{bmatrix} \delta a \\ \delta r \\ \Delta p_{RCS} \\ \Delta r_{RCS} \end{bmatrix}_{i-1\,\text{Control}} \qquad \begin{bmatrix} \delta a \\ \delta r \\ 0 \\ 0 \end{bmatrix}_{i-i\,\text{Guidance}}$$

- $$\underline{v}_i = \hat{\underline{v}}_i + \underline{\underline{K}}_3 (\underline{w}_i - \underline{\underline{N}}\,\hat{\underline{v}}_i) + \underline{\underline{K}}_4 (\underline{\delta}_{2_{i-1}})$$

$$= \begin{bmatrix} \Delta V \\ \Delta\psi \\ \Delta r \\ \Delta\phi \\ \Delta p \end{bmatrix}_{\text{Estimate}}$$

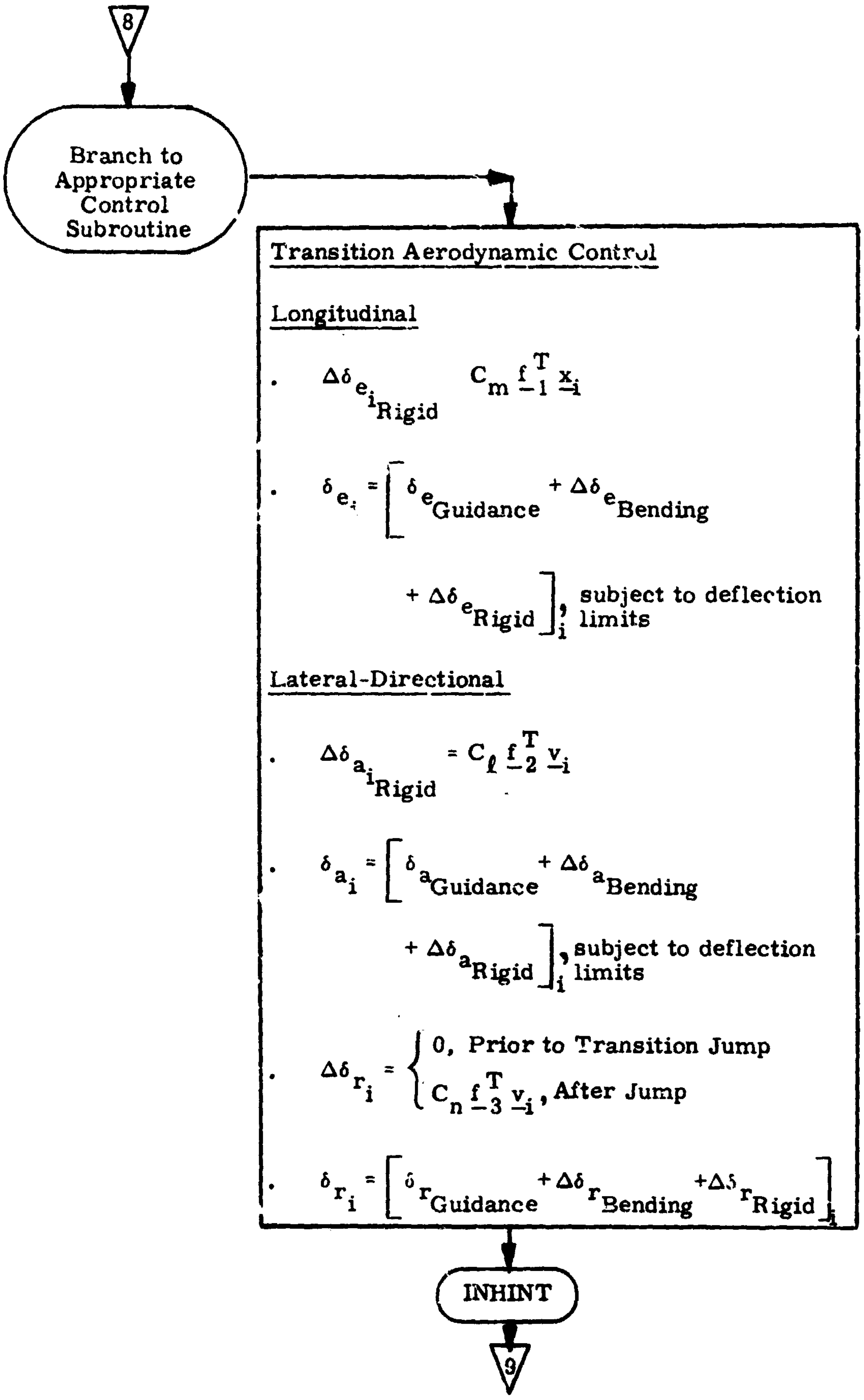
- $$V_{Est} = \Big[(U_{Guid.} + \Delta U_{Est.})^2 + (V_{Guid.} + \Delta V_{Est})^2 + (W_{Guid.} + \Delta W_{Est.})^2 \Big]^{1/2}$$

- $$\beta_{i_{Estimate}} = \sin^{-1}\Big[(V_{Guid.} + \Delta V_{Est.})/V_{Est} \Big]$$

- Return to Executive

Figure 2f DETAILED FLOW DIAGRAM

9.13.1 Autopilot (cont'd)



Transition Aerodynamic Control

Longitudinal

- $\Delta\delta_{e_{i_{Rigid}}} \quad C_m \underline{f}_1^T \underline{x}_i$

- $\delta_{e_i} = \Big[\delta_{e_{Guidance}} + \Delta\delta_{e_{Bending}} + \Delta\delta_{e_{Rigid}} \Big]_i$, subject to deflection limits

Lateral-Directional

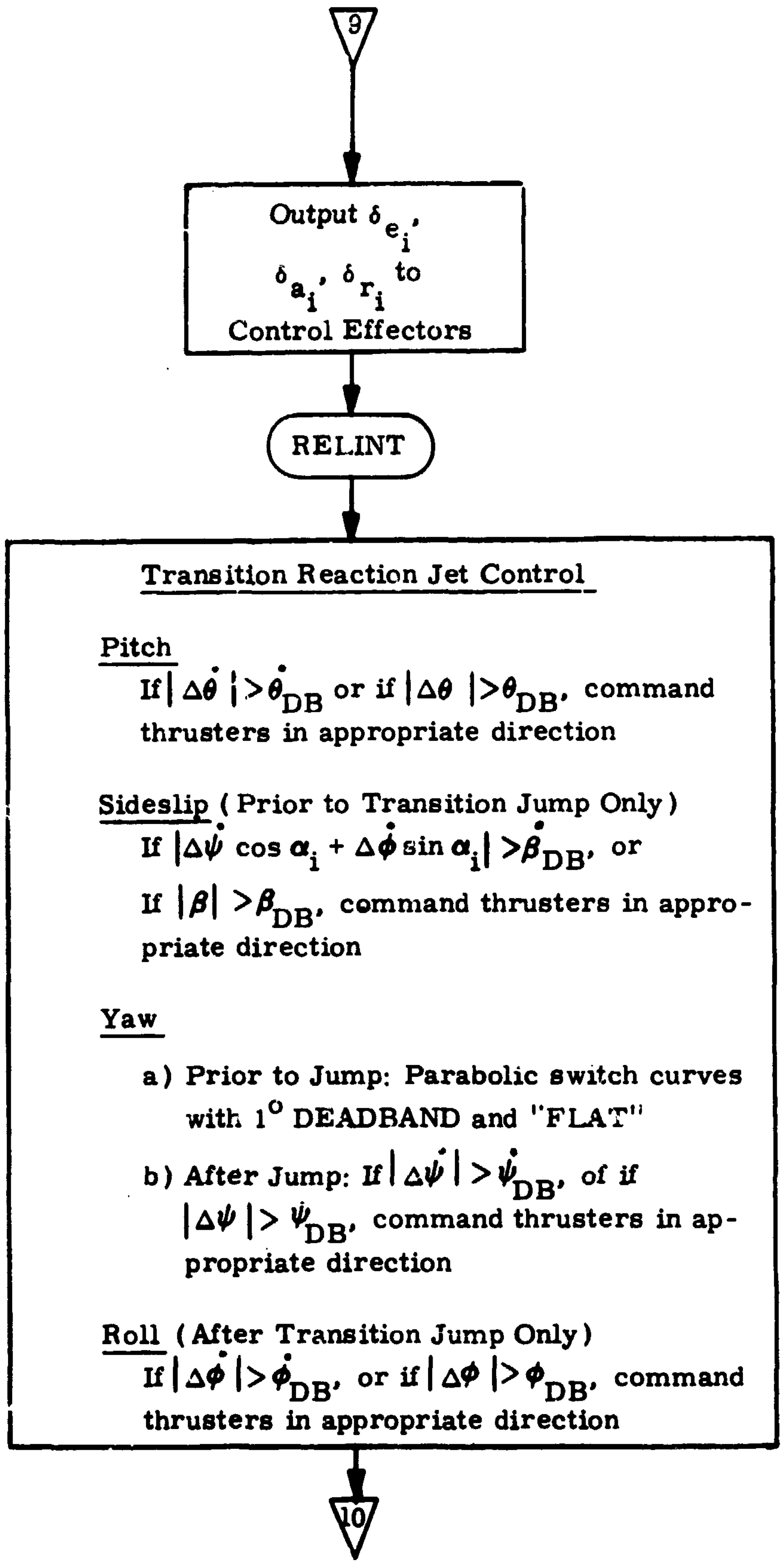
- $\Delta\delta_{a_{i_{Rigid}}} = C_\ell \underline{f}_2^T \underline{v}_i$

- $\delta_{a_i} = \Big[\delta_{a_{Guidance}} + \Delta\delta_{a_{Bending}} + \Delta\delta_{a_{Rigid}} \Big]_i$, subject to deflection limits

- $\Delta\delta_{r_i} = \begin{cases} 0, \text{ Prior to Transition Jump} \\ C_n \underline{f}_3^T \underline{v}_i, \text{ After Jump} \end{cases}$

- $\delta_{r_i} = \Big[\delta_{r_{Guidance}} + \Delta\delta_{r_{Bending}} + \Delta\delta_{r_{Rigid}} \Big]_i$

Figure 2g DETAILED FLOW DIAGRAM



Output δ_{e_i}, δ_{a_i}, δ_{r_i} to Control Effectors

Transition Reaction Jet Control

Pitch

If $|\Delta\dot{\theta}| > \dot{\theta}_{DB}$ or if $|\Delta\theta| > \theta_{DB}$, command thrusters in appropriate direction

Sideslip (Prior to Transition Jump Only)

If $|\Delta\dot{\psi} \cos \alpha_i + \Delta\dot{\phi} \sin \alpha_i| > \dot{\beta}_{DB}$, or

If $|\beta| > \beta_{DB}$, command thrusters in appropriate direction

Yaw

a) Prior to Jump: Parabolic switch curves with 1° DEADBAND and "FLAT"

b) After Jump: If $|\Delta\dot{\psi}| > \dot{\psi}_{DB}$, of if $|\Delta\psi| > \psi_{DB}$, command thrusters in appropriate direction

Roll (After Transition Jump Only)

If $|\Delta\dot{\phi}| > \dot{\phi}_{DB}$, or if $|\Delta\phi| > \phi_{DB}$, command thrusters in appropriate direction

Figure 2h DETAILED FLOW DIAGRAM

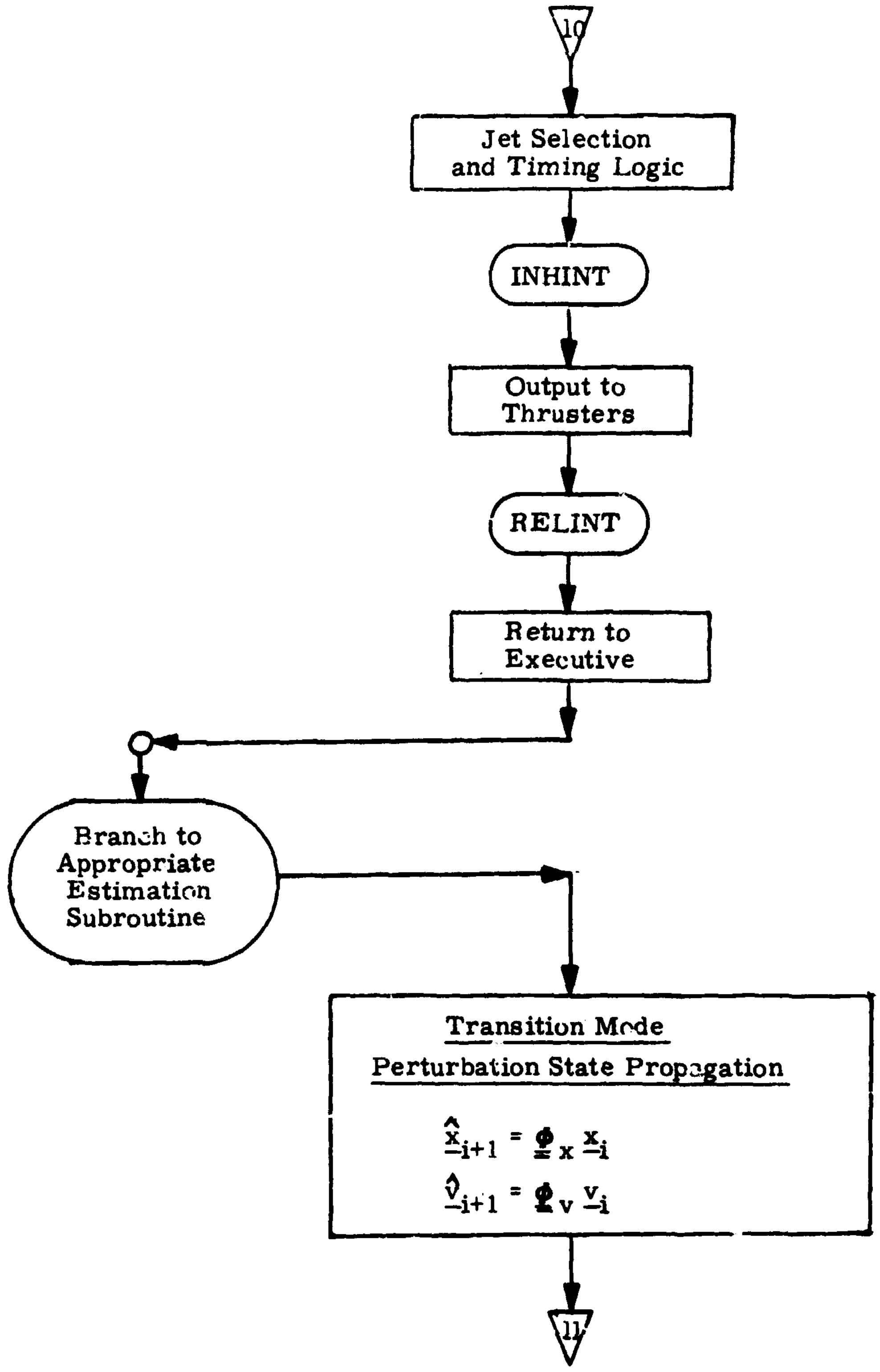
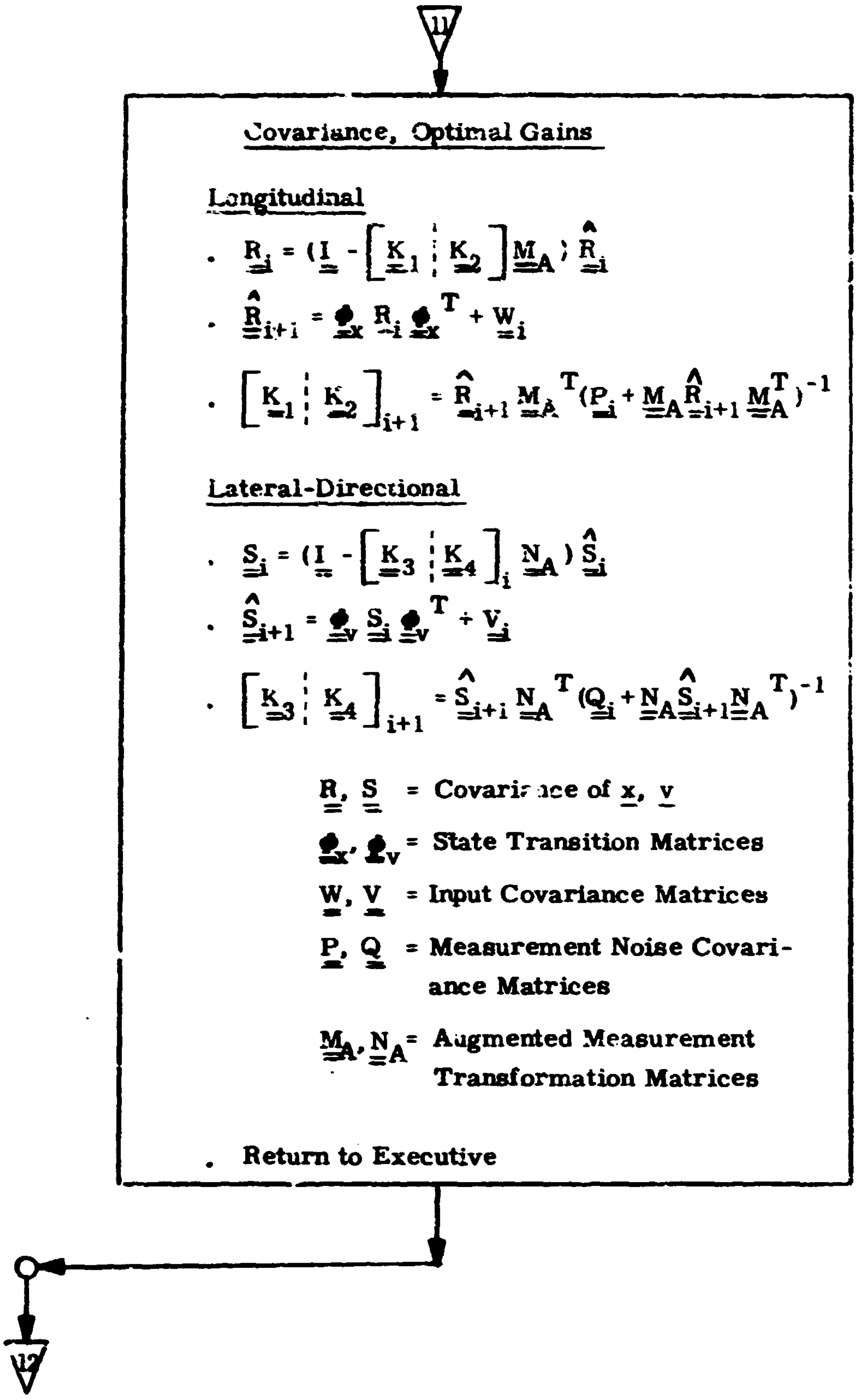

10
Jet Selection and Timing Logic
INHINT
Output to Thrusters
RELINT
Return to Executive
Branch to Appropriate Estimation Subroutine
Transition Mode
Perturbation State Propagation
$\hat{\underline{x}}_{i+1} = \underline{\underline{\Phi}}_x \underline{x}_i$
$\hat{\underline{v}}_{i+1} = \underline{\underline{\Phi}}_v \underline{v}_i$
11

Figure 2i DETAILED FLOW DIAGRAM

9.13.1 Autopilot (cont'd)



Covariance, Optimal Gains

Longitudinal

- $\underline{\underline{R}}_i = (\underline{\underline{I}} - \left[\underline{\underline{K}}_1 \,\vdots\, \underline{\underline{K}}_2 \right] \underline{\underline{M}}_A) \hat{\underline{\underline{R}}}_i$
- $\hat{\underline{\underline{R}}}_{i+i} = \underline{\underline{\Phi}}_x \underline{R}_i \underline{\underline{\Phi}}_x^T + \underline{\underline{W}}_i$
- $\left[\underline{\underline{K}}_1 \,\vdots\, \underline{\underline{K}}_2 \right]_{i+1} = \hat{\underline{\underline{R}}}_{i+1} \underline{\underline{M}}_A^T (\underline{\underline{P}}_i + \underline{\underline{M}}_A \hat{\underline{\underline{R}}}_{i+1} \underline{\underline{M}}_A^T)^{-1}$

Lateral-Directional

- $\underline{\underline{S}}_i = (\underline{\underline{I}} - \left[\underline{\underline{K}}_3 \,\vdots\, \underline{\underline{K}}_4 \right]_i \underline{\underline{N}}_A) \hat{\underline{\underline{S}}}_i$
- $\hat{\underline{\underline{S}}}_{i+1} = \underline{\underline{\Phi}}_v \underline{\underline{S}}_i \underline{\underline{\Phi}}_v^T + \underline{\underline{V}}_i$
- $\left[\underline{\underline{K}}_3 \,\vdots\, \underline{\underline{K}}_4 \right]_{i+1} = \hat{\underline{\underline{S}}}_{i+i} \underline{\underline{N}}_A^T (\underline{\underline{Q}}_i + \underline{\underline{N}}_A \hat{\underline{\underline{S}}}_{i+1} \underline{\underline{N}}_A^T)^{-1}$

$\underline{\underline{R}}, \underline{\underline{S}}$ = Covariance of $\underline{x}, \underline{v}$

$\underline{\underline{\Phi}}_x, \underline{\underline{\Phi}}_v$ = State Transition Matrices

$\underline{\underline{W}}, \underline{\underline{V}}$ = Input Covariance Matrices

$\underline{\underline{P}}, \underline{\underline{Q}}$ = Measurement Noise Covariance Matrices

$\underline{\underline{M}}_A, \underline{\underline{N}}_A$ = Augmented Measurement Transformation Matrices

- Return to Executive

Figure 2j DETAILED FLOW DIAGRAM

Mode Stabilization Parameter Estimation - Part II

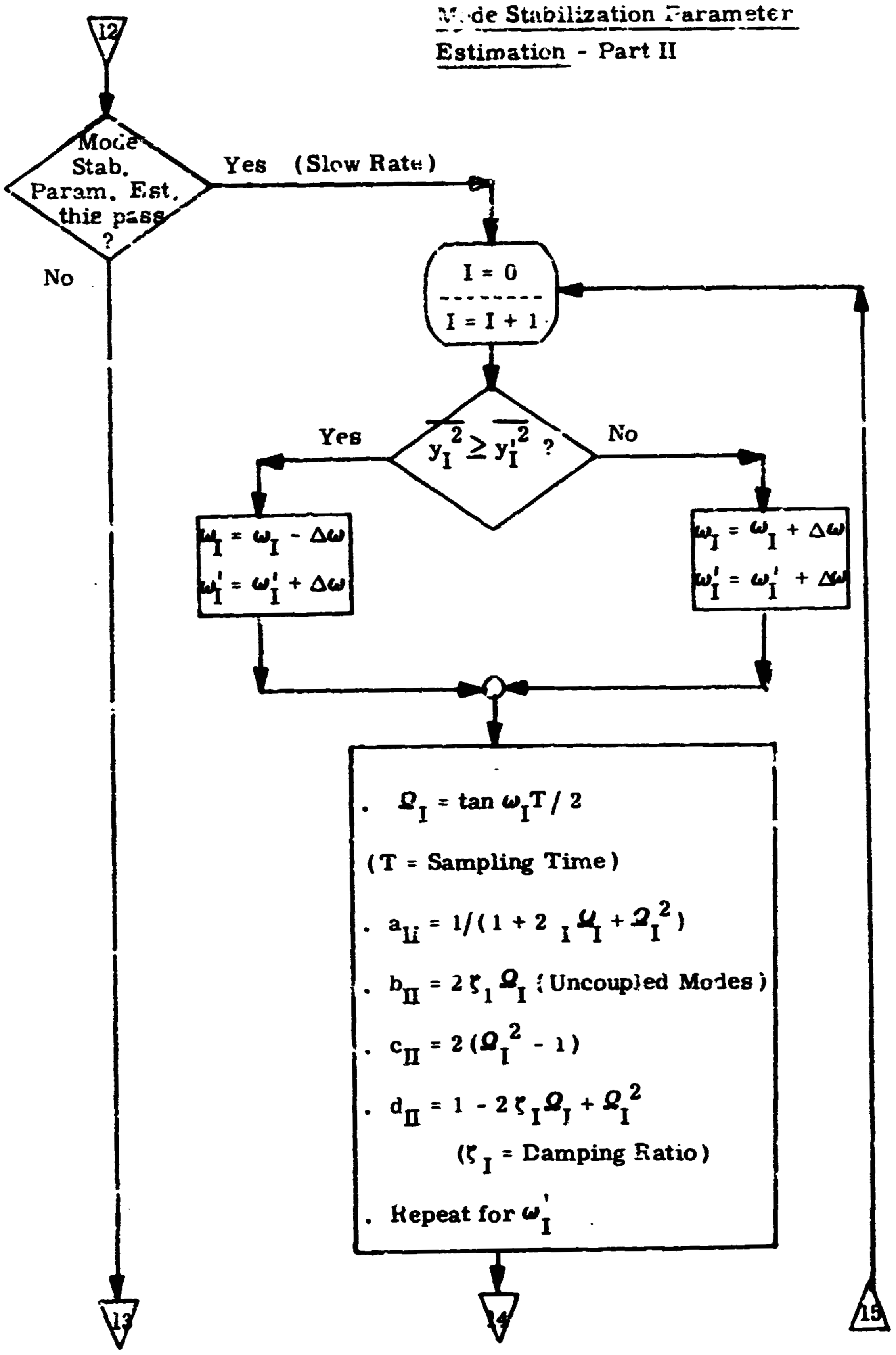

12
Mode Stab. Param. Est. this pass ?
Yes (Slow Rate)
No
I = 0
I = I + 1
$\overline{y_I^2} \geq \overline{y_I'^2}$?
Yes
No
$\omega_I = \omega_I - \Delta\omega$
$\omega_I' = \omega_I' + \Delta\omega$
$\omega_I = \omega_I + \Delta\omega$
$\omega_I' = \omega_I' + \Delta\omega$
. $\Omega_I = \tan \omega_I T / 2$
(T = Sampling Time)
. $a_{1I} = 1/(1 + 2\zeta_I\Omega_I + \Omega_I^2)$
. $b_{II} = 2\zeta_I\Omega_I$ (Uncoupled Modes)
. $c_{II} = 2(\Omega_I^2 - 1)$
. $d_{II} = 1 - 2\zeta_I\Omega_I + \Omega_I^2$
(ζ_I = Damping Ratio)
. Repeat for ω_I'
13
14
15

Figure 2k DETAILED FLOW DIAGRAM

9.13.1 Autopilot (cont'd)

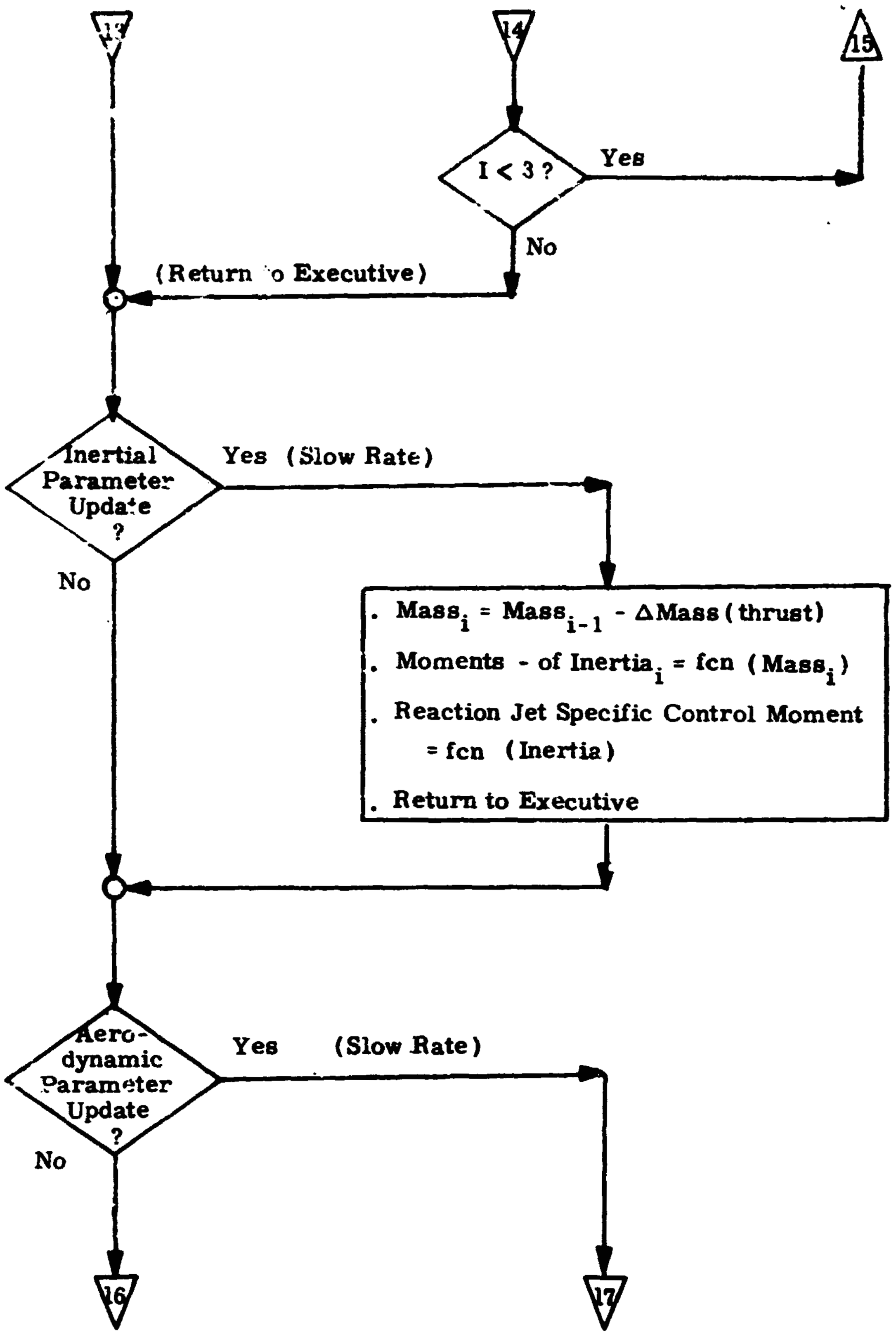

13
14
15
I < 3 ?
Yes
No
(Return to Executive)
Inertial Parameter Update ?
Yes (Slow Rate)
No
. $Mass_i = Mass_{i-1} - \Delta Mass$ (thrust)
. Moments - of $Inertia_i$ = fcn ($Mass_i$)
. Reaction Jet Specific Control Moment = fcn (Inertia)
. Return to Executive
Aero-dynamic Parameter Update ?
Yes (Slow Rate)
No
16
17

Figure 21 DETAILED FLOW DIAGRAM

9.13.1 Autopilot (cont'd)

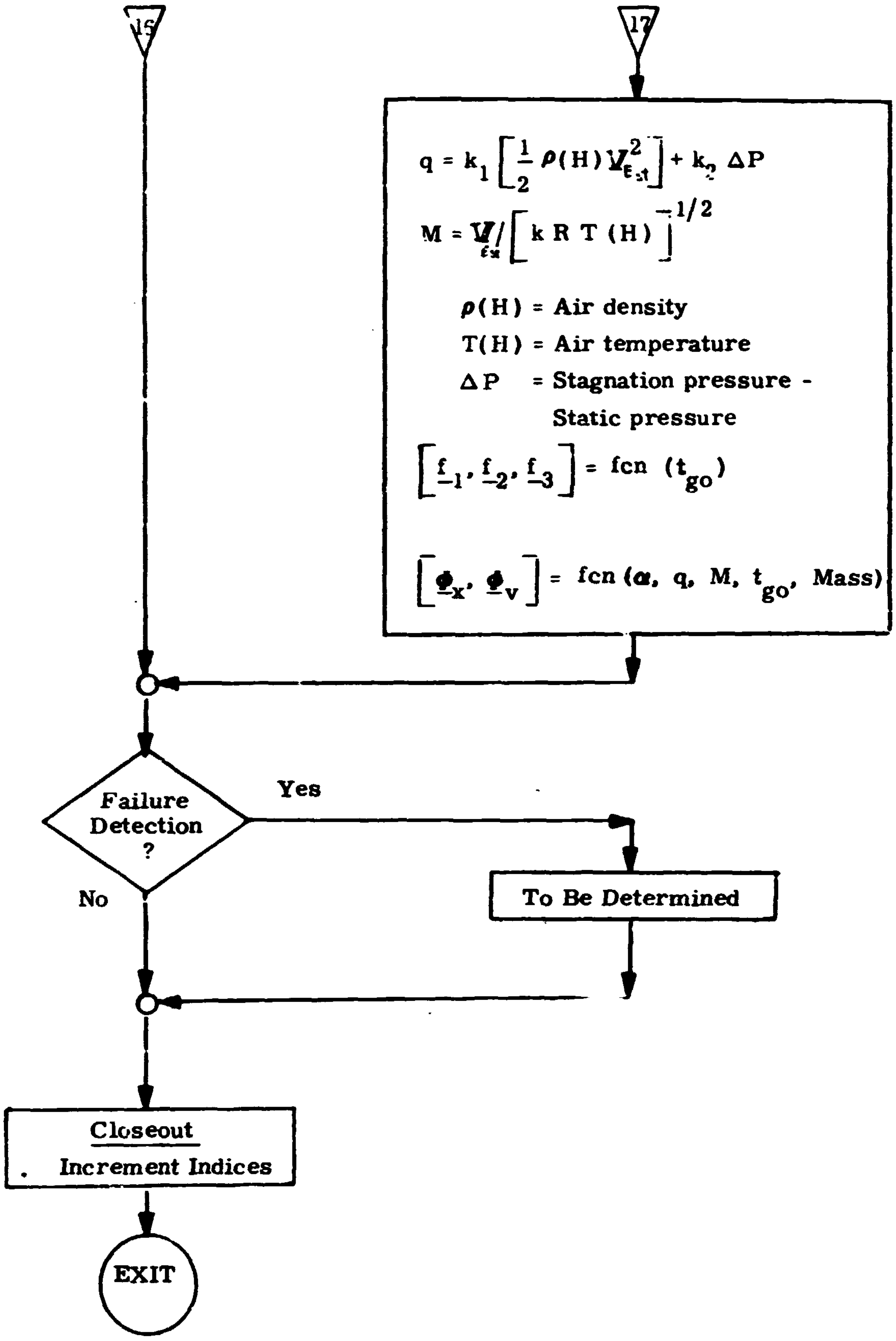

15
17
$q = k_1 \left[\frac{1}{2}\rho(H)\underline{V}_{Est}^{2}\right] + k_2 \Delta P$
$M = \underline{V}_{Ex} / \left[k R T(H)\right]^{1/2}$
$\rho(H)$ = Air density
$T(H)$ = Air temperature
ΔP = Stagnation pressure - Static pressure
$\left[\underline{f}_1, \underline{f}_2, \underline{f}_3\right]$ = fcn (t_{go})
$\left[\underline{\Phi}_x, \underline{\Phi}_v\right]$ = fcn (α, q, M, t_{go}, Mass)
Failure Detection ?
Yes
No
To Be Determined
Closeout
. Increment Indices
EXIT

Figure 2m DETAILED FLOW DIAGRAM

9.13.1 Autopilot (cont'd)

6. Supplementary Information

The functional flow diagram of the Unified Digital Autopilot Program is largely independent of the mission phase and vehicle configuration; however, detailed flow diagrams depend on mission phase. The estimation and control algorithms are further dependent on vehicle configuration, while the parameters for control may vary with vehicle loading as well as flight conditions. Algorithms and control parameters will be further developed as mission profiles and vehicle characteristics are defined.

In addition to standardizing functional flow, the Unified Digital Autopilot Program should allow sharing of subroutines by autopilot logic developed for various mission phases. For example, the transition estimator could be useful during cruising flight and landing approach. It also can provide guidance system monitoring during atmospheric boost and on-line estimation during orbiter powered ascent. Failure detection subroutines will be dedicated to specific measurement and control subsystems; thus the subroutines will be employed as the subsystems are required and may be used during more than one mission phase. In order for the autopilot program to be truly "unified", it will be necessary to consider autopilot requirements for all mission phases in a parallel fashion.

Future work will follow two specific paths. The unified autopilot concept will be refined through simulation and further analysis. The autopilot requirements for specific mission phases will be pursued in accordance with the assigned tasks of NAS9-10268, with particular emphasis on the requirements of high cross-range, delta-wing configurations.

9.13.1 Autopilot (cont'd)

References

1. Stengel, R.F., "Digital Resonant and Notch Filters," MIT DL SAD Memo 16-68, Oct., 1968.

2. Kuo, B.C., Analysis and Synthesis of Sampled-Data Control Systems, Prentice-Hall Englewood Cliffs, 1963.

3. "Guidance System Operations Plan for Manned LM Earth Orbital and Lunar Missions Using Program LUMINARY 1D," Section 3, Digital Autopilot (Rev. 6), MIT CSDL R-567, Feb., 1971.

4. Kalman, R.E., "A New Approach to Linear Filtering and Prediction Problems," ASME Journal of Basic Engineering, 82D, March 1960, pp. 35-45.

5. Liebelt, P.B., An Introduction to Optimal Estimation, Addison-Wesley, Reading, 1967.

6. Stengel, R.F., "Attitude Estimation for a Highly Flexible Vehicle," MIT DL AAP Memo 70-388M-1, Jan., 1970.

7. Stengel, R.F., "Aerodynamic Trim Boundaries and Nonlinear Elevator Effects for the NASA MSC April 1970 Baseline Orbiter," MIT CSDL SSV Memo 70-23C-16, Nov., 1970.

8. Stengel, R. F., "Optimal Transition from Entry to Cruising Flight," MIT CSDL E-2522 (also AIAA Paper 70-1018), July 1970.

9. Stengel, R.F., "Space Shuttle Transition to Cruising Flight," MIT CSDL E-2539, Sept. 1970.

10. Bryson, A.E., Jr. Ho, Y.C., Applied Optimal Control, Blaisdell, Waltham, 1969.

11. Goss, R.D., "Normal Modes for Lateral Directional Motions of the Space Shuttle Vehicle During Entry," MIT CSDL SSV Memo 70-23C-14, Sept. 1970.

12. Cox, K.J., "Orbital operations rotational DAP for the redundant flight control systems simulator," NASA MSC internal memo, Feb. 1971.

13. Crawford, B.S., "Operation and Design of Multi-Jet Spacecraft Control Systems," MIT DL T-509, Sept. 1968.

9.13.1 Autopilot (cont'd)

14. Basile, P.S. "Handling Qualities Parameters for the North American Rockwell Delta-wing Orbiter 134C," MIT CSDL SSV Memo (to be published).

15. Freeman, D.C., "Low-Subsonic Aerodynamic Characteristics of a Space Shuttle-Orbiter Concept with a Blended Delta Wing-Body," NASA TM X-2209, Jan., 1971.

9.14 CRUISE AND FERRY CRUISE

The Cruise phase of the shuttle mission applies to the period of atmospheric flight between transition and approach. The Ferry Cruise phase refers to powered flight betweem airports. These phases include low altitude earth-relative navigation, guidance and aerosurface control. The Software functions required in this mission phase are the following:

1. Powered flight navigation augmented by external data to maintain earth-relative position.
2. Powered flight guidance resulting in thrust level control (for Ferry only) and autopilot (attitude) commands required to achieve desired intersection with the terminal approach path for landing.
3. Autopilot computations to result in proper aerodynamic surface control and trim settings to maintain desired heading and flight path.

9.14.1 Navigation TBD

9.14.2 Guidance TBD

9.14.3 Cruise Autopilot

SPACE SHUTTLE

GN&C SOFTWARE EQUATION SUBMITTAL

Software Equation Section Cruise Autopilot Submittal No. 36

Function Provide characteristics of manual modes

Module No. OC-6 Function No. 3 (MSC 03690)

Submitted For: EG-6 Co. GCD
(Name)

Date: 21 October 1971

NASA Contact: W. H. Peters Organization EG2
(Name)

Approved by Panel III K.T. Date 10/21/71
(Chairman)

Summary Description: This submittal accompanies submittal 35, Approach Guidance. It provides the control system characteristics associated with the manual control employed during verification testing of the Approach Guidance equations.

Shuttle Configuration: (Vehicle, Aero Data, Sensor, Et Cetera)

Coefficients associated with NR 161C orbiter.

Comments:

(Design Status)

(Verification Status)

Panel Comments:

9.14.3.1 Manual Modes

The control system for the delta wing orbiter (as implemented by the Control Requirements Branch, GDC, in their Cruise and Landing Simulator) provides the following modes of operation:

(1) Direct Manual (CM)
(2) Rate Command (RC)
(3) Rate Command/Attitude Hold (RCAH)
(4) Automatic Guidance (AUTO)
(5) Yaw Damper (YD)

Figures 1 through 3 present the complete control systems implemented for pitch, roll and yaw, respectively.

The systems presented include certain modifications over the previously implemented control systems (for the straight wing orbiter.) For the pitch and roll systems, the RC and RCAH control loops are no longer separate and a logic element has been included to preclude switching to attitude hold with large vehicle rates. In the pitch system,the attitide hold function is not engaged unless the selected mode is RCAH, the hand controller is in detent (QIN), and the vehicle pitch rate (q) minus the turn coordinator pitch rate (q_{TC}) is within ± .0175 rad/sec (QLIN). Similar conditions are also required to engage the roll attitude hold function except for the turn coordinate rate that is not required in the roll channel.

The first parts of Figures 1 and 3 are presented to show the turn coordinator for pitch in the RCAH mode and for yaw in the RCAH mode with the yaw damper disengaged. The turn coordinator as implemented provides improved orbiter response during banked manuevers.

In Figure 3, a second feedback loop (in addition to the yaw damper) is shown in the yaw control system. It has been included for studies to evaluate the effectiveness of lateral acceleration feedback during cruise and landing. For baseline purposes, however, the loop gain (R_3) is zero and the loop may be disregarded.

Table 1 presents the gains implemented for all three control systems. Table 2 shows the logic expressions for controlling the numerous switching functions used in the control systems. (The switch designations represent logical conditions, not individual switches; and the same switch is employed in various locations). Integrators controlled by the logical switches are considered in reset, or temporary reset (t≈0.01 sec.), when the switch is closed. Attitude and trim commands are input to the integrator initial condition terminals. The integrator gain is assumed to be plus one.

FIGURE 1

PITCH CONTROL SYSTEM

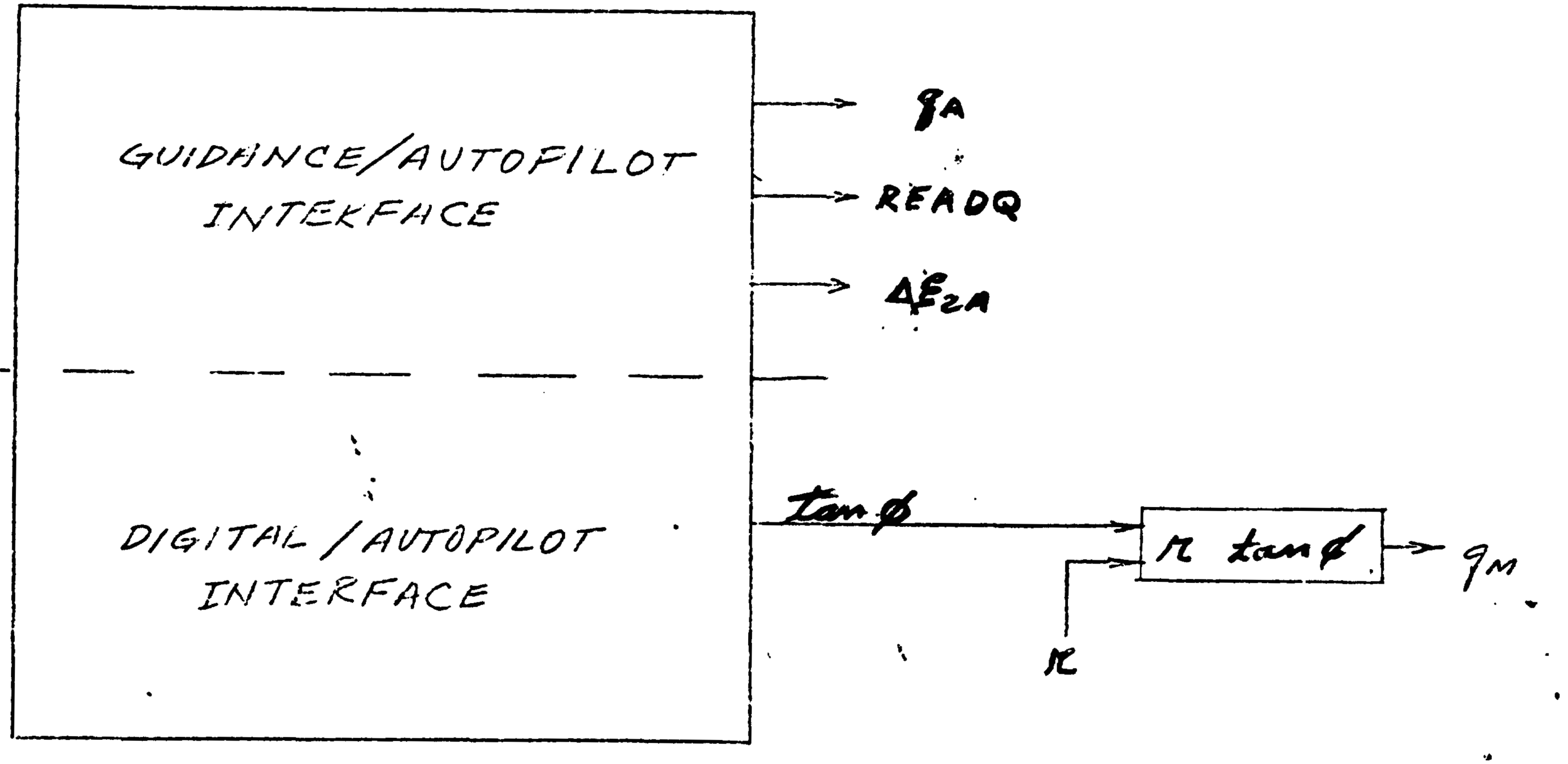


FIGURE 1 (CONTINUED)
PITCH CONTROL SYSTEM

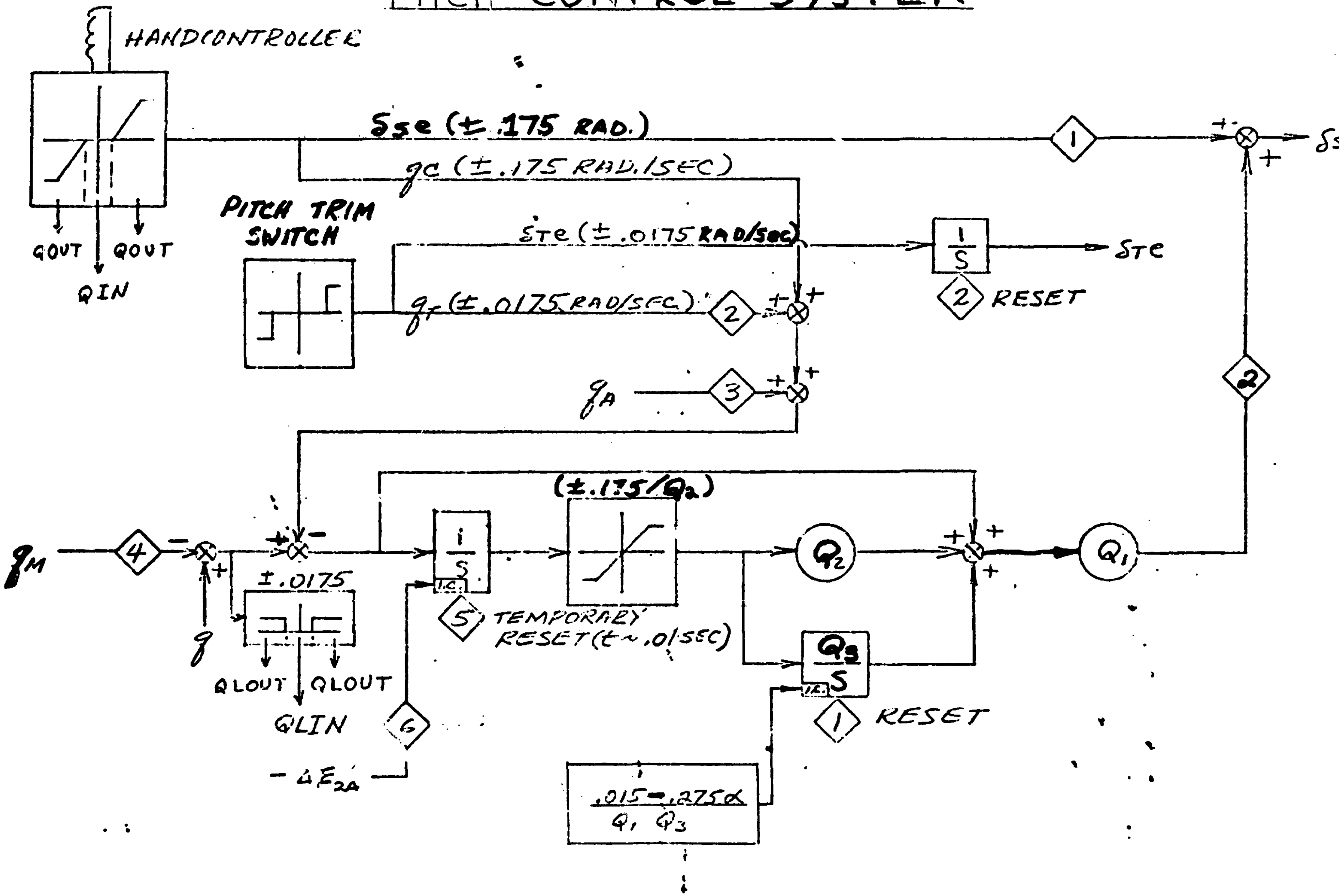


FIGURE 2
ROLL CONTROL SYSTEM

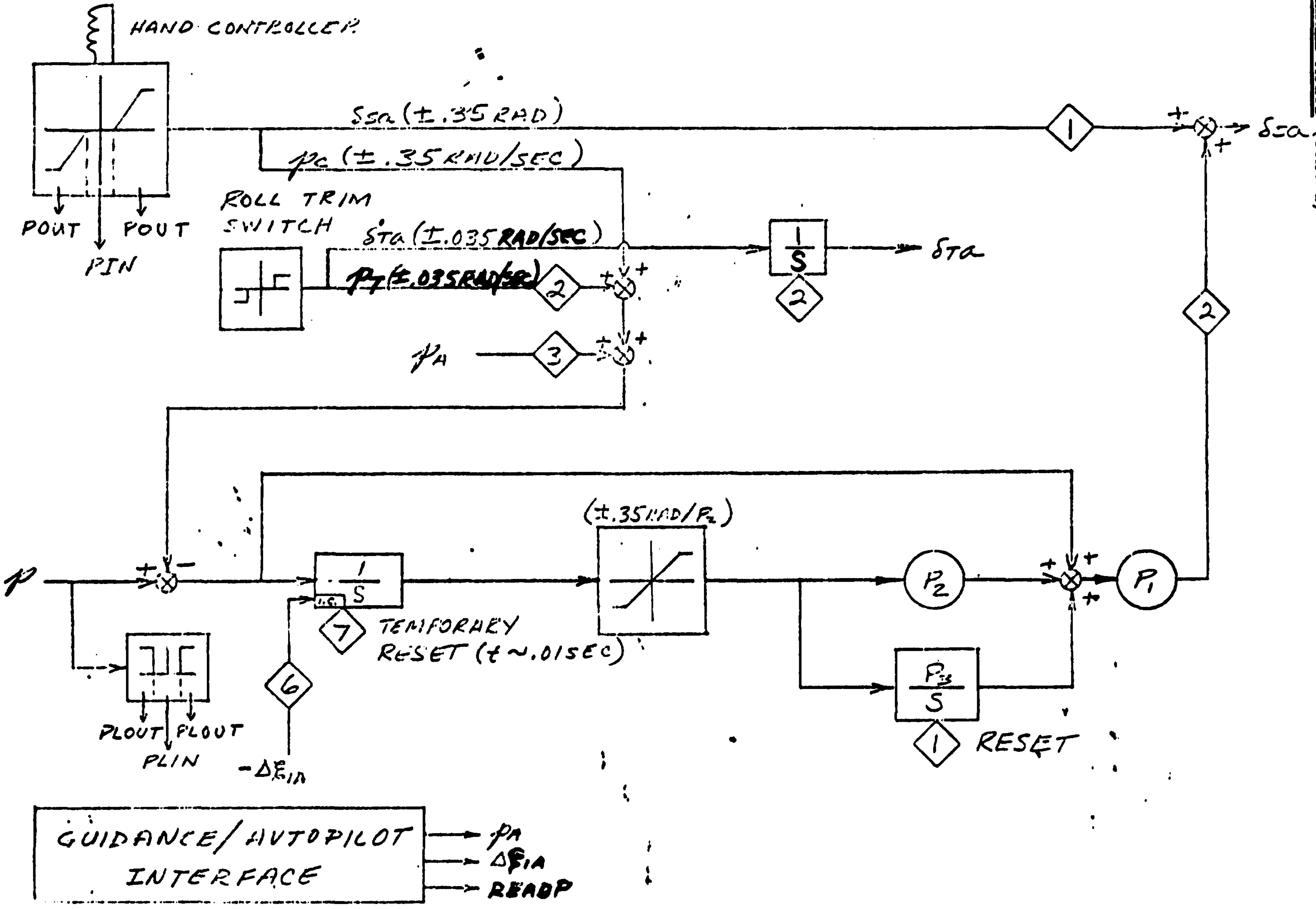


FIGURE 3

YAW CONTROL SYSTEM

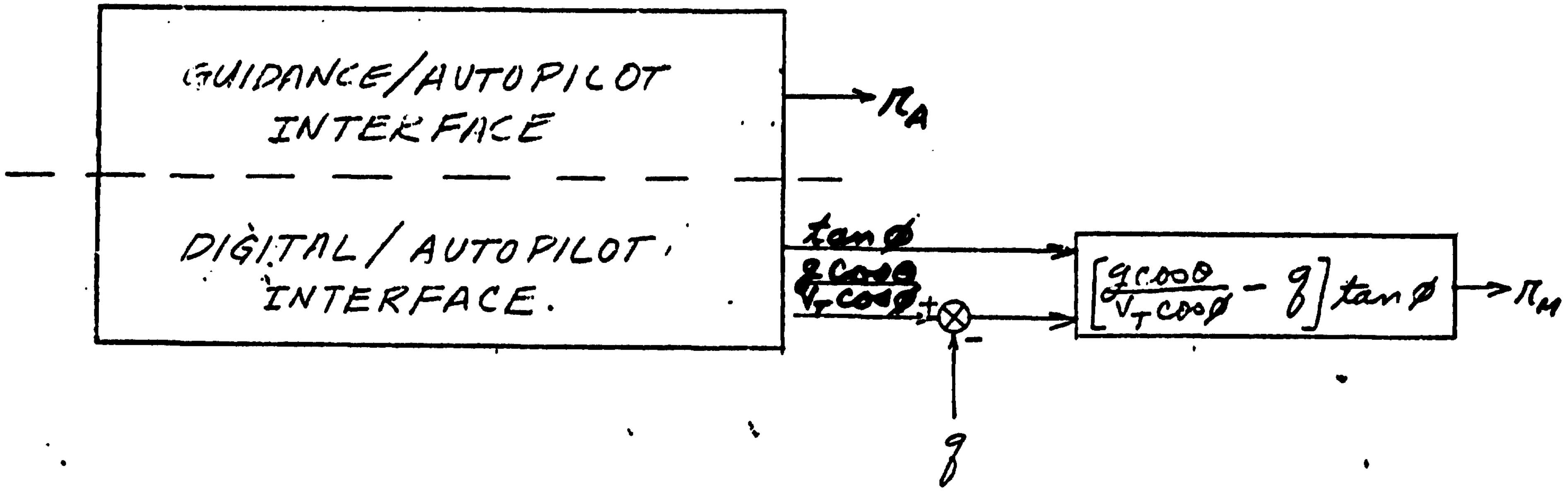

GUIDANCE/AUTOPILOT INTERFACE
r_A
DIGITAL/AUTOPILOT INTERFACE.
$\tan\phi$
$\frac{g\cos\theta}{V_T\cos\phi}$
+
−
g
$\left[\frac{g\cos\theta}{V_T\cos\phi} - g\right]\tan\phi$
r_M

FIGURE 3 (CONTINUED)

YAW CONTROL SYSTEM

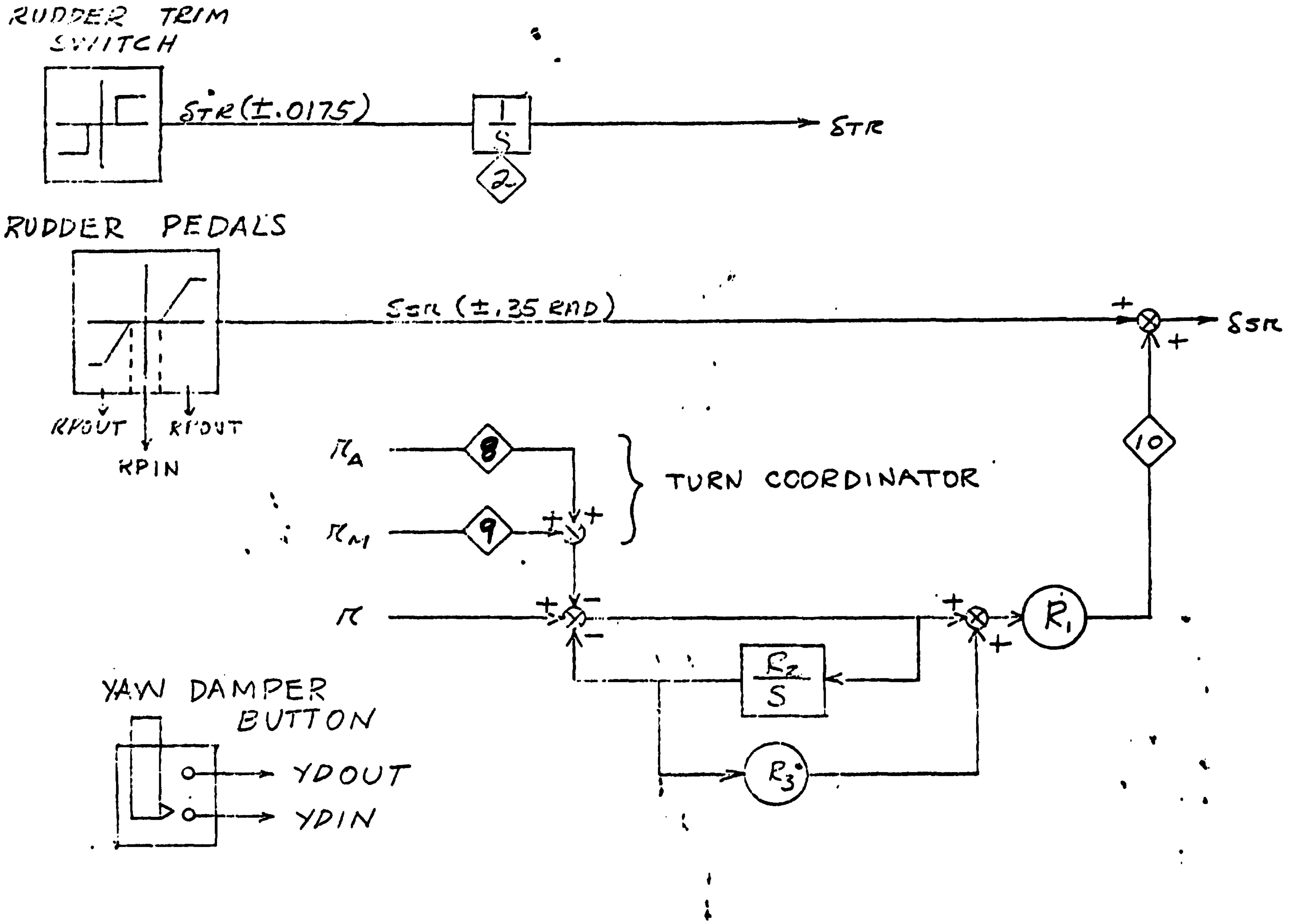


TABLE I

CONTROL SYSTEM GAINS FOR DELTA WING SHUTTLE

ROLL	PITCH	YAW
$P_1 = 1.0$	$Q_1 = 3.2$	$R_1 = 1.2$
$P_2 = 1.61$	$Q_2 = 0.9$	$R_2 = 0.3$
$P_3 = 0.016$	$Q_3 = 0.08$	$R_3 = 0.0$

TABLE 2
LOGICAL SWITCH CONTROL

SWITCH NUMBER	LOGIC CONDITIONS FOR CLOSING SWITCHES
1	DM
2	RC + RCAH + AUTO
3	QIN · AUTO
4	RCAH + QOUT· AUTO
5	QIN·(RCAH· QLIN+AUTO · READQ)
6	AUTO
7	PIN·(RCAH·PLIN + AUTO·READP)
8	YDOUT• RPIN •AUTO
9	YDOUT·RPIN ·(RC+ RCAH)
10	DM•YDIN + RC+RCAH +AUTO

NOTE: LOGIC SWITCHES ASSOCIATED WITH INTEGRATOR MODE CONTROL DENOTE RESET OR TEMPORARY RESET CONDITIONS WHEN SWITCH IS CLOSED.

9.14.3.2 Cruise DFCS

SPACE SHUTTLE

GN&C SOFTWARE EQUATION SUBMITTAL

Software Equation Section Cruise Autopilot Submittal No. 40

Function: Aerodynamic Cruise Digital Flight Control System (DFCS)

Module No. OC5 / OC6 Function No. 1,7 / 3 (MSC 03690 Rev. B)

Submitted by: F. Elam Co. EG2

Date:

NASA Contact: W. H. Peters (Name) Organization EG2

Approved by Panel III K. T. Cox (Chairman) Date 10/21/71

Summary Description: Orbiter DFCS for horizontal aerodynamic flight phase of mission including horizontal take-off, post-entry cruise, approach and landing. The DFCS receives commands from the Guidance modules or pilots and sends commands to the aerodynamic control-surface actuators.

Shuttle Configuration: (Vehicle, Aero Data, Sensor, Et Cetera)

Delta-wing Orbiter; NR161C aero or MSC 040A aero.

DFCS Version 4D.

Comments:

(Design Status)

(Verification Status) Version 4B verified at MIT and at MSC (CRALS) with NR161C aero data.

Panel Comments:

9.14.3.2 Digital Flight Control System

INTRODUCTION

The principal objective of the DFCS, horizontal aerodynamic flight phase, is to provide rotational control and speed brake contol of the Orbiter during those phases of the mission where the Orbiter flight resembles conventional aircraft. These mission phases are the post entry aerodynamic flight, cruise, and landing, and also the horizontal takeoff (without Booster). The aerodynamic DFCS contains an auto mode, direct manual, stability augmentation system (SAS) manual, and a rate command attitude hold RCAH mode.

The horizontal aerodynamic flight phase of the DFCS is in a state of growth itself, even without regard to the incorporation of other mission phases. Modifications presently planned include additional tasks, such as anti-skid, brake control, nose-wheel steering, landing gear extend, throttle, a DFCS self-contained terminal guidance system, and flexible body and fuel slosh control.

This version of the DFCS provides the embrionic structure for the ultimate unified DFCS (see MIT Reports below) which will be used for both rotational and translational control of the Space Shuttle Orbiter and Orbiter/Booster combined in all phases of flight, from launch ascent through orbit to entry and touchdown. The DFCS will evolve into the unified DFCS by sequential incorporation of the presently separate DFCS's for the other mission phases. Thus, the present DFCS contains routines labeled such as:

Transition Mode

Entry Mode

Orbit TVC Mode

Orbit RCS Mode

Insertion TVC

Booster TVC

and such routines, while presently empty, will contain those functions in the unified version.

The DFCS provides a versatile autopilot structure while maintaining simplified communications with other programs, with sensors, and with control effectors by the use of an executive-and-subroutine format. (However, the present version of the DFCS temporarily has demoted these sub-routines, for convenience, to mere routines.) The DFCS reads all external variables (commands, sensors, and supplied data) at a single time point, copying them into dedicated storage,and controls its major support subroutines (presently routines) to be synchronous with the autopilot cycle. As a result, the autopilot program is largely independent of other programs in the guidance computer and is equally insensitive to the characteristics of the processor configuration (dedicated guidance computer versus shared multi-processor).

The sequence of autopilot functional computations is arranged to minimize transportation lag, the time interval between receiving a measurement and effecting a control force. While this lag may be largely due to equipment external to the guidance computer, the time required for control computation can be significant. As a consequence, the filter computations are separated into two sections, one being performed between the "read" and "write," that is, just prior to the control equations. The second filter section, including filter "push down," is performed after writing commands to the control surfaces, and therefore does not contribute to the transport lag.

The DFCS is written in a basic Fortran version (IV) without using features peculiar to a particular computer installation in order to minimize conversion problems by other investigators. Flight versions or special test beds may require reprogramming into machine language to minimize computation time, at the expense,of course, of engineering readability.

Documentation

The baseline equations for the DFCS for the horizontal aerodynamic flight phases of the Orbiter are presented herein as a Fortran listing, version 4D. The listing contains a definition of the alphanumerics, comment cards, and a compilation cross reference printout. An auto-flow chart is also presented, which was made from the exact listing shown. The listing and auto-flow chart constitute version 4D of the DFCS. Other explanatory diagrams herein, the six-page block diagrams of the control modes (version 4B), and the DFCS interface diagram (version 4B), are intended only to provide understanding for the more official listing (version 4D).

MIT Reports

The DFCS was developed by the Draper Laboratory of MIT for MSC. The MIT project leader is Dr. Robert F. Stengel. Additional information describing the DFCS can be found in:

a. Space Shuttle GN&C Equation Document No. 8-71 (Draper Laboratory), "Unified Digital Autopilot With Specific Reference to the Transition Phase," by R. F. Stengel, March 1971. (Sec. 9.13.1 of this Report.)

b. SSV Memorandum No. 71-23C-4 (Draper Laboratory), "Cruise Phase Autopilots for the Straight Wing Orbiter Vehicle," by A. Penchuk and R. W. Schlundt, May 1971.

The first of these reports defines the intended approach for combining all the DFCS from the several mission phases into a unified DFCS, and is written in the context of modern control theory. The second report (memorandum) describes the servo design procedure and results whereby the analog/root-locus selection of servo loops feedback signals, gains, and compensation were selected. The conversion from analog servo-loops to digital servo-loops using the difference equation approach is also detailed. The control equations for the delta wing orbiter in the present DFCS are based on this report, notwithstanding the nomenclature "straight wing" in the report title.

The CRuise And Landing Aerodynamic Simulator (CRALS) at MSC presently contains the DFCS (version 4B) with minor modifications as required for the machine dependent interface signals. Copies of this variation can be obtained by request, both in list form, and as card decks.

DFCS Interfaces

Figure 1 (which is largely self-explanatory) shows the interface signals between the DFCS and other units. The signals actively used in the present version are shown as solid lines. The dotted lines represent signals contained in the read and write statements of version 4D but which are not currently used due to zero gains loaded in their signal flow paths.

Investigators who intend to use the DFCS need not provide the "dotted line" group of interface signals, but if these interface signals are not provided, they should also be omitted from the read statements to prevent program aborts.

The guidance system is shown in Figure 1 as two separate systems; one for commands to the DFCS, and the other for computed information utilized by the DFCS. The computed information source need not be the guidance computer, but could be other sources such as the air data computer, or navigation system, or sensors. Of the latter set, speed, angle of attack, and altitude enable the DFCS to update control gains and filter constants. Speed and "range" (range to go until touchdown) enable the DFCS to control the speed brake during landing approach.

One recurring interface problem traditionally is the arbitrary definitions for control surface position. Within the DFCS and at the interfaces, surface positions of elevator, aileron, and rudder are defined such as to cause positive pitch, roll, and yaw moments. This is opposite to some commonly used sign conventions for elevator and aileron. Elevon position polarity is positive for trailing edge upward. A right rudder pedal push will cause a positive rudder by DFCS definition, which is the same as pilot language for right rudder. The elevators and rudder follow the left hand rule, for thumb of left hand in direction of plus y or plus z direction. The purpose of this selection of signs was to simplify the verification of proper signs since positive gains result for the airplane transfer function. A more compelling reason for the sign selection was that in switching from RCAH to Direct Mode, stick back (using standard signs) calls for plus pitch rate in RCAH but negative elevator position in Direct Mode. Using standard signs, numerous sign changes would confusingly appear and disappear with mode switching. The speed brake angle magnitude is defined in the DFCS as one-half of the included angle between the two surfaces.

Interface signals are of two types, discretes and proportional variables. Figure 1 identifies the discretes and the values each can have. All others are proportional variables. The maximum and minimum permissible values of the interface signals have not been defined.

The "restart" discrete is considered to be generated external to the DFCS. After re-initializing, the DFCS will reset the "restart" signal to zero, which informs all other programs that the DFCS information is trustworthy once more.

The DFCS computes TNEXT, and informs the main computer when the next DFCS cycle is to be called.

Control Modes

The DFCS (horizontal aerodynamic flight phase) provides control at this stage of development for the elevons, rudder, speed brake, and landing gear extend. The major modes are (1) Manual, and (2) Automatic. The Manual mode contains three submodes: (1) Direct, (2) Stability augmentation system (SAS) - a form of rate command, and (3) Rate Command Attitude Hold (RCAH). The SAS mode has been retained because of historical pilot familiarization with it and for purposes of evaluating comparative flight handling qualities.

The manual modes receive commands from the pilot's hand controller and rudder pedal. In the Direct manual mode, the hand controller signals are considered elevator and aileron position commands. In SAS and RCAH manual modes, these same pilot's hand controller signals are considered as attitude rate commands, pitch rate and roll rate. The rudder pedals command rudder position in direct mode, and sideslip in RCAH.

The Automatic mode receives Commands only from the guidance system. Provision is included for many other guidance command signals which may be utilized in later versions if required by guidance. Version 4D "reads" all the guidance commands, but the DFCS internal gains are set at zero for those signals not currently used.

9.14.3.2 DFCS (cont'd)

A yaw attitide or rate command to the DFCS is not used (at present) because it would contradict the sideslip feedback for coordinated turns. Later versions may use yaw commands to avoid crabbing during gusts at landing.

A control option, which may be termed a mode, is "automatic" speed brake control and the elevator altitude trim control available only in the RCAH mode. Later versions will provide the auto speed brake option in all manual modes. It is selectable from the pilot control panel using FLAG 5. A trajectory profile of speed versus distance to touchdown and altitude versus distance, stored in core memory, is used together with externally provided information on speed, altitude, and range-to-go, to compute the speed brake commands and elevator trim commands.

More details on this mode are contained below in the section entitled "Fortran Listing" near the end of that section (see the heading "Landing Approach Trajectory for Manual Modes").

Control Equations

Sheets 1 - 6 contain block diagrams, in servo-loop format, of the digitized equations for the Auto mode, and the RCAH mode, for each of three channels: elevator, aileron, and rudder. Each calculation block is a separate Fortran card, except where several blocks have been enclosed in a larger, dotted-line block on the diagram, which indicates one card.

The Z^{-1} calculation blocks are achieved in the DFCS programming by the relative time sequence of cards, which permits the DFCS cycle time lag to occur at the point so designated. No Fortran cards appear for the Z^{-1} blocks.

The filters are also shown in Sheets 1 - 6, although located in the DFCS under headings called "Filter-1" and "Filter-2," instead of in the "control law" routine. The filters are generalized second order digital filters, but at present, most of the filter gains are zero which results in first-order filters.

The filter network following the summing junction of command and feedback is an integrator, and a first order lead. This function is called (in alternative language) a "proportional plus integral" function.

Washout filters are so labeled on the diagram and correspond to the analog TS/TS+1, which is high pass filter with unity gain at high frequencies. The washout filter is used now only in the yaw rate feedback path for rudder control. It permits the yaw rate feedback to damp the Dutch Roll mode, but not to oppose a steady yaw rate turn. A pitch rate washout filter may be used in later versions to keep the vehicle nose up during banking maneuvers.

RCAH/Elevator Channel (see Sheet No. 4).- The presently active part of this mode/channel has:

a. Pilot's hand controller longitudinal command of pitch rate.

b. Pilot's station trim beeper switch is a discrete integrated within the DFCS. This trims the pitch rate, thus offsetting null biases in the hand controller, rate gyro, or any other source. In direct mode, the trim button provides attitude trim.

Pitch Rate Gyro.- This signal is differenced with the hand controller command and the trim command to provide an error signal. The pitch rate gyro signal is also "gained" and feedback to an inner loop to provide servo damping.

The "attitude hold" aspect of the RCAH mode results only from the fact that with a zero rate commanded, the attitude command is "held." No switching-in of an attitude gyro occurs at stick detent to ensure attitude hold. This has the advantage of permitting the RCAH mode to function even though all attitude references have been lost. Any drifting of attitude due to gusts can be corrected by pilot input.

Additional integration in the servo loops are being considered to improve the attitude hold features of RCAH.

The normal acceleration feedback is not currently used and there is a zero gain in the signal path. This feedback will be used in later versions to provide so-called C-star control during landing where the pilot's controller commands a combination of pitch rate and vertical acceleration.

The normal accelerometer is not used in the Automatic mode.

The elevator command produced by the RCAH longitudinal channel is summed or differenced with the aileron command from the RCAH lateral directional channel to result in individual elevon commands.

The conversion of elevator and aileron signal to elevon signal is accomplished internal to the DFCS to permit usage of non-linear effectiveness gains of the elevons in future versions.

In order to avoid stair-step motion of the control surfaces at the DFCS cycle period, several fixes have been proposed which utilize either local filtering at the actuator or rate commands without position commands or some combination.

Prefiltering done locally at the rate gyro has been proposed to filter body bending without increasing the DFCS sample frequency.

During bank angles (i.e., during turns), the pitch rate gyro picks up body pitch rate and the RCAH gives a nose down output. To correct this, a washout filter in the pitch rate gyro path can be employed, or a more direct cross feed from the lateral RCAH could be added. The DFCS present listing contains the pitch rate washout filter in anticipation of this purpose, but the signal is temporarily deadended, as shown in Sheet no. 4.

RCAH/Aileron Channel (Sheet 5) and RCAH/Rudder Channel (Sheet 6).-

Together, the rudder and aileron channels constitute the lateral-directional RCAH control. The rudder channel is considered an inner loop for the aileron channel. The loop closure sequence is:

a. Yaw rate gyro feedback to rudder, with gain and washout filter.

b. Sideslip feedback to rudder with gain. The "filter" at present is inactive as such.

c. Roll rate feedback with gain (aileron channel) for damping.

d. Roll rate feedback (aileron channel) differenced with pilot's hand controller to produce the RCAH roll rate error.

e. Two feedbacks shown are presently inactive. Future versions may use yaw rate feedback for cross feed into aileron, and roll rate cross feed into rudder.

The forward path, from roll rate command error to aileron, contains a gain, an integrator, and a first order lead.

As discussed above, the pilot's hand controller provides roll rate commands, and the rudder pedals provide sideslip command.

Manual trim is provided separately for roll rate and sideslip in the RCAH mode.

Future versions may substitute lateral acceleration for the sideslip feedback, due to possible difficulties with sideslip sensors, with corresponding changes in compensation.

Auto Mode (sheets 1, 2, and 3).- Less detailed discussion of the Auto mode will be required since it is essentially a position loop closure around the RCAH mode.

Comparisons of the Auto mode and RCAH mode control equations can be made by viewing the corresponding diagrams together; i.e., Sheets 1 and 4 for elevator, Sheets 2 and 5 for aileron, and Sheets 3 and 6 for rudder.

Auto/Elevator vs. RCAH/Elevator (Sheets 1 and 4).- The Auto command is pitch attitude from the guidance system. The forward path contains a gain, integration, and first order lead, all of which operate on the pitch error signal. Damping by the pitch rate gyro is provided. Inactive in the present version are the guidance commands and feedback for angle-of-attack, and guidance command for pitch rate.

Manual trim is not provided in the auto mode since the function is accomplished internal to the guidance system.

Auto/Aileron vs. RCAH/Aileron (Sheets 2 and 5).- Similar remarks to those contained in the preceding paragraph apply to the aileron channel (with an appropriate change in axis nomenclature). The guidance system commands for roll rate, yaw rate, and yaw attitude are currently inactive in the auto mode. The yaw rate gyro cross feed into the aileron channel is also currently inactive.

Auto/Rudder vs. RCAH/Rudder (Sheets 3 and 6).- Presently inactive for the Auto/rudder channel are the guidance commands for yaw, sideslip, yaw rate, and roll rate. The following cross-feeds are also inactive: Roll rate gyro and yaw attitude gyro feedback.

DFCS Listing. - The sections of the program in sequence are:

1 Entry point (for all cycles).

103 "Pad Load"Initialization.

If flagword ITURN = 1, this indicates an initial turn on Constants will be loaded from some external source by "read" statements.

104 Begin standard cycle.

External flagwords are read which indicate restart, modes, gain update option, and automatic speedbrake.

Read clock and compute next DFCS cycle start time.

24 Logic test to determine if DFCS needs re-initialization.

If there has been indication of a restart, or a mode change, the DFCS must be re-initialized.

9

20 Sequence and I/O Initialization.

Reset the restart flags, change to the new modes, and set an internal flag (ISTART) to indicate that this DFCS pass is an initialization cycle. At the end of the entire DFCS cycle, this flag will be set back to normal.

33 Establish Sampling Rates

The DFCS has three different cycles. The control laws and filters are performed on each fast cycle, now set at .10 seconds. The speed brake commands are calculated on each medium cycle. The gains are updated on each slow cycle. A table look-up sets the three cycle times as functions of the DFCS mode and mission phase.

The half-time TF2 for a fast cycle is used in the digital filter gains.

The ratios of the three cycle periods are computed so that countdowns can be used to determine which fast pass coincides with a medium or slow pass. The Modulo function is used for the counting at the very end of the program. The initial count for the medium and slow passes are offset by one cycle to avoid having a medium and slow pass occur simultaneously.

2001
2002 For the initial pass, the gains are set equal to the basic gains in table KFIX, and filter constants are set to the basic GFIX constants. Subsequent passes will update the gains and constants as functions of speed, altitude, etc., which are stored in other tables.

99 Initialize indices.

These are the indices for the Modulo functions which count passes to identify medium and slow cycles.

52 Branch to read, based on mode.

Each mode will result in reading a different set of commands, sensors, and other data.

All "reads" are located below in a read subroutine at statement 300. After reading, the program returns to the executive section for the next branching test.

22 Branch to filter and parameter initialization.

A logic test is made to determine if initialization should be bypassed. Otherwise, a branching is made on mode, initialization is made, and control is passed to the next branching.

Whenever a branch test is made on mode, and the result is Manual mode, a second branch test is made on Manual mode, to determine which manual submode.

Similar branching and returning to executive control is made for:

56 Branch to state filter - part 1 (filter update).

57 Branch to control law.

59 Branch to state filters - part 2 (filter pushdown).

60 Branch to parameter estimation.

This branch depends on mode, and also if Flag 4 has requested parameter estimation.

The parameter estimation routines are located below at statement 7051. These are filters for external information for angle-of-attack, velocity, air density, and dynamic pressure.

Afterward, table lookup is performed to obtain the update for the control law gains.

80 Branch for closeout.

A different closeout is used for the intial pass.

This completes the branching section in the executive program.

The "sub-routines" follow. In this version, as stated above, the subroutines are routines.

Comments will be made only to supplement reading of the listing.

2000 Initialization Routines.

87 Pullup and flare constants for manual approach.

The strategy is to let the DFCS compute these control system constants to avoid human calculation errors when trajectory changes are made.

400 Filter Routines - part 1.

All possible filter calculations are done in Filters part 2 fcllowing the control law calculations to avoid transport lag.

Filters part 1 is performed immediately after "read" and prior to "control laws."

The filter equations and identification of constants by symbol can be seen on Sheets 1 to 6 of the block diagrams.

5000 Control Routines.

The control laws for Auto mode and RCAH are shown in tne block diagrams Sheets 1 to 6.

511 Multi-Rate Speed Brake Control.- An automatic speed brake control mode is available (only in RCAH) to assist in landing approach. A reference trajectory is :tored in the DFCS based on altitude and velocity versus range to go to touchdown. Error signals of velocity and altitude are simply "gained" (without integration or compensation) and added to the pilot's manual speed brake command. The altitude gain is set at zero. Additional information on this subject is below in the routine "Landing Approach Trajectory for Manual Modes" at statement 702.

512
514 Elevator Trim for Landing Approach Trajectory.- DECH and DECU are the delta elevator commands due to altitude and velocity, respectively, from the "Landing Approach Trajectory," for RCAH only. The altitude and velocity errors are "gained" (without integration or compensation) and added to the otherwise total RCAH elevator command. Tht velocity gain is zero at present. The altitude error produces a low gain, slow loop elevator trim which tends to keep the vehicle on the glide slope. The loop is so slow that pilot commands to the RCAH are much faster. Hence, the pilot and not this automatic feature produce the pullup maneuver and landing flare. More information on this subject is below in the routine "Landing Approach Trajectory for Manual Modes" at statement 305.

6000 Filter Routines - Part 2.- This routine containr all the filter calculation that can be done before the "read" on the next cycle. Also included is the "push down" of the digital filter where the quantity at t_n becomes the same quantity at t_{n-1} for the next cycle.

The filter equations for the RCAH and auto modes are shown in the block diagrams, Sheets 1 - 6.

611 Medium Cycle, filter for X-velocity.

612 Slow Cycle, filter for altitudes.

7C2 Landing Approach Trajectory for Manual Modes.- The reference trajectory is used only in RCAH mode, but must be maintained in all modes in case of switchover to RCAH. Two trajectory curves are stored, velocity versus range, and altitude versus range. The velocity trajectory assists in speed brake control of velocity. The altitude trajectory assists in elevator control of altitude.

The range-to-go is broken into four segments. Trajectory information is stored at the endpoints of each segment for velocity, altitude, and altitude slope (slope with respect to range). IR is the index 1 to 4 for each segment of range, with 1 nearest touchdown and 4 at high altitudes.

For altitude, the segments are: Landing flare = 1 (ending in touchdown with a sink rate), straight line glide slope = 2, approach pullup = 3, and steeper glide slope straight line = 4. The glide slope and pullup segments for altitude are algebraic quadratics $(a + bx + cx^2 + dx^3)$ such that the slope and position of the curve coincide with the adjacent straight line segments. Coefficients for these interpolations are computed in the DFCS in the initialization routine "Generate Pullup and Flare Constants for Manual Approach," located near statement 2010.

The velocity segments of the reference landing trajectory are straight lines for the three nearest touchdown. The fourth segment is exponential and results in a linear IAS (indicated airspeed) versus range.

RGO is the range-to-go (to touchdown) and is navigation type data supplied to the DFCS by landing navigation aids or by the guidance system. RLAND (I) is the range-to-go (reference trajectory) at the endpoints of each range segment. RGO segment number IR= 1 is bounded by endpoints I = 1 and I = 2. Segment IR = 2 is bounded by endpoints I = 2 and 3. The program determines IR by comparing RGO - RLAND(I) until it goes negative - then IR = I - 1.

Having found IR, the program proceeds with the interpolation formulas. First, however, each segment of range has its own measure of reference length RL, which,as the listing suggests, RL = RGO - RLAND(IR).

The results of the interpolation equations are UG and HG, a velocity command and altitude command. These commands are used in the control routines (described above), to provide speed brake and elevator commands.

The next version, or soon thereafter, will contain a Mach-trim feature at altitudes above 45,000 ft to provide a steady-state angle-of-attack along a reference trajectory in the RCAH mode.

81 Closeout Routine.- ISTART is the flag which indicates if this pass was an initialization pass for the DFCS. If this was a "first" pass, the executive program will begin the closeout routine at statement 81 and set ISTART = 0.

For the first pass, both the count indices for medium and slow are 0. IS = IS + 1 offsets the slow cycle index from the medium cycle index so that the medium and slow cycles will not occur on the same pass.

82 Increment Indices.- Each fast pass increments the counting indices for the medium and slow cycles.

The Modulo function is a comparison function used in countdowns such that if the index exceeds or equals the second argument, the first argument (index) is set to zero. For example, Modulo 5 would count 1, 2, 3, 4,0 and repeat.

When IM = 0, the DFCS will perform the medium pass operations. in addition to the fast pass operations, and similar results occur for the slow pass when IS = 0.

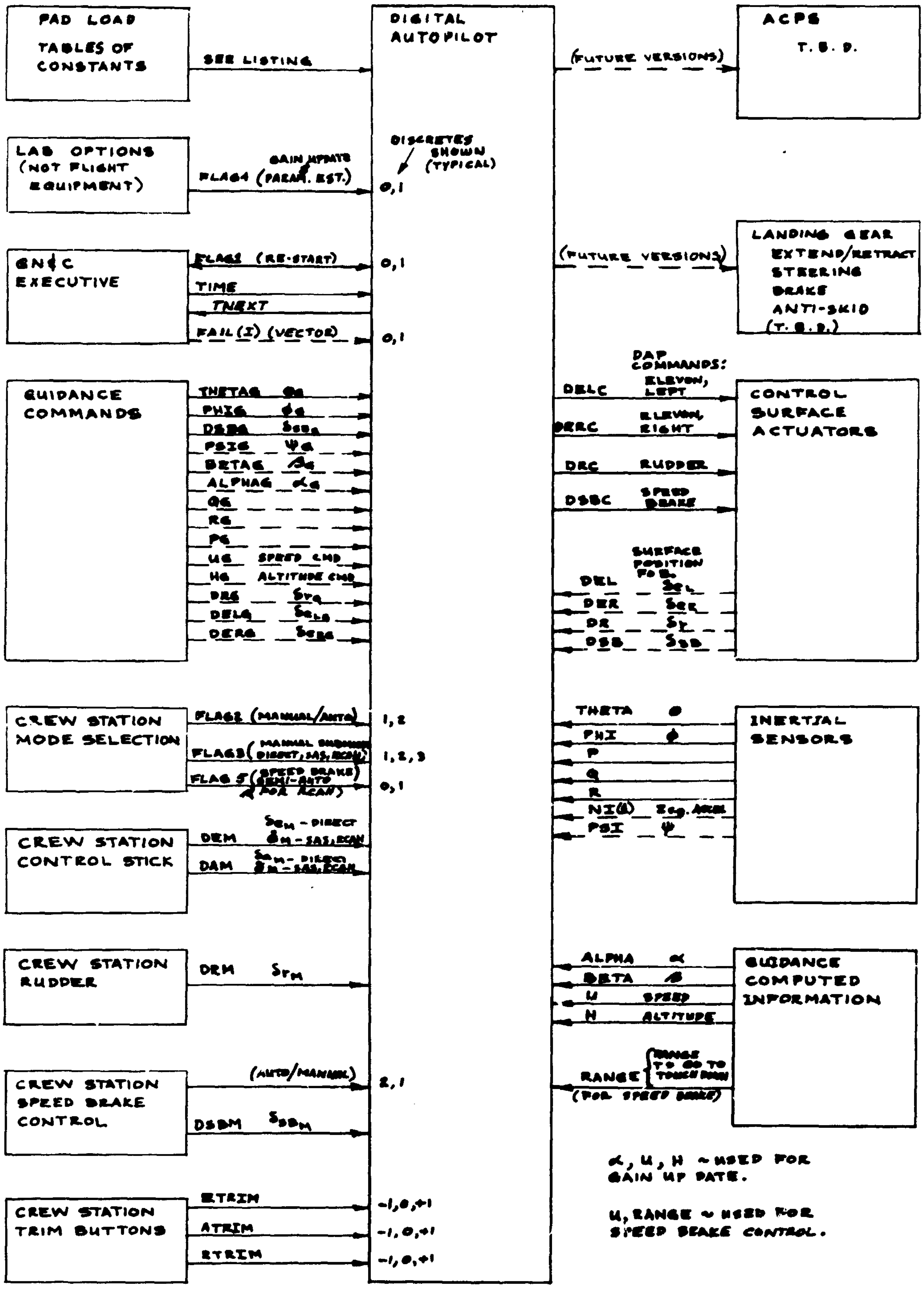
Autoflow Chart

The autoflow chart is a flow diagram made automatically by computer program. Hand notations in ink have been made on the first page of the diagrams to explain the autoflow symbols. The computer mainstream of flow does not always emphasize the same mainstream that the human designer had in mind, so the results are sometimes confusing. However, the autoflow chart is a useful tool.

Pad Load and Constants

The constants have been verified for the NR 161C. Now constants are being determined for the MSC 040A Orbiter. These constants have not been included because they are changing, but are available upon request.

9.14.3.2 DFCS (cont'd)



α, U, H ~ USED FOR GAIN UP DATE.

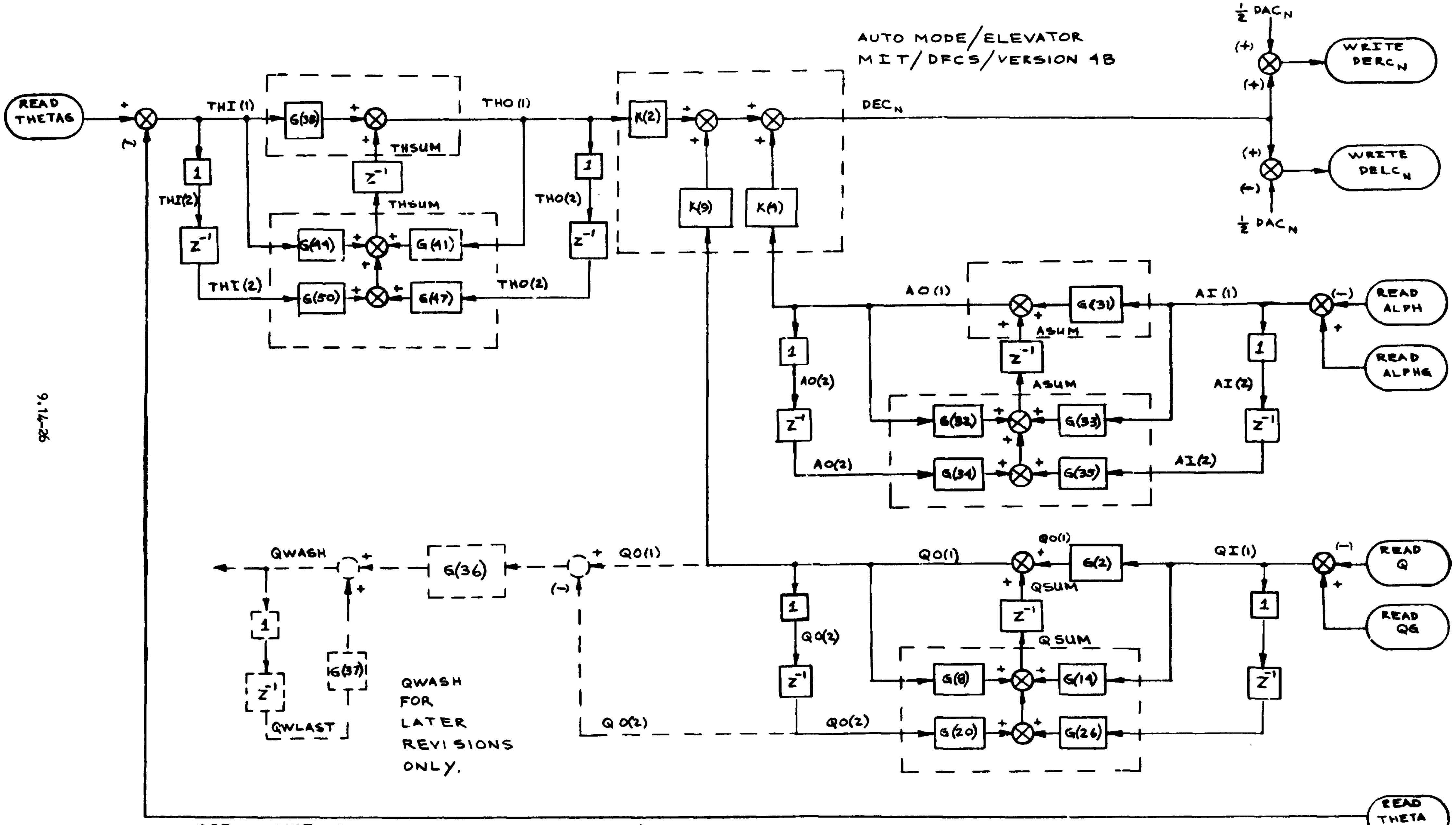
U, RANGE ~ USED FOR SPEED BRAKE CONTROL.

NOTES: 1. DOTTED LINES INDICATE SIGNALS NOT USED IN PRESENT DAP VERSION (#3) - BUT MAY BE IN FUTURE VERSIONS.

FIGURE 1 . INTERFACE SIGNALS TO CRUISE DAP

8-3-71
EG2/F. ELAM

AUTO MODE/ELEVATOR
MIT/DFCS/VERSION 4B



QWASH
FOR
LATER
REVISIONS
ONLY.

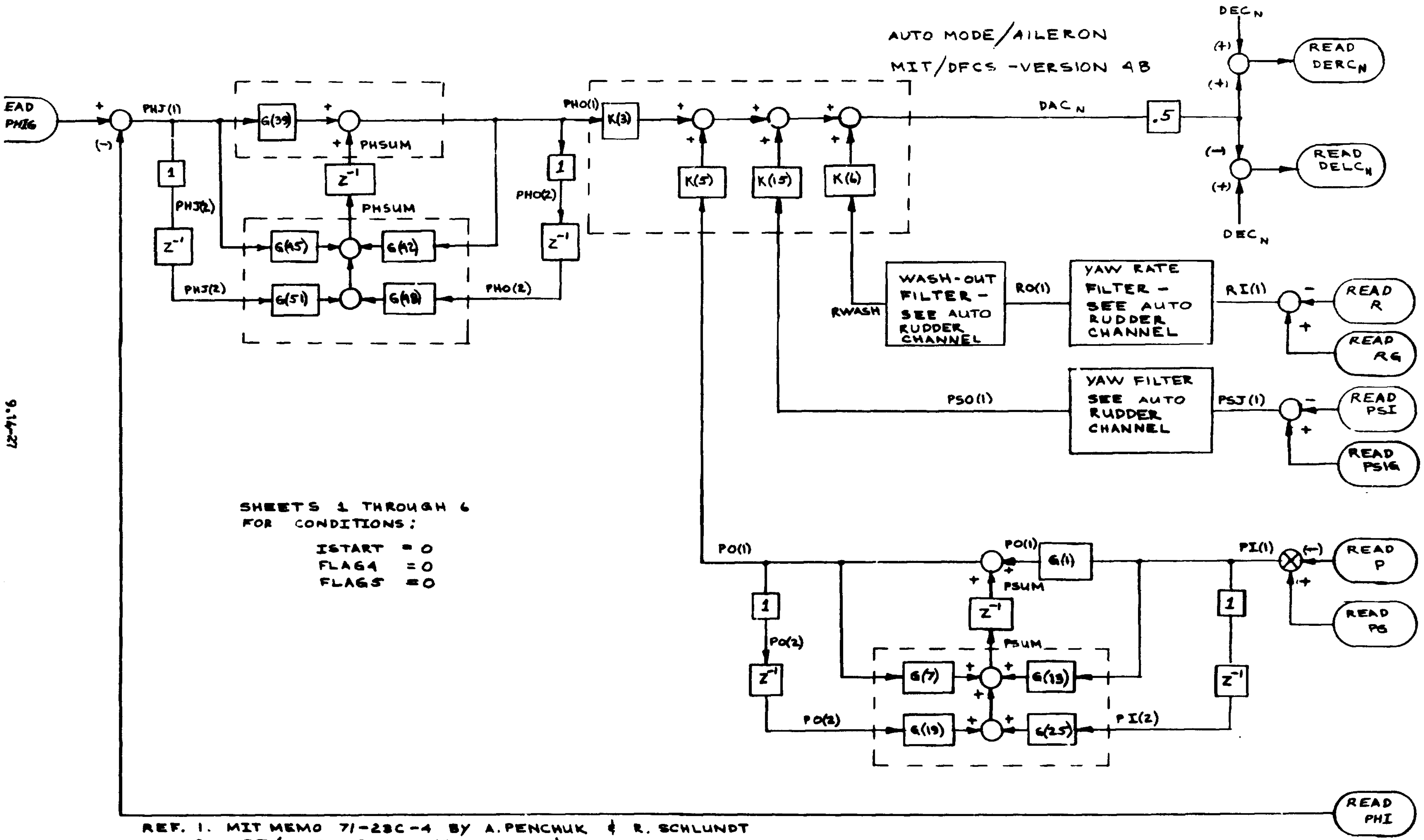
REF. 1. MIT MEMO 71-23C-4 BY A. PENCHUK & R. SCHLUNDT
2. MIT/DFCS VERSION 4B LISTING & AUTOFLOW
BY R. STENGAL

DRAWN BY
MSC/EG2/F. ELAM
10-7-71

SHEET 1 OF 6

AUTO MODE/AILERON

MIT/DFCS -VERSION 4B



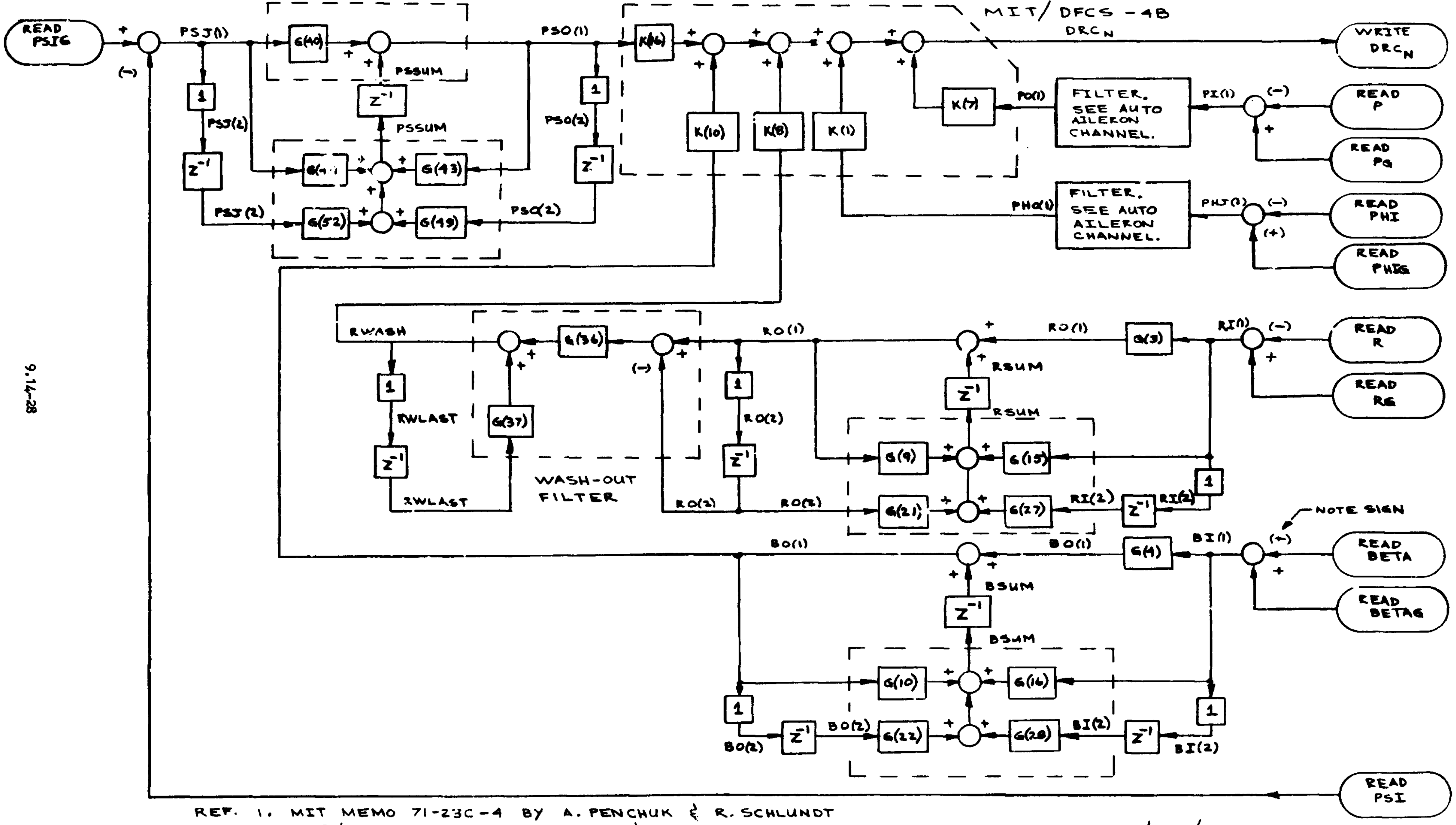
SHEETS 1 THROUGH 6
FOR CONDITIONS:
ISTART = 0
FLAG4 = 0
FLAG5 = 0

REF. 1. MIT MEMO 71-23C-4 BY A.PENCHUK & R. SCHLUNDT
2. MIT/DFCS VERSION 4B LISTING & AUTOFLOW
BY R. STENGAL

MSC/EG2/FRANK ELAM
10-7-71

SHEET 2 OF 6

AUTO MODE/RUDDER
MIT/ DFCS -4B

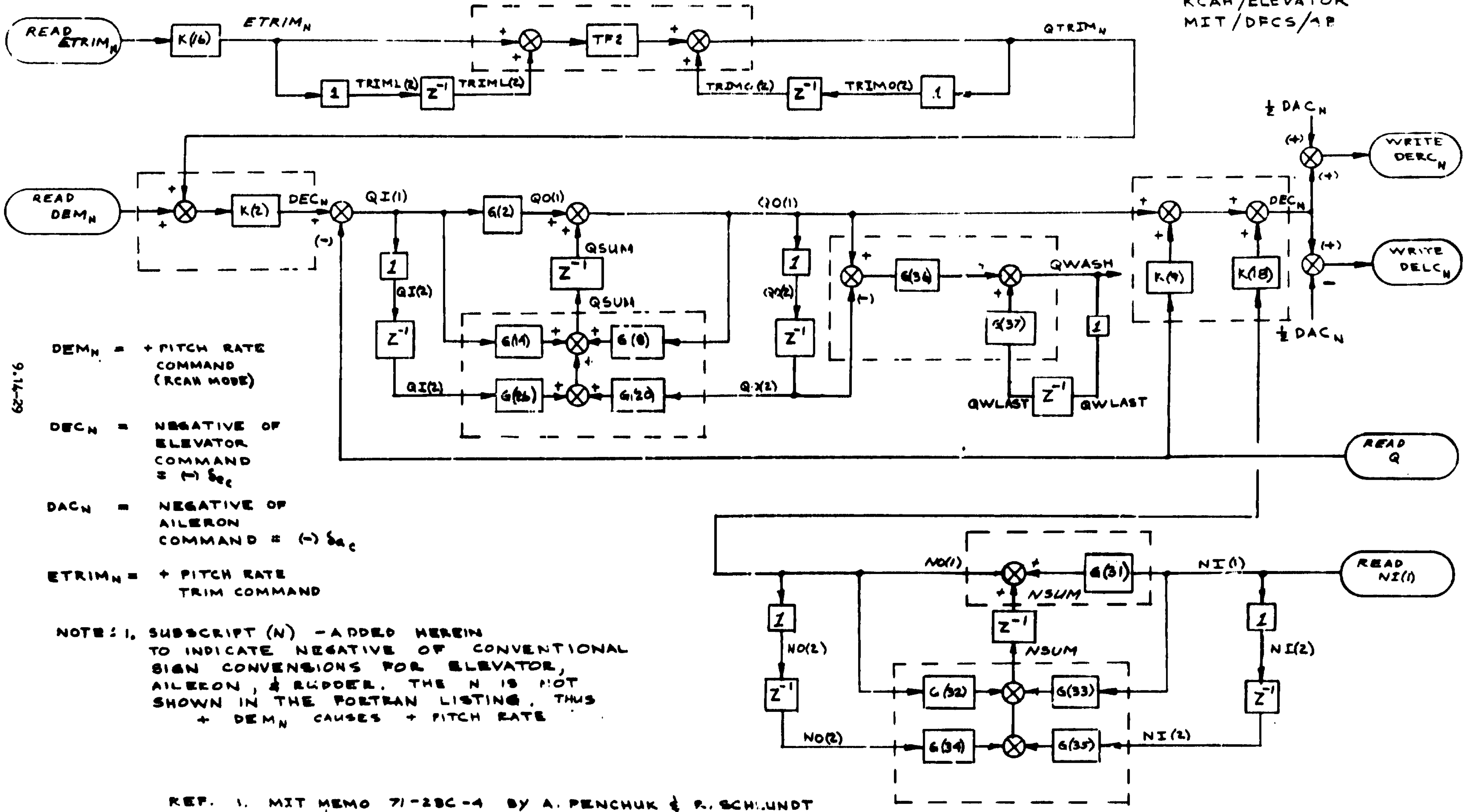


REF. 1. MIT MEMO 71-23C-4 BY A. PENCHUK & R. SCHLUNDT
2. MIT/DFCS VERSION 4B LISTING & AUTOFLOW
BY R. STENGAL

MSC/EG2/FRANK ELAM
10-7-71

SHEET 3 OF 6

RCAH/ELEVATOR
MIT/DFCS/4B



9.14-29

DEM$_N$ = + PITCH RATE COMMAND (RCAH MODE)

DEC$_N$ = NEGATIVE OF ELEVATOR COMMAND = (−) δ_{e_c}

DAC$_N$ = NEGATIVE OF AILERON COMMAND = (−) δ_{a_c}

ETRIM$_N$ = + PITCH RATE TRIM COMMAND

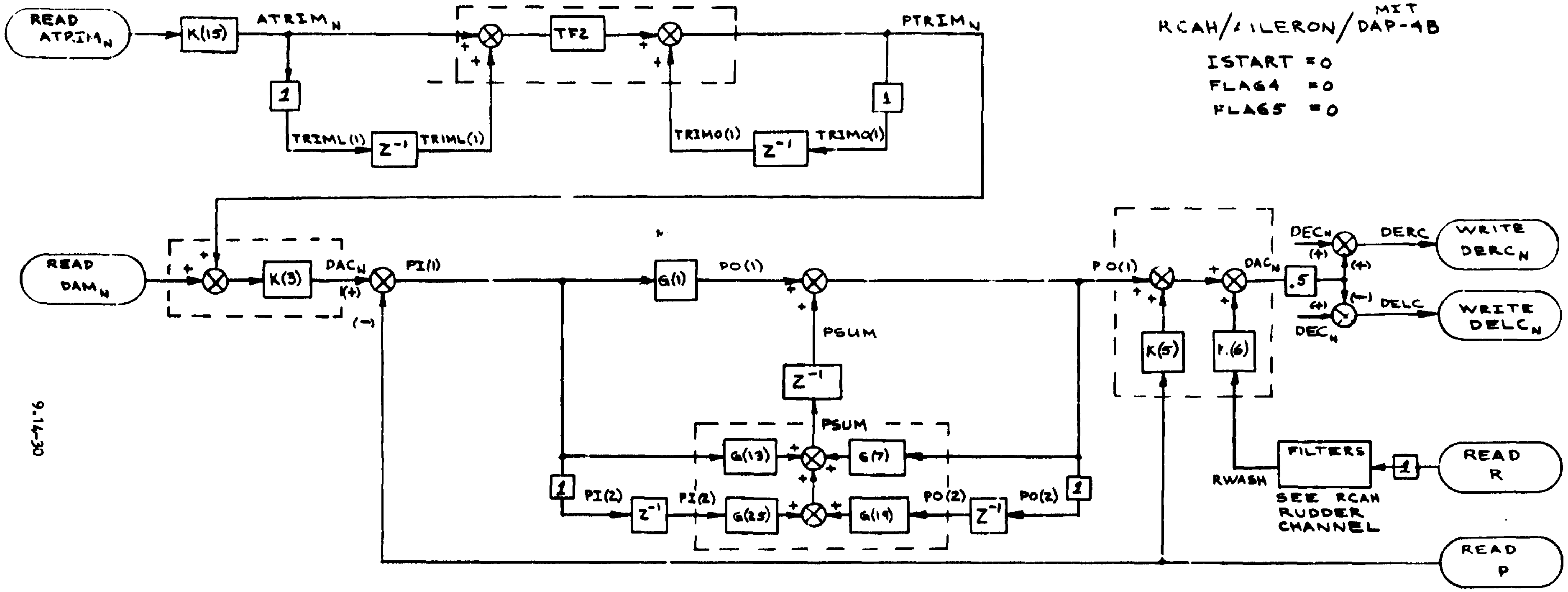
NOTE: 1. SUBSCRIPT (N) - ADDED HEREIN TO INDICATE NEGATIVE OF CONVENTIONAL SIGN CONVENSIONS FOR ELEVATOR, AILERON, & RUDDER. THE N IS NOT SHOWN IN THE FORTRAN LISTING, THUS + DEM$_N$ CAUSES + PITCH RATE

REF. 1. MIT MEMO 71-23C-4 BY A. PENCHUK & R. SCHLUNDT
2. MIT/DFCS VERSION 4B LISTING & AUTOFLOW
BY R. STENGAL

E62/FRANK ELAM
10-7-71
SHEET 4 OF 6

RCAH/AILERON/MIT DAP-4B

ISTART =0
FLAG4 =0
FLAG5 =0



9-14-30

NOTE: THE .5 MULTIPLIER FOR DAC ADDED BY MSC, TO CORRESPOND TO NR DEFINITION OF AILERON, & NR AERO DATA.

$$\delta_a = (\delta_{e_R} - \delta_{e_L})$$

AND

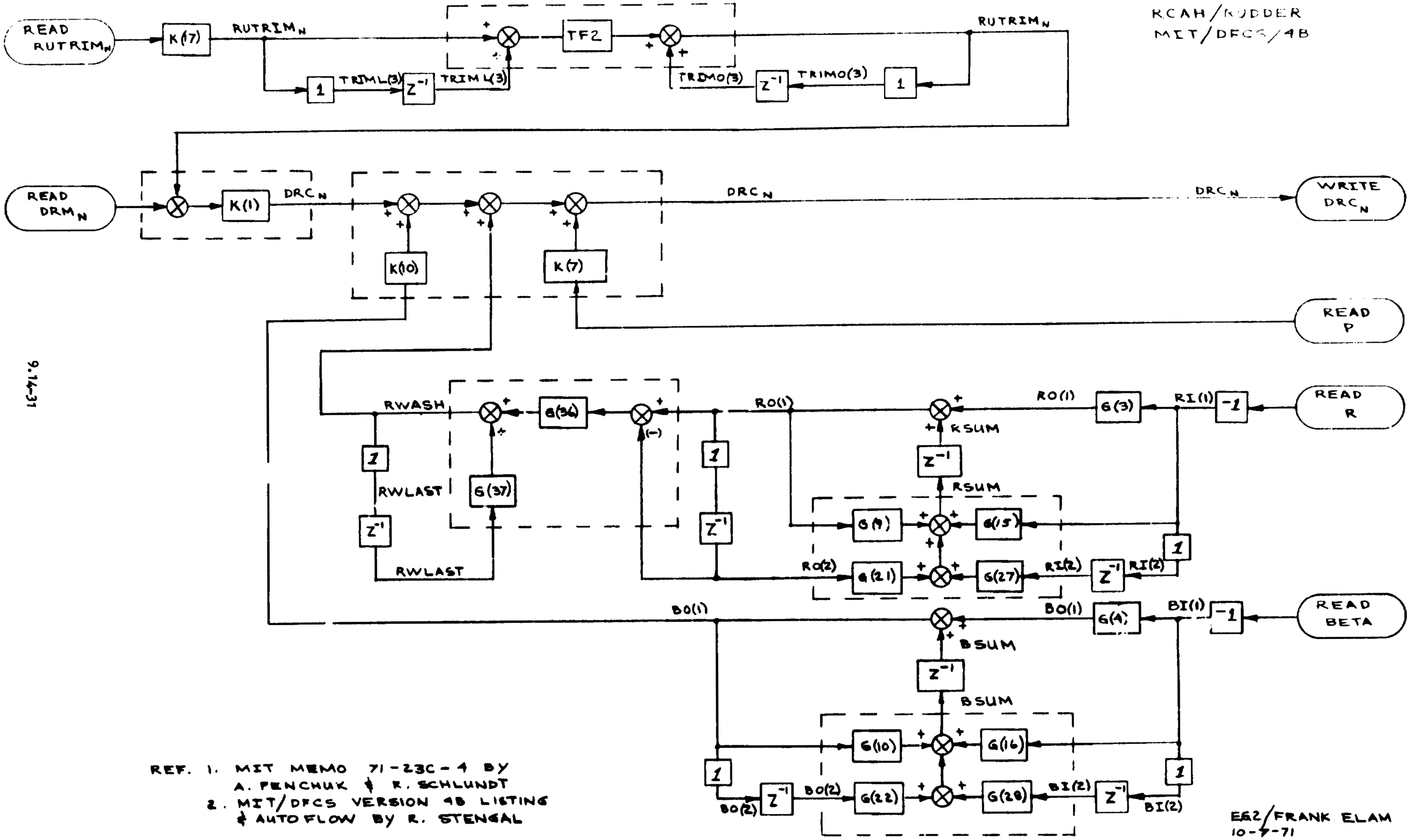
$$DAC_N = (-)\ \delta_{a_c}$$
$$DERC_N = (-)\ \delta_{e_{R_c}}$$
$$DELC_N = (-)\ \delta_{e_{L_c}}$$

REF. 1. MIT MEMO 71-23C-4 BY A. PENCHUK & R. SCHLUNDT
2. MIT/DFCS VERSION 4B LISTING & AUTOFLOW BY R. STENGAL

EG2/F.M. ELAM
10-7-71

SHEET 5 OF 6

RCAH/RUDDER
MIT/DFCS/4B



REF. 1. MIT MEMO 71-23C-4 BY
A. FENCHUK & R. SCHLUNDT
2. MIT/DFCS VERSION 4B LISTING
& AUTOFLOW BY R. STENGAL

EG2/FRANK ELAM
10-7-71
SHEET 6 OF 6

9.14.3.2 DFCS (cont'd)

```
          COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=50,SIZE=0000K,
                             SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,NOID,XREF
      C>>>>>SSV.UNIFIED DIGITAL FLIGHT CONTROL SYSTEM-VERSION 40  10-29-71      00000100
      C      R.F. STENGEL                                         MIT CSDL      00000200
      C      ***************************************************************************00000300
      C                          NOMENCLATURE                                   00000400
      C                          ============                                   00000500
      C      AI(I)              ANGLE OF ATTACK (FILTER INPUT)                  00000600
      C      ALPH                 "    "    "     (MEASUREMENT)                 00000700
      C      ALPHG                "    "    "     (GUIDANCE)                    00000800
      C      AMEAN              MEAN ANGLE OF ATTACK FOR PARAM. EST.            00000900
      C      AO(I)                "    "    "     (FILTER OUTPUT)               00001000
      C      ASUM                 "    "    "     (TEMP)                        00001100
      C      ATRIM              'AILERON' TRIM DISCRETE                         00001200
      C      BETA               SIDESLIP ANGLE (MEASUREMENT)                    00001300
      C      BETAG                "        "    (GUIDANCE)                      00001400
      C      BFIT(I,J)          TABLE OF SIDESLIP GAINS                         00001500
      C      BI(I)                "          "    (FILTER INPUT)                00001600
      C      BO(I)                "          "    (FILTER OUTPUT)               00001700
      C      BSUM                 "          "    (TEMP)                        00001800
      C      DAC                'AILERON' ANGLE COMMAND OUTPUT                  00001900
      C      DACINT             RCAM INTEGRAL IN 'AILERON' ANGLE COMMAND        00002000
      C      DAM                'AILERON' ANGLE CCMMANC (MANUAL)                00002100
      C      DEC                'ELEVATOR' ANGLE COMMAND OUTPUT                 00002200
      C      DECH               'ELEVATOR' COMMAND DUE TO ALTITUCE FEEDBACK     00002300
      C      DECINT             RCAM INTEGRAL IN 'ELEVATOR' ANGLE COMMAND       00002400
      C      DECV               'ELEVATOR' COMMAND DUE TO VELOCITY FEEDBACK     00002500
      C      DEL                LEFT ELEVON ANGLE (MEASUREMENT)                 00002600
      C      DELC                "       "       "    (COMMAND)                 00002700
      C      DELG                "       "       "    (GUIDANCE)                00002800
      C      DEM                'ELEVATOR' ANGLE COMMAND (MANUAL)               00002900
      C      DER                RIGHT ELEVON ANGLE (MEASUREMENT)                00003000
      C      DERC                 "       "       "    (COMMAND)                00003100
      C      DERG                 "       "       "    (GUIDANCE)               00003200
      C      DR                 RUDDER ANGLE (MEASUREMENT)                      00003300
      C      DRC                  "      "   (COMMAND)                          00003400
      C      DRG                  "      "   (GUIDANCE)                         00003500
      C      DRM                RUDDER ANGLE COMMAND (MANUAL)                   00003600
      C      DSB                SPEED BRAKE SETTING (MEASUREMENT)               00003700
      C      DSBC                 "      "      "       (COMMAND)               00003800
      C      DSBG                 "      "      "       (GUIDANCE)              00003900
      C      DSBM               SPFFD BRAKF COMMAND (MANUAL)                    00004000
      C      DYNP               DYNAMIC PRESSURE ESTIMATE                       00004100
      C      ELBIAS             'ELEVATOR' BIAS TO TRIM SPEED BRAKE BIAS        00004200
      C      EL(I)              TABLE OFRELEVATORLBIASESATC TRIM SPEEC BRAKE    00004300
      C      ETRIM              'ELEVATOR' TRIM DISCRETE                        00004400
      C      FAIL(I)            FAILURE CISCRETFS                               00004500
      C      FAST(I)            FAST SAMPLING INTERVAL TABLE                    00004600
```

```
C       FH(I,J)            CONSTANTS FOR FLARE & PULLUP ALTITUDE REFERENCE00004700
C       FIL(I)             FILTER CONSTANTS USED IN PARAMETER ESTIMATION   00004800
C       FLAG1              NEW START DISCRETE (EXTERNAL):0=NORMAL CYCLE    00004900
C                                                    1=NEW START          00005000
C       FLAG2              FLIGHT MODE FLAG (EXTERNAL):1=MANUAL MODE       00005100
C                                                    2=AUTOMATIC MODE     00005200
C                                                    3=TRANSITION MODE    00005300
C                                                    4=ENTRY MODE         00005400
C                                                    5=ORBIT TVC MODE     00005500
C                                                    6=ORBIT RCS MODE     00005600
C                                                    7=INSERTION TVC      00005700
C                                                    8=BOOSTER TVC MODE   00005800
C       FLAG3              MANUAL MODE FLAG (EXTERNAL):1=DIRECT            00005900
[illegible]
C                                                          1=ON           00006300
C       FLAG5              MANUAL MODE AUTO. SPEED BRAKE FLAG(EXTERNAL):   00006400
C                                0=NO AUTO. SPEED BRAKE                    00006500
[illegible]
C       GFIX(I,J)          FILTER GAIN TABLE FOR ALL FLIGHT MODES          00006900
C       H                  ALTITUDE ABOVE RUNWAY (MEASUREMENT)             00007000
C       HG                    "       "      "   (GUIDANCE)                00007100
[illegible]
C       HO(I)              ALTITUDE ABOVE RUNWAY (FILTER OUTPUT)           00007500
C       HSUM                  "       "      "    (TEMP)                   00007600
C       I                  GENERAL PURPOSE INDEX                           00007700
[illegible]
C       ISTART             STARTING CYCLE DISCRETE                         00008100
C       ITM                # OF FAST CYCLES PER MEDIUM CYCLE               00008200
C       ITS                # OF FAST CYCLES PER SLOW CYCLE                 00008300
[illegible]
C       KFIX(I,J)          CONTROL GAIN TABLE FOR ALL FLIGHT MODES         00008700
C       MANMOD             MANUAL MODE FLAG (INTERNAL)                     00008800
C       MEDIUM(I)          MEDIUM SAMPLING INTERVAL TABLE                  00008900
[illegible]
C       NSUM                  "       "        (TEMP)                      00009300
C       P                  ROLL RATE (MEASUREMENT)                         00009400
```

9.14-33

```
C     PCOMP(I,J)          TABLE OF ROLL COMPENSATION GAINS              00009500
C     PFIT(I,J)           TABLE OF ROLL RATE GAINS                      00009600
C     PG                  ROLL RATE (GUIDANCE)                          00009700
C     PHFIT(I,J)          TABLE OF ROLL ATTITUDE GAINS                  00009800
C     PHI                 ROLL ATTITUDE (MEASUREMENT)                   00009900
C     PHIG                 "      "     (GUIDANCE)                      00010000
C     PHJ(I)               "      "     (FILTER INPUT)                  00010100
C     PHO(I)               "      "     (FILTER OUTPUT)                 00010200
C     PHSUM                "      "     (TEMP)                          00010300
C     PI(I)               ROLL RATE (FILTER INPUT)                      00010400
C     PO(I)                "   "   (FILTER OUTPUT)                      00010500
C     PSI                 YAW ATTITUDE (MEASUREMENT)                    00010600
C     PSIG                 "     "     (GUIDANCE)                       00010700
C     PSJ(I)               "     "     (FILTER INPUT)                   00010800
C     PSO(I)               "     "     (FILTER OUTPUT)                  00010900
C     PSSUM                "     "     (TEMP)                           00011000
C     PSUM                ROLL RATE (TEMP)                              00011100
C     PTEMP               TEMPORARY STORAGE IN PARAM. EST.              00011200
C     PTEMP1                 "        "    "    "     "                 00011300
C     PTRIM               ROLL AXIS TRIM COMMAND                        00011400
C     Q                   PITCH RATE (MEASUREMENT)                      00011500
C     QCOMP(I,J)          TABLE OF PITCH COMPENSATION GAINS             00011600
C     QFIT(I,J)           TABLE OF PITCH RATE GAINS                     00011700
C     QG                  PITCH RATE (GUIDANCE)                         00011800
C     QI(I)                 "    "   (FILTER INPUT)                     00011900
C     QO(I)                 "    "   (FILTER OUTPUT)                    00012000
C     QSUM                  "    "   (TEMP)                             00012100
C     QTRIM               PITCH AXIS TRIM COMMAND                       00012200
C     QWASH                 "    "   (WASHOUT FILTER OUTPUT)            00012300
C     QWLAST                "    "   (   "      "    TEMP)              00012400
C     R                   YAW RATE (MEASUREMENT)                        00012500
C     RFIT(I,J)           TABLE OF YAW RATE GAINS                       00012600
C     RG                   "   "   (GUIDANCE)                           00012700
C     RGO                 RANGE TO GO                                   00012800
C     RHO                 REFERENCE AIR DENSITY                         00012900
C     RHOV                AIR DENSITY X VELOCITY                        00013000
C     RI(I)               YAW RATE (FILTER INPUT)                       00013100
C     RL,RL2              TEMPORARY STORAGE IN REFERENCE TRAJECTORY     00013200
C     RLAND(I)            RANGE POINTS ON LANDING APPROACH:             00013300
C                         I= 1:TOUCHDOWN POINT, 2:FLARE INITIATION,     00013400
C                            3:PULLUP TERMINATION, 4:PULLUP INITIATION, 00013500
C                            5:HIGH ALTITUDE REFERENCE                  00013600
C     RO(I)                "   "   (FILTER OUTPUT)                      00013700
C     RSUM                 "   "   (TEMP)                               00013800
C     RTRIM               YAW AXIS TRIM COMMAND                         00013900
C     RUTRIM              RUDDER TRIM DISCRETE                          00014000
C     RWASH               YAW RATE (WASHOUT FILTER OUTPUT)              00014100
C     RWLAST               "   "   (   "      "    TEMP)                00014200
```

9.14-35

```
C       SBH                SPEED BRAKE COMMAND DUE TO ALTITUDE FEEDBACK    00014300
C       SBIAS              SPEED BRAKE BIAS                                00014400
C       SBU                SPEED BRAKE COMMAND DUE TO VELOCITY FEEDBACK    00014500
C       SB(I)              SPEED BRAKE BIAS DURING LANDING APPROACH        00014600
C       SCALF              INVERSF AIR CENSITY SCALE HEIGHT                00014700
C       SLO(I)             TOUCHDOWN,PRE-FLARE,AND APPROACH SLOPES         00014800
C       SLOPE              TOUCHDOWN FLIGHT-PATH SLOPE                     00014900
C       [illegible]        SLOW SAMPLING INTERVAL TABLE                    [illegible]
C       TF                 FAST SAMPLING INTERVAL                          [illegible]
C       TF2                TF/2                                            00015200
C       THETA              PITCH ANGLE (MFASUREMENT)                       00015300
C       THETAG               "     "   (GUICANCE)                          00015400
C       THFIT(I,J)         TABLE OF PITCH ATTITUCE GAINS                   00015500
C       [illegible]          "     "   (FILTER INPUT)                      [illegible]
C       [illegible]          "     "   (FILTER OUTPUT)                     [illegible]
C       [illegible]          "     "   (TEMP)                              [illegible]
C       TIME               CURRFNT TIME                                    00015900
C       TM                 MEDIUM SAMPLING INTERVAL                        J0016000
C       TNEXT              NEXT CAP ENTRY TIME                             00016100
C       [illegible]        [illegible] (TEMP)                              [illegible]
C       [illegible]        [illegible] (TEMP)                              [illegible]
C       TS                 SLOW SAMPLING INTERVAL                          [illegible]
C       U                  EARTH-RELATIVE VELOCITY MAGNITUDE (MEASUREMENT)00016500
C       UG                   "       "        "       "    (GUIDANCE)   00016600
C       UI(I)              EARTH-RELATIVE VELOCITY MAGNITUDE(FILTER INPUT)00016700
C       [illegible]
C       [illegible]
C       [illegible]
C       USLO(I)            CONSTANTS FOR VELOCITY REFERENCE                00017100
C       USUM                 "       "     "        "      (TEMP)         00017200
C       ==================================================================00017300
[illegible]            [illegible](4),[illegible](4),                      [illegible]
      1            [illegible],THO(2),[illegible]                          [illegible]
      2            [illegible],PSJ(2),RO(2),[illegible]                    [illegible]
      3            PC(2),PI(2),K(22),KFIX(22,2),TRIML(3),G(52),            00017700
      4            TRIMO(3),NI(2),NO(2),GFIX(52,2),                        00017800
      5            UC(2),UI(2),HC(2),HI(2),BFIT(2,4),PFIT(2,3),QFIT(2,3),  00017900
      6            RFIT(2,4),THFIT(2,2),PWFIT(2,3),PCOMP([illegible]),QCOMP([illegible]),[illegible]
      7            SLO(3),[illegible],[illegible],[illegible](9),[illegible]   [illegible]
      8            [illegible](2)                                          00018200
ISN 0003    COMMON        TNEXT,DRM,CEM,DAM,DSBM,RUTRIM,ETRIM,ATRIM,       00018300
      1              DRG,DFLG,CERG,DSRG,ALPHG,THETAG,QG,BETAG,PSIG,RG,     00018400
      2              PHIG,PG,THETA,PSI,PHI,CR,DEL,DER,DSB,ALPH,P,Q,R,      00018500
      3              [illegible],[illegible],SL,SS,[illegible],[illegible] [illegible]
      4                  RHO,SCALF,FAST,MEDIUM,SLOW,[illegible]            [illegible]
      5              QFIT,RFIT,THFIT,PCOMP,QCOMP,[illegible]               [illegible]
      6              UG,U,HG,H,     IM,IS,BETA,AO,AI,THO,THI,QO,QI,BO,BI,  00018900
      7              FSC,PSJ,RO,RI,PHO,PHJ,PO,PI,NO,NI,UO,UI,HO,HI,GFAR,   00019000
```

```
                   8              SLCPE,FIL                                                         00019100
ISN 0004       9000 FORMAT  (10F8.0)                                                                00019200
ISN 0005       9001 FORMAT  (F8.0,10I5)                                                             00019300
ISN 0006       9500 FORMAT  (20X,8E12.4)                                                            00019400
ISN 0007            RFAL K,KFIX,MEDIUM,NI,NC,NSUM                                                   00019500
ISN 0008            INTEGER FLAG1,FLAG2,FLAG3,FLAG4,FLAG5,GEAR                                      00019600
              C     **********************************************************************00019700
ISN 0009          1 CONTINUE                                                                        00019800
              C>>>>>FLIGHT CCNTROL SYSTEM EXECUTIVE ROUTINE<<<<<                                    00019900
              C     ====== ======= ====== ========= =======                                         00020000
ISN 0010            GO TO (103,104),ITURN                                                           00020100
              C>>>>>"PAD LCAD" INITIALIZATICN                                                       00020200
ISN 0011        103 CONTINUE                                                                        00020300
ISN 0012            READ (5,9000) KFIX                                                              00020400
ISN 0013            READ (5,9000) GFIX                                                              00020500
ISN 0014            READ (5,9000) S0,FL,SLOPE,RLAND,HLAND,ULAND,RHO,SCALE                           00020600
ISN 0015            RFAD (5,9000) FAST,MEDIUM,SLOW                                                  00020700
ISN 0016            RFAC (5,9000) BFIT,PFIT,PHFIT,QFIT,RFIT,THFIT,PCOMP,QCOMP,FIL                   00020800
ISN 0017            TF = .1                                                                         00020900
ISN 0018            FLAG1 = 1                                                                       00021000
ISN 0019            MANMOD = 3                                                                      00021100
ISN 0020            MODE=1                                                                          00021200
              C     ********************************************************************* 00021300
              C>>>>>READ CURRENT TIME & EXTERNAL FLAGWORDS;SET NEXT ENTRY TIME                      00021400
ISN 0021        104 CONTINUE                                                                        00021500
ISN 0022            READ (5,9001) TIME,FLAG1,FLAG2,FLAG3,FLAG4,FLAG5                                00021600
ISN 0023            TNEXT = TIME+TF                                                                 00021700
              C     ********************************************************************* 00021800
              C>>>>>NEW START, MOCE CHANGE LCGIC                                                    00021900
ISN 0024            IF (FLAG1) 13,13,20                                                             00022000
ISN 0025         13 IF (FLAG2-MODE) 20,14,20                                                        00022100
ISN 0026         14 GO TO (15,30,30,30),MODE                                                        00022200
ISN 0027         15 IF (FLAG3-MANMOD) 20,30,20                                                      00022300
              C     ********************************************************************* 00022400
              C>>>>>SEQUENCE & I/C INITIALIZATION                                                   00022500
ISN 0028         20 FLAG1 = 0                                                                       00022600
ISN 0029            WRITE (6,9001) FLAG1                                                            00022700
ISN 0030            MODE = FLAG2                                                                    00022800
ISN 0031            MANMOD = FLAG3                                                                  00022900
ISN 0032            ISTART = 1                                                                      00023000
              C>>>>>ESTABLISH SAMPLING INTERVALS                                                    00023100
ISN 0033            TF = FAST(MOCE)                                                                 00023200
ISN 0034            TM = MEDIUM(MCDE)                                                               00023300
ISN 0035            TS = SLOW(MODE)                                                                 00023400
ISN 0036            TF2 = TF/2.                                                                     00023500
ISN 0037            ITM = TM/TF                                                                     00023600
ISN 0038            ITS = TS/TF                                                                     00023700
ISN 0039            DO 2001 I=1,22                                                                  00023800
ISN 0040      2001 K(I) = KFIX(I,MODE)                                            00023900
ISN 0041           DO 2002 I=1,52                                                 00024000
ISN 0042      2002 G(I) = GFIX(I,MODE)                                            00024100
ISN 0043           GO TO (2003,2006),MODE                                         00024200
ISN 0044      2003 GO TO (2006,2004,2006),MANMCD                                  00024300
ISN 0045      2004 G(1) = G(1)-K(5)                                               00024400
ISN 0046           G(2) = G(2)-K(9)                                               00024500
ISN 0047           [illegible] = 0.                                               00024600
ISN 0048           [illegible] = 0.                                               00024700
          C>>>>>INITIALIZE INDICES                                                00024800
ISN 0049      2006 IM = 0                                                         00024900
ISN 0050           IS = 0                                                         00025000
ISN 0051           GO TO 30                                                       00025100
          C>>>>>BRANCH TO READ                                                    00025200
ISN 0052        30 GO TO ([illegible],350,360,370),MODE                           00025300
          C>>>>>BRANCH TO FILTER & PARAMETER INITIALIZATION                       00025400
ISN 0053        22 IF (ISTART) 40,40,23                                           00025500
ISN 0054        23 GO TO (200,210,220,230,240,250,260,270),MODE                   00025600
ISN 0055       200 GO TO (2111,2000,2000),MANMOD                                  00025700
          C>>>>>[illegible]-PART 1 (FILTER [illegible])                           [illegible]
ISN [illegible]  40 GO TO ([illegible],440,450,460,470),MODE                      [illegible]
          C>>>>>BRANCH TO CONTROL LAW                                             [illegible]
ISN 0057        50 GO TO (500,510,520,530,540,550,560,570),MODE                   00026100
ISN 0058       500 GO TO (5000,5010,5010),MANMOD                                  00026200
          C>>>>>BRANCH TO S\  E FILTERS-PART 2 (FILTER PUSHDOWN)                  00026300
[illegible]
          C>>>>>BRANCH TO PARAMETER ESTIMATION                                    [illegible]
ISN 0061        70 GO TO (700,705,705,705),MODE                                   00026700
ISN 0062       700 GO TO (80,7000,7000),MANMOD                                    00026800
ISN 0063      7000 IF (FLAG5) 701,701,702                                         00026900
[illegible]
          C>>>>>CLOSEOUT                                                          00027300
ISN 0067        80 IF (ISTART) 82,82,81                                           00027400
          C      ************************************************************00027500
[illegible]
ISN 0068       300 CONTINUE                                                       00027900
ISN 0069           READ (5,9000) CRM,CEM,DAM,DSBM,RUTRIM,ETRIM,ATRIM,NI(1)        00028000
ISN 0070           GO TO 311                                                      00028100
[illegible]
                 1                PHIG,PG,THETA,PSI,PHI,UG,HG                     00028500
ISN 0073       311 CONTINUE                                                       00028600
[illegible]
```

9.14-37

```
ISN 0074          READ (5,9000) DR,DEL,DER,DSB,ALPH,BETA,P,Q,R,U,H,RGO,FAIL          00028700
ISN 0075          GO TO 22                                                            00028800
            C>>>>>TRANSITION MODE READ                                                00028900
ISN 0076      320 GO TO 22                                                            00029000
            C>>>>>ENTRY MODE READ                                                     00029100
ISN 0077      330 GO TO 22                                                            00029200
            C>>>>>ORBIT TVC MODE READ                                                 00029300
ISN 0078      340 GO TO 22                                                            00029400
            C>>>>>ORBIT RCS MODE READ                                                 00029500
ISN 0079      350 GO TO 22                                                            00029600
            C>>>>>INSERTION TVC MODE READ                                             00029700
ISN 0080      360 GO TO 22                                                            00029800
            C>>>>>BOOSTER TVC MODE READ                                               00029900
ISN 0081      370 GO TO 22                                                            00030000
            C     ****************************************************************** 00030100
            C>>>>>INITIALIZATION ROUTINES<<<<<                                        00030200
            C     ============== ========                                            00030300
            C>>>>>MANUAL MODES                                                        00030400
ISN 0082     2000 UG    = U                                                           00030500
ISN 0083          HG    = H                                                           00030600
ISN 0084          NSUM = 0.                                                           00030700
ISN 0085          NO(2) = 0.                                                          00030800
ISN 0086          NI(2) = 0.                                                          00030900
            C>>>>>GENERATE PULLUP & FLARE CONSTANTS FOR MANUAL APPROACH               00031000
            C>>>>>ALTITUDE CONSTANTS                                                  00031100
ISN 0087          SLO(1) = SLOPE                                                      00031200
ISN 0088          DO 2010 I=1,2                                                       00031300
ISN 0089          J = 2*I                                                             00031400
ISN 0090          SLO(I+1) = (HLAND(J+1)-HLAND(J))/(RLAND(J+1)-RLAND(J))              00031500
ISN 0091          FH(1,I) = HLAND(J-1)                                                00031600
ISN 0092          FH(2,I) = SLO(I)                                                    00031700
ISN 0093          RL = RLAND(J)-RLAND(J-1)                                            00031800
ISN 0094          RL2 = RL*RL                                                         00031900
ISN 0095          HL = HLAND(J)-(FH(1,I)+FH(2,I)*RL)                                  00032000
ISN 0096          HLP = SLO(I+1)-FH(2,I)                                              00032100
ISN 0097          FH(3,I) = (3.*HL-RL*HLP)/RL2                                        00032200
ISN 0098     2010 FH(4,I) = (RL*HLP-2.*HL)/(RL2*RL)                                   00032300
            C>>>>>VELOCITY CONSTANTS                                                  00032400
ISN 0099          DO 2012 I=1,3                                                       00032500
ISN 0100     2012 USLO(I) = (ULAND(I+1)-ULAND(I))/(RLAND(I+1)-RLAND(I))               00032600
ISN 0101          USLO(4) = ALOG(ULAND(5)/ULAND(4))/(RLAND(5)-RLAND(4))               00032700
            C>>>>>AUTO MODE                                                           00032800
ISN 0102      210 ELBIAS = 0.                                                         00032900
ISN 0103          SBIAS = 0.                                                          00033000
ISN 0104          PHSUM = 0.                                                          00033100
ISN 0105          THSUM = 0.                                                          00033200
ISN 0106          PSSUM = 0.                                                          00033300
ISN 0107          ASUM = 0.                                                           00033400
ISN 0108             USUM = 0.                                                    00033500
ISN 0109             HSUM = 0.                                                    00033600
ISN 0110             PSUM = 0.                                                    00033700
ISN 0111             QSUM = 0.                                                    00033800
ISN 0112             RSUM = 0.                                                    00033900
ISN 0113             BSUM = 0.                                                    00034000
ISN 0114             AO(2)  = 0.                                                  00034100
ISN 0115             AI(2)  = 0.                                                  [illegible]
[illegible]
[illegible]
ISN 0118             PHO(2) = 0.                                                  00034500
ISN 0119             PHJ(2) = 0.                                                  00034600
ISN 0120             THO(2) = 0.                                                  00034700
[illegible]
[illegible]
[illegible]
ISN 0124             PO(2) = 0.                                                   00035100
ISN 0125             PI(2) = 0.                                                   00035200
ISN 0126             QO(2) = 0.                                                   00035300
[illegible]
[illegible]
[illegible]
ISN 0130             UO(2) = 0.                                                   00035700
ISN 0131             UI(2) = 0.                                                   00035800
ISN 0132             HO(2) = 0.                                                   00035900
[illegible]
[illegible]
[illegible]
ISN 0136             DSBC = 0.                                                    00036300
ISN 0137             QWLAST = 0.                                                  00036400
ISN 0138             RWLAST = 0.                                                  00036500
[illegible]
[illegible]
[illegible]
ISN 0142        2134 CONTINUE                                                     00036900
ISN 0143             I = 5                                                        00037000
ISN 0144        2136 ELBIAS = EL(I-1)                                             00037100
[illegible]
[illegible]
[illegible]
ISN 0148        2142 TRIMO(I) = 0.                                                00037500
ISN 0149             GO TO 40                                                     00037600
           C>>>>>TRANSITION MODE                                                  00037700
[illegible]
[illegible]
[illegible]
           C>>>>>ORBIT TVC MODE                                                   00038100
ISN 0152          240 GO TO 40                                                    00038200
```

9.14-39

```
            C>>>>>ORBIT RCS MODE                                                  00038300
ISN 0153      250 GO TO 40                                                        00038400
            C>>>>>INSERTION TVC MCDE                                              00038500
ISN 0154      260 GO TO 40                                                        00038600
            C>>>>>BOOSTER TVC MCDE                                                00038700
ISN 0155      270 GO TC 40                                                        00038800
            C          ************** ***************************************** *************** 00038900
            C>>>>>FILTER ROUTINES-PART 1<<<<<                                     00039000
            C          ====== ======== ======                                     00039100
            C>>>>>MANUAL FILTERS-PART 1                                           00039200
            C          ADD TRIM DISCRETES TC MANUAL COMMANDS                      00039300
ISN 0156      400 NO(1) = G(31)*NI(1)+NSUM                                        00039400
                  ATRIM =  K(15)*ATRIM                                            00039500
ISN 0157
ISN 0158          ETRIM =  K(16)*ETRIM                                            00039600
ISN 0159          RUTRIM = K(17)*RUTRIM                                           00039700
ISN 0160          PTRIM = TF2*(ATRIM+TRIML(1)) + TRIMC(1)                         00039800
ISN 0161          QTRIM = TF2*(ETRIM+TRIML(2)) + TRIMC(2)                         00039900
ISN 0162          RTRIM = TF2*(RUTRIM+TRIML(3))+ TRIMC(3)                         00040000
ISN 0163          DAC = K(3)*(DAM+PTRIM)                                          00040100
ISN 0164          DEC = K(2)*(DEM+QTRIM)                                          00040200
ISN 0165          DRC = K(1)*(DRM+RTRIM)                                          00040300
ISN 0166          GO TO (50,4002,4002),MANMOD                                     00040400
ISN 0167     4002 PI(1) = DAC-P                                                   00040500
ISN 0168          QI(1) = DEC-Q                                                   00040600
ISN 0169          RI(1) = -R                                                      00040700
ISN 0170          BI(1) = -BETA                                                   00040800
ISN 0171          GO TO 4104                                                      00040900
            C>>>>>AUTO FILYERS-PART 1                                             00041009
ISN 0172      410 PHJ(1) = PHIG-PHI                                               00041100
ISN 0173          THI(1) = THETAG-THETA                                           00041200
ISN 0174          PSJ(1) = PSIG-PSI                                               00041300
ISN 0175          AI(1) = ALPHG-ALPH                                              00041400
ISN 0176          PI(1) = PG-P                                                    00041500
ISN 0177          QI(1) = QG-Q                                                    00041600
ISN 0178          RI(1) = RG-R                                                    00041700
ISN 0179          BI(1) = BETAG-BETA                                              00041800
ISN 0180          PHO(1) = G(39)*PHJ(1)+PHSUM                                     00041900
ISN 0181          THO(1) = G(38)*THI(1)+THSUM                                     00042000
ISN 0182          PSO(1) = G(40)*PSJ(1)+PSSUM                                     00042100
ISN 0183          AO(1) = G(41)*AI(1)+ASUM                                        00042200
ISN 0184     4104 IF (IM) 412,411,412                                             00042300
ISN 0185      411 UI(1) = UG-U                                                    00042400
ISN 0186          UO(1) = G( 5)*UI(1)+USUM                                        00042500
ISN 0187      412 IF (IS) 414,413,414                                             00042600
ISN 0188      413 HI(1) = HG-H                                                    00042700
ISN 0189          HO(1) = G( 6)*HI(1)+HSUM                                        00042800
ISN 0190      414 PO(1) = G(1)*PI(1)                                              00042900
ISN 0191          QO(1) = G(2)*QI(1)                                              00043000
ISN 0192           RO(1) = G(3)*RI(1)                                                          00043100
ISN 0193           BO(1) = G(4)*BI(1)                                                          00043200
ISN 0194           GO TO (416,427),MODE                                                        00043300
ISN 0195       416 GO TO (90,418,417),MANMOD                                                   00043400
ISN 0196       417 PO(1) = PO(1)+PSUM                                                          00043500
ISN 0197           QO(1) = QO(1)+QSUM                                                          0004[illegible]0
ISN 0198           RO(1) = RO(1)+RSUM                                                          00043700
ISN 0199           BO(1) = BO(1)+BSUM                                                          [illegible]
         C>>>>>YAW AND PITCH RATE WASHOUT FILTERS                                              [illegible]
ISN 0200       418 QWASH = G(36)*(QO(1)-QO(2))+G(37)*QWLAST                                    [illegible]
ISN 0201           RWASH = G(36)*(RO(1)-RO(2))+G(37)*RWLAST                                    00044100
ISN 0202           GO TO 50                                                                    00044200
         C>>>>>TRANSITION FILTERS-PART 1                                                       00044300
ISN 0203       420 GO TO 50                                                                    00044400
         [illegible]                                                                           [illegible]
ISN 0204       [illegible]                                                                     [illegible]
         C>>>>>ORBIT TVC FILTERS-PART 1                                                        00044700
ISN 0205       440 GO TO 50                                                                    00044800
         C>>>>>ORBIT RCS FILTERS-PART 1                                                        00044900
ISN 0206       [illegible]                                                                     [illegible]
         [illegible]                                                                           [illegible]
ISN 0207       [illegible]                                                                     [illegible]
         C>>>>>BOOSTER TVC FILTERS-PART 1                                                      00045300
ISN 0208       470 GO TO 50                                                                    00045400
         C**************************************************************************************** 00045500
         [illegible]
ISN 0209      5000 DRC = K(1)*DRM                                                              00045900
ISN 0210           DEC = K(2)*DEM                                                              00046000
ISN 0211           DAC = K(3)*DAM                                                              00046100
         [illegible]
ISN 0214      5010 DAC = PO(1)+K(5)*P+K(6)*RWASH                                               00046500
ISN 0215           DEC = QO(1)+K(9)*Q+K(18)*NO(1)                                              00046600
ISN 0216           DRC = RWASH+K(7)*P+K(10)*BO(1)+DRC                                          00046700
ISN 0217       [illegible]                                                                     [illegible]
         [illegible]
ISN 0218       [illegible]                                                                     [illegible]
ISN 0219           DEC = K(2)*THC(1)+K(9)*CC(1)+K(4)*AO(1)                                     00047100
ISN 0220           DRC = K( 1)*PHO(1)+K( 7)*?O(1)+K(16)*PSO(1)+K( 8)*RWASH+                    00047200
                  1     K(10)*BO(1)                                                            00047300
         [illegible]
ISN 0221       [illegible]                                                                     [illegible]
ISN 0222      5111 IF [illegible] 5112,5112,5114                                               [illegible]
ISN 0223      5112 DSBC = K(4)*DSBM                                                            00047700
ISN 0224           GO TO 517                                                                   00047800
```

9.14-43

```
                                                                                                                  PAGE 011
ISN 0225      5114 IF (IM) 513,512,513                                                           00047900
ISN 0226       512 DECU = K(11)*UO(1)                                                            00048000
ISN 0227           SBU = K(12)*UO(1)                                                             00048100
ISN 0228       513 IF (IS) 515,514,515                                                           00048200
ISN 0229       514 DECH = K(13)*HO(1)+ELBIAS                                                     00048300
ISN 0230           SBH = K(14)*HO(1)+SBIAS                                                       00048400
ISN 0231       515 DEC = DEC+DECU+DECH                                                           00048500
ISN 0232           DSBC = SBU+SBH                                                                00048600
ISN 0233           IF (MODE-1) 517,516,517                                                       00048700
ISN 0234       516 DSBC = DSBC+K(4)*DSBM                                                         00048800
ISN 0235       517 DELC = DEC-DAC                                                                00048900
ISN 0236           DERC = DEC+DAC                                                                00049000
ISN 0237           WRITE (6,9500) CRC,DELC,DERC,DSBC,TNEXT                                       00049100
ISN 0238           GO TO (518,60),MODE                                                           00049200
ISN 0239       518 GO TO (60,519,519),MANMOD                                                     00049300
ISN 0240       519 TRIML(1) = ATRIM                                                              00049400
ISN 0241           TRIML(2) = ETRIM                                                              00049500
ISN 0242           TRIML(3) = RUTRIM                                                             00049600
ISN 0243           TRIMO(1) = PTRIM                                                              00049700
ISN 0244           TRIMO(2) = QTRIM                                                              00049800
ISN 0245           TRIMO(3) = RTRIM                                                              00049900
ISN 0246           GO TO 60                                                                      00050000
             C>>>>>TRANSITION CONTROL                                                            00050100
ISN 0247       520 GO TO 60                                                                      00050200
             C>>>>>ENTRY CONTROL                                                                 00050300
ISN 0248       530 GO TO 60                                                                      00050400
             C>>>>>ORBIT TVC CONTROL                                                             00050500
ISN 0249       540 GO TO 60                                                                      00050600
             C>>>>>ORBIT RCS CONTROL                                                             00050700
ISN 0250       550 GO TO 60                                                                      00050800
             C>>>>>INSERTION TVC CONTROL                                                         00050900
ISN 0251       560 GO TO 60                                                                      00051000
             C>>>>>BOOSTER TVC CONTROL                                                           00051100
ISN 0252       570 GO TO 60                                                                      00051200
             C     ******************************************************************** 00051300
             C>>>>>FILTER ROUTINES-PART 2<<<<<                                                   00051400
             C     ====== ======== ======                                                        00051500
             C>>>>>MANUAL FILTERS-PART 2                                                         00051600
ISN 0253      6000 NSUM = G(33)*NI(1)+G(35)*NI(2)+G(32)*NO(1)+G(34)*NO(2)                        00051700
ISN 0254           NI(2) = NI(1)                                                                 00051800
ISN 0255           NO(2) = NO(1)                                                                 00051900
ISN 0256           GO TO 611                                                                     00052000
             C>>>>>AUTO FILTERS-PART 2                                                           00052100
ISN 0257       610 PHSUM = G(45)*PHJ(1)+G(51)*PHJ(2)+G(42)*PHO(1)+G(48)*PHO(2)                   00052200
ISN 0258           THSUM = G(44)*THI(1)+G(50)*THI(2)+G(41)*THO(1)+G(47)*THO(2)                   00052300
ISN 0259           PSSUM = G(46)*PSJ(1)+G(52)*PSJ(2)+G(43)*PSO(1)+G(49)*PSO(2)                   00052400
ISN 0260           ASUM = G(33)*AI(1)+G(35)*AI(2)+G(32)*AO(1)+G(34)*AO(2)                        00052500
ISN 0261           PHJ(2) = PHJ(1)                                                               00052600
```

9.14-63

```
ISN 0262            PHO(2) = PHO(1)                                                 00052700
ISN 0263            THI(2) = THI(1)                                                 00052800
ISN 0264            THO(2) = THO(1)                                                 00052900
ISN 0265            PSJ(2) = PSJ(1)                                                 00053000
ISN 0266            PSO(2) = PSC(1)                                                 00053100
ISN 0267            AI(2) = AI(1)                                                   00053200
ISN 0268            AO(2) = AO(1)                                                   00053300
ISN 0269        611 IF (IMI) 613,612,613                                            00053400
ISN 0270        612 UNOW = G(17)*UI(1)+G(29)*UI(2)+G(11)*UO(1)+G(23)*UO(2)          00053500
ISN 0271            UI(2) = UI(1)                                                   00053600
ISN 0272            UO(2) = UO(1)                                                   00053700
ISN 0273        613 IF (IS) 615,614,615                                             00053800
ISN 0274        614 HSUM = G(18)*HI(1)+G(30)*HI(2)+G(12)*HO(1)+G(24)*HO(2)          00053900
ISN 0275            HI(2) = HI(1)                                                   00054000
ISN 0276            HO(2) = HO(1)                                                   00054100
ISN 0277        615 GO TO (616,617),MCODE                                           00054200
ISN 0278        616 GO TO (70,618,617),MANMCD                                       00054300
ISN 0279        617 PSUM = G(13)*PI(1)+G(25)*PI(2)+G(7)*PO(1)+G(19)*PO(2)           00054400
ISN 0280            QSUM = G(14)*QI(1)+G(26)*QI(2)+G( 8)*QO(1)+G(20)*QO(2)          00054500
ISN 0281            RSUM = [illegible](2)+G( 9)*[illegible]                         00054600
ISN 0282            [illegible]                                                     00054700
ISN 0283            PO(2) = PO(1)                                                   00054800
ISN 0284            QI(2) = QI(1)                                                   00054900
ISN 0285            QO(2) = QO(1)                                                   00055000
ISN 0286            RI(2) = RI(1)                                                   00055100
[illegible]
ISN 0290            BO(2) = BO(1)                                                   00055500
ISN 0291        618 QWLAST = QWASH                                                  00055600
ISN 0292            RWLAST = RWASH                                                  00055700
[illegible]
                C>>>>>ENTRY FILTERS-PART 2                                          00056100
ISN 0295        630 GO TO 70                                                        00056200
                C>>>>>ORBIT TVC FILTERS-PART 2                                      00056300
ISN 0296        640 GO TO 70                                                        [illegible]
                C>>>>>[illegible] FILTERS-PART 2                                    [illegible]
ISN 0297        650 GO TO 70                                                        [illegible]
                C>>>>>INSERTION TVC FILTERS-PART 2                                  00056700
ISN 0298        660 GO TO 70                                                        00056800
                C>>>>>BOOSTER TVC FILTERS-PART 2                                    00056900
ISN 0299        [illegible]                                                         [illegible]
                C                                                                   [illegible]
                C>>>>>NULL REFERENCE TRAJECTORY<<<<<                                [illegible]
                C     ==== ========= ==========                                     00057300
ISN 0300        701 UG = U                                                          00057400
ISN 0301                HG = H                                                          00057500
ISN 0302                ELBIAS = 0.                                                     00057600
ISN 0303                SBIAS = 0.                                                      00057700
ISN 0304                GO TO 705                                                       00057800
              C>>>>>LANDING APPROACH TRAJECTORY FOR MANUAL MCDES<<<<<                   00057900
              C         ======= ======== ========== === ====== ====-                    00058000
ISN 0305            702 DO 7022 I=1,5                                                   00058100
ISN 0306                IF (RGO-RLAND(I)) 7024,7024,7022                                00058200
ISN 0307           7022 CONTINUE                                                        00058300
ISN 0308                I = 5                                                           00058400
ISN 0309           7024 IR = I-1                                                        00058500
ISN 0310                RL = RGO-RLAND(IR)                                              00058600
ISN 0311                GO TO (703,704,703,704),IR                                      00058700
              C>>>>>FLARE(IR=1) AND PULLUP(IR=3)                                        00058800
ISN 0312            703 GEAR = 1                                                        00058900
ISN 0313                IF (IR-2) 7032,7032,7034                                        00059000
ISN 0314           7032 J = 1                                                           00059100
ISN 0315                GC TC 7036                                                      00059200
ISN 0316           7034 J = 2                                                           00059300
ISN 0317           7036 RL2 = RL*RL                                                     00059400
ISN 0318                UG = ULAND(IR)+USLO(IR)*RL                                      00059500
ISN 0319                HG = FH(1,J)+FH(2,J)*RL+FH(3,J)*RL2+FH(4,J)*RL2*RL              00059600
ISN 0320                GC TC 7046                                                      00059700
              C>>>>>APPROACH GLIDE(IR=4) ANC PRE-FLARE GLIDE(IR=2)                      00059800
ISN 0321            704 IF (IR-3) 7042,7042,7044                                        00059900
ISN 0322           7042 J = 2                                                           00060000
ISN 0323                UG = ULAND(2)+USLO(3)*RL                                        00060100
ISN 0324                GEAR = 1                                                        00060200
ISN 0325                GO TC 7046                                                      00060300
ISN 0326           7044 J = 3                                                           00060400
ISN 0327                UG = ULAND(4)*EXP(USLC(4)*RL)                                   00060500
ISN 0328                HG = HLAND(IR)+SLO(J)*RL                                        00060600
ISN 0329                GEAR = 0                                                        00060700
ISN 0330           7046 ELBIAS = EL(IR)                                                 00060800
ISN 0331                SBIAS = SB(IR)                                                  00060900
ISN 0332                GO TC 705                                                       00061000
              C         *******************************************************************00061100
              C>>>>>PARAMETER UPDATE ROUTINES<<<<<                                      00061200
              C         ********* ****** ********                                       00061300
ISN 0333           7051 AMFAN = AMFAN+.05*(ALPH-AMEAN)                                  00061400
ISN 0334                UMEAN = UMFAN+.05*(U-UMFAN)                                     00061500
ISN 0335                IF (IS) 80,7052,80                                              00061600
ISN 0336           7052 PTEMP = RHO*EXP(-SCALE*H)                                       00061700
ISN 0337                RHOV = PTEMP*UMEAN                                              00061800
ISN 0338                DYNP = .5*RHOV*UMEAN                                            00061900
ISN 0339                GO TO 7053                                                      00062000
              C>>>>>PARAMETER ESTIMATION FOR MANUAL MODES                               00062100
ISN 0340            706 CALL TABLE(PCOMP,AMFAN,PTEMP,4)                                 00062200
ISN 0341            G(1) =  PTEMP                                                          00062300
ISN 0342            G(13) = -PTEMP*FIL(1)                                                  00062400
ISN 0343            CALL TABLE(QCOMP,DYNP,PTEMP,3)                                         00062500
ISN 0344            G(2) =  PTEMP                                                          00062600
ISN 0345            G(14) = -PTFMP*FIL(2)                                                  00062700
ISN 0346            CALL TABLE(PFIT,AMEAN,PTEMP,3)                                         00062800
ISN 0347            K(5) = PTEMP                                                           00062900
ISN 0348            CALL TABLE([illegible]FIT,AMEAN,PTEMP,3)                               [illegible]
[illegible]
[illegible]         CALL TABLE([illegible]FIT,DYNP,PTEMP,4)                                [illegible]
ISN 0351            G(3) = PTEMP                                                           00063300
ISN 0352            GO TO 712
                                                                                           00063400
          C>>>>>PARAMETER ESTIMATION FOR AUTO MODE                                         00063500
ISN 0353       716  CALL TABLE([illegible]FIT,[illegible],PTEMP,3)                         [illegible]
[illegible]
[illegible]
ISN 0356            G(50) = G(38)*(FIL(2)-1.)                                              00063900
ISN 0357            CALL TABLE(QFIT,AMEAN,PTEMP,3)                                         00064000
ISN 0358            G(2) = PTEMP1-PTEMP                                                    00064100
[illegible]
[illegible]
ISN 0361            CALL TABLE([illegible],PTEMP,4)                                        [illegible]
ISN 0362            G(39) = PTEMP*PTEMP1                                                   00064500
ISN 0363            G(45) = G(39)*(FIL(1)-1.)                                              00064600
ISN 0364            CALL TABLE(RFIT,AMEAN,PTEMP,3)                                         00064700
[illegible]
[illegible]
ISN 0367            CALL TABLE([illegible])                                                [illegible]
ISN 0368            K(8) = PTEMP                                                           00065100
ISN 0369       712  CALL TABLE(BFIT,DYNP,PTEMP,4)                                          00065200
ISN 0370            K(10) = PTEMP                                                          00065300
[illegible]
ISN 0372            [illegible] GO TO 80                                                   [illegible]
          C>>>>>PARAMETER ESTIMATION FOR ENTRY MODE                                        00065700
ISN 0373       730  GO TO 80                                                               00065800
          C>>>>>PARAMETER ESTIMATICN FCR ORBIT TVC MODE                                    00065900
[illegible]
ISN 0375       750  GO TO 80                                                               [illegible]
          C>>>>>PARAMETER ESTIMATION FOR INSERTION TVC MCDE                                00066300
ISN 0376       760  GO TO 80                                                               00066400
          C>>>>>PARAMETER ESTIMATION FCR BOCSTER TVC MODE                                  00066500
[illegible]
          C>>>>>[illegible]<<<<<                                                           [illegible]
          C         ========                                                               00066900
ISN 0378        81  ISTART = 0                                                             00067000
```

0.14–45

```
ISN 0379          IS = IS+1                                                 00067100
            C>>>>>>INCREMENT INDICES                                        00067200
ISN 0380       82 IM = IM+1                                                 00067300
ISN 0381          IS = IS+1                                                 00067400
ISN 0382          IM = MOD(IM,ITM)                                          00067500
ISN 0383          IS = MOD(IS,ITS)                                          00067600
            C>>>>>>RETURN TO GN&C EXECUTIVE<<<<<                            00067700
ISN 0384          RETURN                                                    00067800
ISN 0385          END                                                       00067900
```

```
LEVEL 20.1 (MAY 71)                         OS/360  FORTRAN H                                        DATF  71.302/13.25.56

          COMPILER CPTIONS - NAME=  MAIN,OPT=02,LINECNT=50,SIZE=0000K,
                             SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,NOID,XREF
  ISN 0002                SUBROUTINE TABLE(A,Z,B,N)                                          00068000
                    C>>>>>LINEAR INTERPOLATION<<<<<                                          00068100
                    C>>>>>A(I,J)=TABLF, Z=INDEPENDENT VARIABLE, B=CUTPUT                     00068200
                    C     N=NUMBER OF TABULATED PAIRS                                        00068300
  ISN 0003                DIMENSICN A(2,1)                                                   00068400
  ISN 0004                DO 1 J=1,N                                                         00068500
  ISN 0005                IF (Z-A(2,J)) 3,2,1                                                00068600
  ISN 0006              1 CONTINUE                                                           00068700
  ISN 0007                J=N                                                                00068800
  ISN 0008              2 B = A(1,J)                                                         00068900
  ISN 0009                GO TO 5                                                            00069000
  ISN 0010              3 IF (J-1) 2,2,4                                                     00069100
  ISN 0011              4 B =(A(1,J)-A(1,J-1))*(Z-A(2,J-1))/(A(2,J)-A(2,J-1))+A(1,J-1)       00069200
  ISN 0012              5 CONTINUE                                                           00069300
  ISN 0013                RETURN                                                             00069400
  ISN 0014                END                                                                00069500
```

9.14-47

```
CHART TITLE - INTRODUCTORY COMMENTS


>>>>>SSV UNIFIED DIGITAL FLIGHT CONTROL SYSTEM-VERSION 4D  10-29-71
     R.F. STENGEL                                          MIT CSDL
     *******************************************************************
                        NOMENCLATURE
                        ============
     AI(I)             ANGLE OF ATTACK (FILTER INPUT)
     ALPH                "   "   "   (MEASUREMENT)
     ALPHG               "   "   "   (GUIDANCE)
     AMEAN             MEAN ANGLE OF ATTACK FOR PARAM. EST.
     AO(I)               "   "   "   (FILTER OUTPUT)
     ASUM                "   "   "   (TEMP)
     ATRIM             'AILERON' TRIM DISCRETE
     BETA              SIDESLIP ANGLE (MEASUREMENT)
     BETAG              "       "   (GUIDANCE)
     BFIT(I,J)         TABLE OF SIDESLIP GAINS
     BI(I)              "       "   (FILTER INPUT)
     BO(I)              "       "   (FILTER OUTPUT)
     BSUM               "       "   (TEMP)
     DAC               'AILERON' ANGLE COMMAND OUTPUT
     DACINT            RCAH INTEGRAL IN 'AILERON' ANGLE COMMAND
     DAM               'AILERON' ANGLE COMMAND (MANUAL)
     DEC               'ELEVATOR' ANGLE COMMAND OUTPUT
     DECH              'ELEVATOR' COMMAND DUE TO ALTITUDE FEEDBACK
     DECINT            RCAH INTEGRAL IN 'ELEVATOR' ANGLE COMMAND
     DECU              'ELEVATOR' COMMAND DUE TO VELOCITY FEEDBACK
     DEL               LEFT ELEVON ANGLE (MEASUREMENT)
```

DELC	" " " (COMMAND)
DELG	" " " (GUIDANCE)
DEM	'ELEVATOR' ANGLE COMMAND (MANUAL)
DER	RIGHT ELEVON ANGLE (MEASUREMENT)
DERC	" " " (COMMAND)
DERG	" " " (GUIDANCE)
DR	RUDDER ANGLE (MEASUREMENT)
DRC	" " (COMMAND)
DRG	" " (GUIDANCE)
DRM	RUDDER ANGLE COMMAND (MANUAL)
DSB	SPEED BRAKE SETTING (MEASUREMENT)
DSBC	" " " (COMMAND)
DSBG	" " " (GUIDANCE)
DSBM	SPEED BRAKE COMMAND (MANUAL)
DYNP	DYNAMIC PRESSURE ESTIMATE
ELBIAS	'ELEVATOR' BIAS TO TRIM SPEED BRAKE BIAS
EL(I)	TABLE OF'ELEVATOR'BIASES TO TRIM SPEED BRAKE
ETRIM	'ELEVATOR' TRIM DISCRETE
FAIL(I)	FAILURE DISCRETES
FAST(I)	FAST SAMPLING INTERVAL TABLE
FH(I,J)	CONSTANTS FOR FLARE & PULLUP ALTITUDE REFERENCE
FIL(I)	FILTER CONSTANTS USED IN PARAMETER ESTIMATION
FLAG1	NEW START DISCRETE (EXTERNAL):0=NORMAL CYCLE 1=NEW START
FLAG2	FLIGHT MODE FLAG (EXTERNAL):1=MANUAL MODE 2=AUTOMATIC MODE

9.14-69

9.14.3.2 DFCS (cont'd)

```
CHART TITLE - INTRODUCTORY COMMENTS

                                                3=TRANSITION MODE
                                                4=ENTRY MODE
                                                5=ORBIT TVC MODE
                                                6=ORBIT RCS MODE
                                                7=INSERTION TVC
                                                8=BOOSTER TVC MODE
FLAG3           MANUAL MODE FLAG (EXTERNAL):1=DIRECT
                                                2=RATE COMMAND
                                                3=RCAH
FLAG4           PARAMETER ESTIMATION FLAG (EXTERNAL):0=SKIP
                                                     1=DO
FLAG5           MANUAL MODE AUTO. SPEED BRAKE FLAG(EXTERNAL):
                       0=NO AUTO. SPEED BRAKE
                       1=AUTO. SPEED BRAKE
G(I)            FILTER GAINS FOR CURRENT FLIGHT MODE
GEAR            LANDING GEAR DISCRETE
GFIX(I,J)       FILTER GAIN TABLE FOR ALL FLIGHT MODES
H               ALTITUDE ABOVE RUNWAY (MEASUREMENT)
HG                 "       "     "    (GUIDANCE)
HI(I)           ALTITUDE ABOVE RUNWAY (FILTER INPUT)
HL,HLP          TEMPORARY STORAGE IN REFERENCE TRAJECTORY
HLAND(I)        ALTITUDE POINTS ON LANDING APPROACH
HO(I)           ALTITUDE ABOVE RUNWAY (FILTER OUTPUT)
HSUM               "       "     "    (TEMP)
I               GENERAL PURPOSE INDEX
IM              MEDIUM SAMPLING RATE INDEX
```

IR	REFERENCE TRAJECTORY INDEX
IS	SLOW SAMPLING RATE INDEX
ISTART	STARTING CYCLE DISCRETE
ITM	# OF FAST CYCLES PER MEDIUM CYCLE
ITS	# OF FAST CYCLES PER SLOW CYCLE
ITURN	TURN-ON CYCLE DISCRETE
J	REFERENCE TRAJECTORY INDEX
K(I)	CONTROL GAINS FOR CURRENT FLIGHT MODE
KFIX(I,J)	CONTROL GAIN TABLE FOR ALL FLIGHT MODES
MANMOD	MANUAL MODE FLAG (INTERNAL)
MEDIUM(I)	MEDIUM SAMPLING INTERVAL TABLE
MODE	FLIGHT MODE FLAG (INTERNAL)
NI(I)	NORMAL ACCELERATION (FILTER INPUT)
NO(I)	NORMAL ACCELERATION (FILTER OUTPUT)
NSUM	" " (TEMP)
P	ROLL RATE (MEASUREMENT)
PCOMP(I,J)	TABLE OF ROLL COMPENSATION GAINS
PFIT(I,J)	TABLE OF ROLL RATE GAINS
PG	ROLL RATE (GUIDANCE)
PHFIT(I,J)	TABLE OF ROLL ATTITUDE GAINS
PHI	ROLL ATTITUDE (MEASUREMENT)
PHIG	" " (GUIDANCE)
PHJ(I)	" " (FILTER INPUT)
PHO(I)	" " (FILTER OUTPUT)
PHSUM	" " (TEMP)
PI(I)	ROLL RATE (FILTER INPUT)

```
FO(I)                  "    "   (FILTER OUTPUT)
PSI                   YAW ATTITUDE (MEASUREMENT)
PSIG                   "     "     (GUIDANCE)
PSJ(I)                 "     "     (FILTER INPUT)
PSO(I)                 "     "     (FILTER OUTPUT)
PSSUM                  "     "     (TEMP)
PSUM                  ROLL RATE (TEMP)
PTEMP                 TEMPORARY STORAGE IN PARAM. EST.
PTEMP1                    "        "    "    "     "
PTRIM                 ROLL AXIS TRIM COMMAND
Q                     PITCH RATE (MEASUREMENT)
QCOMP(I,J)            TABLE OF PITCH COMPENSATION GAINS
QFIT(I,J)             TABLE OF PITCH RATE GAINS
QG                    PITCH RATE (GUIDANCE)
QI(I)                   "    "   (FILTER INPUT)
QO(I)                   "    "   (FILTER OUTPUT)
QSUM                    "        (TEMP)
QTRIM                 PITCH AXIS TRIM COMMAND
QWASH                   "    "   (WASHOUT FILTER OUTPUT)
QWLAST                  "    "   (    "       "     TEMP)
R                     YAW RATE (MEASUREMENT)
RFIT(I,J)             TABLE OF YAW RATE GAINS
RG                     "    "   (GUIDANCE)
RGO                   RANGE TO GO
RHO                   REFERENCE AIR DENSITY
RHOV                  AIR DENSITY X VELOCITY
```

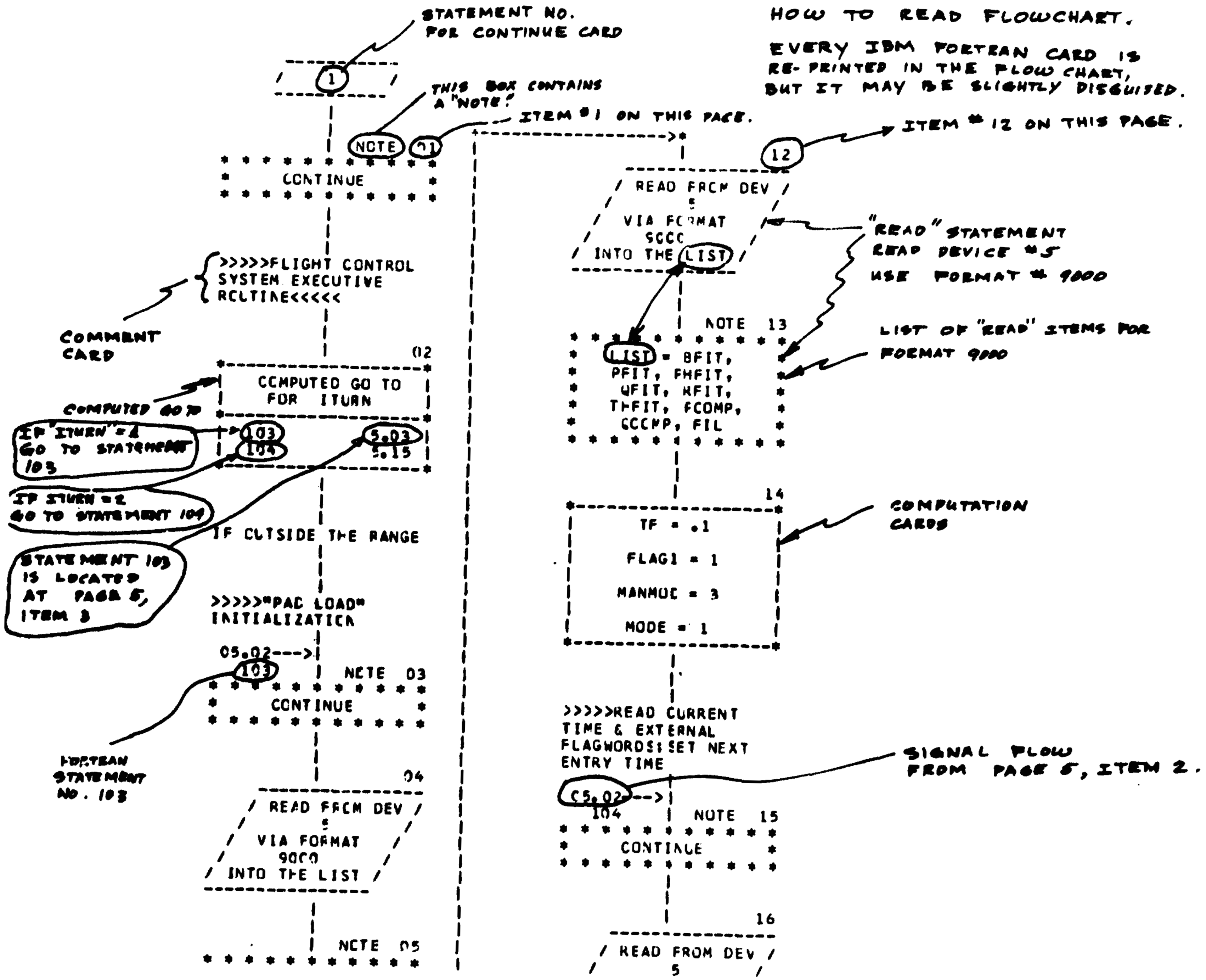
RI(I)	YAW RATE (FILTER INPUT)
RL,RL2	TEMPORARY STORAGE IN REFERENCE TRAJECTORY
RLAND(I)	RANGE POINTS ON LANDING APPROACH: I= 1:TOUCHDOWN POINT, 2:FLARE INITIATION, 3:PULLUP TERMINATION, 4:PULLUP INITIATION, 5:HIGH ALTITUDE REFERENCE
RO(I)	" " (FILTER OUTPUT)
RSUM	" " (TEMP)
RTRIM	YAW AXIS TRIM COMMAND
RUTRIM	RUDDER TRIM DISCRETE
RWASH	YAW RATE (WASHOUT FILTER OUTPUT)
RWLAST	" " (" " TEMP)
SBH	SPEED BRAKE COMMAND DUE TO ALTITUDE FEEDBACK
SBIAS	SPEED BRAKE BIAS
SBU	SPEED BRAKE COMMAND DUE TO VELOCITY FEEDBACK
SB(I)	SPEED BRAKE BIAS DURING LANDING APPROACH
SCALE	INVERSE AIR DENSITY SCALE HEIGHT
SLO(I)	TOUCHDOWN,PRE-FLARE,AND APPROACH SLOPES
SLOPE	TOUCHDOWN FLIGHT-PATH SLOPE
SLOW(I)	SLOW SAMPLING INTERVAL TABLE
TF	FAST SAMPLING INTERVAL
TF2	TF/2
THETA	PITCH ANGLE (MEASUREMENT)
THETAG	" " (GUIDANCE)
THFIT(I,J)	TABLE OF PITCH ATTITUDE GAINS
THI(I)	" " (FILTER INPUT)

```
11/01/71                                  AUTOFLOW CHART SET -     MAIN                       PAGE  04

CHART TITLE - INTRODUCTORY COMMENTS


                THO(I)          "    "   (FILTER OUTPUT)
                THSUM           "    "   (TEMP)
                TIME          CURRENT TIME
                TM            MEDIUM SAMPLING INTERVAL
                TNEXT         NEXT DAP ENTRY TIME
                TRIML(I)      TRIM FILTER (TEMP)
                TRIMO(I)      TRIM FILTER (TEMP)
                TS            SLOW SAMPLING INTERVAL
                U             EARTH-RELATIVE VELOCITY MAGNITUDE (MEASUREMENT)
                UG              "     "       "         "     (GUIDANCE)
                UI(I)         EARTH-RELATIVE VELOCITY MAGNITUDE(FILTER INPUT)
                ULAND(I)      VELOCITY POINTS ON LANDING APPROACH
                UMEAN         MEAN VELOCITY FOR PARAM. EST.
                UO(I)         EARTH-RELATIVE VELOCITY MAGNITUDE(FILTER OUTPUT
                USLO(I)       CONSTANTS FOR VELOCITY REFERENCE
                USUM            "     "       "         "     (TEMP)
                **************************************************************
                **************************************************************
```

CHART TITLE - PROCEDURES

MARKINGS INDICATE HOW TO READ FLOWCHART.

EVERY IBM FORTRAN CARD IS RE-PRINTED IN THE FLOW CHART, BUT IT MAY BE SLIGHTLY DISGUISED.



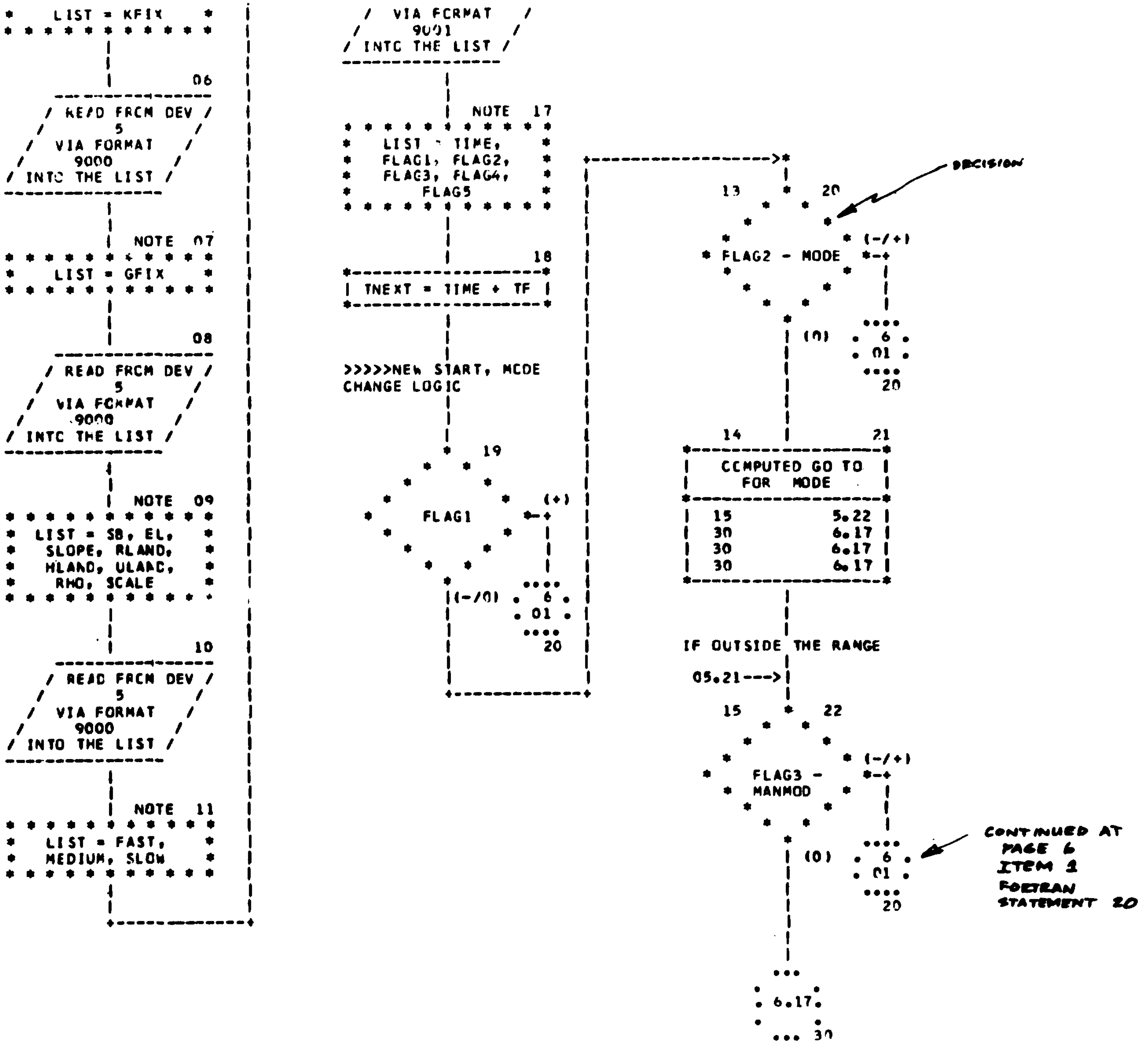
STATEMENT NO. FOR CONTINUE CARD
1
THIS BOX CONTAINS A "NOTE"
ITEM #1 ON THIS PAGE.
NOTE 01
CONTINUE
>>>>>FLIGHT CONTROL SYSTEM EXECUTIVE ROUTINE<<<<<
COMMENT CARD
02
COMPUTED GO TO FOR ITURN
COMPUTED GO TO
IF "ITURN"=1 GO TO STATEMENT 103
IF ITURN=2 GO TO STATEMENT 104
103
104
5.03
5.15
IF OUTSIDE THE RANGE
STATEMENT 103 IS LOCATED AT PAGE 5, ITEM 3
>>>>>"PAD LOAD" INITIALIZATION
05.02--->
103
NOTE 03
CONTINUE
FORTRAN STATEMENT NO. 103
04
READ FROM DEV 5 VIA FORMAT 9000 INTO THE LIST
NOTE 05
ITEM # 12 ON THIS PAGE.
12
READ FROM DEV 5 VIA FORMAT 9000 INTO THE LIST
"READ" STATEMENT
READ DEVICE #5
USE FORMAT # 9000
NOTE 13
LIST = BFIT, PFIT, FHFIT, QFIT, RFIT, THFIT, FCOMP, CCCMP, FIL
LIST OF "READ" ITEMS FOR FORMAT 9000
14
TF = .1
FLAG1 = 1
MANMOD = 3
MODE = 1
COMPUTATION CARDS
>>>>>READ CURRENT TIME & EXTERNAL FLAGWORDS; SET NEXT ENTRY TIME
05.02--->
104
NOTE 15
CONTINUE
SIGNAL FLOW FROM PAGE 5, ITEM 2.
16
READ FROM DEV 5

9.14-56

9.14.3.2 DPCS (cont'd)

PRECEDING PAGE BLANK NOT FILMED

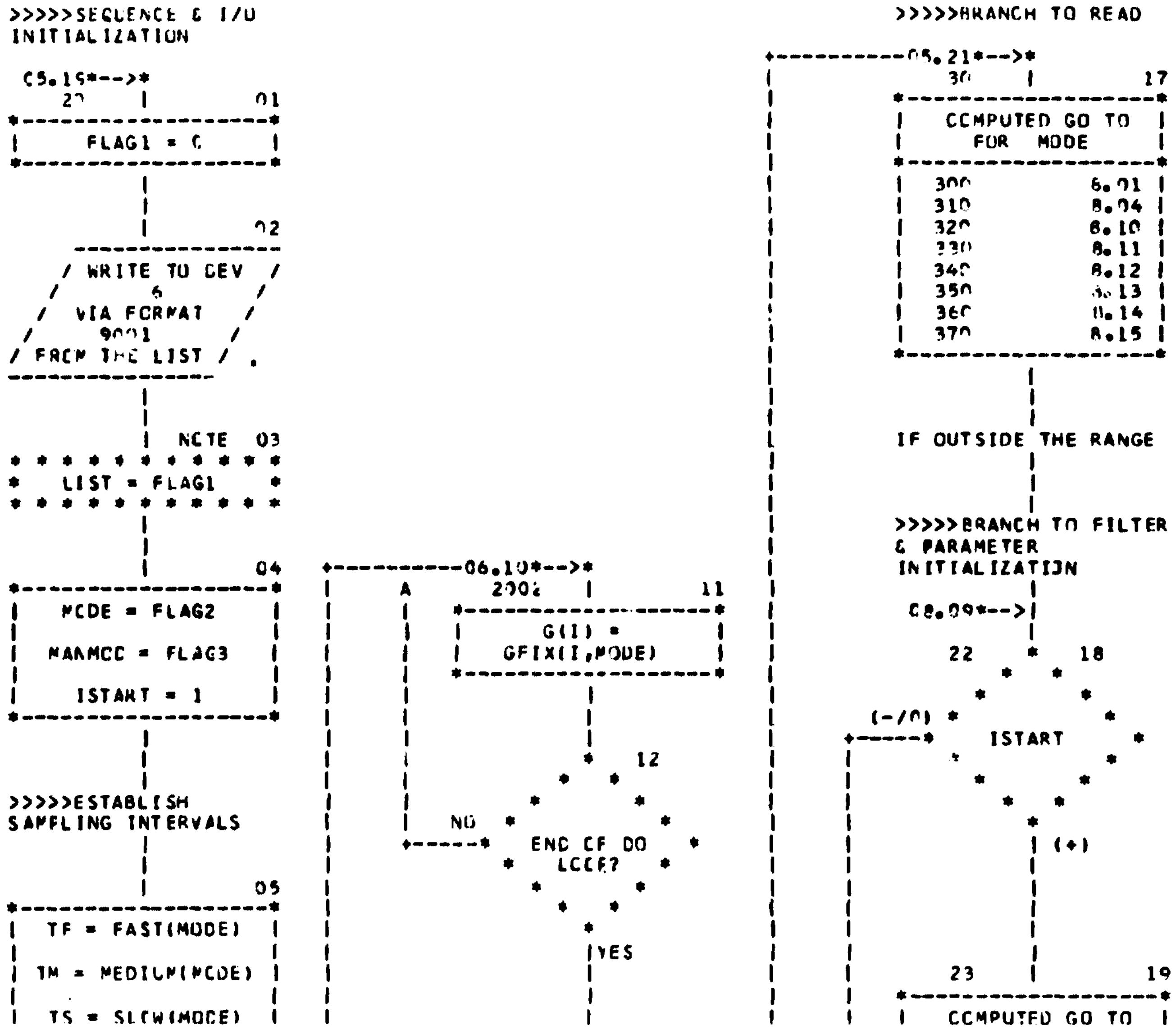
9-16-77


LIST = KFIX
06
READ FRCM DEV 5 VIA FORMAT 9000 INTO THE LIST
NOTE 07
LIST = GFIX
08
READ FRCM DEV 5 VIA FCRMAT 9000 INTC THE LIST
NOTE 09
LIST = SB, EL, SLOPE, RLAND, HLAND, ULAND, RHO, SCALE
10
READ FRCM DEV 5 VIA FORMAT 9000 INTO THE LIST
NOTE 11
LIST = FAST, MEDIUM, SLOW
VIA FCRMAT 9001 INTC THE LIST
NOTE 17
LIST = TIME, FLAG1, FLAG2, FLAG3, FLAG4, FLAG5
18
TNEXT = TIME + TF
>>>>>NEW START, MCDE CHANGE LOGIC
19
FLAG1
(+)
(-/0)
6 01 20
13 20
FLAG2 - MODE
(-/+)
(0)
6 01 20
14 21
CCMPUTED GO TO FOR MODE
15 5.22
30 6.17
30 6.17
30 6.17
IF OUTSIDE THE RANGE
05.21--->
15 22
FLAG3 - MANMOD
(-/+)
(0)
6 01 20
6.17 30

DECISION

CONTINUED AT PAGE 6 ITEM 1 FORTRAN STATEMENT 20

CHART TITLE - PROCEDURES



9.14.3.2 DFCS (cont'd)

```
 06.21--->*
          |
IF OUTSIDE THE RANGE
          |
          |
          |
>>>>>BRANCH TO
CONTROL LAW
          |
 10.05*-->|
   50     |            01
*----------------------*
|   COMPUTED GO TO     |
|     FOR  MODE        |
*----------------------*
|  500            7.02 |
|  510           11.04 |
|  520           12.01 |
|  530           12.02 |
|  540           12.03 |
|  550           12.04 |
|  560           12.05 |
|  570           12.06 |
*----------------------*
          |
          |
          |
IF OUTSIDE THE RANGE
          |
 07.01--->|
   500    |            02
*----------------------*
|   COMPUTED GO TO     |
|    FOR  MANMOD       |
*----------------------*
|  5000          11.02 |
|  5010          11.03 |
|  5010          11.03 |
*----------------------*
          |
          |
          |
IF OUTSIDE THE RANGE
          |
          |
          |
>>>>>BRANCH TO STATE
FILTERS-PART 2
(FILTER PUSHDOWN)
          |
```

```
                        ------------
                       /   705    /
                       ------------
                             |
*-----------------07.05*---->|
|                            |
|                            *    09
|                         *     *
|                       *         *
|          (-/0)      *             *
|         *---------*     FLAG4       *
|         |           *             *
|         |             *         *
|         |               *     *
|         |                  *
|         |                  |  (+)
|         |                  |
|         |                  |
|         |                  |
|         |                  |
|         |                 ...
|         |               .     .
|         |               .15.01.
|         |               .     .
|         |                 ... 7051
|         |
|         |
|         |
|         |             ------------
|         |            /   7053   /
|         |            ------------
|         |                  |
|         |        15.03---->|
|         |                  |            10
|         |       *----------------------*
|         |       |   COMPUTED GO TO     |
|         |       |     FOR  MODE        |
|         |       *----------------------*
|         |       |  7054           7.11 |
|         |       |  710           15.14 |
|         |       |  720           16.01 |
|         |       |  730           16.02 |
|         |       |  740           16.03 |
|         |       |  750           16.04 |
|         |       |  760           16.05 |
|         |       |  770           16.06 |
|         |       *----------------------*
|         |                  |
|         |                  |
```

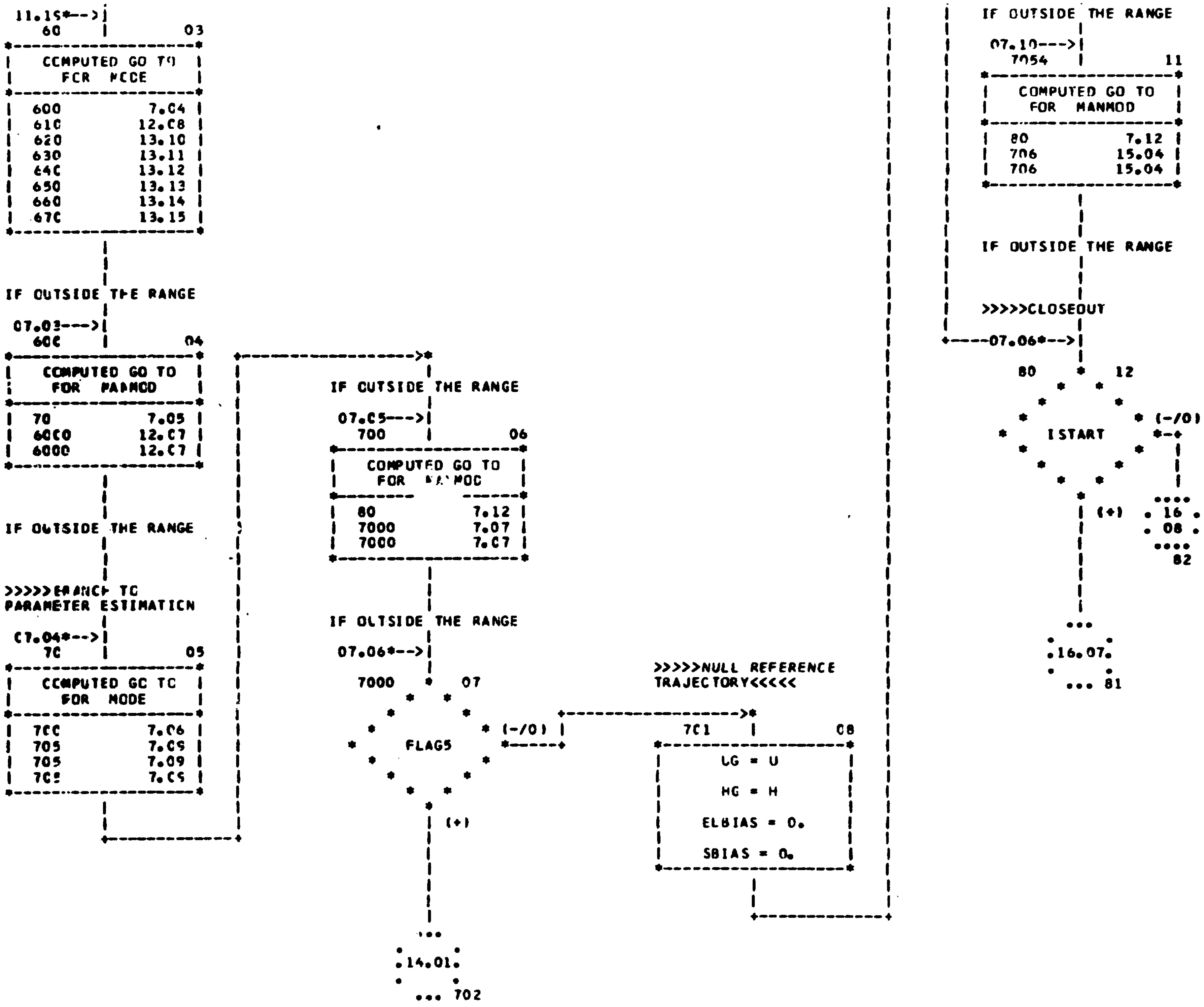

11.15*-->
60
03
CCMPUTED GO TO
FCR MCDE
600 7.C4
61C 12.C8
620 13.10
630 13.11
64C 13.12
650 13.13
660 13.14
67C 13.15
IF OUTSIDE THE RANGE
07.03--->
60C
04
COMPUTED GO TO
FOR PARMOD
70 7.05
60C0 12.C7
6000 12.C7
IF OUTSIDE THE RANGE
>>>>>BRANCH TC
PARAMETER ESTIMATICN
C7.04*-->
7C
05
CCMPUTED GC TC
FOR MODE
7CC 7.06
705 7.C9
705 7.09
7C5 7.C9
IF OUTSIDE THE RANGE
07.C5--->
700
06
COMPUTED GO TO
FOR MANMOD
80 7.12
7000 7.07
7000 7.C7
IF OUTSIDE THE RANGE
07.06*-->
7000
07
FLAGS
(-/0)
(+)
.14.01.
702
>>>>>NULL REFERENCE
TRAJECTORY<<<<<
7C1
08
LG = U
HG = H
ELBIAS = 0.
SBIAS = 0.
IF OUTSIDE THE RANGE
07.10--->
7054
11
COMPUTED GO TO
FOR MANMOD
80 7.12
706 15.04
706 15.04
IF OUTSIDE THE RANGE
>>>>>CLOSEOUT
07.06*-->
80
12
ISTART
(-/0)
(+)
. 16 .
. 08 .
82
.16.07.
81

CHART TITLE - PROCEDURES

```
>>>>>READ
ROUTINES<<<<<


>>>>>CRUISING FLIGHT
READ (MANUAL)

 06.17--->*
   300    | NOTE  01
* * * * * * * * * * *
*     CONTINUE      *
* * * * * * * * * * *
          |
          |
          |          02
     -----------------
    / READ FROM DEV /
   /       5       /
  /  VIA FORMAT   /
 /     9000      /
/ INTO THE LIST /
-----------------
          |
          |
          | NOTE  03
* * * * * * * * * * *
* LIST = CRM, CEM,  *
*    CAM, DSBM,     *
*  RUTRIM, ETRIM,   *
*   ATRIM, NI(1)    *
* * * * * * * * * * *   +------------------>|
          |             |
          |             |
          +-------------+
```

```
>>>>>CRUISING FLIGHT
READ (AUTO)

 06.17--->*
   210    | NOTE  04
* * * * * * * * * * *
*     CONTINUE      *
* * * * * * * * * * *
          |
          |
          |          05
     -----------------
    / READ FROM DEV /
   /       5       /
  /  VIA FORMAT   /
 /     9000      /
/ INTO THE LIST /
-----------------
          |
          |
          | NOTE  06
* * * * * * * * * * *
* LIST = DRG, DELG, *
*    CERG, OSBG,    *
*  ALPHG, THETAG,   *
* QG, BETAG, PSIG,  *
*   RG, PHIG, PG,   *
* THETA, PSI, PHI,  *
*      UG, HG       *
* * * * * * * * * * *
--------------->|
   211    | NOTE  07
* * * * * * * * * * *
*     CONTINUE      *
* * * * * * * * * * *
          |
          |
          |          08
     -----------------
    / READ FROM DEV /
   /       5       /
  /  VIA FORMAT   /
 /     9000      /
/ INTO THE LIST /
-----------------
          |
          |
```

```
>>>>>ORBIT TVC MODE
READ

 06.17--->*
   340    | 12
         ...
       .     .
       . 6.18.
       .     .
         ... 22




>>>>>ORBIT RCS MODE
READ

 06.17--->*
   350    | 13
         ...
       .     .
       . 6.18.
       .     .
         ... 22




>>>>>INSERTION TVC
MODE READ

 06.17--->*
   360    | 14
         ...
```

```
>>>>>INITIALIZATION
ROUTINES<<<<<


>>>>>MANUAL MODES

 06.20*-->*
   2000   |          16
*-------------------*
|      UG = U       |
|                   |
|      HG = H       |
|                   |
|     NSUM = 0.     |
|                   |
|    NO(2) = 0.     |
|                   |
|    NI(2) = 0.     |
*-------------------*
          |
          |
          |
>>>>>GENERATE PULLUP
& FLARE CONSTANTS FOR
MANUAL APPROACH
>>>>>ALTITUDE
CONSTANTS |
          |
          |          17
*-------------------*
|  SLO(1) = SLOPE   |
*-------------------*
          |
          |
          | NOTE  18
* * * * * * * * * * *
*   BEGIN DO LOOP   *
*   2010 I = 1, 2   *
* * * * * * * * * * *
          |
 09.01--->|
          |          19
*-------------------*
|      J = 2*I      |
|                   |
|   SLO(I + 1) =    |
            | NOTE  09
* * * * * * * * * * * *
*  LIST = CR, DEL,    *
*  DER, DSB, ALPH,    *
* BETA, P, Q, R, L,   *
*    F, RGO, FAIL     *
* * * * * * * * * * * *
            |
            |
            |
           ...
         .     .
         . 6.18.
         .     .
           ... 22
```

```
>>>>>TRANSITICN MCCE
READ

 06.17--->*
   320    | 10
         ...
       .     .
       . 6.19.
       .     .
         ... 22
```

```
>>>>>ENTRY MCCE READ

 06.17--->*
   330    | 11
         ...
       .     .
       . 6.18.
       .     .
         ... 22
```

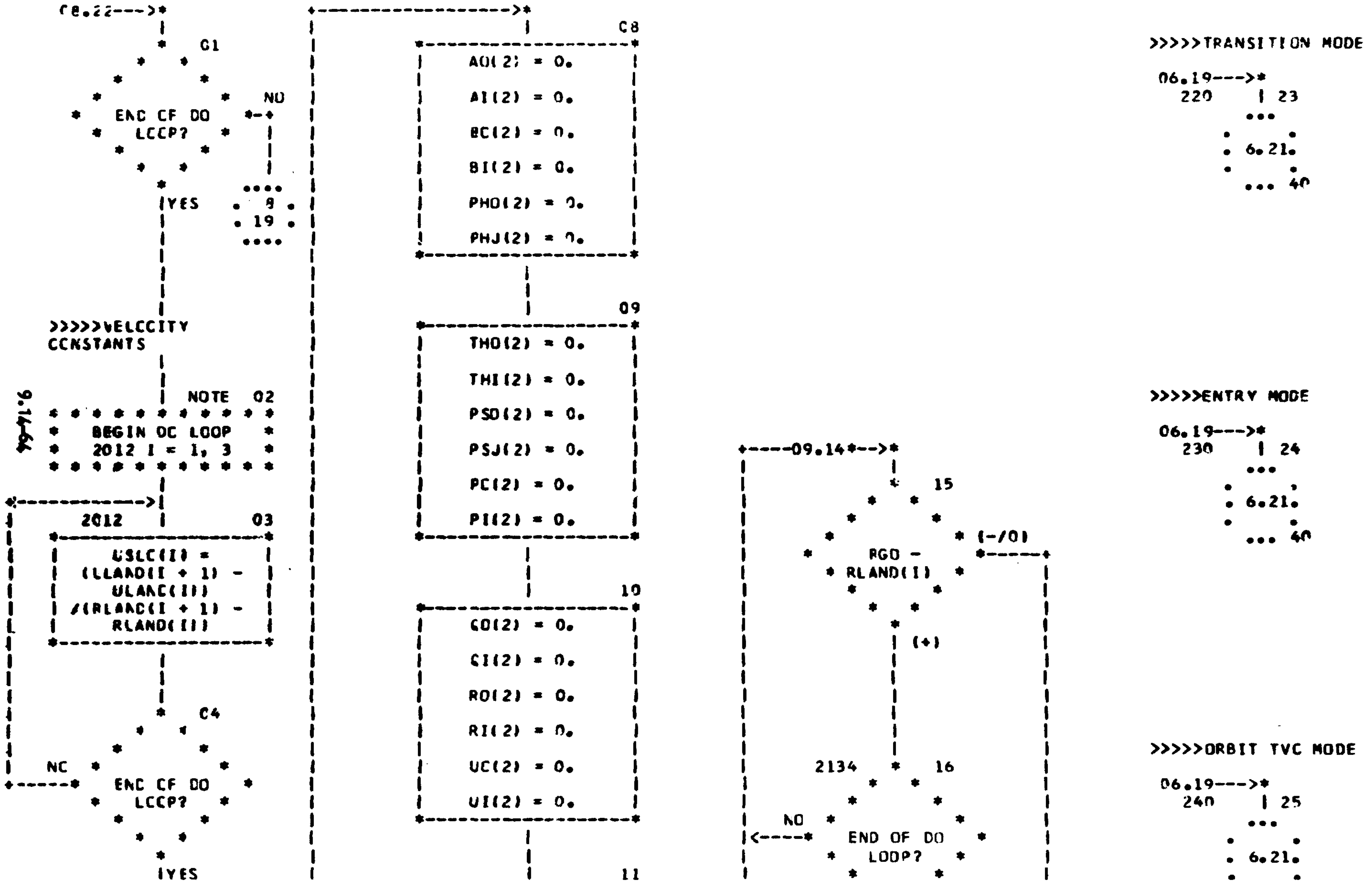
```
       .     .
       . 6.18.
       .     .
         ... 22
```

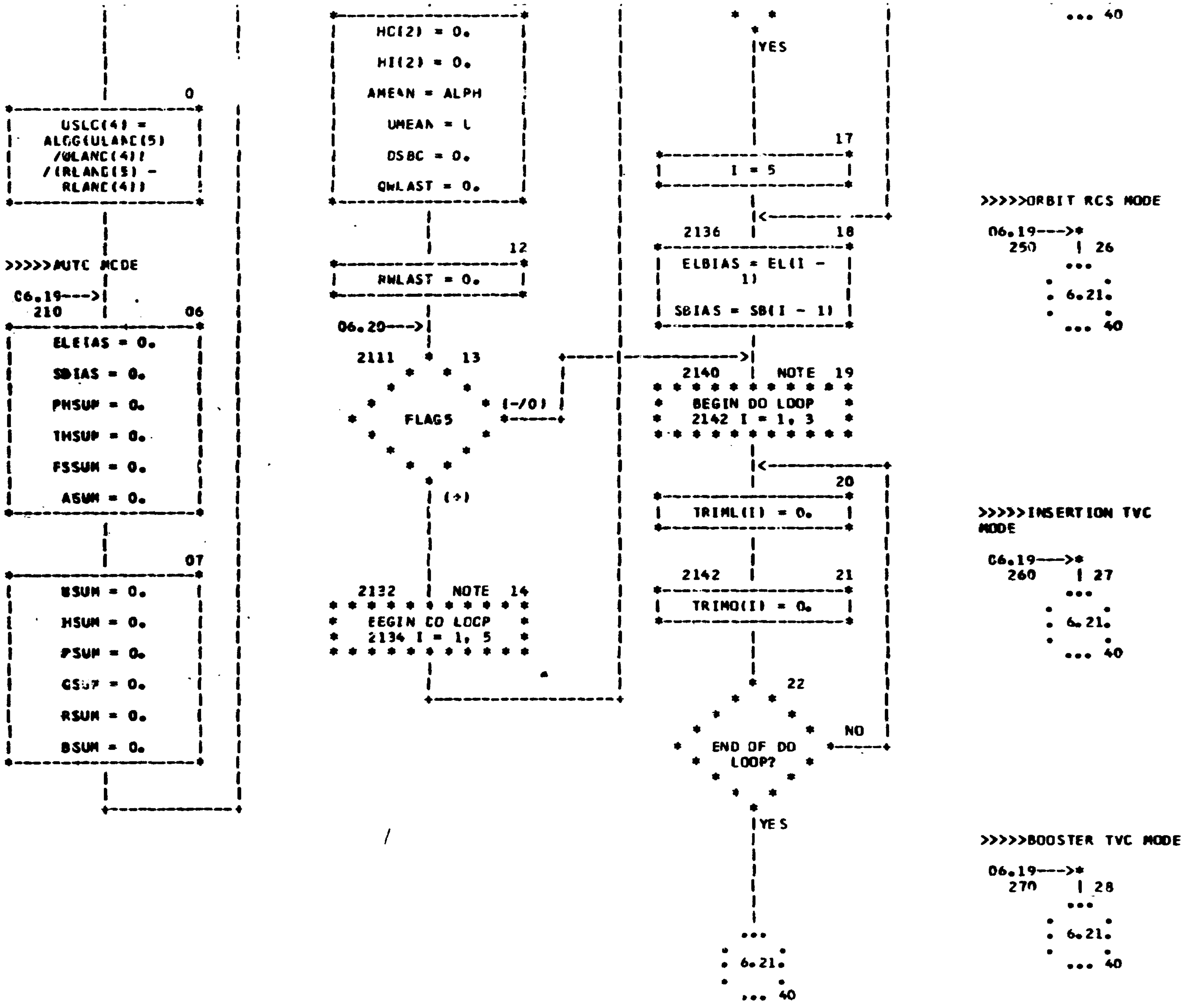
```
>>>>>BOCSTER TVC MOCE
READ

 06.17--->*
   370    | 15
         ...
       .     .
       . 6.18.
       .     .
         ... 22
```

```
|  (HLAND(J + 1) -   |
|     HLAND(J))      |
| /(RLAND(J + 1) -   |
|     RLAND(J))      |
|                    |
|     FH(1,I) =      |
|   HLAND(J - 1)     |
*--------------------*
          |
          |
          |             20
*--------------------*
| FH(2,I) = SLO(I)   |
|                    |
|  RL = RLAND(J) -   |
|   RLAND(J - 1)     |
|                    |
|    RL2 = RL*RL     |
*--------------------*
          |
          |
          |             21
*--------------------*
|  HL = HLAND(J) --  |
|    (FH(1,I) +      |
|    FH(2,I)*RL)     |
|                    |
|   HLP = SLO(I +    |
|   1) - FH(2,I)     |
|                    |
|     FH(3,I) =      |
|     (3.*HL -       |
|   RL*HLP)/RL2      |
*--------------------*
          |
          |
   2010   |             22
*--------------------*
|     FH(4,I) =      |
|     (RL*HLP -      |
|  2.*HL)/(RL2*RL)   |
*--------------------*
          |
          |
          |
          |  /
       -----
      / 9.01
```

CHART TITLE - PROCEDURES

9.14.3.2 DDCS (cont'd)





9-16-65

0
USLC(4) = ALOG(ULANC(5) /ULANC(4)) /(RLANC(5) - RLANC(4))

>>>>>AUTC MCDE
C6.19--->
210 06
ELEIAS = 0.
SBIAS = 0.
PHSUM = 0.
THSUM = 0.
FSSUM = 0.
ASUM = 0.

07
WSUM = 0.
HSUM = 0.
PSUM = 0.
GSUM = 0.
RSUM = 0.
BSUM = 0.

HC(2) = 0.
HI(2) = 0.
AMEAN = ALPH
UMEAN = U
DSBC = 0.
QWLAST = 0.

12
RWLAST = 0.

06.20--->
2111 13
FLAG5
(-/0)
(+)

2132 NOTE 14
EEGIN CO LOCP 2134 I = 1, 5

YES
17
I = 5

2136 18
ELBIAS = EL(I - 1)
SBIAS = SB(I - 1)

2140 NOTE 19
BEGIN DO LOOP 2142 I = 1, 3

20
TRIML(I) = 0.

2142 21
TRIMO(I) = 0.

22
END OF DO LOOP?
NO
YES
6.21.
40

... 40

>>>>>ORBIT RCS MODE
06.19--->
250 26
6.21.
40

>>>>>INSERTION TVC MODE
06.19--->
260 27
6.21.
40

>>>>>BOOSTER TVC MODE
06.19--->
270 28
6.21.
40

```
>>>>>FILTER                        >>>>>ALTO                                                            >>>>>TRANSITION
ROUTINES-PART 1<<<<<               FILTERS-PART 1                                                       FILTERS-PART 1

                                    06.21--->*                                                            06.21--->*
                                     410     I        07                                                    420    I 20
>>>>>MANUAL                        *-------------------*                                                          ...
FILTERS-PART 1                     I  PHJ(1) = PHIG -  I                                                        .     .
ACC TRIM DISCRETES TO              I        PHI        I                                                        . 7.01.
MANUAL COMMANDS                    I                   I                                                        .     .
                                   I THI(1) = THETAG - I                                                          ... 50
                                   I       THETA       I
 06.21--->*                        I                   I
   400     I       01              I  PSJ(1) = PSIG -  I
*-------------------*              I        PSI        I
I      NO(1) =      I              I                   I
I   G(31)*NI(1) +   I              I  AI(1) = ALPHG -  I
I       NSUM        I              I       ALPH        I          -------------->*
I                   I              *-------------------*          I      413    I        14
I      ATRIM =      I                        I                    I   *-------------------*
I    K(15)*ATRIM    I                        I                    I   I  HI(1) = HG - H   I
I                   I                        I        08          I   I                   I
I      ETRIM =      I              *-------------------*          I   I      HO(1) =      I              >>>>>ENTRY
I    K(16)*ETRIM    I              I  PI(1) = PG - P   I          I   I G(6)*HI(1) + HSUM I              FILTERS-PART 1
I                   I              I                   I          I   *-------------------*
I     RUTRIM =      I              I  QI(1) = QG - Q   I          I             I                         06.21--->*
I    K(17)*RUTRIM   I              I                   I   +------I------------>I                           430    I 21
*-------------------*              I  RI(1) = RG - R   I   I      I      414    I        15                       ...
          I                        I                   I   I      I   *-------------------*                     .     .
          I        .02             I  BI(1) = BETAG -  I   I      I   I      PO(1) =      I                     . 7.01.
*-------------------*              I       BETA        I   I      I   I    G(1)*PI(1)     I                     .     .
I      PTRIM =      I              *-------------------*   I      I   I                   I                       ... 50
I  TF2*(ATRIM +     I                        I             I      I   I      QO(1) =      I
I     TRIML(1)) +   I                        I             I      I   I    G(2)*QI(1)     I
I     TRIMO(1)      I                        I        09   I      I   I                   I
I                   I              *-------------------*   I      I   I      RO(1) =      I
I      QTRIM =      I              I      PHO(1) =     I   I      I   I    G(3)*RI(1)     I
I  TF2*(ETRIM +     I              I  G(35)*PHJ(1) +   I   I      I   I                   I
I     TRIML(2)) +   I              I       PHSUM       I   I      I   I      BO(1) =      I
I     TRIMO(2)      I              I                   I   I      I   I    G(4)*BI(1)     I
*-------------------*              I      THO(1) =     I   I      I   *-------------------*
          I                        I  G(38)*THI(1) +   I   I      I             I
          I                        I       THSUM       I   I      I             I
          I         03             I                   I   I      I             I        16              >>>>>ORBIT TVC
*-------------------*              I      PSO(1) =     I   I      I   *-------------------*              FILTERS-PART 1
I      RTRIM =      I              I  G(40)*PSJ(1) +   I   I      I   I  COMPUTED GO TO   I
I  TF2*(RUTRIM +    I              I       PSSUM       I   I      I   I     FOR  MODE     I               06.21--->*
I     TRIML(3)) +   I              *-------------------*   I      I   *-------------------*                 440    I 22
I     TRIMC(3)      I                        I             I      I   I 416         10.17 I                       ...
```

C.5

9.1467

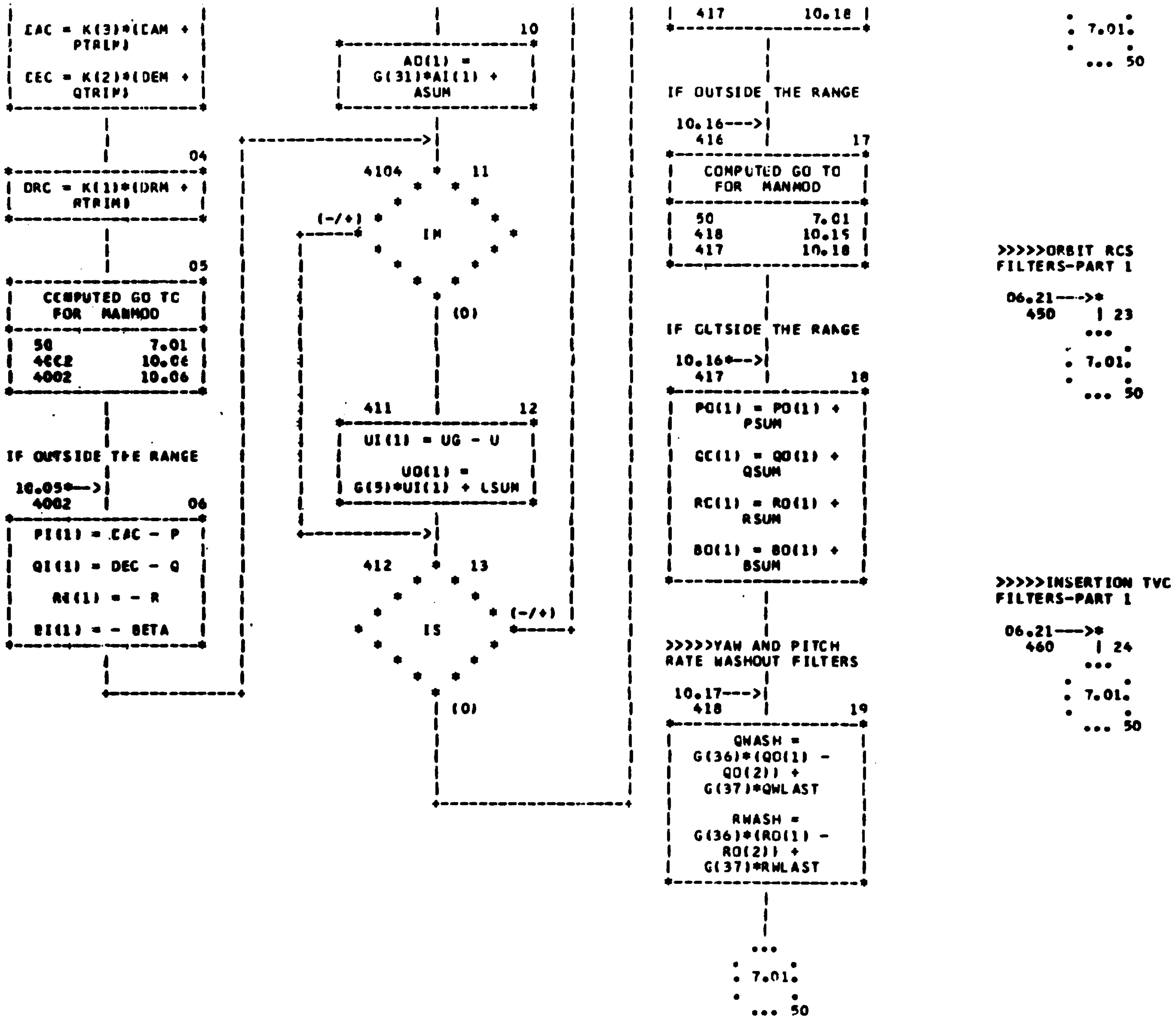
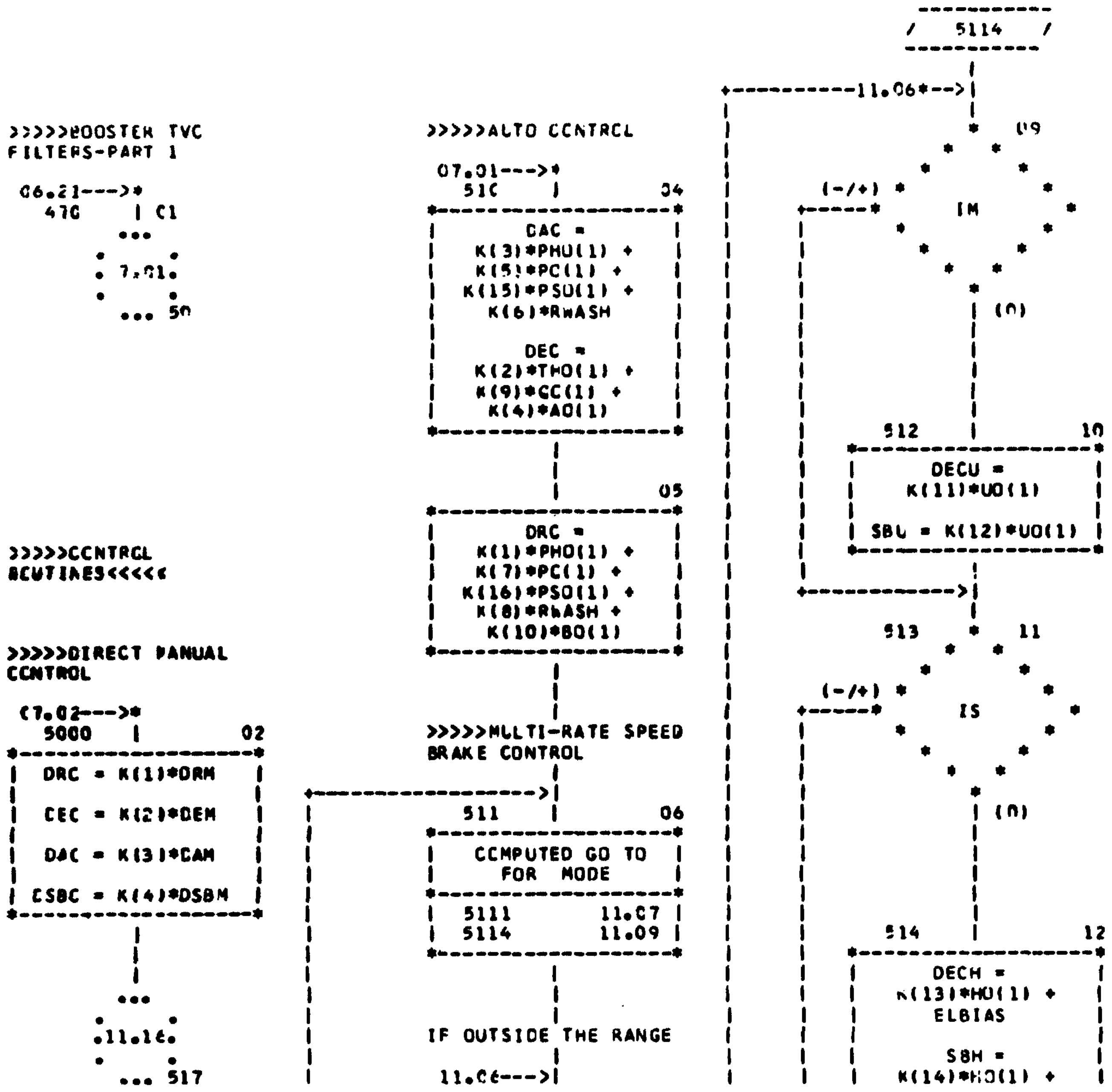
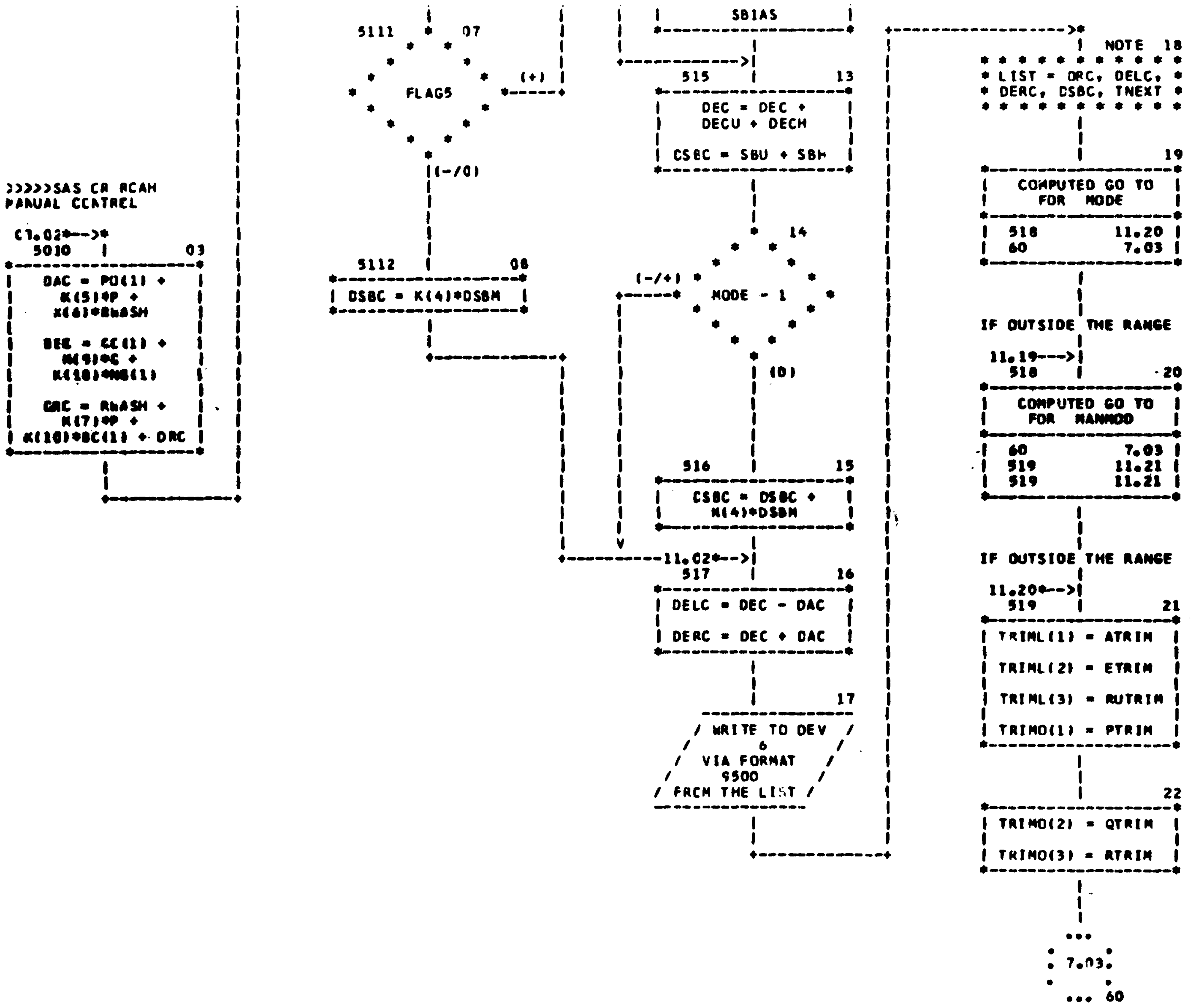

EAC = K(3)*(EAM + PTRIM)
DEC = K(2)*(DEM + QTRIM)
04
DRC = K(1)*(DRM + RTRIM)
05
COMPUTED GO TO FOR MANMOD
50 7.01
4002 10.06
4002 10.06
IF OUTSIDE THE RANGE
10.05--->
4002
06
PI(1) = EAC - P
QI(1) = DEC - Q
RI(1) = - R
BI(1) = - BETA
10
AO(1) = G(31)*AI(1) + ASUM
4104
11
(-/+)
IM
(0)
411
12
UI(1) = UG - U
UO(1) = G(5)*UI(1) + LSUM
412
13
IS
(-/+)
(0)
417 10.18
IF OUTSIDE THE RANGE
10.16--->
416
17
COMPUTED GO TO FOR MANMOD
50 7.01
418 10.15
417 10.18
IF OUTSIDE THE RANGE
10.16--->
417
18
PO(1) = PO(1) + PSUM
QO(1) = QO(1) + QSUM
RO(1) = RO(1) + RSUM
BO(1) = BO(1) + BSUM
>>>>>YAW AND PITCH RATE WASHOUT FILTERS
10.17--->
418
19
QWASH = G(36)*(QO(1) - QO(2)) + G(37)*QWLAST
RWASH = G(36)*(RO(1) - RO(2)) + G(37)*RWLAST
7.01
50
7.01
50
>>>>>ORBIT RCS FILTERS-PART 1
06.21--->
450
23
7.01
50
>>>>>INSERTION TVC FILTERS-PART 1
06.21--->
460
24
7.01
50

CHART TITLE - PROCEDURES

9.14.3.2 DECS (cont'd)




>>>>>SAS CR RCAH
PANUAL CCNTREL
C1.02*-->*
5010
03
DAC = PO(1) +
K(5)*P +
K(6)*RWASH
BEE = GC(1) +
K(9)*C +
K(10)*NB(1)
GRC = RWASH +
K(7)*P +
K(10)*BC(1) + DRC
5111
07
FLAGS
(+)
(-/0)
5112
08
DSBC = K(4)*DSBM
SBIAS
515
13
DEC = DEC +
DECU + DECH
CSBC = SBU + SBH
14
(-/+)
MODE - 1
(0)
516
15
CSBC = DSBC +
K(4)*DSBM
11.02*-->
517
16
DELC = DEC - DAC
DERC = DEC + DAC
17
WRITE TO DEV
6
VIA FORMAT
9500
FRCM THE LIST
NOTE 18
LIST = DRC, DELC,
DERC, DSBC, TNEXT
19
COMPUTED GO TO
FOR MODE
518 11.20
60 7.03
IF OUTSIDE THE RANGE
11.19--->
518
20
COMPUTED GO TO
FOR MANMOD
60 7.03
519 11.21
519 11.21
IF OUTSIDE THE RANGE
11.20*-->
519
21
TRIML(1) = ATRIM
TRIML(2) = ETRIM
TRIML(3) = RUTRIM
TRIMO(1) = PTRIM
22
TRIMO(2) = QTRIM
TRIMO(3) = RTRIM
7.03.
60

9-14-69

CHART TITLE - PROCEDURES

9.14.3.2 DFCS (cont'd)

```
>>>>>TRANSITION                   >>>>>BOOSTER TVC                  >>>>>AUTO
CONTROL                           CONTROL                           FILTERS-PART 2

 C7.01--->*                        C7.01--->*                        07.03--->*
   520    | 01                       570    | 06                       610    |         08
         ...                                ...                     *--------------------*
       .     .                            .     .                   |      PHSUM =       |
       . 7.03.                            . 7.03.                   |  G(45)*PHJ(1) +    |
       .     .                            .     .                   |  G(51)*PHJ(2) +    |
         ... 60                             ... 60                  |  G(42)*PHO(1) +    |
                                                                    |   G(48)*PHO(2)     |
                                                                    *--------------------*
                                                                              |
                                                                              |
                                                                              |         09
                                                                    *--------------------*
                                                                    |      THSUM =       |
                                                                    |  G(44)*THI(1) +    |
                                                                    |  G(50)*THI(2) +    |
                                                                    |  G(41)*THO(1) +    |
                                                                    |   G(47)*THO(2)     |
>>>>>ENTRY CONTROL                >>>>>FILTER                       *--------------------*
                                  ROUTINES-PART 2<<<<<                        |
 07.01--->*                                                                   |
   530    | 02                                                                |         10
         ...                                                        *--------------------*
       .     .                    >>>>>MANUAL                       |      PSSUM =       |
       . 7.03.                    FILTERS-PART 2                    |  G(46)*PSJ(1) +    |
       .     .                                                      |  G(52)*PSJ(2) +    |
         ... 60                    C7.04*-->*                       |  G(43)*PSO(1) +    |
                                     6000   |         07            |   G(49)*PSO(2)     |
                                  *--------------------*            |                    |
                                  |      NSUM =        |            |      ASUM =        |
                                  |  G(33)*NI(1) +     |            |   G(33)*AI(1) +    |
                                  |  G(35)*NI(2) +     |            |   G(35)*AI(2) +    |
                                  |  G(32)*NO(1) +     |            |   G(32)*AO(1) +    |
                                  |   G(34)*NO(2)      |            |    G(34)*AO(2)     |
                                  |                    |            *--------------------*
                                  |  NI(2) = NI(1)     |                      |
                                  |                    |                      |
                                  |  NO(2) = NO(1)     |                      |         11
>>>>>ORBIT TVC                    *--------------------*            *--------------------*
CONTROL                                     |                       |  PHJ(2) = PHJ(1)   |
                                            |                       |                    |
 C7.01--->*                                 +--------------+        |  PHO(2) = PHO(1)   |
   540    | 03                                             |        |                    |
         ...                                               |        |  THI(2) = THI(1)   |
       .     .                                             |        |                    |
```

9-14-72

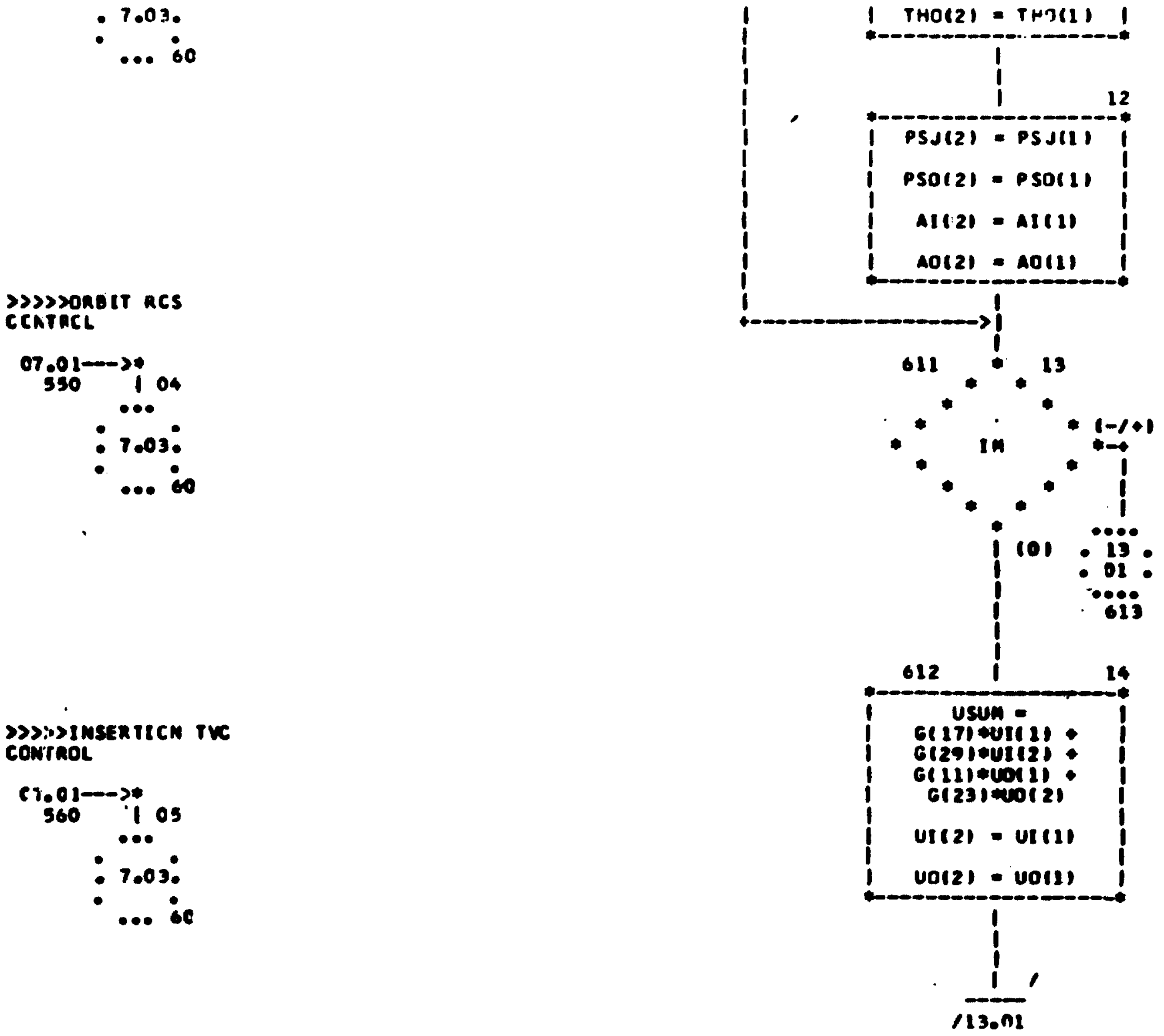
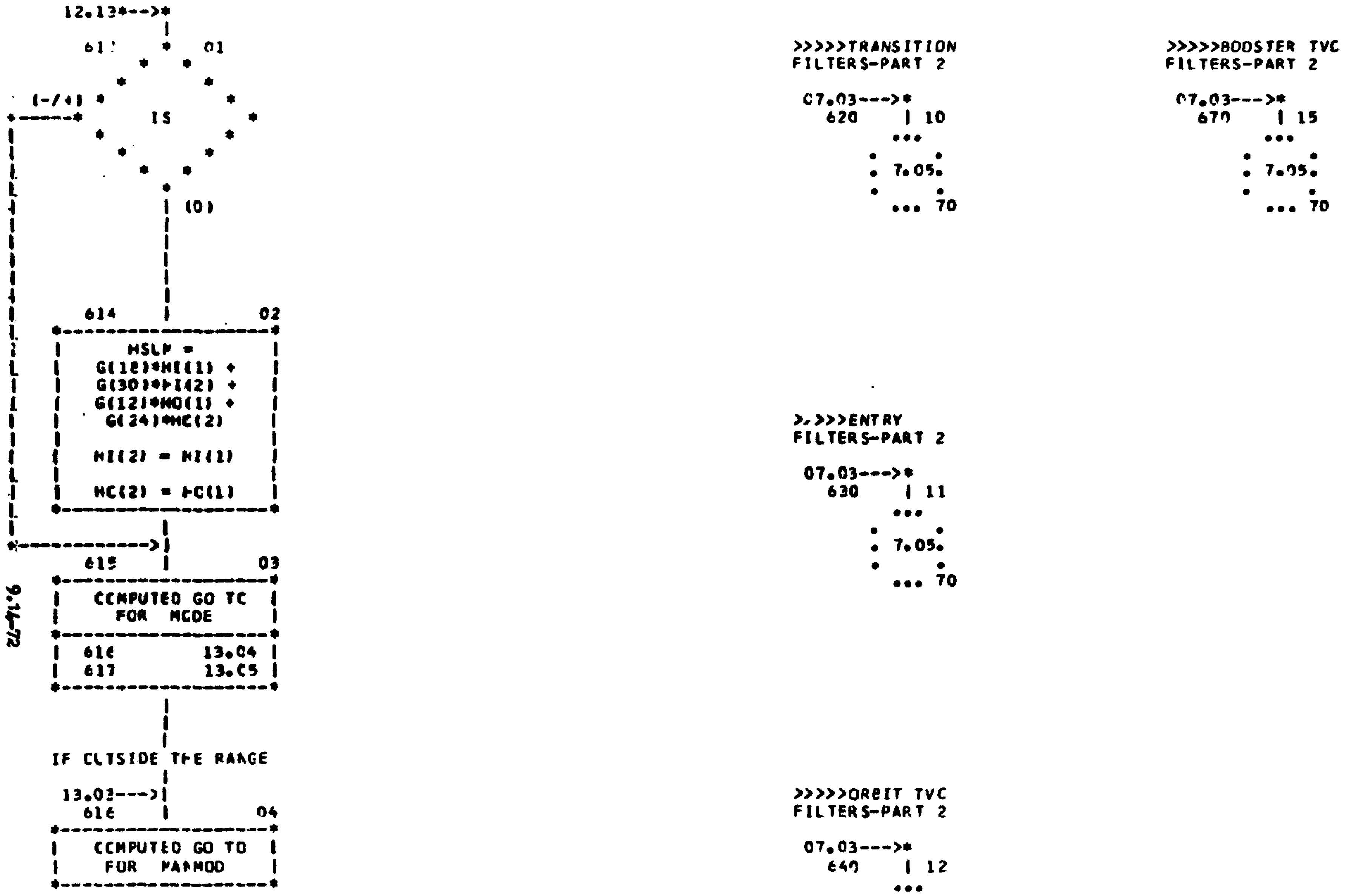

. 7.03.
... 60
>>>>>ORBIT RCS
CCNTRCL
07.01--->*
550
| 04
...
. 7.03.
... 60
>>>>>INSERTICN TVC
CONTROL
C1.01--->*
560
| 05
...
. 7.03.
... 60
THO(2) = TH?(1)
12
PSJ(2) = PSJ(1)
PSO(2) = PSO(1)
AI(2) = AI(1)
AO(2) = AO(1)
611
13
IM
(-/+)
(0)
13
01
613
612
14
USUM =
G(17)*UI(1) +
G(29)*UI(2) +
G(11)*UO(1) +
G(23)*UO(2)
UI(2) = UI(1)
UO(2) = UO(1)
/13.01

CHART TITLE - PROCEDURES

9.14.3.2 DFCS (cont'd)



```
| 70            7.05 |
| 618          13.09 |
! 617          13.C5 |
*--------------------*
          |
          |
          |
IF CLTSIDE THE RANGE
          |
 13.03*-->|
  617     |         05
*--------------------*
|       PSUM =       |
|   G(13)*FI(1) +    |
|   G(25)*PI(2) +    |
|   G(7)*FO(1) +     |
|    G(19)*PC(2)     |
|                    |
|       CSUM =       |
|   G(14)*CI(1) +    |
|   G(26)*QI(2) +    |
|   G(8)*CO(1) +     |
|    G(20)*GE(2)     |
*--------------------*
          |
          |
          |         06
*--------------------*
|       RSUM =       |
|   G(15)*RI(1) +    |
|   G(27)*BI(2) +    |
|   G(9)*RC(1) +     |
|    G(21)*RC(2)     |
|                    |
|   PI(2) = FI(1)    |
|                    |              *------------------>*
|   FC(2) = EC(1)    |              |                   |         08
*--------------------*              |        *--------------------*
          |                         |        |       BSUM =       |
          |                         |        |   G(16)*BI(1) +    |
          |         07              |        |   G(28)*BI(2) +    |
*--------------------*              |        |   G(10)*BO(1) +    |
|   QI(2) = CI(1)    |              |        |    G(22)*BO(2)     |
|                    |              |        |                    |
|   CC(2) = EO(1)    |              |        |   BI(2) = BI(1)    |
|                    |              |        |                    |
|   RI(2) = RI(1)    |              |        |   BC(2) = BO(1)    |
|                    |              |        *--------------------*
|   RC(2) = RO(1)    |              |                   |
*--------------------*              |         13.C4---->|
          |                         |          618      |         09
          |                         |        *--------------------*
          *-------------------------*        |  QWLAST = QWASH    |
                                             |                    |
                                             |  RWLAST = RWASH    |
                                             *--------------------*
                                                        |
                                                        |
                                                        |
                                                       ...
                                                     .     .
                                                     . 7.05.
                                                     .     .
                                                       ... 70
```

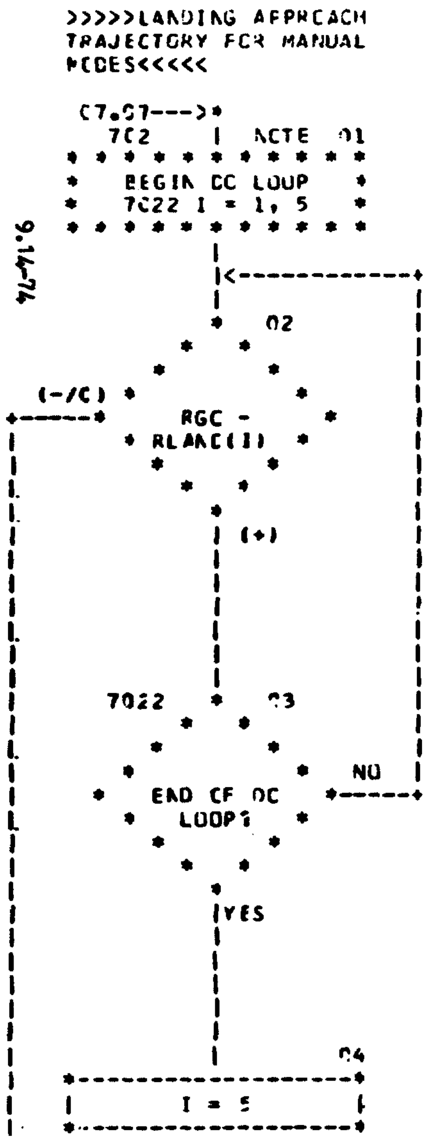
```
  .     .
  . 7.05.
  .     .
    ... 70


>>>>>ORBIT RCS
FILTERS-PART 2

 C7.03--->*
   650    | 13
         ...
       .     .
       . 7.05.
       .     .
         ... 70


>>>>>INSERTION TVC
FILTERS-PART 2

 07.03--->*
   660    | 14
         ...
       .     .
       . 7.05.
       .     .
         ... 70
```

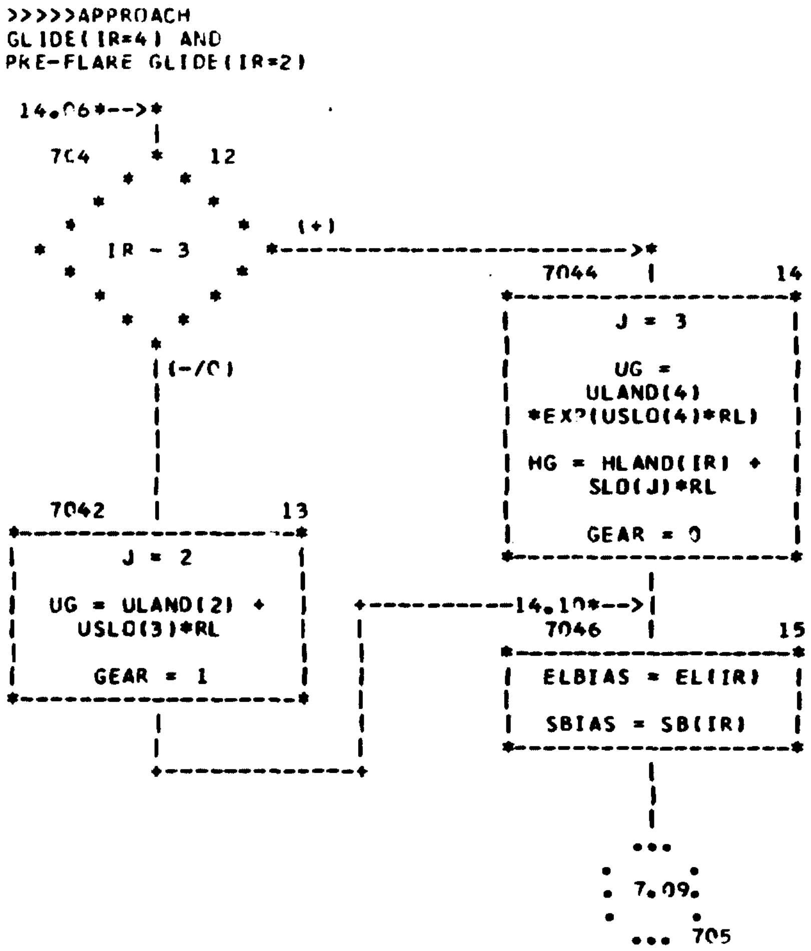
9-14-73

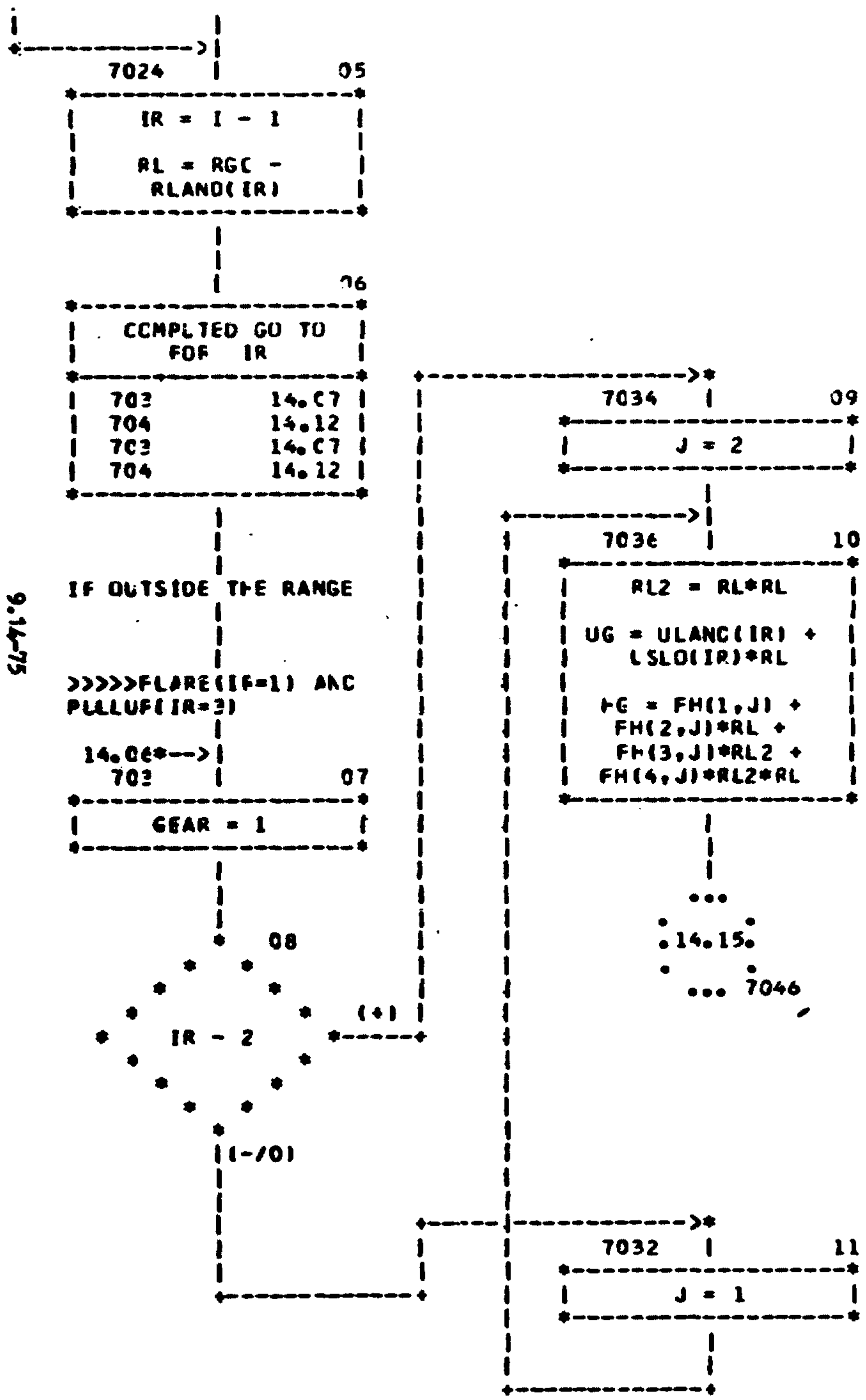
CHART TITLE - PROCEDURES

9.14.3.2 DPCS (cont'd)

9.14-74






7024
05
IR = I - 1
RL = RGC - RLAND(IR)
06
COMPUTED GO TO FOR IR
703 14.07
704 14.12
703 14.07
704 14.12
IF OUTSIDE THE RANGE
>>>>>FLARE(IF=1) AND PULLUP(IR=3)
14.06*-->
703
07
GEAR = 1
08
IR - 2
(+)
(-/0)
7034
09
J = 2
7036
10
RL2 = RL*RL
UG = ULAND(IR) + USLO(IR)*RL
FG = FH(1,J) + FH(2,J)*RL + FH(3,J)*RL2 + FH(4,J)*RL2*RL
14.15.
7046
7032
11
J = 1

9.14-75

CHART TITLE - PROCEDURES

9.14.3.2 DFCS (cont'd)

```
>>>>>PARAMETER UPDATE
ROUTINES<<<<<

 C7.05--->*
   7051    |          01
*--------------------*
|  AMEAN = AMEAN +   |
|    .05*(ALPH -     |
|      AMEAN)        |
|                    |
|  UMEAN = UMEAN +   |
|  .05*(U - UMEAN)   |
*--------------------*
           |
           |
           |
           *    02
        *     *
     *           *
   *               *  (-/+)
 *        IS         *-*
   *               *   |
     *           *     |
        *     *        |
           *          ....
           | (0)    .  7 .
           |        . 12 .
           |          ....
           |            8C
           |
   7052    |          03
*--------------------*
|      PTEMP =       |
| RHC*EXP(-SCALE*H)  |
|                    |
|      RHCV =        |
|    PTEMP*UMEAN     |
|                    |
|      DYNP =        |
|   .5*RHCV*UMEAN    |
*--------------------*
           |
           |
           |
          ...
        .     .
        . 7.10.
```

```
>>>>>PARAMETER
ESTIMATION FOR MANUAL
MODES

 C7.11*-->*
   706     |          04
*--------------------*
|1 |                 H
|8 |     TABLE       H
|. |  (PCOMP,AMEAN,  H
|0 |    PTEMP,4)     H
|1 |                 H
*--------------------*
           |
           |
           |          05
*--------------------*
|   G(1) = PTEMP     |
|                    |
|     G(13) = -      |
|   PTEMP*FIL(1)     |
*--------------------*
           |
           |
           |          06
*--------------------*
|1 |                 H
|8 |     TABLE       H
|. |  (QCOMP,DYNP,   H
|0 |    PTEMP,3)     H
|1 |                 H
*--------------------*
           |
           |
           |          07
*--------------------*
|   G(2) = PTEMP     |
|                    |
|     G(14) = -      |
|   PTEMP*FIL(2)     |
*--------------------*
           |
           |
           |          08
*--------------------*
|1 |                 H
|8 |     TABLE       H
|. |  (PFIT,AMEAN,   H
```

```
>>>>>PARAMETER
ESTIMATION FOR AUTO
MODE

 C7.10--->*
   710     |          14
*--------------------*
|1 |                 H
|8 |     TABLE       H
|. |  (THFIT,DYNP,   H
|0 |    PTEMP,5)     H
|1 |                 H
*--------------------*
           |
           |
           |          15
*--------------------*
|1 |                 H
|8 |     TABLE       H
|. |  (CCOMP,DYNP,   H
|0 |    PTEMP1,3)    H
|1 |                 H
*--------------------*
           |
           |
           |          16
*--------------------*
|      G(38) =       |
|   PTEMP*PTEMP1     |
|                    |
|      G(50) =       |
|  G(38)*(FIL(2) -   |
|        1.)         |
*--------------------*
           |
           |
           |          17
*--------------------*
|1 |                 H
|8 |     TABLE       H
|. |  (QFIT,AMEAN,   H
|0 |    PTEMP,3)     H
|1 |                 H
*--------------------*
           |
           |
           |          18
*--------------------*
```

```
 .    .
 ... 7053
```

9-14-77

```
|0 |    PTEMP,3)    H
|1 |                H
*-------------------*
          |
          |
          |        09
*-------------------*
|  K(5)  = PTEMP    |
*-------------------*
          |
          |
          |        10
*-------------------*
|1 |                H
|8 |    TABLE       H
|. |  (QFIT,AMEAN,  H
|0 |    PTEMP,3)    H
|1 |                H
*-------------------*
          |
          |
          |        11
*-------------------*
|  K(9)  = PTEMP    |
*-------------------*
          |
          |
          |        12
*-------------------*
|1 |                H
|8 |    TABLE       H
|. |  (RFIT,DYNP,   H
|0 |    FTEMP,4)    H
|1 |                H
*-------------------*
          |
          |
          |        13
*-------------------*
|  G(3)  = PTEMP    |
*-------------------*
          |
          |
          |
         ...
       .     .
       .15.27.
       .     .
         ... 712
```

```
|  G(2) = PTEMP1 -  |
|      PTEMP        |
|                   |
|     G(26) =       |
|  PTEMP1*FIL(2) -  |
|      G(2)         |
*-------------------*
          |
          |
          |        19
*-------------------*
|1 |                H
|8 |    TABLE       H
|. |  (PHFIT,RHOV,  H
|0 |    PTEMP,3)    H
|1 |                H
*-------------------*
          |
          |
          |        20
*-------------------*
|1 |                H
|3 |    TABLE       H
|. |  (PCOMP,AMEAN, H
|0 |    PTEMP1,4)   H
|1 |                H
*-------------------*
          |
          |
          |        21
*-------------------*
|     G(39) =       |
|   FTEMP*PTEMP1    |
|                   |
|     G(45) =       |
|  G(39)^(FIL(1) -  |
|       1.)         |
*-------------------*
          |
          |
          |        22
*-------------------*
|1 |                H
|8 |    TABLE       H
|. |  (PFIT,AMEAN,  H
|0 |    PTEMP,3)    H
|1 |                H
*-------------------*
          |
          |
          *----------------*
```

```
*------------------->*
                     |        24
           *-------------------*
           |  G(1) = PTEMP1 -  |
           |      PTEMP        |
           |                   |
           |     G(13) =       |
           |  PTEMP1*FIL(1) -  |
           |      G(1)         |
           *-------------------*
                     |
                     |
                     |        25
           *-------------------*
           |1 |                H
           |8 |    TABLE       H
           |. |  (RFIT,DYNP,   H
           |0 |    PTEMP,4)    H
           |1 |                H
           *-------------------*
                     |
                     |
                     |        26
           *-------------------*
           |  K(8)  = PTEMP    |
           *-------------------*
                     |
           15.13--->|
           712       |        27
           *-------------------*
           |1 |                H
           |8 |    TABLE       H
           |. |  (BFIT,DYNP,   H
           |0 |    PTEMP,4)    H
           |1 |                H
           *-------------------*
                     |
                     |
                     |        28
           *-------------------*
           |  K(10)  = PTEMP   |
           *-------------------*
                     |
                     |
                     |
                    ...
                  .     .
                  . 7.12.
                  .     .
                    ... 80
```

9.14.3.2 DFCS (cont'd)

```
CHART TITLE - PROCECURES



          >>>>>PARAMETER              >>>>>PAKAMETER
          ESTIMATICN FCR              ESTIMATION FOR
          TRANSITION MCDE             BOOSTER TVC MODE

           C7.10--->*                  07.1C--->*
             720    | C1                 770    | 06
                   ...                         ...
                 .     .                     .     .
                 . 7.12.                     . 7.12.
                 .     .                     .     .
                   ... 80                      ... 80




          >>>>>PARAMETER              >>>>>CLOSECUT<<<<<
          ESTIMATION FOR ENTRY
          MODE                         07.12--->*
                                         81     |          07
           07.1C--->*                 *-------------------*
             730    | 02              |     ISTART = 0    |
                   ...                |                   |
                 .     .              |     IS = IS + 1   |
                 . 7.12.              *-------------------*
                 .     .                        |
                   ... 80                       |
                                                |
                                      >>>>>INCREMENT
                                      INDICES
                                                |
                                       07.12--->|
                                         82     |          08
                                      *-------------------*
                                      |     IM = IM + 1   |
                                      |                   |
                                      |     IS = IS + 1   |
          >>>>>PAKAMETER              |                   |
          ESTIMATICN FCR ORBIT        | IM = MCD(IM,ITM)  |
          TVC MODE                    |                   |
                                      | IS = MCD(IS,ITS)  |
                                      *-------------------*
```

```
 :

07.10--->*
  740    | C3
        ...
       .     .
       . 7.12.
       .     .
        ... 80
```

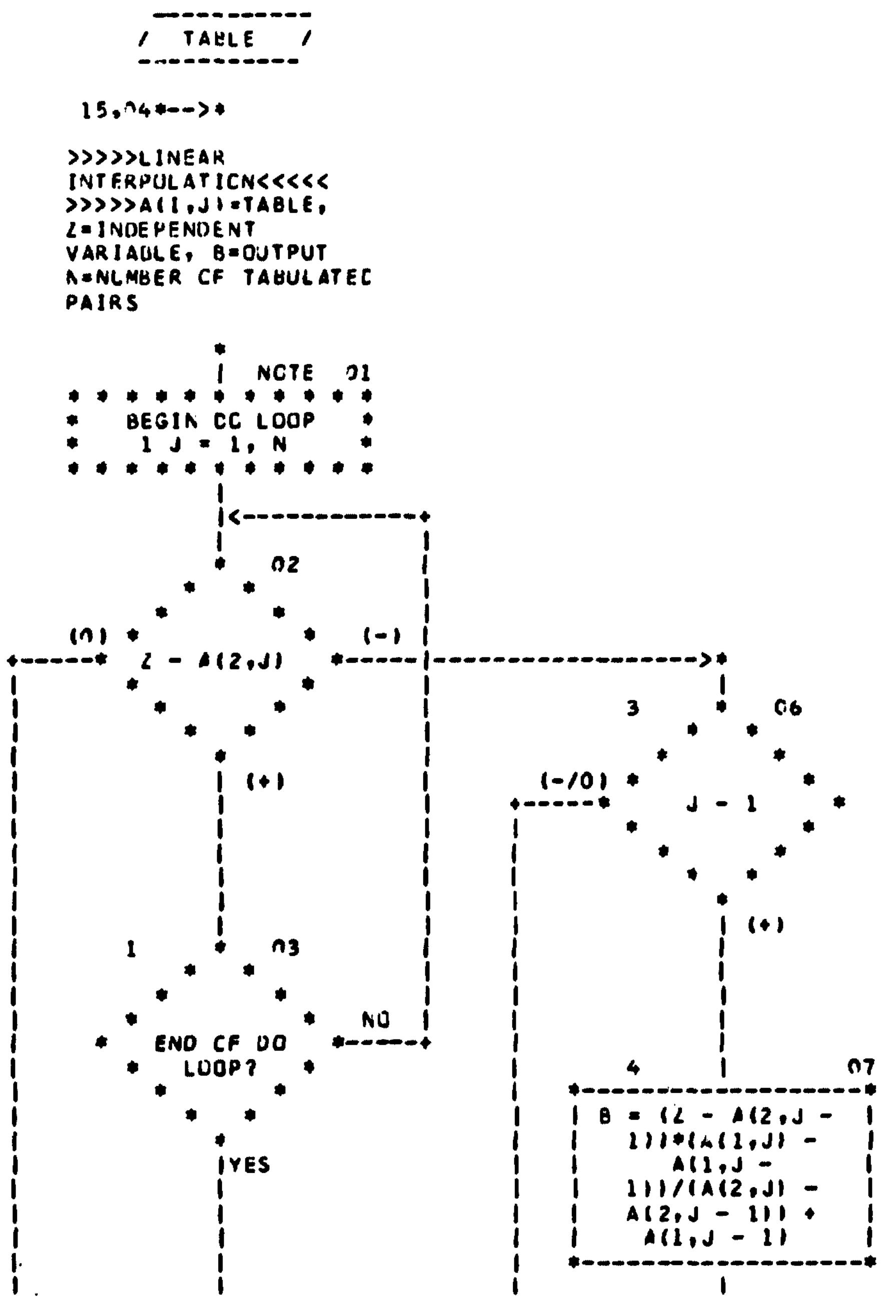
```
>>>>>PARAMETER
ESTIMATICN FCR ORBIT
RCS MODE

 C7.10--->*
   750    | 04
         ...
        .     .
        . 7.12.
        .     .
         ... 80
```

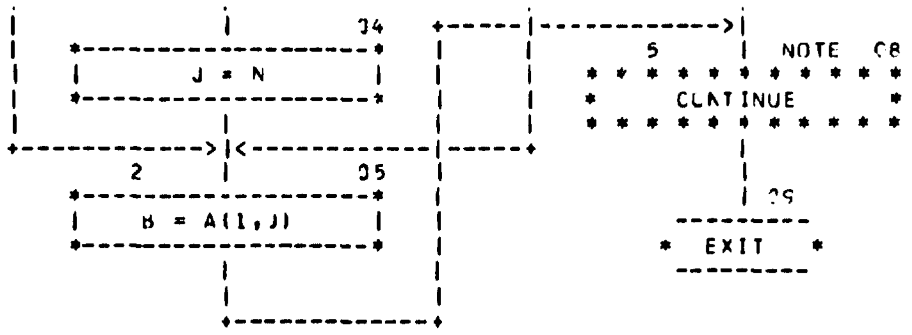
9-14-79

```
>>>>>PARAMETER
ESTIMATION FOR
INSERTION TVC MODE

 C7.10--->*
   760    | 05
         ...
        .     .
        . 7.12.
        .     .
         ... 80
```

```
             |
             |
             |
>>>>>RETURN TO GN&C
EXECUTIVE<<<<<
             |
             |
             | 09
        ---------
       *  EXIT   *
        ---------
```

CHART TITLE - SUBROUTINE TABLE(A,Z,J,N)


/ TABLE /
15,04*-->*
>>>>>LINEAR INTERPOLATION<<<<< >>>>>A(I,J)=TABLE, Z=INDEPENDENT VARIABLE, B=OUTPUT N=NUMBER OF TABULATED PAIRS
NOTE 01
BEGIN DO LOOP 1 J = 1, N
02
(0)
Z - A(2,J)
(-)
(+)
1
03
END OF DO LOOP?
NO
YES
3
06
(-/0)
J - 1
(+)
4
07
B = (Z - A(2,J - 1))*(A(1,J) - A(1,J - 1))/(A(2,J) - A(2,J - 1)) + A(1,J - 1)



9.14-81

9.15 APPROACH AND LANDING

Approach commences with the vehicle in the vicinity of the landing site and includes the manuevers performed in the process of achieving the final approach trajectory in the final approach plane. Final Approach terminates at touchdown on the runway. Landing begins at touchdown and continues until the craft reaches ground taxi speed. The GN&C software functions to be performed during this phase are:

1. Approach navigation using measurements from the IMU and the ground transponders. Both the shuttle state and the landing site state may be estimated.
2. Approach and landing guidance resulting in commands to the autopilot (attitude), thrust throttle commands, braking and/or lift flap commands.
3. Autopilot computations to compute control surface commands in order to achieve the command attitude.

9.15.1 Navigation (TBD)

9.15.2 Guidance

Submittals are included herein for Approach Guidance and Final Approach Guidance. Landing Guidance requirements have not yet been defined.

SPACE SHUTTLE

GN&C SOFTWARE EQUATION SUBMITTAL

Software Equation Section Approach Guidance Submittal No. 35

Function: Terminal Area Guidance with variable point of entry in final approach plane.

Module No. OG-6 Function No. 1,2,4,7 (MSC 03690 Rev. A)

Submitted for: T. E. Moore Co. EG6 (MSC-05121)

Date October 21, 1971

NASA Contact: J. Suddath (Name) Organization EG-2

Approved by Panel III V.T. Cox (Chairman) Date 10/21/71

Summary Description: Near optimal approach for puting vehicle satisfactorily on final approach plane from any bearing and heading within footprint of capability. Two-turn approach with nominal L/D flight path. Alternate approaches for borderline initial conditions include three-turn approach and L/D_{MAX} flight paths.

Shuttle Configuration: (Vehicle, Aero Data, Sensor, Et Cetera)

NR-161C delta wing orbiter with flight control characteristics described in Sec. 9.14.3.1

Comments:

(Design Status)

(Verification Status) Considerable test data : 6DOF Hybrid and All Digital

Panel Comments:

9.15.2.1 Approach Guidance

SUMMARY

A technique is presented for guiding an aircraft in subsonic flight (with or without power) from any bearing and heading with respect to a runway to a reference trajectory in the final approach plane (FAP), that contains the runway. This technique produces three phases: an initial turn toward the FAP, wings-level flight to the FAP, and a final turn into the FAP. Each phase is flown with a nearly constant angle-of-attack and indicated airspeed. The guidance selects the most efficient set out of four possible sets of turn direction combinations; i.e., left initial turn with right final turn, etc. The energy management parameter is the range of the FAP entry point from the runway.

The initial condition requirements or the end condition constraints imposed on the supersonic flight phase are that the vehicle be within the footprint of capability (of the order of 20-30 n mi radius from the landing site), and be at an energy state which will remain subsonic during the initial terminal area turn which is made at maximum L/D and 45 degrees bank. (The heading angle at this point need not be constrained.)

This guidance technique has been successfully flown on a six-degrees-of-freedom hybrid simulation for a straight-wing orbiter. Results from an all-digital program for both a straight and delta-wing orbiter are presented.

INTRODUCTION

The Shuttle Orbiter is presumed to have re-entered the atmosphere and received a navigation update following the communications blackout period. A "post-blackout" guidance technique which is the subject of a subsequent study will have steered the vehicle to a subsonic state (figure 1) which is considered the beginning of the terminal area mission phase.

Since the Orbiter, in its operational configuration, will not have a cruise capability, it is necessary to provide a terminal area guidance technique which makes most effective use of the available energy. Also required in this mission phase is the necessity to operate in instrument weather conditions. Because of this requirement, the guidance must be designed for ease of pilot monitoring and/or control when visual pilotage cues are not available.

The guidance technique presented herein will guide either the powered or unpowered vehicle from a high subsonic energy state in the most efficient manner (within the constraints imposed) from any position and heading with reference to a runway that is within the vehicle footprint of capability, to a reference trajectory in final approach plane. From that state, the Orbiter can complete an unpowered landing.

The design includes logic which makes it compatible with good aircraft instrument flight procedure and results in maximizing the periods of wings-level flight at nearly constant airspeed and angle-of-attack.

The objective of tnis internal note is to present the derivation of guidance equations. To demonstrate this guidance, some results of an all-digital program are presented for both a straight and delta-wing orbiter.

DISCUSSION

To guide an aircraft from any state relative to a runway to a linear reference trajectory in the FAP containing the runway, a two-turn maneuver is determined as shown on figure 2a. A constant and equal radius is used for both the initial turn toward the FAP and the final turn into the FAP. There are four possible turn direction combinations of two-turn maneuvers (figure 2b). At any guidance computation cycle, the guidance selects from these four sets that which will place the vehicle in the FAP with the greatest energy. The FAP entry point is then adjusted until this energy is the amount required to intercept the FAP reference trajectory once the vehicle is in FAP.

The two main parts of the guidance are determining the reference trajectory to the FAP, which is continually computed, and the attitude and attitude-rate commands to guide to this reference trajectory. Attitude commands might be processed more often than the reference trajectory determina.ion equations. Simulations to date have used equal computation time intervals of two seconds.

Determine Reference Trajectory

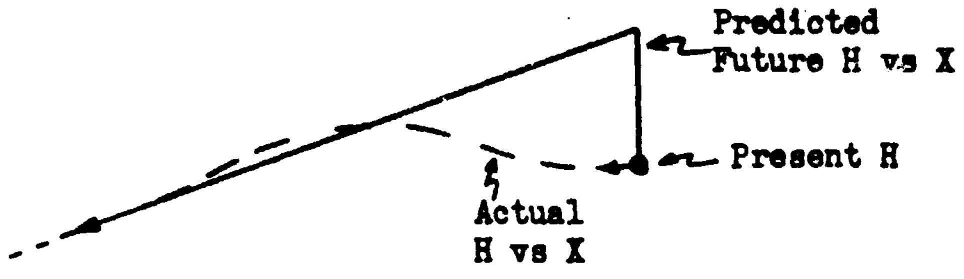
Determine ground tracks. - With the initial and the final turn maneuvers of the same radius, it is a fairly simple problem of geometry to solve for the four possible ground tracks of figure 2b. The vector equations to be solved are shown in appendix A. The answers required are the angles of the initial and of the final turns and the distances between turns for each of the four possible turn combinations.

If either of the opposite turn combinations (i.e., left initial, right final turn) has overlapping turn circles, then that particular combination is eliminated as a candidate for the reference trajectory.

Determine altitude profiles. - In planning the flight to the FAP, some margin of flight path controllability should be maintained for control against errors; i.e., wind, aero assumptions, etc. This requires a nominal angle-of-attack (α) less than that for L/D_{max}. To maintain this control margin would require the nominal α to be constant. The guidance plans each segment for a specified constant α, and assumes the vehicle will achieve a steady-state flight path (γ_{ss}). Examples of numbers used with the straight-wing orbiter are:

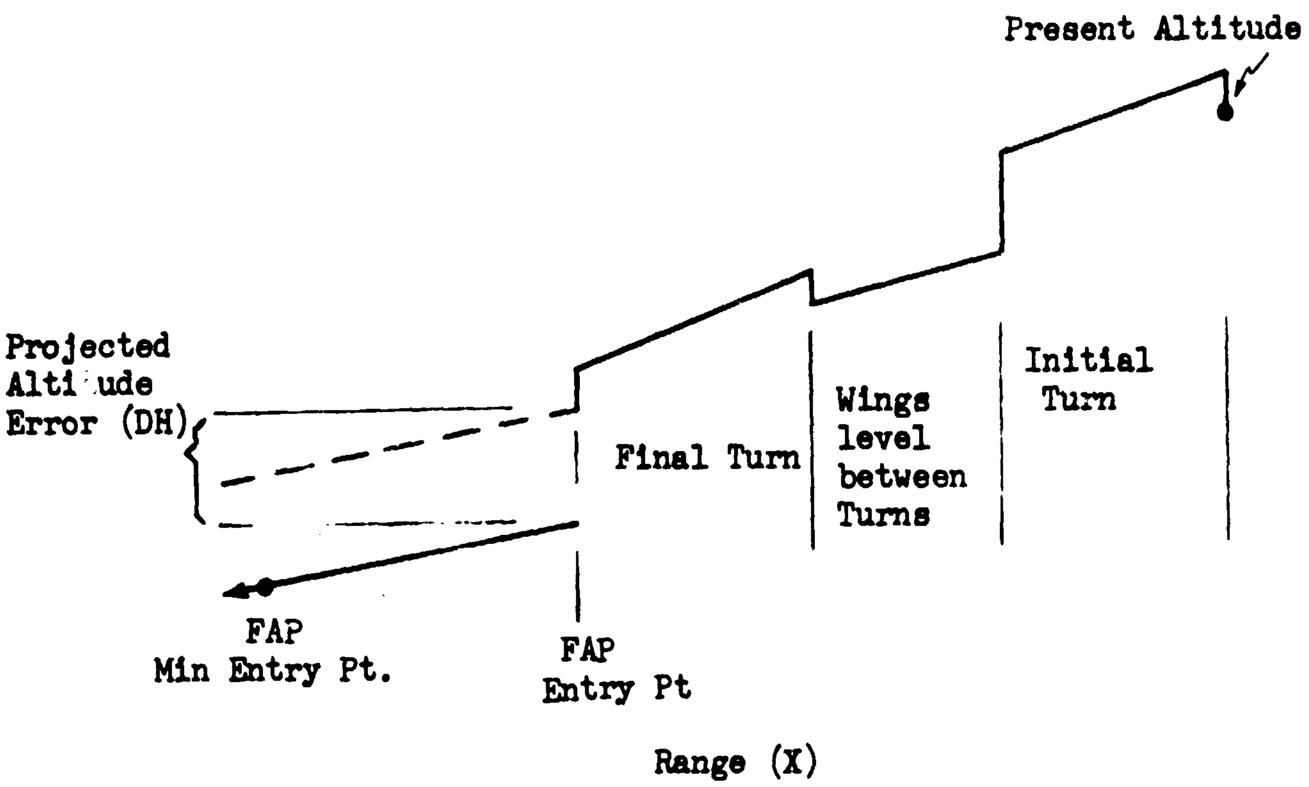
	Initial Turn	Wings-level between Turns	Final Turn	Final Approach with Landing Gear down & Speedbrakes @ 50%
α_N	7.5	4	6	4.5
γ_{ss}	8	7.2	8	10

Justification of this assumption and equations that determine magnitudes of γ_{ss} for data load into guidance computer are presented in appendix B. Equation four, which specifies the velocity required for γ to be constant, and equation seven, which predicts the trajectory with any velocity magnitude, are utilized in the guidance. For the initial turn, the ground plane distance X = (Turn Radius · Initial Turn Angle). The predicted trajectory would be



The incremented shift in predicted altitude (H) accounts for the general case wherein the existing vehicle energy state is not exactly that required to fly at the new angle-of-attack for the next segment. The guidance predicts an appropriate shift in altitude for an assumed constant γ segment. The actual trajectory, of course, approaches the predicted trajectory smoothly and reaches the same end conditions, very closely.

Equations four and seven are reprocessed with the predicted end conditions of each phase to predict the entire altitude profile:



The minimum entry point is a point in FAP which is a specified arbitrary minimum time; i.e., 60 sec, before touchdown on runway.

Select most efficient maneuvers. - Out of the four possible sets of turn combinations, the set that produces the highest projected altitude, at a given location of FAP entry point, is selected as the mosu effi-cient. The projected altitude e-ror of this set is then used to compute a new FAP entry point.

If the FAP entry point does not require adjustment, then this selected set will determine the turn direction commands.

Adjust FAP entry point. - If the most efficient projected altitude error is within a deadband range (±100 ft has been used), the entry point will not be adjusted. Otherwise, the adjustment is made according to the guidance equations derived in appendix C. The lorp control logic shown on the guidance flow diagram of figure 3 will prevent infinite looping in the guidance.

Convergence failure. - The guidance could fail to compute a satisfactory reference trajectory as shown in the logic flow of figure 3, for two reasons. If the vehicle were outside the footprint of capability with nominal flight paths, a landing might still be possible by flying the most efficient trajectory with L/D_{max} rather than nominal values. Another type of failure can occur if the vehicle were too high to fly directly to the minimum entry point without diving ($\gamma > 15^{o}$-20^{o}), and too low to fly any other two-turn maneuvers. A three-turn solution, as sketched on figure 2d, is achieved by commanding an initial turn opposite in direction to that of the most efficient trajectory to the minimum entry point, until a two-turn solution is achieved.

Reference trajectory. - The reference trajectory is basically the most efficient altitude profile previously defined. More specifically though, the guidance output at this point is merely the altitude (H_c) and rate ($\dot{H}_c$) commands for the present instant of time. During the initial turn,

$$H_c = H + \text{SHIFT}$$

$$\dot{H}_c = -V \cdot \text{Sin } \gamma_{ss}$$

where

$$\text{SHIFT} = \frac{V^2 - V_c^2}{2g},$$

i.e., the energy difference between present and desired conditions expressed as an altitude shift and where V_c is the V_c of equation (4), appendix B.

The initial turn is made at max L/D (although this is not a constraint) to conserve altitude during turn and, therefore, there is no control margin to remove residual altitude error (that error not removed by entry point adjustment, DH), but after the initial turn,

$$H_c = H + \text{SHIFT} - DH$$

$$\dot{H}_c = -V \cdot \text{Sin } \gamma_{ss}.$$

The lateral part of the reference trajectory is the ground track that goes with the most efficient altitude profile. The lateral command is by definition the actual position of the vehicle, because the ground track is defined from the present vehicle position. For the initial turn, it is necessary only to fly a constant radius turn in the commanded direction in order to maintain the same projected ground track, and for the flight between turns to fly in the commanded direction toward the final turn. There is an exception, though, for the final turn, during which time the entry point is no longer computed; i.e., remains fixed. The lateral position is then commanded to be a fixed radius from the center of the turn.

Initial turn: The lateral axis definition of the reference trajectory is chosen as the perpendicular to the true airspeed ($\underline{VA}$).

$$Y_c = Y = 0$$

$$\dot{Y}_c = \dot{Y} = 0$$

In a following section, an acceleration command, Y_c will be computed independent of Y and $\dot{Y}$ to hold a constant radius turn.

Flight between turns: The lateral axis definition of the reference trajectory is chosen as the perpendicular to the selected most efficient flight direction vector (FD) as computed in appendix A (i.e., FD = AC, or AD, etc.).

$$Y_c = Y = 0$$

$$\dot{Y}_c = 0$$

$$\dot{Y} = \underline{VA} \cdot \underline{U} \ (\underline{FD} \times \underline{LV})$$

where LV is the local vertical and U( ) means the unit of the vector within the parenthesis.

Final turn: The final turn can be flown in the same semi-open-loop manner as the initial turn. But for more precise guidance in the presence of errors, the lateral axis direction is chosen along the vector between the present vehicle position (RP) and the present location of the center of the air turn circle (TRNCNT) of radius R. During the final turn, TRNCNT is a fixed vector only if the apriori wind is zero (discussed in later section).

$$Y_c = 0$$

$$Y = R - \left| \underline{TRNCNT} - \underline{RP} \right|$$

$$\dot{Y}_c = 0$$

$$Y = \underline{VA} \cdot \underline{U} \ (\underline{TRNCNT} - \underline{RP})$$

Phase control. - Definition of the reference trajectory and vehicle position relative to it has just been shown to be phase dependent, initial turn (IT), between turns (FD, flight direction), or final turn (FT). An estimate of phase time is necessary for phase control.

From the geometry (defined in appendix A) of the most efficient ground track selected by guidance, the ground plane distances for each phase can be computed. In the process of predicting (using appendix B) the trajectory, airspeeds (horizontal component) at each phase change point have been computed.

The guidance trajectory determination assumes instantaneous attitude response at phase change points; i.e., $\phi = 35 \longrightarrow 0$. To compensate for attitude response time delay, the maneuvers are led. The phase (PH) control logic with examples for lead times are:

At first processing of Guid, PH = IT

At subsequent processing,

If PH = IT and TGO(IT) < 3, then PH = FD

If PH = FD and TGO(IT) > 5, then PH = IT

If PH = FD and TGO(FD) < 3, then PH = FT

If PH = FT and TGO(FT) < 3, then EXIT Term Area Guid;
i.e., PH = Final Approach Guid.

Wind considerations. - The guidance will compensate for wind errors in two simultaneous manners. With error compensation networks (to be discussed), the angle-of-attack will be adjusted in order to hold the reference trajectory. Also, the FAP entry point will readjust during IT or FD phases if the vehicle is blown off the predicted course.

The performance of the guidance in the presence of wind is greatly enhanced by providing the guidance with knowledge of the wind (apriori wind magnitude and direction versus altitude). The reference trajectory can be adjusted for wind in order to maintain nominal angle-of-attack, in which case, α - margin is allotted for wind uncertainty of apriori profile rather than on full magnitude of wind.

In addition to apriori wind profile, a routine is desirable which will compute the average wind over two specified variable levels of altitude.

To incorporate this wind in the guidance, appendix A requires modification of final turn circle (S) as shown on figure 2c. To keep the turn bank angles nominal, the final turn circle center is first rotated so that ground speed at end of final turn is in FAP, and second, translated against the wind by the sum of the average wind of each of three maneuvers times the time of each maneuver.

Guide to the Reference Trajectory

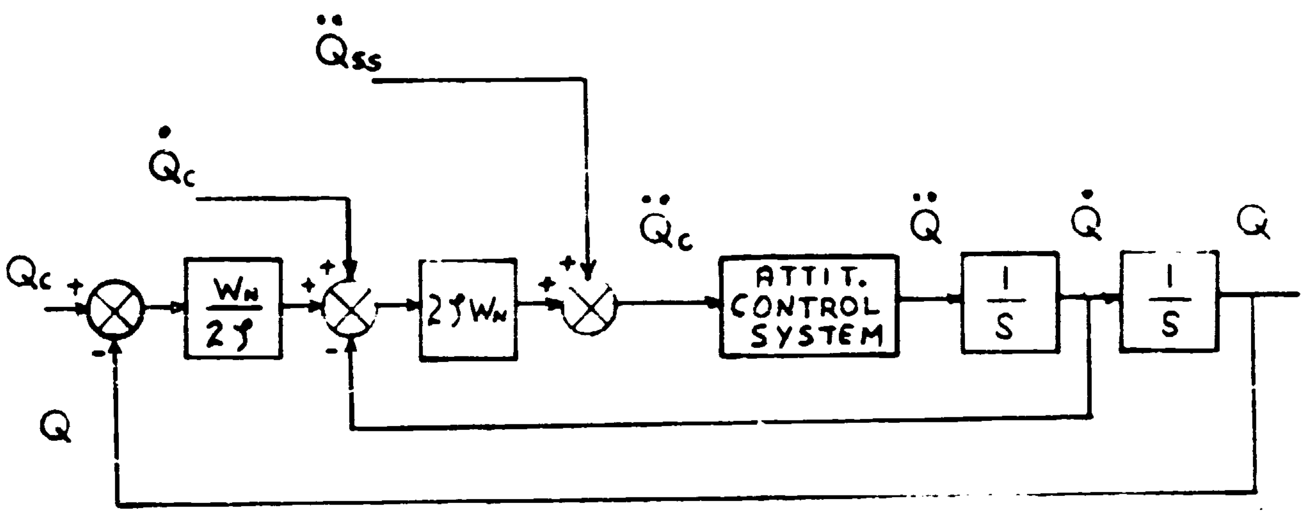
Each time the reference trajectory equations of the previous section are processed, vehicle state commands (h_c, $\dot{h}_c$, Y_c, and $\dot{Y}_c$) are generated and present vehicle state components (h, $\dot{h}$, Y and $\dot{Y}$) are defined. The task is now to guide to the commanded state. The guidance equations of this section are derived in detail and presented in appendix D. A functional description of the guidance output commands (two components of Euler attitude and three body rates), and guidance-autopilot interface follows.

Attitude commands. - Pitch (θ_c) and roll (ϕ_c) attitude commands of the standard aircraft Euler attitude sequence, ψ, θ, ϕ, are generated. An azimuth command, ψ_c, is not generated. The control requirement on ψ is that sideslip be zero relative to actual, not navigated airspeed.

To control the vehicle state, a second-order control law is u.ed,

$$\ddot{Q}_c = W_n^2(Q_c - Q) + 2\zeta W_n(\dot{Q}_c - \dot{Q}) + \ddot{Q}_{ss}$$

which corresponds to a system of the form shown:



where Q = is either h or Y

W_n = natural frequency of desired response (of the order of 10 times slower than vehicle attitude response)

ζ = damping ratio of desired response

$\ddot{Q}_{ss} = \frac{d\dot{Q}_c}{dt}$ = steady state acceleration required to maintain the two error signals at zero. For example, this term for the lateral axis during a turn would be the centripetal acceleration.

The acceleration commands $\ddot{H}_c$ and $\ddot{Y}_c$ are converted respectively to angle-of-attack α_c and bank angle ϕ_{vc} commands. Pitch θ_c and roll ϕ_c

commands are computed from these two angles and the navigated airspeed; i.e., VA = VG(Nav gnd speed) + VW (apriori wind).

b. Attitude rate commands. - The state vector acceleration commands (Q_c) are by definition variable. The vehicle can be commanded to a linearly varying acceleration state by computing body attitude rates which will produce the commanded jerk ($\dddot{Q}_c$), where an expression for jerk command is obtained by differentiating $\ddot{Q}_c$,

$$\dddot{Q}_c = W_n^2(\dot{Q}_c - \dot{Q}) + 2\zeta W_n(\ddot{Q}_{ss} - \ddot{Q}_c) + \dddot{Q}_{ss}$$

Body rate commands, P_c, q_c, and r_c, as functions of jerk command are obtained by differentiating (as done in appendix D) α_c and ϕ_{VC}.

Error compensation. - Acceleration errors can exist from errors in the aero coefficient assumptions, navigation, wind, attitude response time delays, etc. Any of these errors that exist in a form of a bias can be detected by comparing the present vehicle state with the state that should exist had the acceleration and jerk commands of the previous computation been flown. A bias error term is obtained by weighting and accumulating this measured error, compensated for expected acceleration error due to attitude time delay. This bias is then fed back into the acceleration command so that the actual acceleration will converge on the command acceleration in the presence of any bias errors.

Guidance-Autopilot interface. - The guidance can operate at a relatively large computation cycle time such as 2 sec or probably even larger. There are autopilot interface equations that must be processed more often as shown on figure 4. If the sequence of attitude of the inertial measurement unit (IMU) is not standard aircraft relative to landing site, then the three angles must be converted to the required θ and ϕ. Attitude error is then converted to body axes error. Coordinated turn rate commands must also be added to the guidance rate commands. The autopilot is shown on the figure simply as a second-order system merely to show that the guidance commands on the autopilot are attitude ramps in body axes pitch and roll. Body axis yaw must control to a rate and also for sideslip $\beta = 0$. β could be replaced with either an integral of accelerometer measurement or merely a signal proportional to bank angle.

Selection of Turn Radius

With respect to altitude lost per unit of turn angle, a 45° bank turn is most efficient. With respect to comfort though, a 30° bank turn is more desirable. A constant radius turn tends to satisfy both. For example, a 9200 ft radius for the straight-wing orbiter requires

45° bank for an initial turn at 20,000 ft altitude with max L/D, and a 30° bank for a final turn if at 4000 ft altitude with $L/D < L/D_{max}$. This same situation exists for the delta-wing orbiter with a 12,000 ft radius of turn. These numbers were used for the results that follow in another section. At initial turn altitudes greater than 20,000 ft, the bank command was limited to 45°. This results in a minor adjustment of entry point in FAP because the actual is greater than guided turn radius. The actual converges to the guided trajectory at either the completion of initial turn or altitude less than 20,000 ft, whichever occurs first.

Turn radius could be selected so that it would be flyable with 45° bank at the maximum altitude for terminal area guidance; i.e., about 40,000 ft for delta-wing. But this two to three times larger radius would give an inefficient final turn which would require of the order of 15° bank.

DEMONSTRATION RUNS

Description of Simulation

The results presented were made on a six-degrees-of-freedom 1108 computer program. This program contains aerodynamic forces as a function of angle-of-attack for a Mach number of .25. In place of aerodynamic moments, an autopilot is assumed which drives the vehicle to the command attitude and rate with a second-order response, where the natural frequency is 1 rad/sec with .707 damping. The results of this program, though, are representative of results obtained on a more detailed real-time hybrid simulation.

Guidance reference trajectory computations and attitude commands are processed every 2 sec. The autopilot and environment equations are processed every 1/16 sec. The natural frequency of the guidance commands for driving to the reference trajectory is selected to be 1/10 that of attitude response, or .1 rad/sec with .707 damping.

Phase (IT, FD, or FT) control is monitored in the fast time loop so that phase change will occur within 1/16 sec of the time computed in the slow time loop. The guidance equations are processed immediately at phase change.

This report presents only the guidance to FAP. The end condition energy state should be such that the reference trajectory in FAP will be intersected after entry into FAP. The simulation contains a complete guidance to touchdown, and the results show part of the FAP guidance in order to demonstrate the intersection with the reference teajectory. Simulation data applicable to all runs made is shown on table I.

Results

Results of runs are presented to demonstrate:

For straight-wing orbiter. -

(1) Ground tracks from various initial position relative to runway.
(2) Effects of wind errors on trajectory and control parameters.
(3) Effects of very high winds with guidance knowledge of such, on trajectory and control parameters.
(4) Altitude convergence to reference trajectory.

For delta-wing orbiter. -

(1) A nominal ground track from a maximum energy state initial condition for terminal area guidance.
(2) Same as (1) except with a guidance modification for a maximum entry point.

Initial position. - Run 1 on figure 5, which consists of two right-turn maneuvers to an entry point at 50,000 ft range from runway, is the nominal run that is used in the next sections. Run 2 demonstrates a left turn-right turn maneuver. For run 3, which is already in the FAP, the altitude high error at MEP would be greater than 100 ft if the vehicle were to proceed directly to MEP with nominal angle-of-attack. But also, the energy state is too low for any other two-turn solution to MEP; i.e., a 360° initial turn. The guidance sets the flag IOPOST to 1 when it detects this condition, and the vehicle is commanded to make an opposite turn; i.e., initial turn direction opposite in direction to the most efficient path to MEP. After approximately 110° of turn, a two-turn (right turn-left turn) solution is achieved, and the flag is then set to zero. For the above case, a possibility under study exists to modulate speed brakes and thus increase the flight path angle. This technique could also provide a satisfactory two-turn solution and minimize the necessity for a non-standard procedure.

The initial energy state for run 4 is too low for nominal angles-of-attack flight to MEP (IOPOST = -1). The vehicle is then commanded along the best path, but at max L/D. At a later point, the vehicle intersects the reference trajectory, at which point nominal angles-of-attack are flown (IOPOST = 0).

Wind errors. - Run 1 (no wind) is replotted on figure 6 along with ±40 ft/sec wind errors (zero wind assumed). This wind is 45° relative to the runway. Entry point varied between 46,000 and 52,000 ft. Altitude is shown to converge on the reference trajectory in FAP.

Control parameter variation is shown on figure 6b for the nominal and one wind error case. Angle-of-attack holds very near the reference values for each phase (see table I for comparison).

Apriori wind. - With guidance computer knowledge of the high winds, a successful descent was made as shown on figure 7. Note the altitude climbed during the initial turn, the reason being that this run started as run 1 with same ground speed, and in effect, it instantaneously encountered the wind which then increased the initial airspeed by the wind velocity. Note also the change of final turn direction from run 1. This change of direction was nct one which occurred after initiation, but was computed as the best during the first pass through the guidance, The turns no longer appear circular because the plot is relative to the ground. Relative to the moving air mass, the turns should remain as circles.

One thing the guidance does with its knowledge of wind is adjust the reference trajectory slope in FAP in order to maintain the nominal α for that phase. This slope adjustment is shown on figure 7a. The altitude is shown to converge on the reference trajectory in FAP.

Control parameter variation is shown on figure 7b. Angle-of-attack converges very near the reference values for each phase (compare with figure 6b).

Altitude convergence. - The guidance computes altitude command as a function of the potential energy equivalent of the kinetic energy difference between present vehicle state and that required to hold the reference flight path with nominal angle-of-attack. At phase changes then, there can be a discontinuity in H_c as shown on figure 8a, for run 1. Note, though, for the nearly constant α (see figure 6b), H converges to H_c (figure 8a). The discontinuity at other phase change points is small because of the choice of targeting conditions, but this discontinuity is not required to be small.

The same type plot is shown on figure 8b for the apriori wind run. Heie there are discontinuities at each phase change, but, still, the trajectory is shown to converge smoothly to the reference. The reason for these discontinuities on figure 8b and not on figure 8a is that the nominal final turn angle-of-attack is replaced with the higher initial turn α. The guidance automatically makes this targeting change whenever it estimates the altitude of the final turn to be greater than 15,000 ft (13,000 ft for delta-wing). This keeps bank angle required from exceeding the maximum in order to hold turn radius.

Nominal delta-wing orbiter. - For a constraint of Mach $\leq$ .9 during the initial turn with $\alpha = 7.5^{o}$, there is a maximum energy state from which the initial turn can be started. The runs on figure 9 are approximately from this maximum energy state. Note that even though the 12,000 ft turn radius cannot be flown at h = 41,000 ft, a smooth convergence to a solution is achieved. The entry point with the nominal guidance is 80,000 ft from the landing site.

There should not be any fear that the entry point is too far from the runway, because there is considerable margin in controllability against errors once in the FAP. For example, at the 80,000 ft point for the nominal trajectory, the landing gear goes down and speedbrakes at one-half maximum drag. And even if this were not enough control margin, then the reference trajectory slope to MEP could be made steeper—runs were made with the same slope to and from MEP. But, to demonstrate that entry point can be limited, the dashed run on figure 9 activated the opposite turn logic whenever entry point greater than 60,000 ft.

CONCLUSIONS

A guidance technique is derived which will guide an unpowered aircraft in a nearly optimum manner from any bearing and heading with respect to a runway to a reference trajectory in the final approach plane from which a landing can be made. The entry point into this plane is computed so that the aircraft can fly at nearly constant, specified angles-of-attack. Constant turn radii are utilized.

Simulations of both straight and delta-wing orbiters demonstrate the ability of the technique to guide in the presence of large known winds and of wind errors.

Subsequent study will be concerned with procedural variations within the guidance which might be desirable for use under different weather and wind environments.

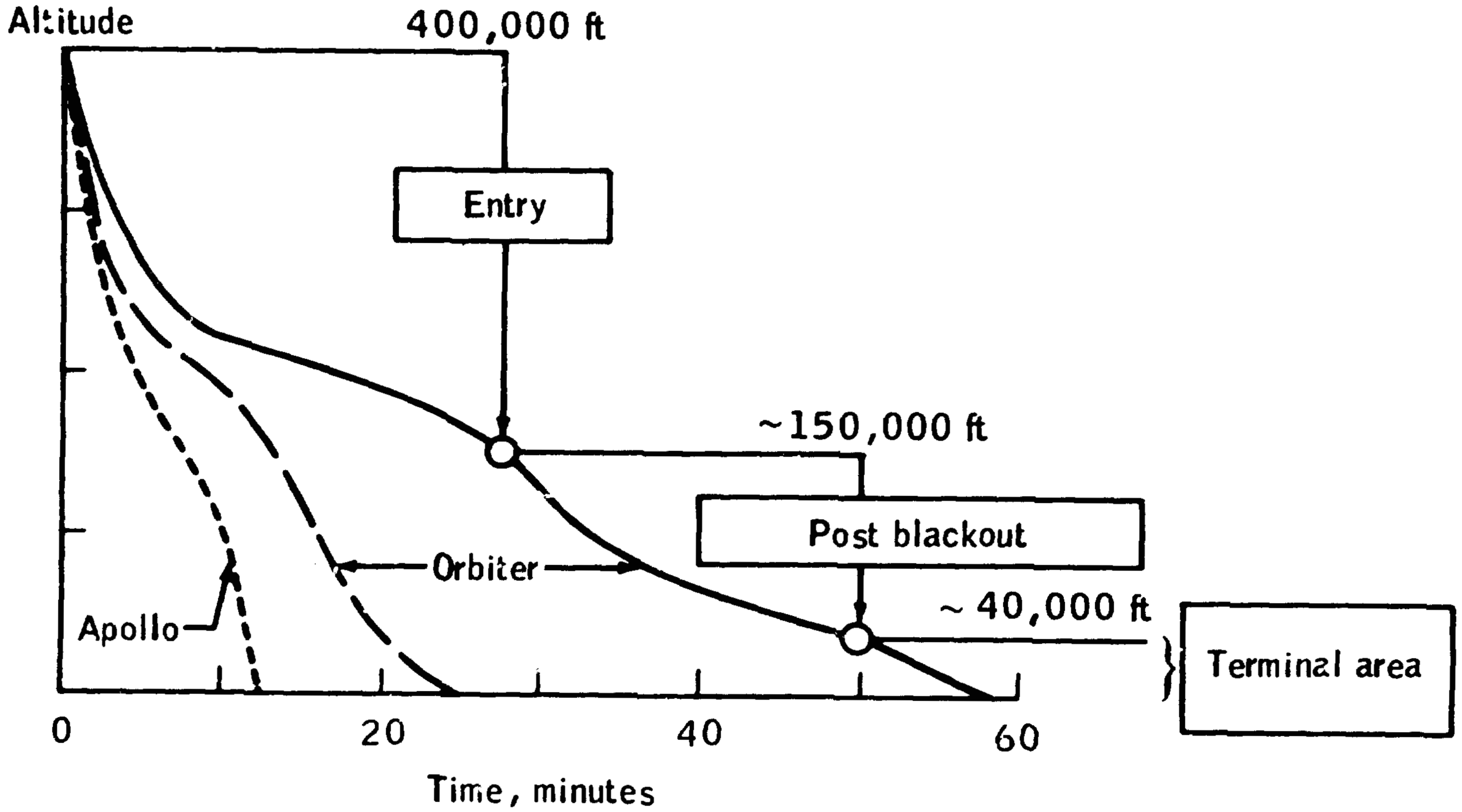
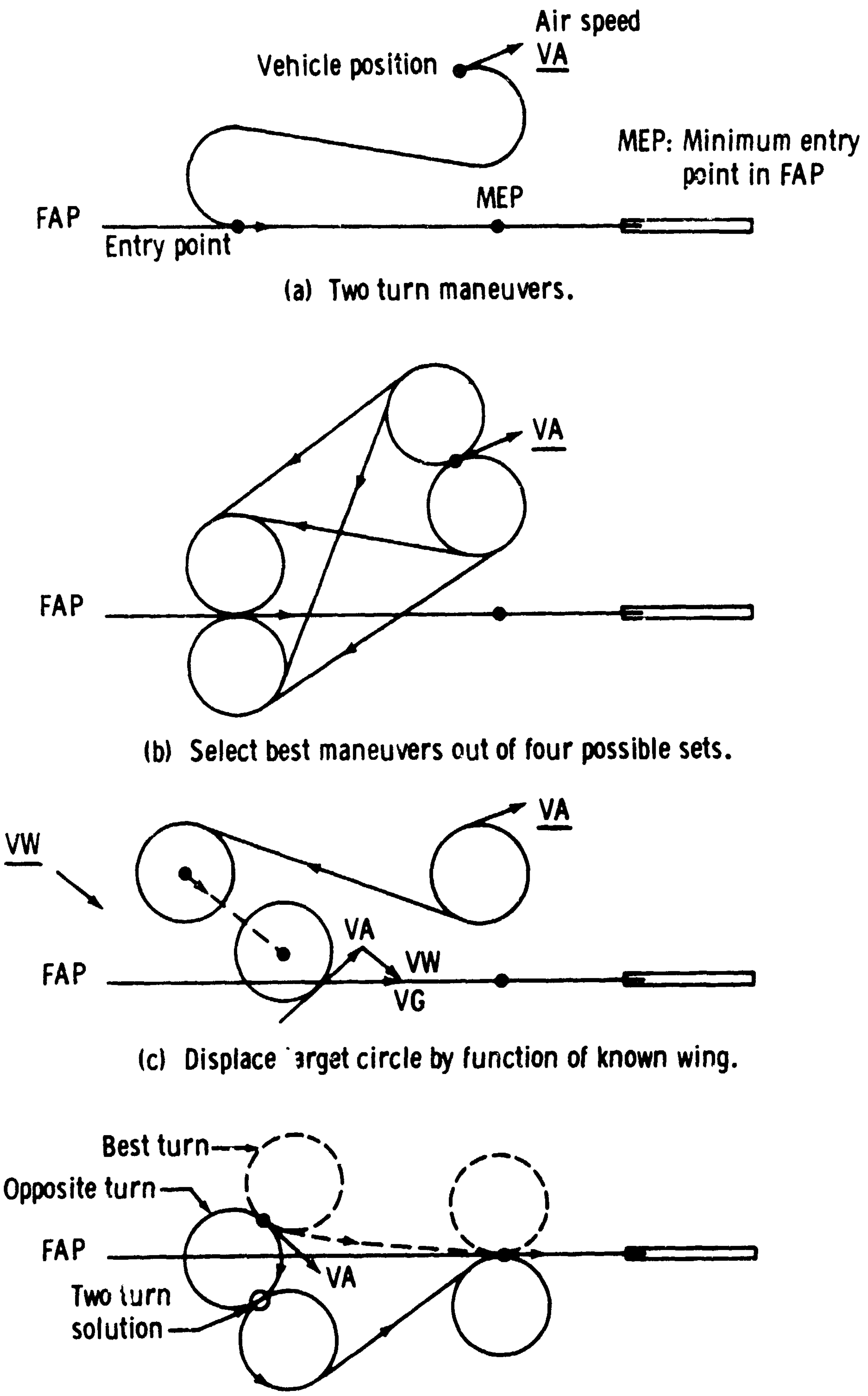


Figure 1 .- Guidance phases.



(a) Two turn maneuvers.

(b) Select best maneuvers out of four possible sets.

(c) Displace 'arget circle by function of known wing.

(d) Three turn maneuver capability.

Figure 2. - Sketches demonstrating ground tracks considered by guidance.

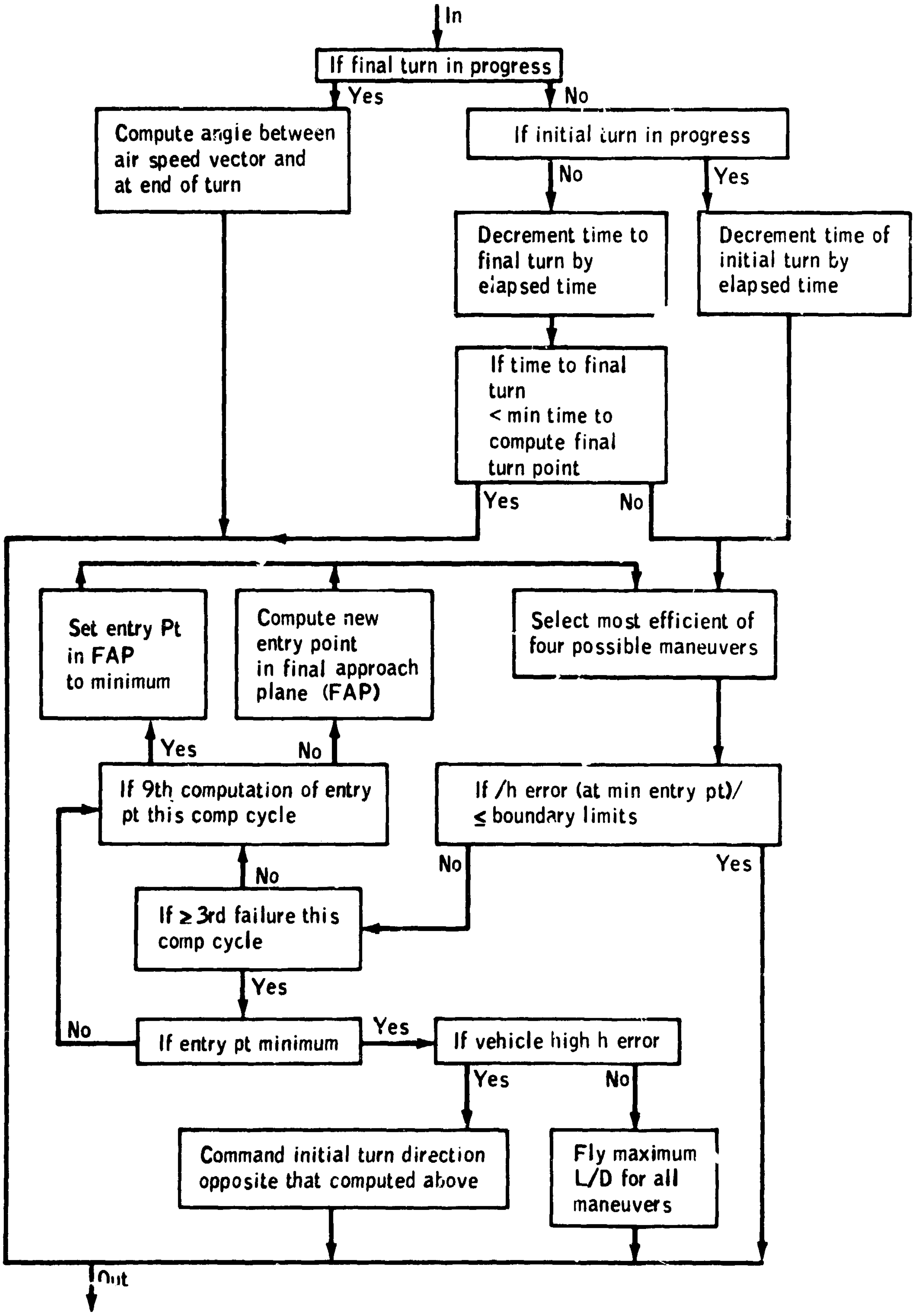

In
If final turn in progress
Yes
No
Compute angie between air speed vector and at end of turn
If initial turn in progress
No
Yes
Decrement time to final turn by elapsed time
Decrement time of initial turn by elapsed time
If time to final turn < min time to compute final turn point
Yes
No
Set entry Pt in FAP to minimum
Compute new entry point in final approach plane (FAP)
Select most efficient of four possible maneuvers
Yes
No
If 9th computation of entry pt this comp cycle
If /h error (at min entry pt)/ ≤ boundary limits
No
Yes
No
If ≥3rd failure this comp cycle
Yes
No
If entry pt minimum
Yes
If vehicle high h error
Yes
No
Command initial turn direction opposite that computed above
Fly maximum L/D for all maneuvers
Out

Figure 3.-Flow diagram of guidance to final approach plane.

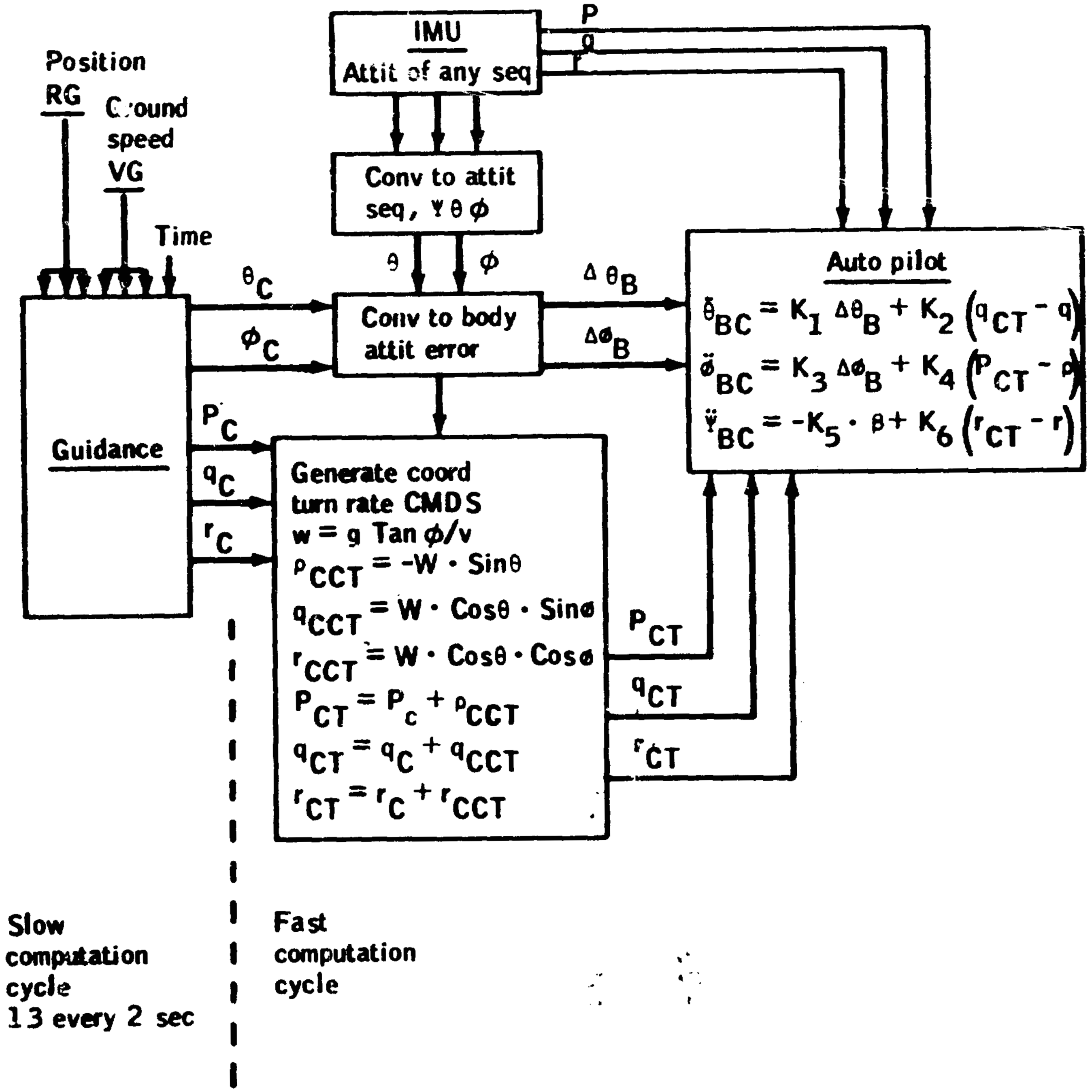


Figure 4.- Guidance-autopilot interface.

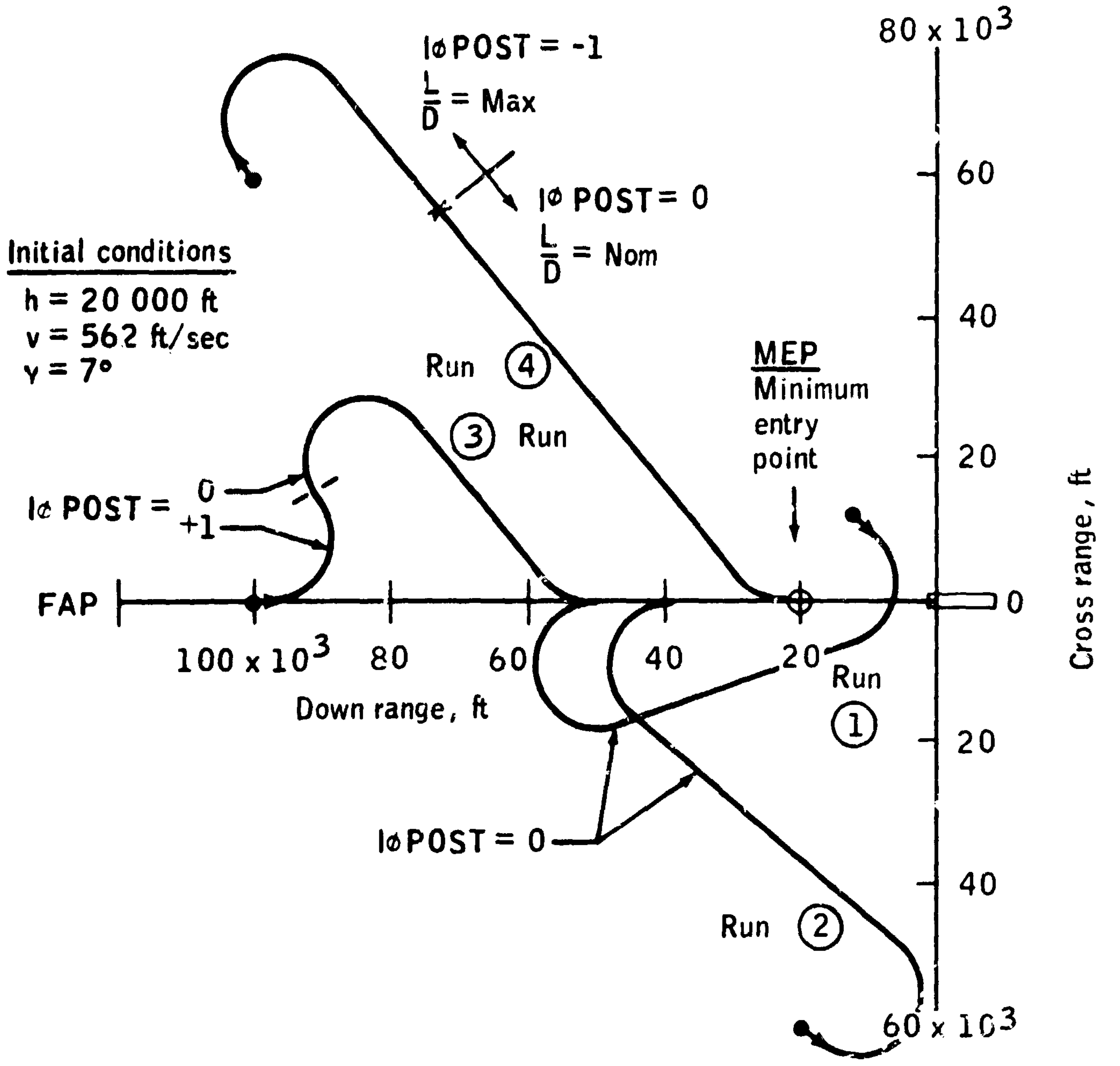


Figure 5.- Terminal area guidance of straight wing vehicle from various positions relative to runway.

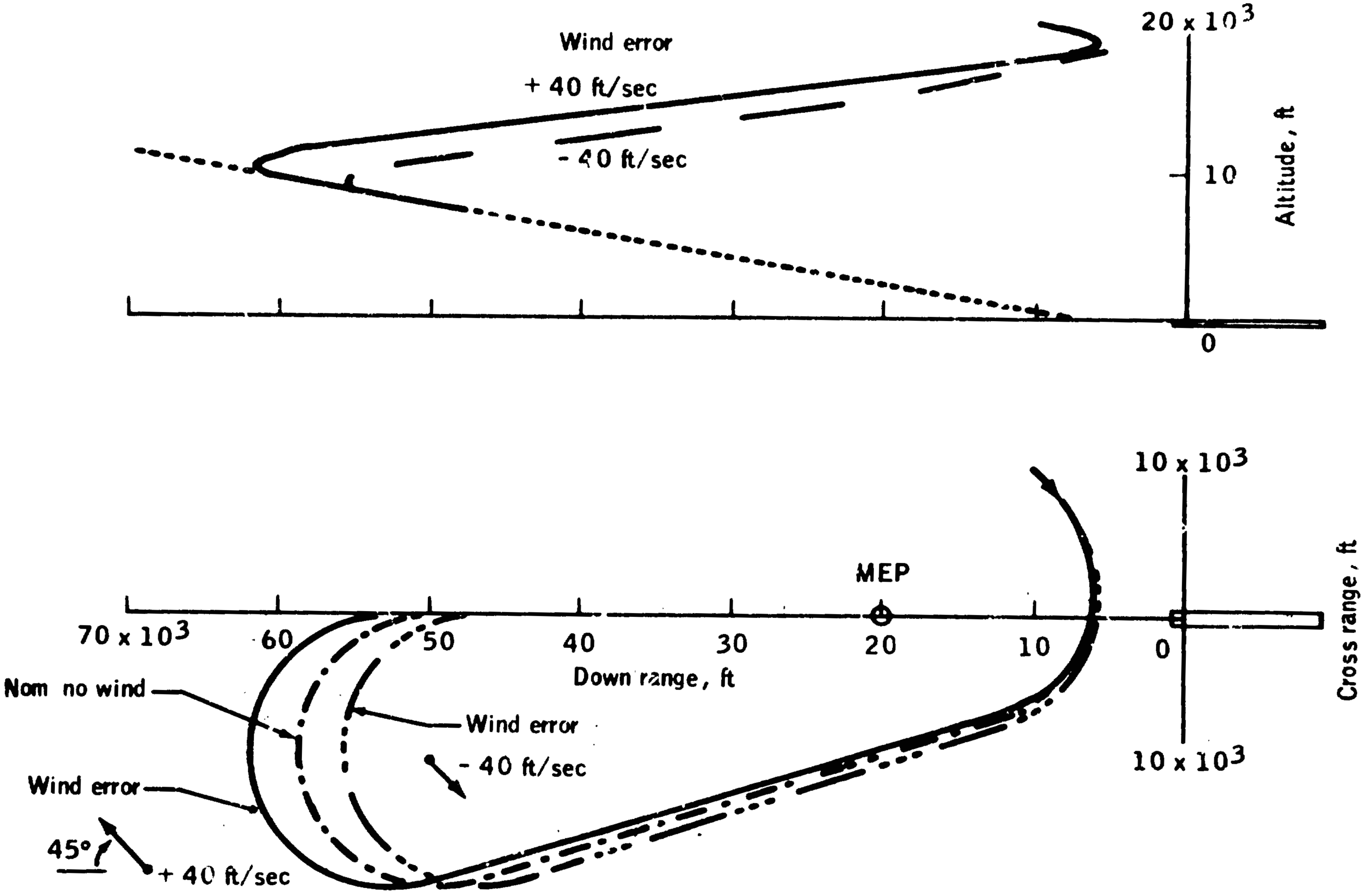


Figure 6a.- Terminal area guidance of straight wing vehicle with wind errors. - Trajectories.

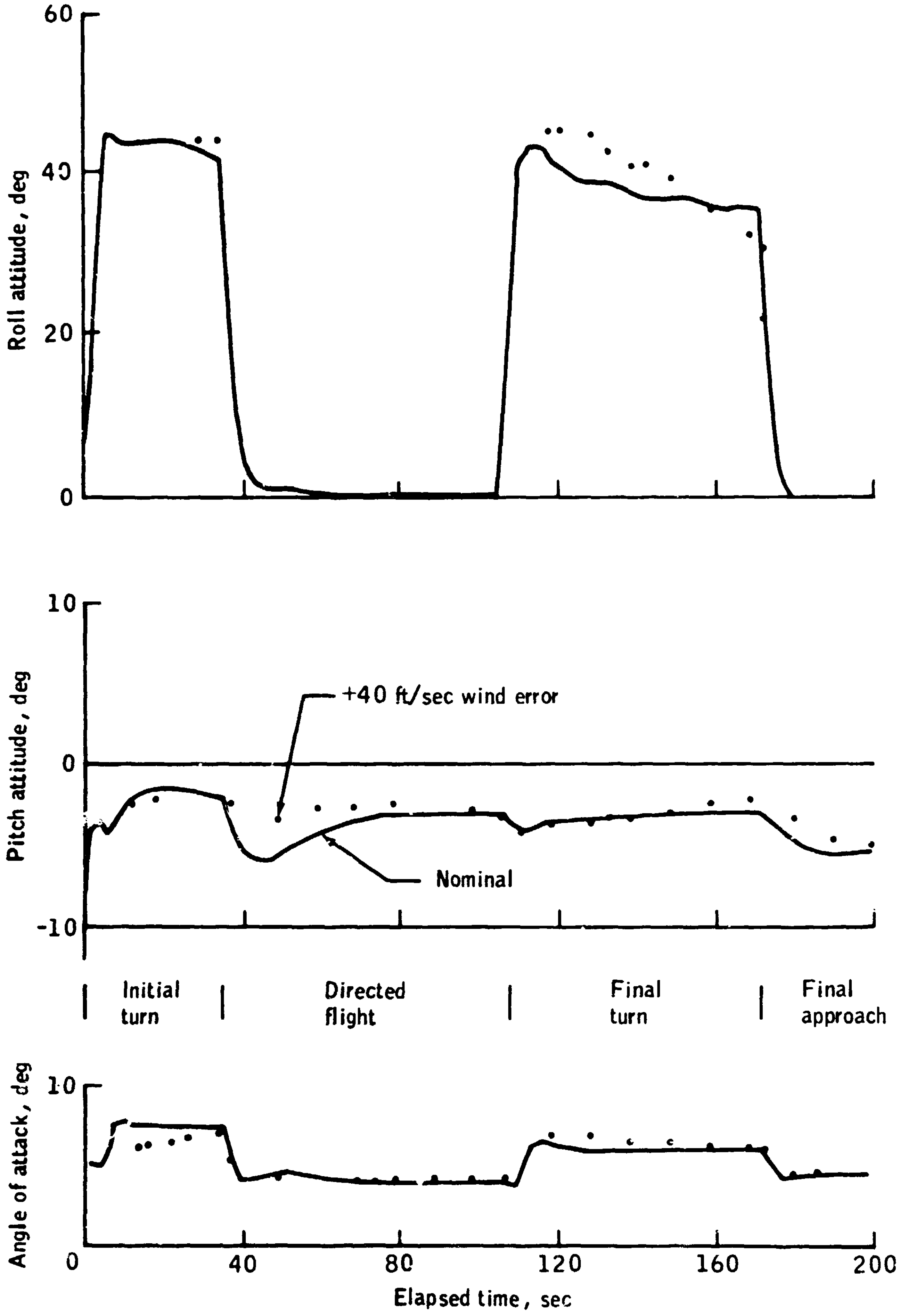


Figure 6b.- Terminal area guidance of straight wing vehicle, nominal and with wind error. Trajectory control parameters.

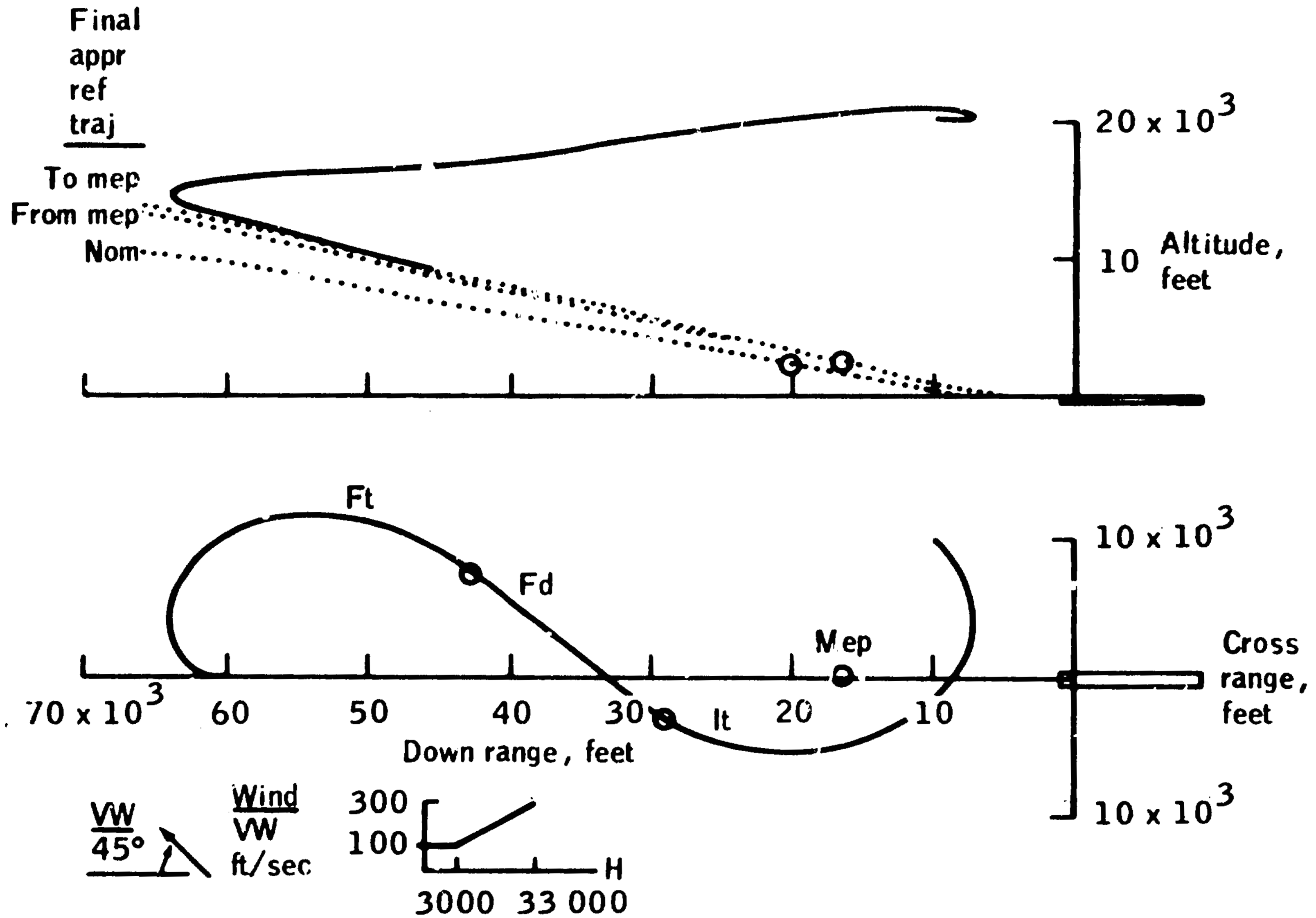


Figure 7a.- Terminal area guidance of straight wing vehicle with apriori wind equal actual wind. - Trajectory.

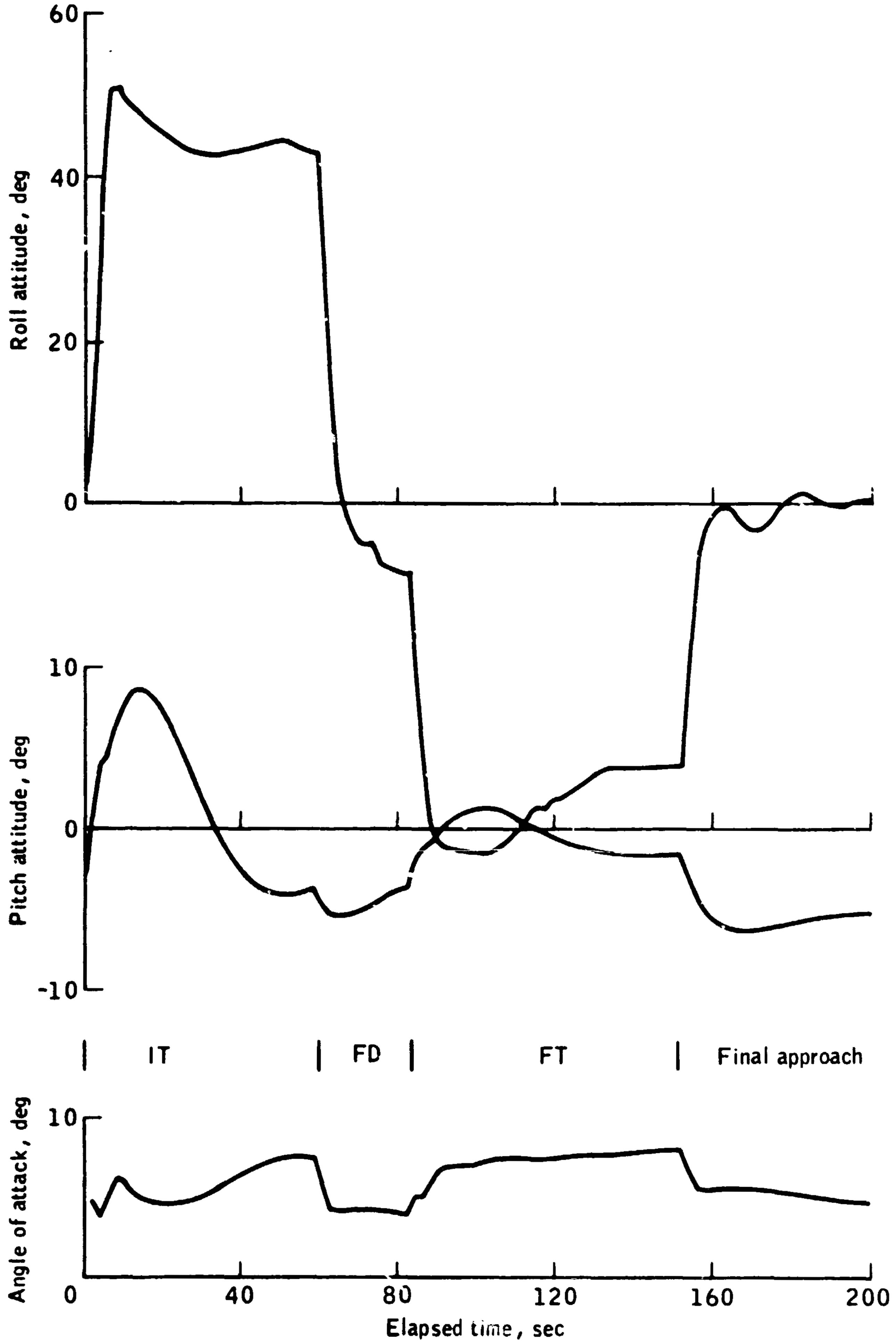


Figure 7b.- Terminal a a guidance of straight wing vehicle with apriori wind equal actual wind.- Trajectory control parameters.

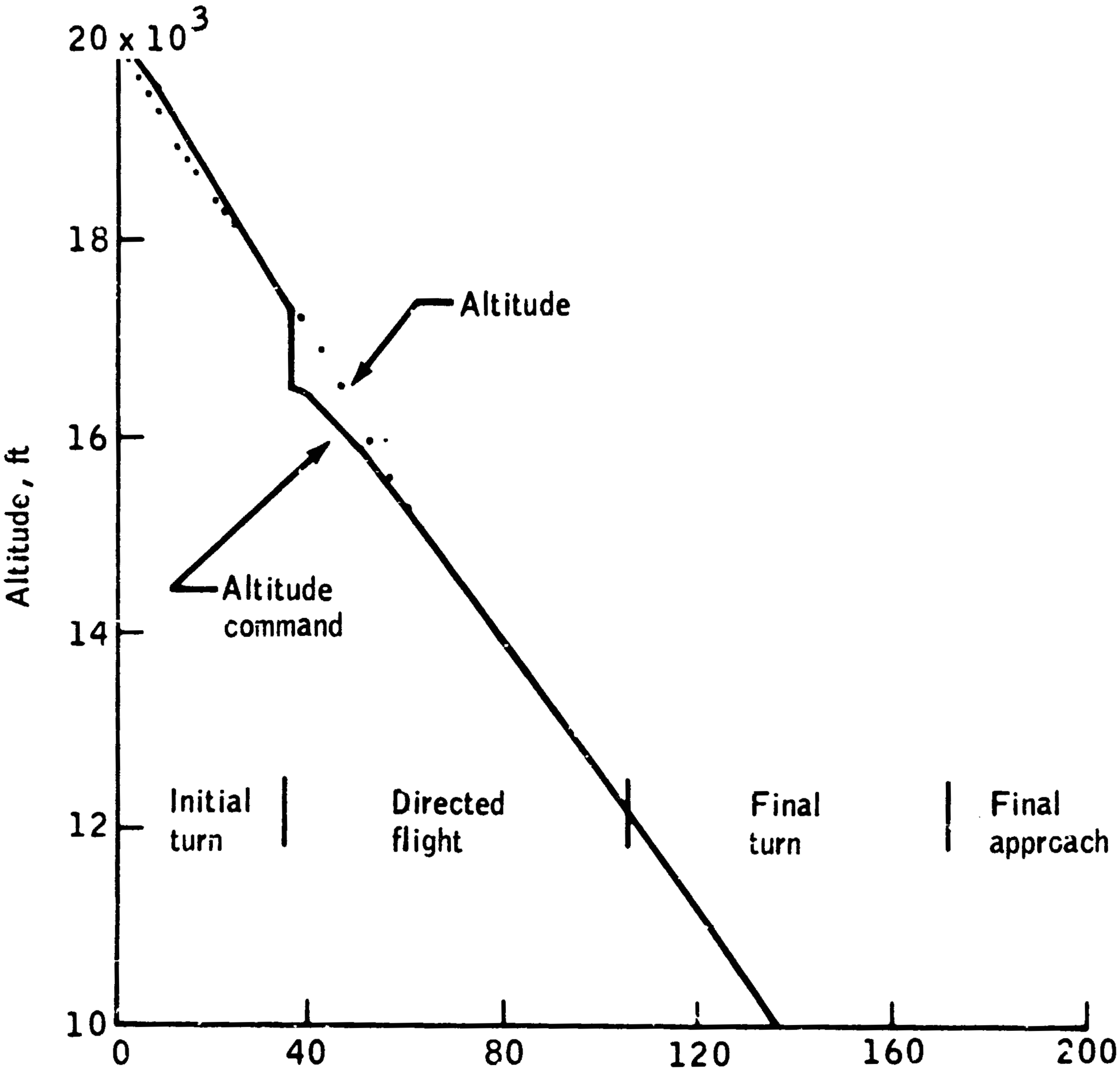


Figure 8a.- Terminal area guidance of straight wing vehicle, nominal - altitude and command.

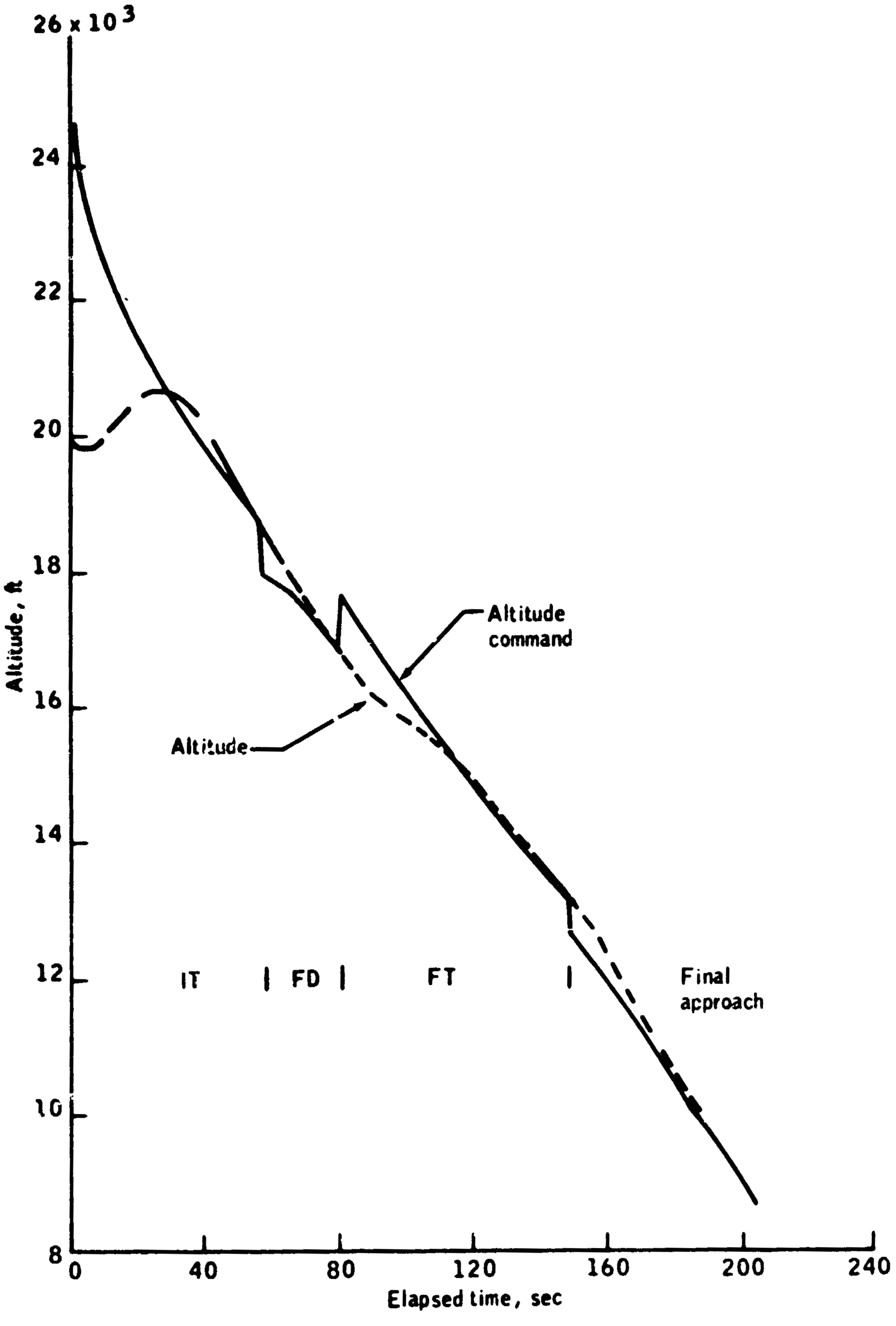


Figure 8b. - Terminal ⁄ rea guidance of straight-wing vehicle with apriori wind equal actual wind - altitude and command

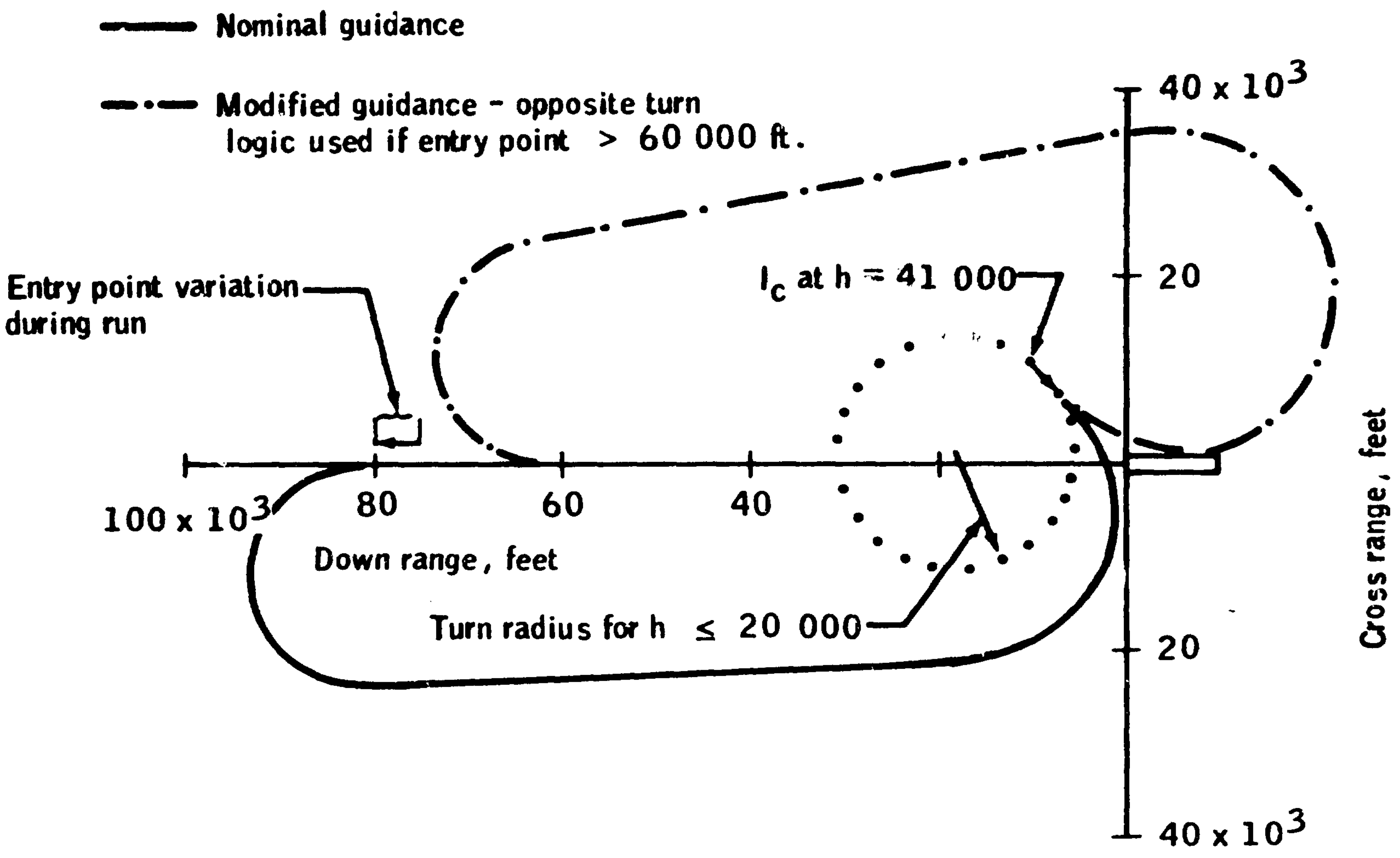
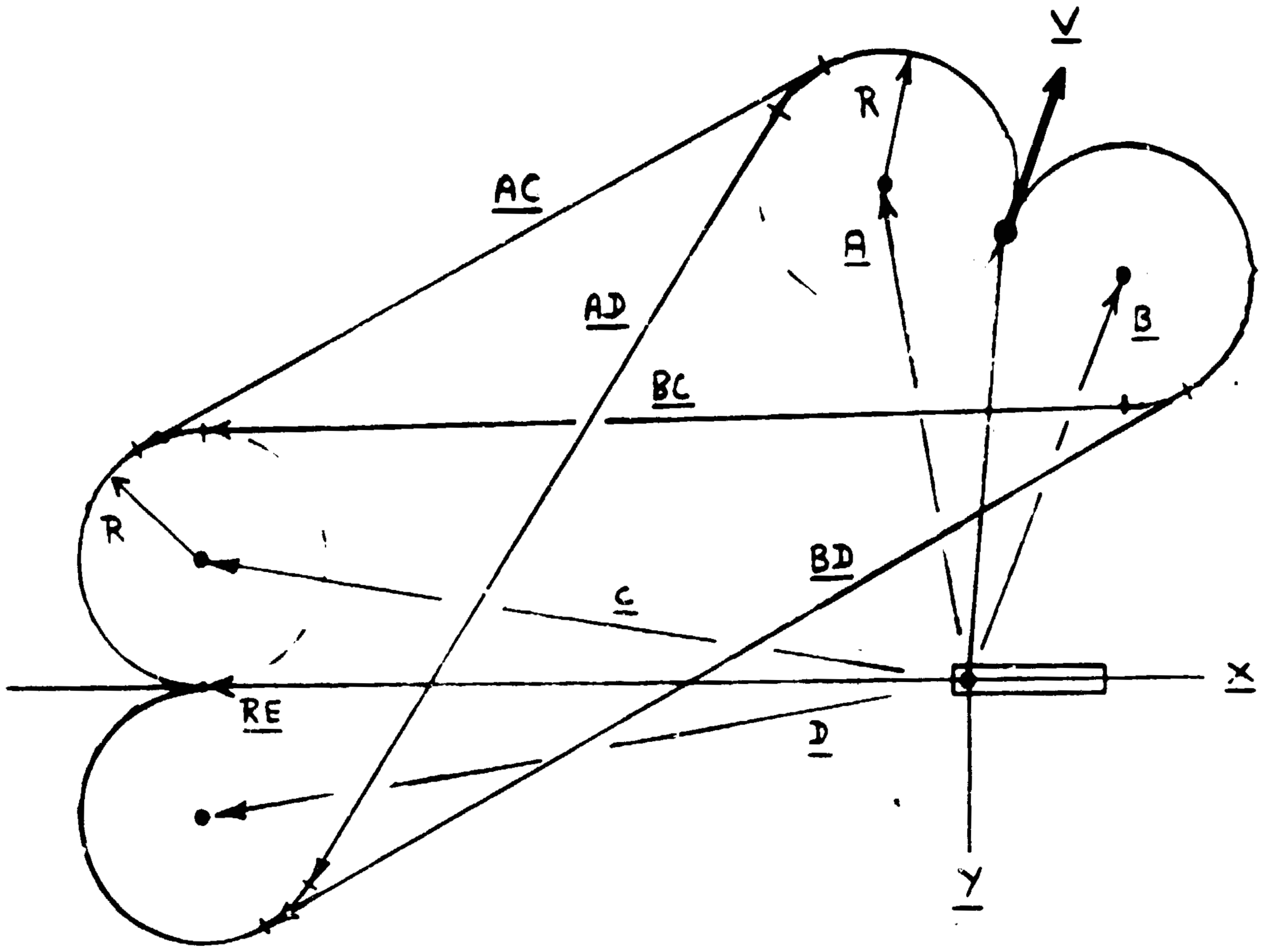


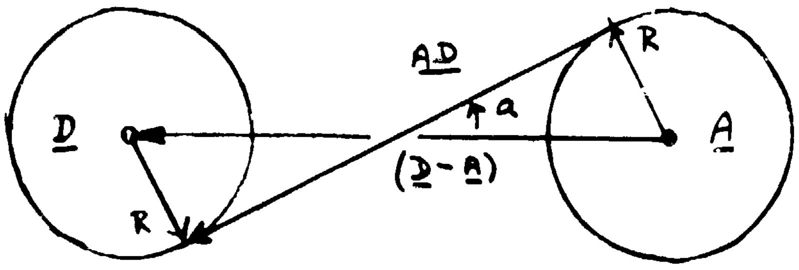
Figure 9.- Terminal area guidance of delta wing vehicle.

APPENDIX A

Ground Track Vector Equations



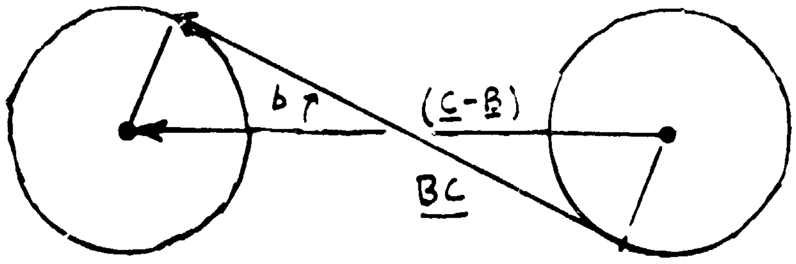
$\underline{A} = \underline{RP} + R \cdot \underline{U}\ (\underline{V}\ X\ \underline{k})$	$\underline{U}\ (\ ) =$ Unit Vector
$\underline{B} = \underline{RP} + R \cdot \underline{U}\ (\underline{k}\ X\ \underline{V})$	$\underline{i} = \underline{U}\ (X)$
$\underline{C} = \underline{RE} - R \cdot \underline{j}$	$\underline{j} = \underline{U}\ (Y)$
$\underline{D} = \underline{RE} + R \cdot \underline{j}$	$\underline{k} = \underline{U}\ (Z)$
$\underline{AC} = \underline{C} - \underline{A}$	$\underline{ULV} = -\underline{k}$
$\underline{BD} = \underline{D} - \underline{B}$	= Unit Local Vertical



$$a = \text{ASIN}\left(\frac{2R}{|\underline{D} - \underline{A}|}\right)$$

$$\underline{AD} = (|\underline{D} - \underline{A}| - 2 \cdot R \cdot \text{SIN } a) \cdot \underline{U}\,(\underline{D} - \underline{A}) + (2 \cdot R \cdot \text{COS } a) \cdot \underline{U}\,((\underline{D} - \underline{A}) \times \underline{k}))$$

If $|\underline{D} - \underline{A}| < 2R$, set flag to eliminate this turn comb.



$$b = \text{ASIN}\left(\frac{2R}{|\underline{C} - \underline{B}|}\right)$$

$$BC = (|\underline{C} - \underline{B}| - 2 \cdot R \cdot \text{SIN } b) \cdot \underline{U}\,(\underline{C} - \underline{B}) + (2 \cdot R \cdot \text{COS } b) \cdot \underline{U}\,(\underline{k} \times (\underline{C} - \underline{B}))$$

If $|\underline{C} - \underline{B}| < 2R$, set flag to eliminate this turn comb.

Initial Turn Angles - Solve for + angle (0 to 360°) between $\underline{V}$ and vector between turns.

Final Turn Angles - Solve for + angle (0 to 360°) between vector between turns and $\underline{i}$.

9.15.2.1 Approach Guidance (cont'd)

APPENDIX B

B-1

Trajectory Prediction of Aircraft Flying Constant Angle-of-Attack and Turn Radius

Steady state flight path angle. - The equations presented hsre are not (except equations (4 and 7)) a direct part of the guidance, but are equations to be externally processed for data load in the flight plan portion of the onboard guidance. The guidance makes simple use of the fact that the steady state flight path angle is nearly constant for constant angle-of-attack. The approximation

$(\gamma = \frac{1}{L/D \text{ Cos } \phi})$ is not satisfactory.

To hold a constant turn radius (R), the centripetal acceleration,

$$\ddot{Y} = \frac{(V \text{ Cos } \gamma)^2}{R} = \frac{L}{M} \text{ Sin } \phi$$

but lift,

$$L = \tfrac{1}{2} C_L S \rho V^2$$

$\therefore$,

$$\text{(1)} \quad \text{Sin } \phi = \frac{2M \text{ Cos}^2 \gamma}{R C_L S \rho}$$

and by differentiating (1)

$$\dot{\phi} = -\frac{\dot{\rho}}{\rho} \cdot \text{Tan } \phi$$

Standard atmosphere air density,

$$\text{(2)} \quad \rho = .002378 \; e^{\frac{1}{.235} \ln(1 - 6.875 \cdot 10^{-6} \cdot h)}$$

by differentiating (2)

$$\text{(3)} \quad \frac{\dot{\rho}}{\rho} = \frac{6.875 \cdot 10^{-6} \cdot V \cdot \text{Sin } \gamma}{.235(1 - 6.875 \cdot 10^{-6} \cdot h)}$$

If there is a steady value of γ, then

$$\dot{\gamma} = 0 = \frac{g \cdot \cos \gamma - \frac{L}{M} \cos \phi}{V}$$

SIGN CONVENTION

γ (+) for Descent

or

$$g \cos \gamma = \frac{L}{M} \cos \phi$$

from this and the lift equation velocity is constrained,

$$(4) \quad V = \sqrt{\frac{2 \cdot M \cdot g \cdot \cos \gamma}{C_L \cdot S \cdot \rho \cdot \cos \phi}}$$

The rate of change of velocity by definition;

$$\dot{V} = g \cdot \sin \gamma - \frac{D}{M}$$

or

$$g \sin \gamma = \frac{D}{M} + \dot{V}$$

Also, for a steady γ ,

$$\ddot{\gamma} = 0$$

or

$$d(L \cdot \cos \phi) = 0 = \dot{L} \operatorname{Cos} \phi - L \operatorname{Sin} \phi \cdot \dot{\phi}$$

$$\dot{L} = L \cdot \frac{\dot{\rho}}{\rho} (1 + 2 \frac{\rho}{\dot{\rho}} \frac{\dot{V}}{V}) = L \operatorname{Tan} \phi \cdot \dot{\phi} = -L \cdot Ta^2n \phi \cdot \frac{\dot{\rho}}{\rho}$$

$$(5) \quad \dot{V} = -\frac{V}{2} \cdot \frac{\dot{\rho}}{\rho} (1 + \operatorname{Tan}^2 \phi)$$

The flight path that produces $\ddot{\gamma} = 0$ is then,

$$\operatorname{Tan} \gamma = \frac{g \operatorname{Sin} \gamma}{g \operatorname{Cos} \gamma} = \frac{\frac{D}{M} + \dot{V}}{\frac{L}{M} \operatorname{Cos} \phi} = \frac{1}{\frac{L}{D} \operatorname{Cos} \phi} + \frac{\dot{V}}{g \operatorname{Cos} \gamma} \cdot \frac{\operatorname{Sin} \gamma}{\operatorname{Sin} \gamma}$$

by rearranging and collecting Tan γ terms, this equation becomes

$$(6) \quad \operatorname{Tan} \gamma = \frac{1}{\frac{L}{D} \operatorname{Cos} \phi} \Big/ (1 - \frac{\dot{V}}{g \operatorname{Sin} \gamma})$$

An iterative processing of equations (1 to 6) provides the state of velocity and flight path that must exist for the first and second derivative of flight path to be zero at any altitude, radius of turn, and angle-of-attack. Two questions to be answered empirically are:

a. Is there correlation between the predicted γ of equation (6) and actual γ ?

b. Is there a steady state value for γ ?

Results of equations (1 - 6) for three conditions of α and R are shown as a function of altitude on figure B-1. The aero-coefficient used are those of a straight-wing orbiter. The wings-level case (infinite radius) corresponds to within the resolution of the plot with an actual flight. A steady state value of γ is not achieved, but for practical guidance purposes, the variation is sufficiently small so that a constant γ can be assumed. For turns under 20,000 ft altitude, if α and R are selected so that bank $\leq 45^{\circ}$, the γ holds fairly steady at 8°. The simplified estimate of

$$\mathrm{Tan}\,\gamma = \frac{1}{\frac{L}{D}\,\mathrm{Cos}\,\phi}$$

is shown to be greatly in error.

Turns starting at higher altitudes and with turn radii so that maximum banks are 46.5 degrees are shown on figure B-2, along with actual results for one case. More variation is seen here for γ, but still, for practical guidance purposes, a steady state γ can be assumed. For example, at 50,000 ft altitude, a 180° turn would lose 8000 ft altitude with a resultant γ from 4.4 to 5.75. The guidance can estimate altitude at end of maneuver and γ (5.75) for that h can be assumed. The angle-of-attack required to hold this assumed γ for $\alpha = 7.5$ will be ≤ 7.5. This is a conservative approach because the kinetic energy at end of maneuver must be $\geq$ aim condition of guidance.

Trajectory prediction. - At any altitude with a velocity as given by equation (4), the future flight path can be predicted and it is sufficiently constant so that a linear trajectory (γ_{ss}) can be assumed. The remaining problem, though, is to predict the trajectory from a state with V $\neq$ equation (4). A solution is to shift the linear trajectory by the potential energy equivalent of the kinetic energy difference of vehicles present state and that of equation (4);

$$\tfrac{1}{2}M\,(V^2 - V_c^{\,2}) = W\,\Delta h$$

where

V_c = V of equation (4)

W = Mg

$$\Delta h = \frac{V^2 - V_c^2}{2g}$$

the predicted reference trajectory for constant α flight

$$(7) \quad h_c = h + \frac{(V^2 - V_c^2)}{2g} - X \cdot \text{Tan } \gamma_{ss}.$$

Some results of this assumption are shown on figure B-3. A γ_{ss} of 7.16 was selected from figure B-1 for an altitude of 10,000 ft. The three actual trajectories of figure B-3 were started at h = 20,000 ft, γ = 6.8, and V = 579.3 (from equation 4), 632.4 and 989.5 corresponding to 0, 1000, and 10,000 ft equivalent potential energy difference. The 1000 ft case steadies out on the reference trajectory, but the shift for the 10,000 ft case is only about 50 percent of predicted.

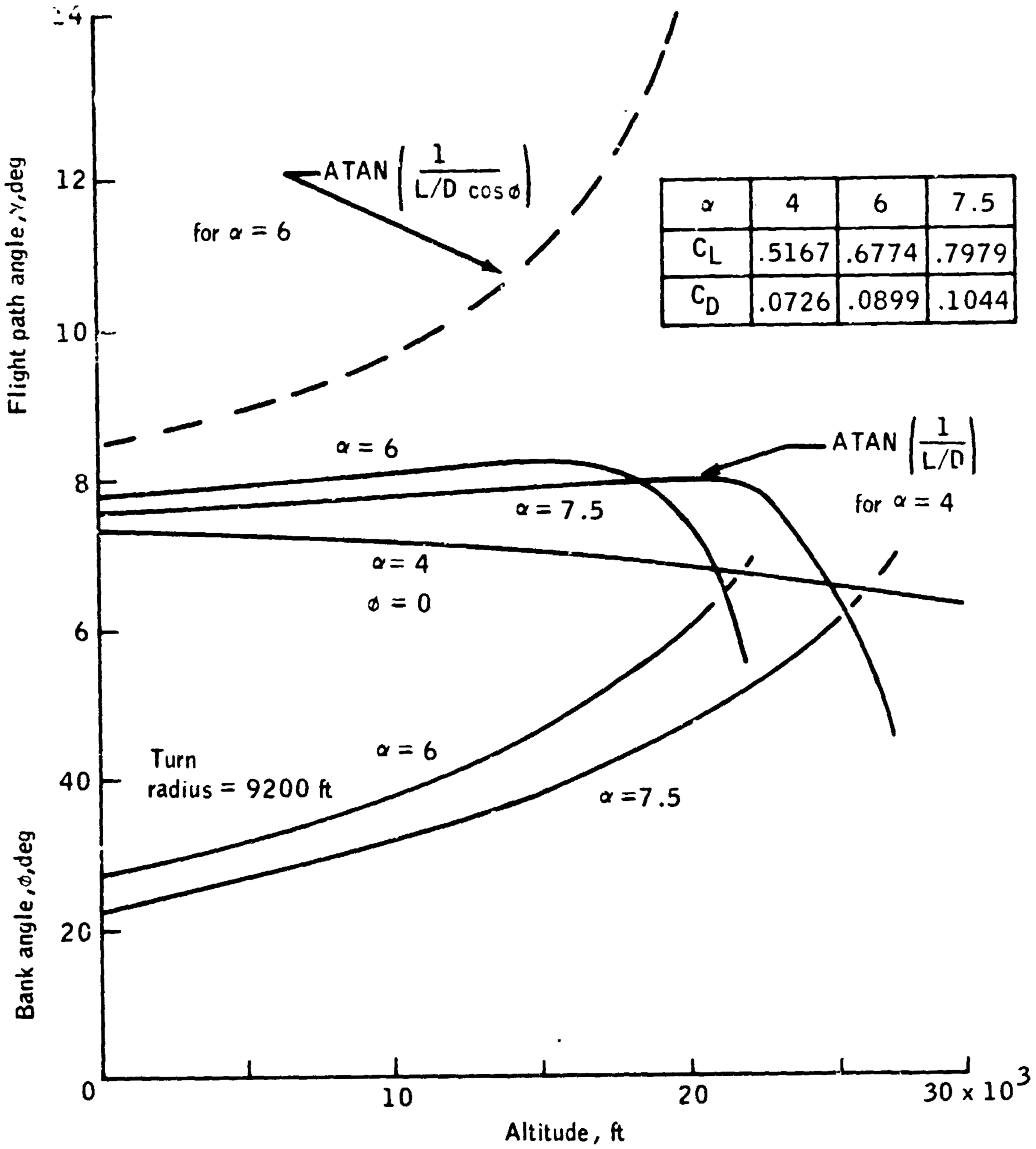
Energy is dissipated during flight because of drag force (D).

$$\Delta E = \int D \, dx$$

The trajectory prediction assumes no additional energy dissipation over this nominal dissipation, but actually, there is additional drag with a greater velocity, and therefore,

$$\Delta E = \int (D + \Delta D) \, dx$$

It is possible to predict this additional energy loss to better predict the trajectory but an iteration solution would be required. Also, this better solution indicates that the 1000 ft case is only 25 ft in error, and 1000 ft is about the maximum value encountered with the guidance system presented in this report, and therefore, the simpler prediction (7) is valid.



α	4	6	7.5
C_L	.5167	.6774	.7979
C_D	.0726	.0899	.1044

Figure B-1.- Predicted flight path for constant angle of attack flight.

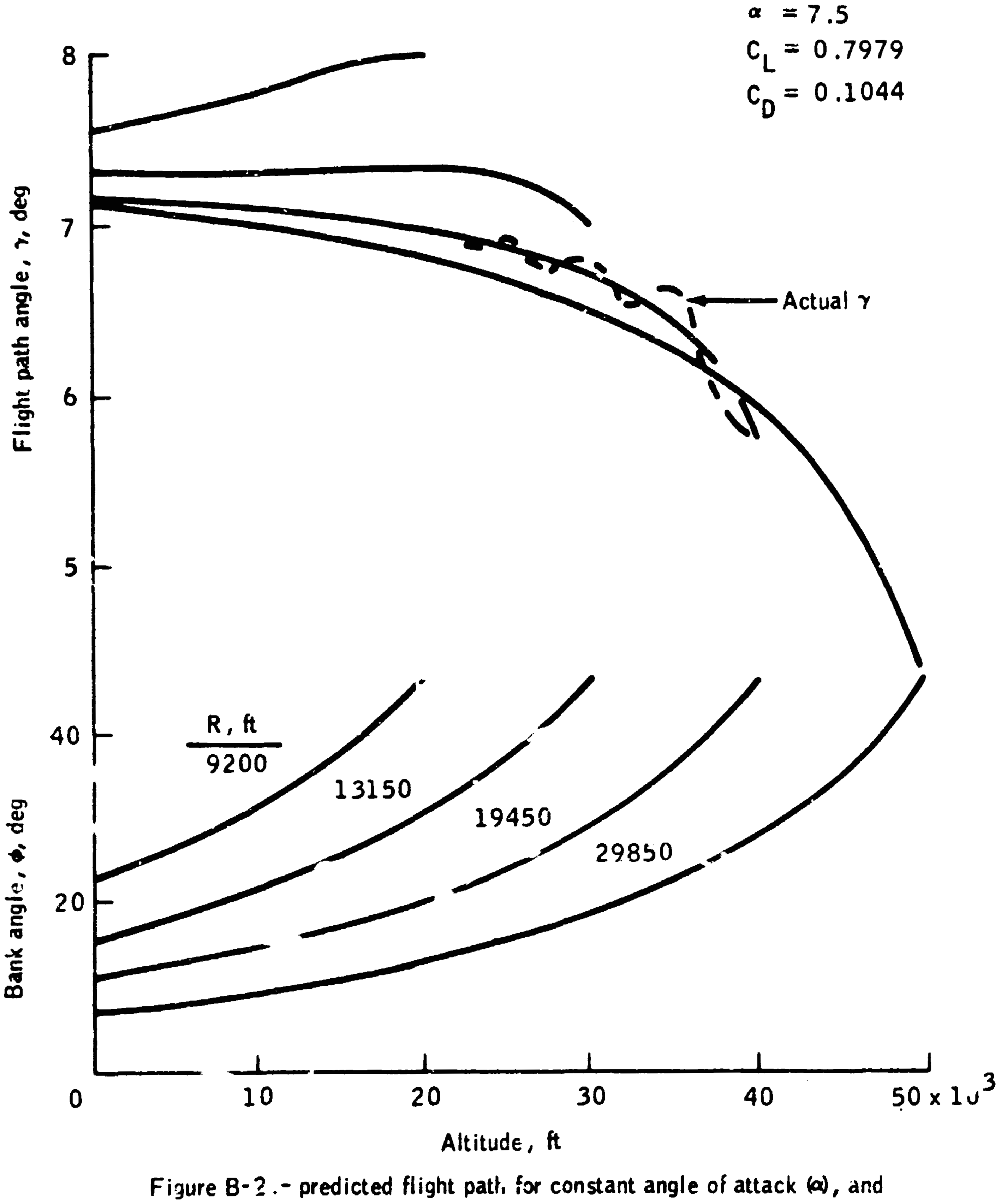


Figure B-2.- predicted flight path for constant angle of attack (α), and turn radius (R).

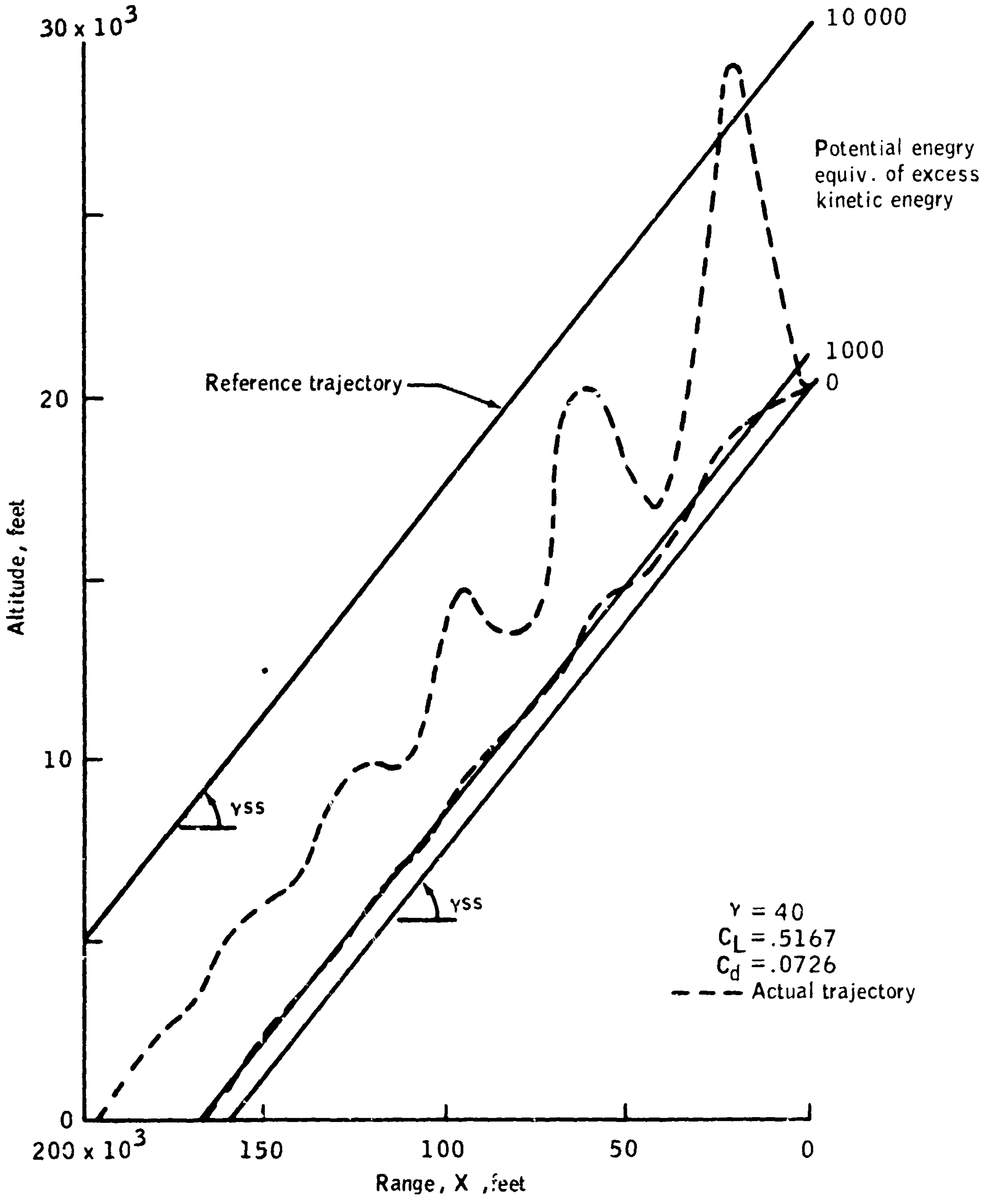


Figure B-3.- Predicted trajectory for constant angle of attack flight.

9.15.2.1 Approach Guidance (cont'd)

APPENDIX C

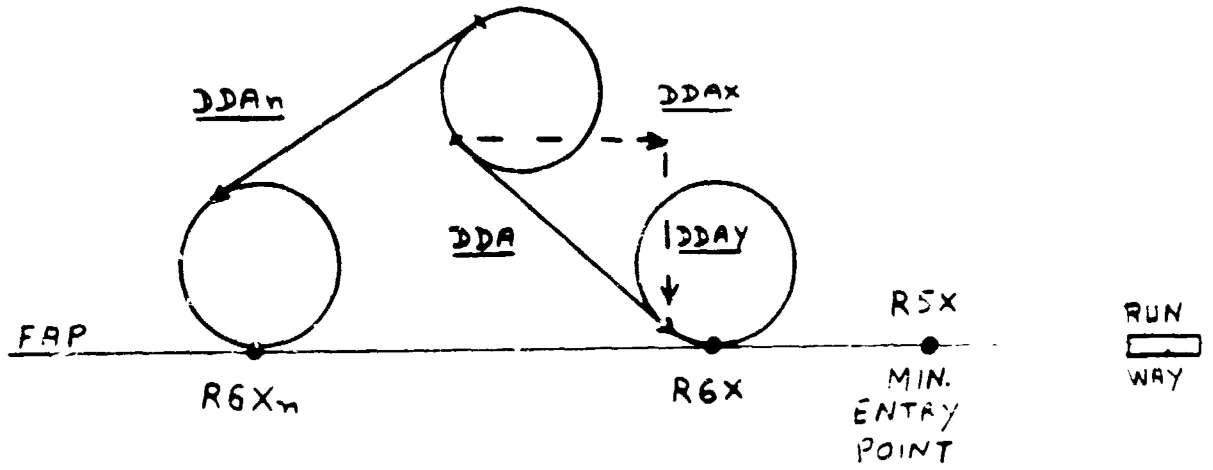
Compute New FAP Entry Point

Given: 1. A vector DDA from exit point of initial turn to entry point of final turn.

2. Exit point of final turn ($R6X$ = FAP entry point).

3. TAN of flight path angles; K_1 to FAP and K_2 in FAP.

4. Projected vehicle high altitude error (DH) at minimum FAP entry point ($R5X$).

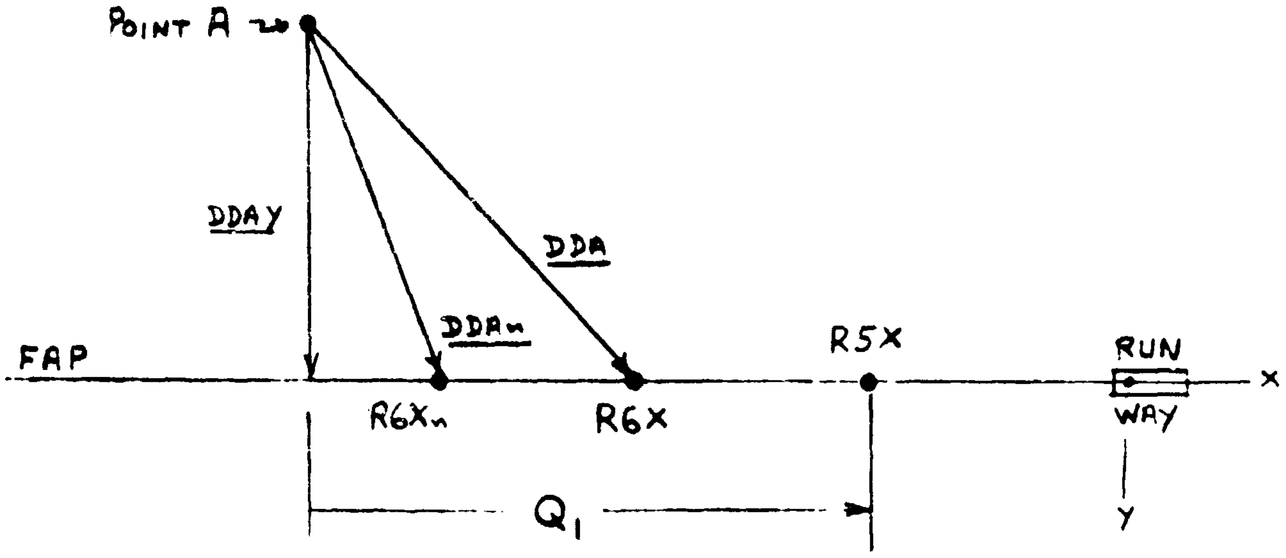


Assumptions: 1. A new $\underline{DDA}_N$ solution for a new $R6X_N$ will have the same DDAY component.

2. The sum of both the initial and final turn angle remain constant with variation of R6X, and therefore, altitude loss of turns is approximately independent of R6X.

NOTE: These are better assumptions for turns with same direction (i.e., left initial and final turns) than for opposite turns —this should merely mean that convergence on proper R6X will be faster if vehicle is in position for alike turns.

Because of the assumptions, the problem becomes one of determining intersection(s) of two straight lines—trajectory to FAP and in FAP.



$$Q_1 = R5X - R6X + DDAX$$

The constant altitude difference between point A and the desired altitude @ R5 is

$$\Delta H = K_1 \sqrt{DDAX^2 + DDAY^2} + K_2(Q_1 - DDAX) + DH$$

The objective of the new $R6X_N$ is to have a zero DH, therefore,

$$K_1 \sqrt{DDAX_N{}^2 + DDAY^2} + (Q_1 - DDAX_N)K_2 = K_1 \sqrt{DDAX^2 + DDAY^2}$$

$$+ K_2 (Q_1 - DDAX) + DH$$

rearranging equation

(1) $K_2 \sqrt{DDAX_N{}^2 + DDAY^2} = K_2 \cdot DDAX_N + Q^2$

where

(2) $\boxed{Q_2 = K_1 \sqrt{DDAX^2 + DDAY^2} - K_2 \cdot DDAX + DH}$

By squaring equation (1)

$$K1^2\ (DDAX_N^2 + DDAY^2) = (K_2 \cdot DDAX_N)^2 + 2Q_2 \cdot K_2 \cdot DDAX + Q_2^2$$

which is of quadratic form

$$A \cdot DDAX_N^2 + B \cdot DDAX + C = 0$$

where

$$A = (K_1^2 - K_2^2)$$

$$B = -2Q_2K_2$$

$$C = (K_1 \cdot DDAY)^2 - Q_2^2$$

the solution of which is

$$DDAX_N = \frac{-B \pm \sqrt{B^2 - 4AC}}{2A}$$

this equation presents a problem though, when A = 0, which is likely for a no-wind condition; therefore, a better form is

$$DDAX_N = \frac{-B \pm \sqrt{B^2 - 4AC}}{2A} \cdot \frac{(-B \mp \sqrt{B^2 - 4AC})}{(-B \mp \ B^2 - 4AC)}$$

$$(3) \qquad = \frac{-2C/B}{1 \pm \sqrt{1 - \frac{4AC}{B^2}}}$$

Before using equation (3), some constraints must be applied.

The roots of (3) must be real,

$$\boxed{1 - 4AC/B^2 \geq 0}$$

There are at most two real roots of equation (3), and there can be eight sets of two real roots depending on the signs of the coefficients (A, B, & C). Four of these sets where B is positive, although real for equation (3) are not real for the physical problem; i.e., equation (1). These can be avoided with constraint

$$B < 0$$

or

$$\boxed{Q_2 > 0}$$

Two other sets of roots where A is negative contain two real roots for equation (3), but only one is valid for equation (1). The valid one is obtained by using the positive sign in equation (3).

The remaining two sets are valid for both equations, but the more negative root (FAP entry at greatest distance from R5X) is desired, and it also requires positive sign in equation (3), and therefore, if no constraint violated,

$$\boxed{DDAX_N = \frac{-2C/B}{1 + \sqrt{1 - 4AC/B^2}}}$$

$$\boxed{\begin{array}{l} R6X_N = R5X - Q_1 + DDAX_N \\ \\ \text{If} \quad (R6X_N > R5X) \; R6X = R5X \end{array}}$$

If any constraint was violated, then FAP entry point set to minimum,

$$\boxed{R6X = R5X}$$

APPENDIX D

Guidance to Reference Trajectory

The reference trajectory $(H_c, \dot{H}_c, Y_c, \dot{Y}_c)$ and actual vehicle state $(H, \dot{H}, Y, \dot{Y})$ have been defined for vertical and lateral directions. The guidance equations are derived that produce guidance output commands of θ_c, ϕ_c of standard aircraft sequence ψ, θ, ϕ and body rate commands p_c, q_c, r_c. A compilation of just the guidance equations is summarized at the end of this appendix.

General Physical Equations Utilized

Airspeed

$$\underline{V} = \underline{V} \text{ (Nav relative to ground)} - \underline{VW} \text{ (apriori wind)}$$

$$V = |\underline{V}|$$

Dynamic pressure

$$\bar{q} = \tfrac{1}{2} \rho V^2$$

Lift coefficient

(1) $$C_L = C_{L_0} + \text{pCLA} \cdot \alpha$$

Lift

(2) $$L = C_L S \bar{q}$$

Airspeed acceleration

$$\dot{V} \doteq (V - V_0)/\Delta t$$

where Δt is computation cycle time (i.e., 2 sec)

and V_0 is airspeed at previous computation time.

Flight path

$$\gamma = \text{Tan}^{-1} \left(VZ/\sqrt{V_X^2 + V_Y^2}\right)$$ Note: γ+ DOWN

C.6

Drag

(3) $$\frac{D}{M} = g \sin \gamma - \dot{V}$$

Total vertical acceleration

(4) $$\ddot{H}_T = \frac{L}{M} \text{Cos}\ \phi_V \text{ Cos}\,\gamma + \frac{D}{M} \text{Sin}\,\gamma - g$$

where

ϕ_V = bank angle with reference to V.

from (3) and (4)

(5) $$\frac{L}{M} \text{Cos}\ \phi_V = \frac{\ddot{H}_T + \dot{V} \text{ Sin}\,\gamma}{\text{Cos}} + g \text{ Cos}\,\gamma$$

a component of vertical acceleration defined as

(6) $$\ddot{H} = \ddot{H}_T + \dot{V} \text{ Sin}\,\gamma$$

angle of attack from (1, 2, 5, and 6)

(7) $$\alpha = \frac{M}{pCLA \cdot S \cdot \bar{q} \cdot \text{Cos}\ \phi_V} \left(\frac{\ddot{H}}{\text{Cos}\gamma} + g \text{ Cos}\,\gamma\right) - \frac{C_{L_O}}{pC_{L_A}}$$

by differentiating (7)

(8) $$\dot{\alpha} = \frac{M}{pCLA \cdot S \cdot \bar{q} \cdot \text{Cos}\ \phi_V} \left(\frac{\dddot{H}}{\text{Cos}\,\gamma} + \dot{\gamma} \text{ Tan}\gamma \left(\frac{\ddot{H}}{\text{Cos}\gamma} - g \text{ Cos}\,\gamma\right) - \left(\frac{\ddot{H}}{\text{Cos}\,\gamma} + g \text{ Cos}\,\gamma\right) \cdot \left(\frac{2\dot{V}}{V} + \frac{\dot{\rho}}{\rho} - \dot{\phi}_V \text{ Tan}\ \phi_V\right)\right)$$

also rate of change of angle-of-attack

(9) $$\dot{\alpha} = \dot{\gamma} \text{Cos}\ \phi_V + q$$

where q is body pitch rate.

rate of change of flight path

$$\dot{\gamma} = \frac{g \text{ Cos } \gamma - \frac{L}{M} \text{ Cos } \phi_v}{V}$$

$$(10) \quad \dot{\gamma} = \frac{-\ddot{H}}{V \text{ Cos } \gamma}$$

Lateral acceleration

$$(11) \quad \ddot{Y} = \frac{L}{M} \text{ Sin } \phi_v$$

Bank angle from (5, 6, and 8)

$$(12) \quad \phi_v = \text{Tan}^{-1}\left(\frac{\ddot{Y}}{\left(\frac{H}{\text{Cos } \gamma} + g \text{ Cos } \gamma\right)}\right)$$

by differentiating (12)

$$(13) \quad \dot{\phi}_v = \frac{\text{Cos}^2 \phi_v}{\left(\frac{H}{\text{Cos } \gamma} + g \text{ Cos } \gamma\right)} \left(\dddot{Y} - \text{Tan } \phi_v \left(\frac{\dddot{H}}{\text{Cos } \gamma} + \dot{\gamma} \text{Tan } \gamma \left(\frac{\ddot{H}}{\text{Cos } \gamma} - g \text{ Cos } \gamma\right)\right)\right)$$

<u>Vertical axis guidance</u>. - To guide the vehicle altitude to the command reference trajectory with specified response characteristics of natural frequency (WN) and damping ratio (ζ), a second-order acceleration command is generated

$$\ddot{H}_{CT} = W_n^2(H_c - H) + 2\zeta W_n(\dot{H}_c - \dot{H}) - \dot{V} \text{ Sin } \gamma$$

The last term of this equation is the steady state acceleration required to hold the error signals of the first two terms at zero. But, if this acceleration is applied to equation (5), the $\dot{V}$ term will cancel, and therefore, just the following component of the command need be generated.

(14) $$\ddot{H}_c = W_n^2(H_c - H) + 2\zeta W_n(\dot{H}_c - \dot{H})$$

Equation (7) will give the resulting α_c for this accel command,

(15) $$\alpha_c = \frac{M}{P_{CLA} \cdot S \cdot \bar{q} \cdot \text{Cos}\,\phi_{vc}} \left(\frac{\ddot{H}_c}{\text{Cos}\,\gamma} + g\,\text{Cos}\,\gamma\right) - \frac{C_{lo}}{PCLA}$$

where ϕ_{vc}, which is discussed later, must be generated first for use in this equation.

Given γ, α_c, and ϕ_{vc}, the local vertical component of a unit vector along the commanded direction of the fuselage is,

(16) $$UXBLV = \text{Sin}\alpha_c \cdot \text{Cos}\,\phi_{vc} \cdot \text{Cos}\gamma - \text{Cos}\alpha_c \cdot \text{Sin}\gamma$$

The pitch component of this vector for the standard aircraft sequence of $\psi\ \theta\ \phi$ is,

(17) $$\theta_c = \text{Sin}^{-1}\ (UXBLV)$$

The acceleration command (14) is a variable, the slope of which at a given time,

$$\dddot{H}_c = W_n^2\ (\dot{H}_c - \dot{H}) + 2\zeta W_n\ \left(\frac{d\dot{H}_c}{dt} - \ddot{H}\right)$$

$$\frac{d\dot{H}_c}{dt} = -\dot{V}\ \text{Sin}\,\gamma_{ss} \quad \text{(as defined for reference trajectory)}$$

assume,

$$\ddot{H} = \ddot{H}_{CT} = \ddot{H}_c + V\ \text{Sin}\,\gamma$$

$\therefore$,

$$\dddot{H}_c \doteq W_n^2\ (\dot{H}_c - \dot{H}) - 2\zeta W_n\ (\ddot{H}_c)$$

Substituting this into equation (8) and neglecting the last two terms of (8), which is an imperically determined good approximation,

$$(19) \qquad \dot{\alpha}_c = \frac{M}{P_{CLA} \cdot S \cdot \bar{q}} \frac{}{\cdot \text{ Cos } \phi_{vc}} \left(\frac{\ddot{H}_c}{\text{Cos}\gamma}\right)$$

From (9 and 10), the pitch rate command,

$$(20) \qquad q_c = \frac{\ddot{H}_c \cdot \text{Cos } \phi_{vc}}{V \cdot \text{Cos } \gamma} + \dot{\alpha}_c$$

If these commands were perfectly applied, then the total vertical acceleration would be,

$$\ddot{H}_T = \ddot{H}_c - \dot{V} \text{ Sin } \gamma + \dddot{H}_c \cdot t$$

Assume, though, that there are bias type errors present resulting from errors in aero coefficients, navigation, wind, etc.,

∴ the actual acceleration,

$$\ddot{H}_{T_A} = \ddot{H}_T + \ddot{H}_\epsilon$$

This error can be computed by comparing the altitude rate from one computation cycle to the next with the rate that should exist,

$$\dot{H} = \dot{H}_o + \int_o^{\Delta t} \ddot{H}_{T_A} dt$$

$$(21) \qquad \ddot{H}_\epsilon = (\dot{H} - \dot{H}_o - \ddot{H}_{co} \cdot \Delta t + \dot{V} \text{ Sin } \gamma \cdot \Delta t - \tfrac{1}{2}\dddot{H}_{co} \Delta t^2)/\Delta t$$

The objective is to remove bias errors, but not transient error such as caused attitude response time delays. For a step change of attitude command, the instantaneous error would be,

$$(22) \qquad \ddot{H}'_{\epsilon} \doteq -\frac{P_{CLA} \cdot S \cdot \bar{q}}{M}\,(\theta_{co} - \theta_o)$$

As that system responds to the command, the error will reduce, so that the net expected error over Δt sec will be some fraction (i.e., .4) of H'_{ϵ} . The error to be compensated should then be reduced by the expected error,

$$(23) \qquad \ddot{H}_{T_{\epsilon}} = \ddot{H}_{\epsilon} - .4 \cdot \ddot{H}'_{\epsilon}$$

By accumulating this error through a gain, i.e., .8,

$$(24) \qquad \ddot{H}_{BIAS} = \ddot{H}_{BIAS} + .8 \cdot \ddot{H}_{T_{\epsilon}}$$

and feeding back a corrected command into (15) only

$$(25) \qquad \ddot{H}_C = \ddot{H}_C \text{ (of (14))} - \ddot{H}_{BIAS}$$

the bias will be removed and the actual acceleration will converge on the desired level of (14).

Lateral Axis Guidance. - To guide the vehicle lateral position to the command reference trajectory with specified response characteristics of natural frequency (W_n) and damping ratio (ζ), a second order acceleration command is generated,

$$\ddot{Y}_C = W_n^2(Y_c - Y) + 2\zeta W_n(\dot{Y}_c - \dot{Y}) + \ddot{Y}_{ss}$$

The last term of this equation is the steady state acceleration required to hold the error signals of the first two terms at zero. This last term exists only for turn maneuvers and is the centripetal force,

(26) $$\ddot{Y}_{ss} = (V \cdot \text{Cos } \gamma)^2/R \text{ (radius of turn)}$$

The reference trajectory section of the guidance has selected the Y-axis direction so that the position and rate commands Y_c and $\dot{Y}_c$ are sero, ∴

(27) $$\ddot{Y}_c = -W_n^2 Y - 2\zeta W_n \dot{Y} + \ddot{Y}_{ss}$$

The number of terms of this equation that are utilized is phase dependent. The full equation is used for the final turn, but only the last term for initial turn and only second term for flight between turns. The acceleration and jerk computation phase dependency is shown in detail at the end of this appendix, but just the final turn phase will be discussed here.

The bank angle command is obtained by substituting command values into equation (12),

(28) $$\phi_{vc} = \text{Tan}^{-1} \frac{\ddot{Y}_c}{\left(\frac{\ddot{H}_c}{\text{Cos}\gamma} + g \text{ Cos}\gamma\right)}$$

Given γ and ϕ_{vc}, the local vertical component of a unit vector along the commanded direction of the wings is,

(29) $$\text{UYBLV} = -\text{Cos } \gamma \cdot \text{Sin } \phi_{vc}$$

The roll component of this vector for the standard aircraft sequence of ψ θ ϕ is,

(30) $$\phi_c = \text{Sin}^{-1}(-\text{UYBLV}/\text{Cos } \theta_c)$$

The rate of change of $\dot{Y}$ which is referenced to a rotating axis system,

$$\frac{d\dot{Y}}{dt} = \ddot{Y}_c - \ddot{Y}_{ss}$$

The rate of change of the acceleration command

(31) $$\dddot{Y}_c = -W_n^2\dot{Y} - 2\zeta W_n(\ddot{Y}_c - \ddot{Y}_{ss}) + \frac{2V \cdot \cos\gamma}{R}(\dot{V}\cos\gamma + \ddot{H}_c\tan\gamma)$$

Substituting this into equation (13) and neglecting the negligible last term,

(32) $$\dot{\phi}_{vc} = (\dddot{Y}_c - \tan\phi_{vc} \cdot \dddot{H}_c/\cos\gamma)\cos^2\phi_{vc}/(\frac{\ddot{H}_c}{\cos\gamma} + g\cos\gamma)$$

Projecting this on body axes,

(33) $$p_c = \dot{\phi}_{vc} \cdot \cos\alpha_c$$
$$r_c = \dot{\phi}_{vc} \cdot \sin\alpha_c$$

As done with the vertical axis, a bias acceleration error can be measured and compensated with feedback through the acceleration command,

(34) $$\ddot{Y}_\epsilon = (\dot{Y} - \dot{Y}_0 - \ddot{Y} \cdot \Delta t - \frac{\dddot{Y}_0\Delta t^2}{2})/\Delta t$$

(35) $$\ddot{Y}BIAS = \ddot{Y}BIAS + \frac{1}{2}\ddot{Y}_\epsilon$$

where the gain of $\frac{1}{2}$ is empirically determined for stable removal of errors

(36) $$\ddot{Y}_c = \ddot{Y}_c \text{ (of (27))} - \ddot{Y}BIAS$$

This compensated value of acceleration command is used only in equation (28).

The guidance to the reference trajectory equations and logic are now summarized for each axis.

Vertical Axis Guidance Equations

H_c and $\dot{H}_c$ from reference trajectory section of Guidance

$$\rho = .002378^{\frac{1}{.235}\ln(1 - 6.875 \cdot 10^{-6} \cdot H)}$$

$$\bar{q} = \tfrac{1}{2}\rho V^2$$

Flight Path Sign Conv.

(+) Down

$\dot{V} = (V - V_o)/\Delta t$, where Δt is comp cycle time; i.e., 2 sec

$$V_o = V$$

$$\ddot{H}_c = W_n{}^2(H_c - H) + 2\zeta W_n(\dot{H}_c - \dot{H})$$

$$\dddot{H}_c = W_n{}^2(\dot{H}_c - \dot{H}) - 2\zeta W_n(\ddot{H}_c)$$

$$\ddot{H}_\epsilon = (\dot{H} - \dot{H}_o - \ddot{H}_{c_o}\Delta t + \dot{V} \cdot \text{Sin}\,\gamma \cdot \Delta t - \tfrac{1}{2}\dddot{H}_{co}\Delta t^2)/\Delta t$$

$$\ddot{H}_{co} = \ddot{H}_c,\ \dddot{H}_{co} = \dddot{H}_c,\ \dot{H}_c = \dot{H}$$

$$\ddot{H}_\epsilon' = \frac{-\rho C_{LA} \cdot S \cdot \bar{q}}{M} \cdot (\theta_c - \theta_o)$$

$$\theta_o = \theta$$

$$\ddot{H}_{T_\epsilon} = \ddot{H}_\epsilon - .4 \cdot \ddot{H}_\epsilon'$$

If $(\ddot{H}_{T_\epsilon} \cdot \ddot{H}_\epsilon < 0)$, then $\ddot{H}_{T_\epsilon} = 0$.

If (Not at a phase change, and $\alpha_c < \alpha_{cmax}$),

then, $HBIAS = HBIAS + .8 \cdot \ddot{H}_{T_\epsilon}$

$$\ddot{H}_c = \ddot{H}_c - HBIAS$$

Note: Compute ϕ_{vc} from lat. guidance first for use here.

D-10

$$\left[\alpha_c = \frac{M}{S\,\bar{q}}\;\frac{(\ddot{H}_c/\mathrm{Cos}\,\gamma + g\,\mathrm{Cos}\,\gamma)}{P_{CLA}\cdot\mathrm{Cos}\,\phi_{vc}} - \frac{C_{LO}}{PCLA}\right]_{\text{Limit to }\alpha_{cmax}}$$

If (Flight director phase between turns and max L/D discrete from reference trajectory section of guidance), then $\alpha_c = \alpha_{cmax}$.

$$UXBCZ = -\mathrm{Sin}\,\alpha_c\cdot\mathrm{Cos}\,\phi_{vc}\,\mathrm{Cos}\,\gamma + \mathrm{Cos}\alpha_c\cdot\mathrm{Sin}\,\gamma$$

$$\theta_c = \mathrm{Sin}^{-1}\,(-UXBCZ)$$

$$q_c = \frac{\ddot{H}_{co}}{V}\cdot\frac{\mathrm{Cos}\,\phi_{vc}}{\mathrm{Cos}\,\gamma} + \frac{\dddot{H}_c M}{P_{CLA}\cdot\mathrm{Cos}\,\phi_{vc}\,S\,\bar{q}\,\mathrm{Cos}\,\gamma}$$

D-11

Lateral Axis Guidance Equations

Y_c, Y, $\dot{Y}_c$, and $\dot{Y}$ from reference trajectory section of guidance

Limit Y to ±2000 ft

Flight Path Sign Conv.
(+) DOWN

$\ddot{Y}_{ss} = \dddot{Y}_{ss} = 0$

If (Initial or final turn), then $\ddot{Y}_{ss} = (V \cdot \operatorname{Cos} \Upsilon)^2/R$,

and, $\dddot{Y}_{ss} = \frac{2V \operatorname{Cos} \Upsilon}{R} (\dot{V} \operatorname{Cos} \Upsilon + \ddot{H}_{co} \operatorname{Tan} \Upsilon)$

$\ddot{Y}_c = W_n^2 (Y_c - Y) + 2 \zeta W_n (\dot{Y}_c - \dot{Y})$

$\dddot{Y}_c = -2 \zeta W_n \cdot \ddot{Y}_c + \dddot{Y}_{ss}$

$\ddot{Y}_c = \ddot{Y}_c + \ddot{Y}_{ss}$

If (final turn or in FAP, and $/ Y / < 2000$ ft)

then, $\dddot{Y}_c = \dddot{Y}_c - W_n^2 \cdot \dot{Y}$

$\ddot{Y}_\epsilon = (\dot{Y} - \dot{Y}_o - \ddot{Y}_o \cdot \Delta t - \frac{\dddot{Y}_o \Delta t^2}{2}) / \Delta t$

$\dot{Y}_o = \dot{Y}$

$\ddot{Y}_o = \ddot{Y}_c$

$\dddot{Y}_o = \dddot{Y}_c$

$YBIAS = YBIAS + \frac{1}{8} \ddot{Y}_\epsilon$

If (initial turn or flight director phase) YBIAS = 0

$\ddot{Y}_c = \ddot{Y}_c - YBIAS$

If (initial or final turn and a left turn command from reference trajectory section of guid) then, $\ddot{Y}_c = -\ddot{Y}_o$ and $\dddot{Y}_c = -\dddot{Y}_c$

If (initial turn and an opposite turn command for three-turn meneuver),

then, $\ddot{Y}_c = -\ddot{Y}_c$ and $\dddot{Y}_c = -\dddot{Y}_c$

$$\emptyset_{vc} = \left[\mathrm{Tan}^{-1} \left(\frac{\ddot{Y}_c}{\ddot{H}_{co}/\mathrm{Cos}\ \gamma + g\ \mathrm{Cos}\ \gamma} \right) \right] \text{Limit to } \pm\emptyset_{max}$$

$$\mathrm{UYBCZ} = \mathrm{Sin}\ \emptyset_{vc} \cdot \mathrm{Cos}\ \gamma$$

$\emptyset_c = \mathrm{Sin}^{-1}\ (\mathrm{UYBCZ}/\mathrm{Cos}\ \theta_c)$ Note: Compute θ_c first, then use here.

$$\dot{\emptyset}_{vc} = (\dddot{Y}_c - \mathrm{Tan}\ \emptyset_{vc} \cdot \dddot{H}_c/\mathrm{Cos}\ \gamma)\ \mathrm{Cos}^2\ \emptyset_{vc}/(\ddot{H}_{co}/\mathrm{Cos}\ \gamma + g\ \mathrm{Cos}\ \gamma)$$

$$p_c = \dot{\emptyset}_{vc} \cdot \mathrm{Cos}\ \alpha_c,\ r_c = \dot{\emptyset}_{vc} \cdot \mathrm{Sin}\ \alpha_c$$

9.15.2.1 Approach Guidance (cont'd)

TABLE 1. - SIMULATION INPUT DATA FOR DEMONSTRATION RUNS

		VEHICLE							
		STRAIGHT WING ORBITER				DELTA WING ORBITER			
Initial Conditions	H	20000 ft				41000 ft			
	V_G	562 ft/sec				703 ft/sec			
	γ	7 deg				6 deg			
Guid. Knowledge of Trim Aero.	Term. Area CL	$.1954 + 4.60257 \cdot \alpha_{(RAD)}$				$-.026746 + 1.958083 \cdot \alpha_{(RAD)}$			
	Final Appr. CL	$.1954 + 4.60257 \cdot \alpha_{(RAD)}$				$-.014995 + 1.936128 \cdot \alpha_{(RAD)}$			
	S	1175 ft^2				6650 ft^2			
	M	4041 slugs				8330 slugs			
Guid. Targeting	Phase	IT	FD	FT	Final Appr.	IT	FD	FT	Final Approach
	α_N	7.5°	4°	6°	4.5°	7.5°	5°	6°	6.3°
	γ_{SS}	8°	7.2°	8°	10°	7.46°	7.83°	8.22°	14.5°
	Speed Brakes	0	0	0	30°	0	0	0	20°
	Landg. Gear	Up	Up	Up	Down	Up	Up	Up	Down
	Turn Radius	9200 ft				12000 ft			
MEP	X	-21000 ft				-22000 ft			
	h	2600 ft				4200 ft			
Other Guid. Items	Min $\ddot{H}_c$	$-\frac{1}{2}$ g				$-\frac{1}{2}$ g			
	Max α_c	8°				8°			
	Max Bank Cmd.	46°				46°			
	H @ which ϕ_{max} for turn	≥ 20000 ft				≥ 20000 ft			

SPACE SHUTTLE

GN&C SOFTWARE EQUATION SUBMITTAL

Software Equation Section Final Approach Guidance Submittal No. 29

Function: Lateral and Longitudinal Autoland Guidance

Module No. OG7 Function No. 1, unpowered (MSC 03690 Rev. B)

Submitted by: D. Dyer (Name) Co. EG2

Date: August 24 and October 21, 1971

NASA Contact: J. Suddath (Name) Organization EG2

Approved by Panel III K. J. Cox (Chairman) Date 10/21/71

Summary Description: Pitch rate and speed-brake commands are computed and used to control in-plane approach. Lateral position error and its integral plus heading-angle error are used to form vehicle roll command.

Shuttle Configuration: (Vehicle, Aero Data, Sensor, Et Cetera)

NR161C orbiter aero data

Comments:

(Design Status)

(Verification Status)

Panel Comments:

9.15.2.2 Final Approach Guidance

1. Introduction

These equations are submitted as candidates to fulfill the unpowered Final Approach Guidance requirements for the space-shuttle Orbiter. They include Autoland lateral and longitudinal guidance equations. The scheme is all-inertial; navigation aids are used only to update the navigated vehicle state. Pitch rate and speed-brake commands are computed and issued to control in-plane approach. Lateral position error and its integral plus heading-angle error are used to form the vehicle roll command. (There is no decrab or wings level manuever; the assumption is made that the gear is designed to accommodate the stress for crabbed landings in design winds.)

2. Functional Diagram

A general description of Autoland Guidance is contained in Ref. 1. Figure 1 is a functional diagram. Fig. 2 is a block diagram. (For general information, the autopilots being used in simulation runs are included in Fig. 2.)

Inputs to the Guidance module (OG-7) are from the Final Approach and Guidance Navigation module (ON-5); the inputs are the navigated state in the Earth-fixed landing coordinate system. From this are calculated the range to touchdown target, altitude, inertial velocity magnitude, inertial flight-path angle, lateral position and heading angle. Outputs are pitch rate command, speed-brake position command and vehicle roll command to the autopilot. The guidance roll command drives a roll-rate aileron-autopilot inner loop with roll attitude outer loop. Roll rate command is interconnected to a rate command rudder autopilot with turn coordination and normal acceleration inputs. The acceleration and heading-angle signals are instrumental in holding the orbiter to the final approach plane in crosswinds.

9.15.2.2 Final Approach Guidance (cont'd)

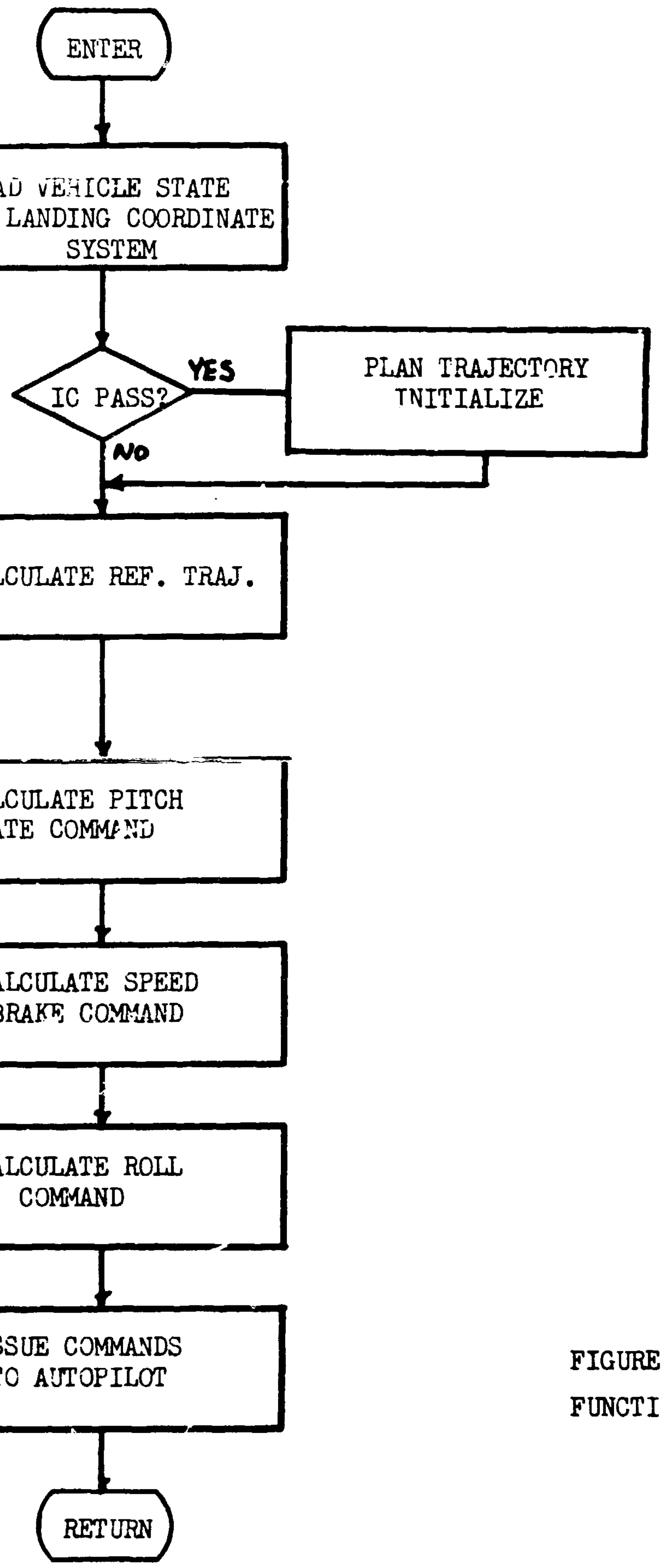

ENTER
READ VEHICLE STATE IN LANDING COORDINATE SYSTEM
IC PASS?
YES
NO
PLAN TRAJECTORY INITIALIZE
CALCULATE REF. TRAJ.
CALCULATE PITCH RATE COMMAND
CALCULATE SPEED BRAKE COMMAND
CALCULATE ROLL COMMAND
ISSUE COMMANDS TO AUTOPILOT
RETURN

FIGURE I

FUNCTIONAL DIAGRAM

9.15.2.2 Final Approach Guidance (cont'd)

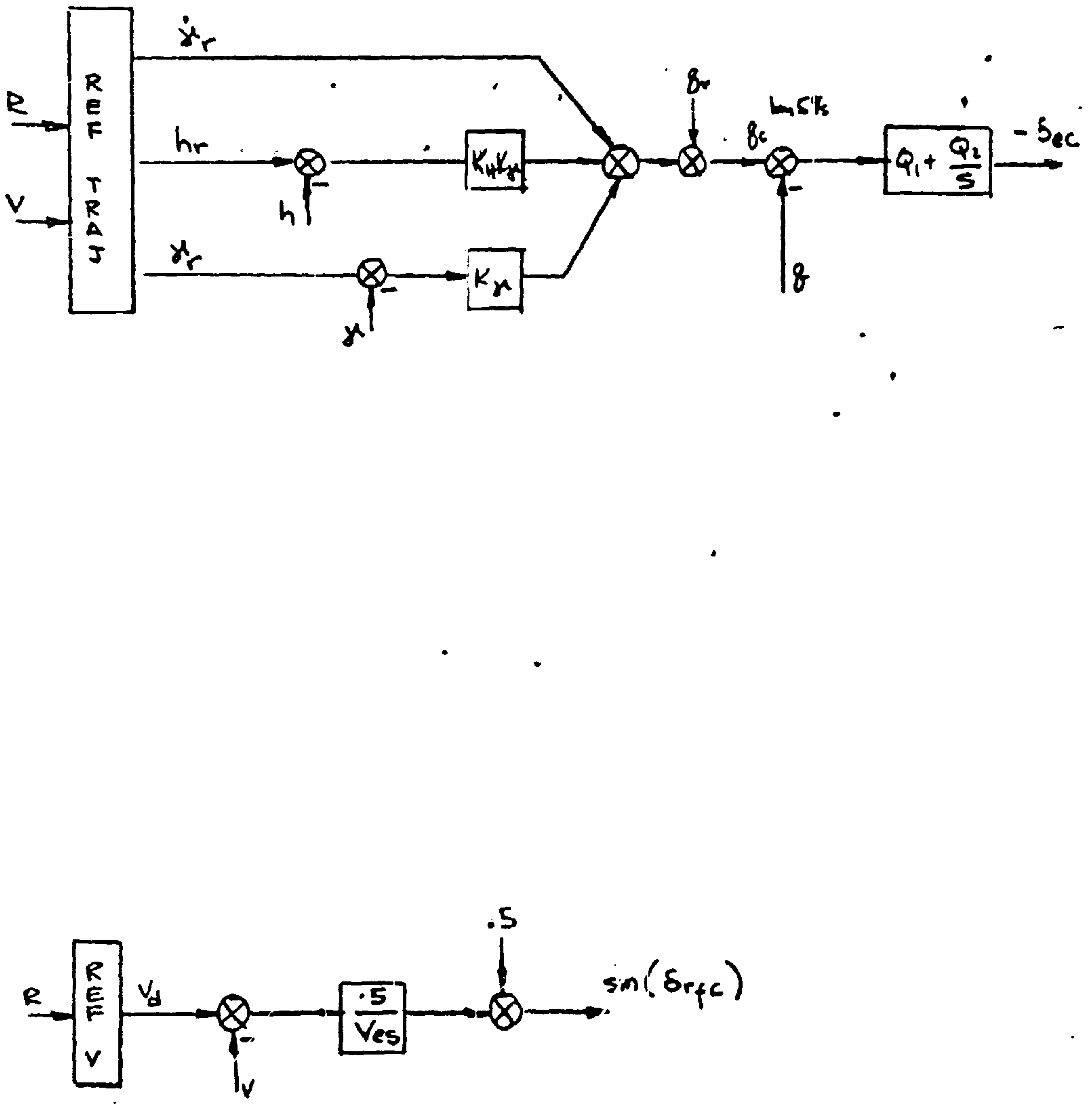


Figure 2a Block Diagram, Longitudinal

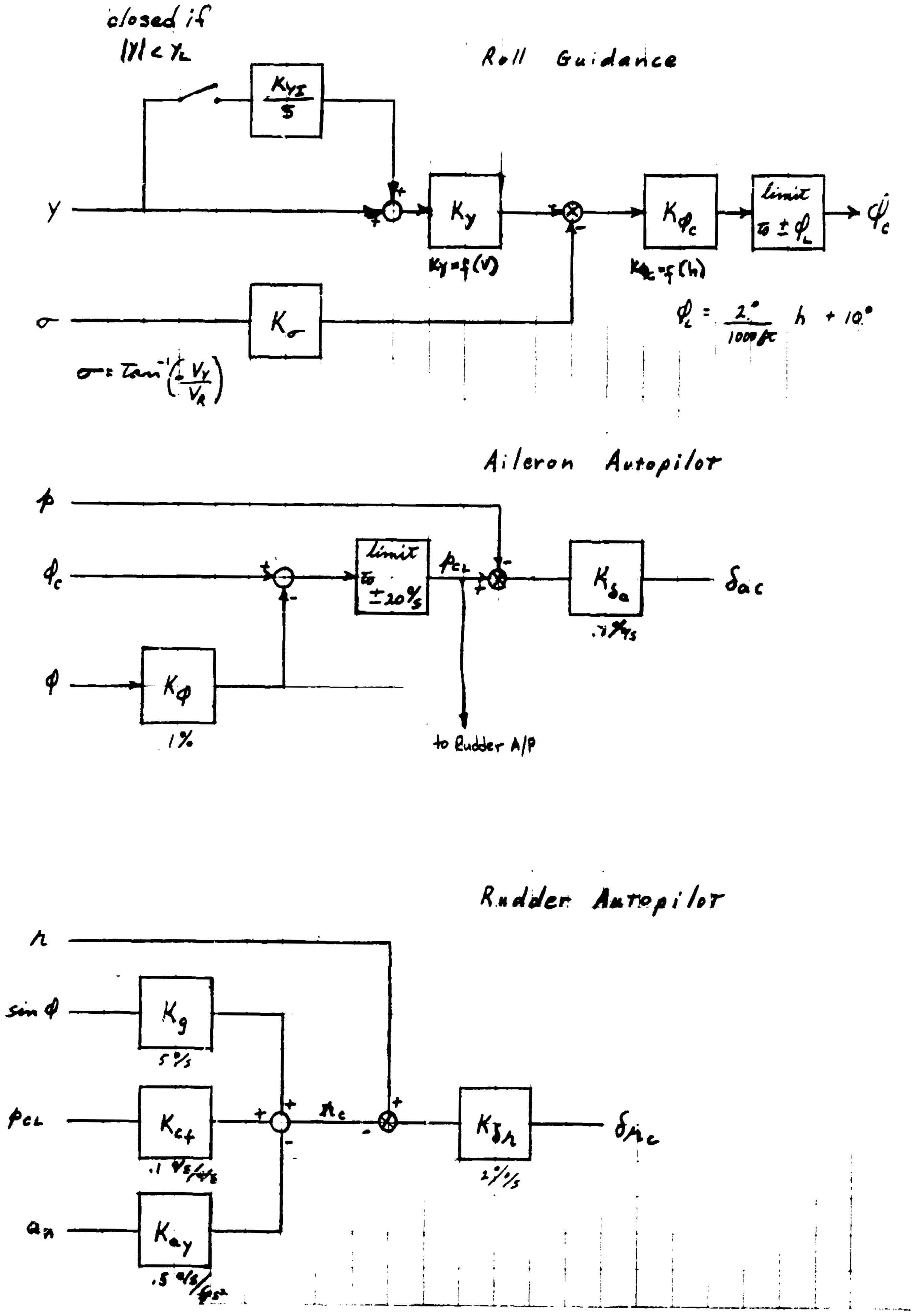


Figure 2b Block Diagram, Lateral

3. Coordinate System

The autoland guidance uses vehicle position and velocity relative to a runway coordinate system,as shown in Figure 3. Figure 3 also indicates longitudinal sign convention for the equations. The "altitude of the IMU" at touchdown is represented in the equations as c.g. altitude.

4. Equations and Flow

4.1 The longitudinal guidance equations are presented in Figure 4.

The guidance routine (as currently implemented) is entered four times per second. (This frequency is to be verified by planned guidance/autopilot interaction and wind gust response studies.) Stability of the guidance equations and minimization of the vehicle rotational motions set the minimum rate. There are no switching requirements in the scheme.

The vehicle's state is in a landing coordinate system. On the initial pass, the target sink rate, touchdown ground speed, and shallow flight path angle are used to compute reference trajectory constants. The reference trajectory subroutine is entered with the targeted range at pullup completion, and flare and touchdown constants, to find the reference altitude at pullup completion. At TARGET, a steep flight path angle is computed which will carry the vehicle from its current altitude through pullup onto the shallow slope. This capability may also be used to replan the reference trajectory during the steep phase to minimize transients from large navigation updates or to replan when forced by emergency conditions. A linear reference velocity is defined to carry the vehicle from its current velocity to its targeted value at the completion of the pullup.

9.15.2.2 Final Approach Guidance (cont'd)

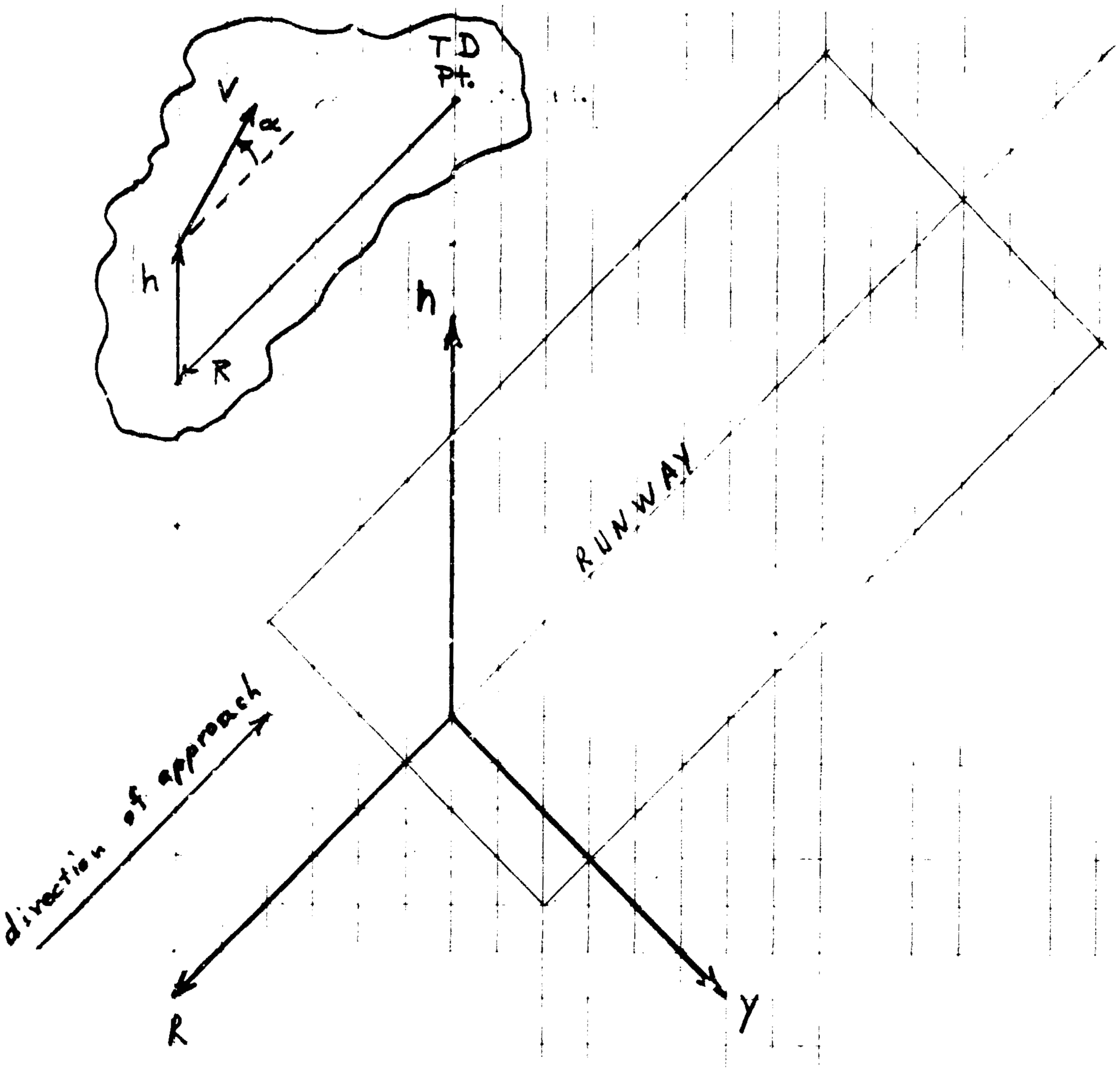


Figure 3. Runway Coordinate System

	NAME	DATE	LOCKHEED ELECTRONICS COMPANY HOUSTON AEROSPACE SYSTEMS DIVISION	Page	TEMP	PERM
Prepared	ALDER	8/19				
Checked			TITLE Fig 4a GUIDANCE EQNS (1)	Model:		
Approved				Report No.		

Enter guidance routine each dt_g (.25 sec)

Read vehicle state R, h, V, γ

go to IC pass one time.

<u>Reference Trajectory</u>

```
GO TO:  S&P  IF  PHASE = STEEP
        S&F  IF  PHASE = SHALLOW

TARGET    $\sigma_p = \frac{R-R_p}{\rho_p}$

          $m_p = \left( \frac{h-h_p}{\rho_p} - m_{bp}\sigma_p \right) / \left( \sigma_p - 2.5066 \, ERF(\sigma_p) \right)$
          $R_b = R_p$ , $V_b = V_p$ , $dvdr = (V-V_b)/(R-R_b)$
S&P       CALL RFTRAJ ( $V$, $R-R_p$ , $\rho_p$, $m_p$, $m_{bp}$, $h_p$, $h_r$, $\gamma_r$, $\dot{\gamma}_r$ )

          IF ( $R<R_p$ ) PHASE = SHALLOW , $dvdr = \frac{V-V_{td}}{R}$ , $R_b = 0$, $V_b = V_{td}$ , $g_v = .0052$
          GO TO LOOPS

S&F       CALL RFTRAJ ( $V$, $R$, $\rho_f$ , $m_f$, $m_{bf}$, $cg$ , $h_r$, $\gamma_r$, $\dot{\gamma}_r$ )
          $\dot{\gamma}_r = 0.$
          GO TO LOOPS
```

9-16-65

	NAME	DATE	LOCKHEED ELECTRONICS COMPANY HOUSTON AEROSPACE SYSTEMS DIVISION	Page	TEMP	PERM
Prepared	A. [illegible]	[illegible]				
Checked			TITLE FIG 4b GUIDANCE EQNS (2)	Model		
Approved				Report No.		

GUIDANCE LOOPS

LOOPS $K_\gamma = 1.$ $\quad K_H = \frac{K_{HV}}{V} = \frac{.25}{V}$

$$g_c = \left[K_\gamma(\gamma_r - \gamma) + K_\gamma K_H(h_r - h) + g_v\right]_{\lim \pm g_{lim}}$$

Speed brakes

$V_d = dvdr\,(R - R_b) + V_b \qquad V_d \geq V_b$

$\delta_{rfc} = \sin^{-1}\left[.5 + \frac{.5}{V_{es}}(V - V_d)\right]$

If $(h \leq h_{fb})$ freeze brake

Issue cmnds to A/P: g_c, δ_{rfc}

Return

	NAME	DATE	LOCKHEED ELECTRONICS COMPANY HOUSTON AEROSPACE SYSTEMS DIVISION	Page	TEMP.	PERM
Prepared	A. Diaz	9-15				
Checked			TITLE FIG 4C GUIDANCE EQNS (3)	Model		
Approved				Report No.		

IC PASS

PHASE = STEEP

$m_{bf} = -\dot{h}_{td} / V_{td}$

$m_f = -\gamma_2 - m_{bf}$

$m_{bp} = -\gamma_2$

CALL RFTRAJ(– , R_p , ρ_f , m_f, m_{bf} , cg, h_p , – , –) gets h_p

go to TARGET

9-15-67

	NAME	DATE	LOCKHEED ELECTRONICS COMPANY HOUSTON AEROSPACE SYSTEMS DIVISION	Page	TEMP	PERM
Prepared	Ann	0/7				
Checked			TITLE FIG 4d	Model		
Approved			REFERENCE TRAJECTORY EQNS	Report No.		

SUBROUTINE RFTRAJ$(\underbrace{V, R, \rho, m : m_b, h_b}_{\text{inputs}},\ \underbrace{h_r, \gamma_r, \dot{\gamma}_r}_{\text{outputs}})$

Input variables are not changed

$b_1 = .99729$ $\quad b_4 = .08496$

$b_2 = .41156$ $\quad b_5 = .023008$

$b_3 = -.25434$

IF $(R \leq 0)$ $m = 0$

$\sigma = R/\rho \qquad \omega = \frac{\pi\sigma}{5}$

$m_e = m e^{(-.5\sigma^2)}$

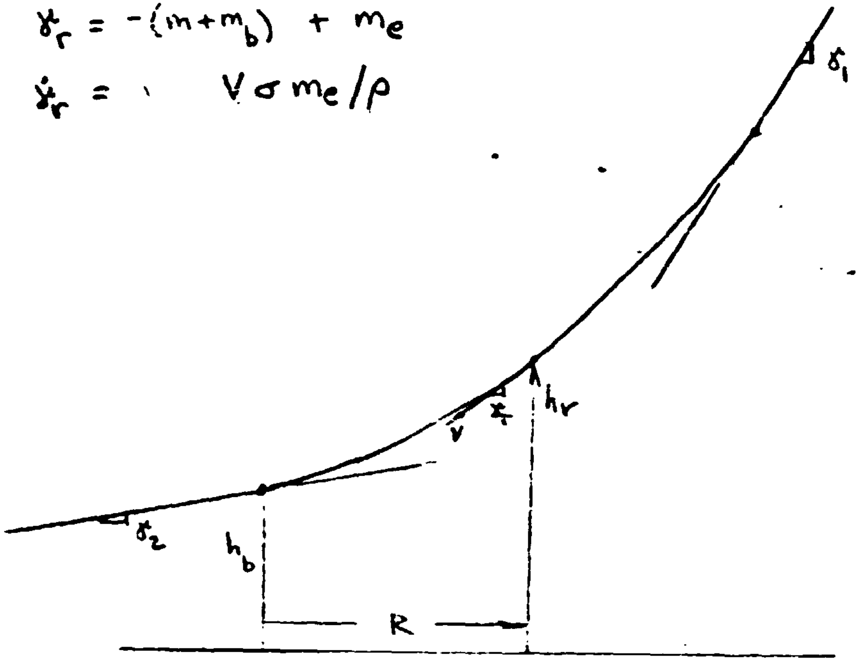
$$erf(\sigma) = \sigma/10 + \frac{\sin\omega}{\pi}\left\{b_1 + \cos\omega\left[b_2 + b_4\cos^2\omega\right] + \sin^2\omega\left[b_3 + b_5\sin^2\omega\right]\right\}$$

$erf(\sigma > 4) = .5 \qquad erf(-) = 0.$

$h_r = R(m + m_b) - \sqrt{2\pi}\, m\rho\, erf(\sigma) + h_b$

$\gamma_r = -(m + m_b) + m_e$

$\dot{\gamma}_r = V\sigma m_e/\rho$



$m_b = -\gamma_2$ $\qquad$ γ's are negative, small angles

$m = -(\gamma_1 - \gamma_2)$

9.15-68

	NAME	DATE	LOCKHEED ELECTRONICS COMPANY HOUSTON AEROSPACE SYSTEMS DIVISION	Page	TEMP	PFRM
Prepared	[illegible]	[illegible]				
Checked			TITLE Fig 4e Reference Traj. Derivation	Model		
Approved				Report No.		

$$\mathrm{erf}\left(\frac{R}{\rho}\right)=\frac{1}{\sqrt{2\pi}}\int^{R/\rho} e^{-\lambda^2/2}\,d\lambda$$

$$h(R) = R(m+m_b) - m\,\rho\int_0^{R/\rho} e^{-\lambda^2/2}\,d\lambda \;+ h_b$$

$$\gamma(R) = -\frac{dh}{dR} = -(m+m_b) + m\,\rho\, e^{-\frac{1}{2}\left(\frac{R}{\rho}\right)^2}(1/\rho) \;=\; -(m+m_b) + m\,e^{-\frac{1}{2}\left(\frac{R}{\rho}\right)^2}$$

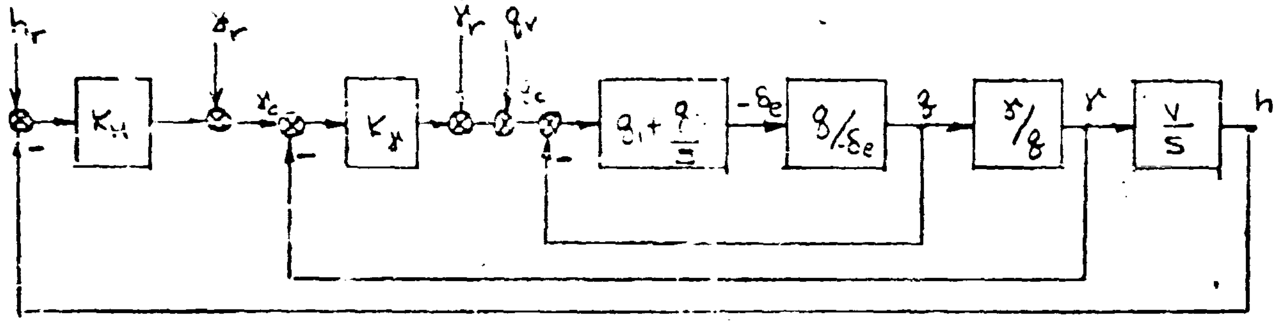
$$\dot\gamma(R) = \frac{d\gamma}{dR}\,\frac{dR}{dt} = \frac{mR}{\rho^2}\,e^{-\frac{1}{2}\left(\frac{R}{\rho}\right)^2} V$$

small angle

- ~~constant vel~~

- signs of Fig 3

Prepared	ALDER	SEPT 71	LOCKHEED ELECTRONICS COMPANY HOUSTON AEROSPACE SYSTEMS DIVISION	Page		
Checked			TITLE FIG 4f	Model		
Approved			LINEAR REPRESENTATION	Report No		



V=350 fps

$$q/-\delta_e = \frac{4.5\left(\frac{s}{.68}+1\right)}{\frac{s^2}{.44^2} + \frac{2(.61)}{.44}s + 1} \qquad \frac{\gamma}{q} = \frac{\left(\frac{s}{1.75}+1\right)\left(\frac{s}{-2.0}+1\right)}{s\left(\frac{s}{.68}+1\right)}$$

$$q/q_c = \frac{\left(\frac{s}{.33}+1\right)\left(\frac{s}{.68}+1\right)}{\left(\frac{s}{.31}+1\right)\left(\frac{s}{.89}+1\right)\left(\frac{s}{3.2}+1\right)} \qquad g_1=3,\ g_2=1$$

$$\gamma/\gamma_c = \frac{\left(\frac{s}{.33}+1\right)\left(\frac{s}{1.75}+1\right)\left(\frac{s}{-2.0}+1\right)}{\left(\frac{s}{.34}+1\right)\left(\frac{s}{2.5}+1\right)\left(\frac{s^2}{1^2}+\frac{2(.31)}{1}s+1\right)} \qquad K_\gamma=1$$

$$h/h_r = \frac{\left(\frac{s}{.33}+1\right)\left(\frac{s}{1.75}+1\right)\left(\frac{s}{-2.0}+1\right)}{\left(\frac{s}{.23}+1\right)\left(\frac{s}{.36}+1\right)\left(\frac{s}{2.59}+1\right)\left(\frac{s^2}{.9^2}+\frac{2(.25)}{.9}s+1\right)} \qquad K_\gamma=1,\ K_H=\frac{.2}{V}$$

9-15-70

9.15.2.2 <u>Final Approach Guidance</u> (cont'd)

On a normal call, after the vehicle state is determined, if phase is steep the reference trajectory is defined through pullup. When the pullup-complete range has been reached, phase is set to shallow and a new reference velocity profile is defined to decelerate the vehicle to its targeted touchdown velocity. The two velocity slopes approximate the expected vehicle deleleration along the steep and shallow glide paths with gear down and half speed brakes. The switch to shallow is not critical since the two portions of the reference trajectory overlap smoothly.

At LOOPS, velocity dependent gains are computed. A pitch rate command is formed and limited. A speed brake command is formed such that zero error gets half brakes and both commands are issued to the autopilot.

The reference trajectory subroutine is entered with one of two sets of targets; pullup over and touchdown. Calling arguments are as follows:

V - Current vehicle velocity, used for $\dot{\gamma}$.

R - Range to go to end of current maneuver. Pullup, $R-R_b$, and touchdown, R.

ρ - Constant which sets maneuver curvature, normal acceleration level. ρ_p , ρ_s .

m - Negative of difference between flight path angles entering and leaving the maneuver. Steep minus shallow and shallow minus touchdown.

m_b- Negative of flight path angle leaving the maneuver. Shallow glide and touchdown glide slope.

h_b- Altitude at which maneuver is complete and vehicle flying $-m_b$. Altitude at pullup complete and cg height at touchdown.

9.15.2.2 Final Approach Guidance (cont'd)

h_r - Reference altitude

γ_r - Reference flight path angle

$\dot{\gamma}_r$ - Reference flight path angle rate

ERF is an approximation to the error function devised in Ref. 2. Its use is illustrated in Fig. 4.

Linear approximations of the system transfer functions are presented in Fig. 4.

4.2 The lateral guidance equations are presented in Fig. 5. On the initial pass, the roll gain, crossrange integral gain, and the heading gain are stored. On a normal pass the crossrange gain, K_y, is calculated as a function of velocity. When altitude becomes less than 50 ft, the roll command gain is decreased from 1 to .5 over a 2-second period. The roll command is the sum of a crossrange, integral of crossrange, and velocity heading angle term. It is limited and issued to the autopilot.

5. Constants/Variables Summary

Figure 6 summarizes variables and constants. Precision and quantization requirements have not been finalized. Figure 7 describes the principal variables. Figure 8 lists typical input values.

6. FORTRAN Coding

The equations have been coded in FORTRAN for simulation on the SSFS (Space Shuttle Functional Simulator).

7. Simulation Status

The scheme is operational on the SSFS.

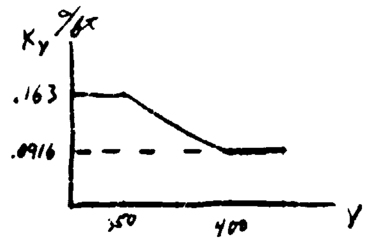
Enter guidance routine each dt_g (.25 sec)

go through IC on first pass only

Read Vehicle state: h, Y, and σ

GAINS

$K_y = -2.49 \times 10^{-5}\, V + 1.16 \times 10^{-2}$



IF($V > 400.$ fps) $K_y = 1.599 \times 10^{-3}$
IF($V < 350.$ fps) $K_y = 2.845 \times 10^{-3}$

IF($h < 50.$ ft) $K_{\phi_c} = K_{\phi_c} - \frac{1}{4}\, dt_g$ — K_{ϕ_c} goes from 1 to .5
IF($K_{\phi_c} < .5$) $K_{\phi_c} = .5$ — in 2 sec at $h = 50$ ft.

CROSS R

$\phi_L = 3.49 \times 10^{-5}\, h + .1745$ — ϕ_L is 10° + 2° per 1000 feet of altitude

IF (|y| < Y_L) $Y_I = \int Y\, dt$

$\phi_c = K_{\phi_c} \{ -K_\sigma \sigma - K_y (Y + K_{YI}\, Y_I) \}$

IF($\phi_c > \phi_L$ or $\phi_c < -\phi_L$) $\phi_c = \dfrac{\phi_L \cdot \phi_c}{|\phi_c|}$

Issue ϕ_c to A/P

Return

IC

$K_{\phi c} = 1$
$K_{YI} = 1/15$
$K_\sigma = 6$

Figure 5 Lateral Guidance Equations

9-15-74

FIG 6a CONSTANTS/VARIABLES SUMMARY ①

SYMBOL	MNEMONIC	UNITS	NOMINAL VALUE OR EXPECTED RANGE	PROGRAMMED RANGE ± number or ZERO to number					
R	R	ft	-2000, 20,000	± 50,000					
h	Y	ft	0 , 2000	20,000					
V	V	ft/sec	250, 500	1000					
γ	GAMRAD	rad	0, -.35	± 1					
Δt_g	DELTAT	sec.	.25	10					
σ_p	SIGMAP	—	0, 4	10					
R_p	RP3	ft	7500	20,000					
h_p	FEP3	ft	400	10,000					
V_p	VP3	ft/sec	400	1,000					
ρ_p	PRHO	ft	1500	10,000					
ρ_f	FRHO	ft	1000	10,000					
	ERFP	–	0, 4	10.					
m_p	PM	—	.26	± 1					
m_f	FM	—	.06	± 1					
m_{bp}	PMB	—	.06	± 1					
m_{bf}	FMB	—	.011	± 1					
c_g	CGBIAS	ft	18	100					
h_r	FE	ft	0, 2000	20000					
γ_r	GAMR	rad	0, -.35	± 1					
$\dot{\gamma}_r$	GAMDTR	rad/sec	0, .035	1					
$\dot{h}_{td}$	HDTD	ft/sec	-3	-10					
γ_2	GAM2	rad	-.06	-.1					

DATA SHEET

9.15.2.2 Final Approach Guidance (cont'd)

Fig 6b

SYMBOL	MNEMONIC	UNITS	NOM. VALUE	RANGE					
K_γ	GKG	1/sec	1.5	10					
K_h	GKH	1/ft	~~.0006~~, .0009	.01					
K_{hv}	GKHV	1/sec	.25	10					
q_c	QC	rad/sec	±.1	±.35					
q_{clim}	QCLIM	rad/sec	.088	.35					
	INIT	–	0, 1	1					
	PHASE	–	1, 2, 3, 4	10					
q_v	QCV	rad/sec	.0052	.1					
	B(1)	–	.99729	±1					
	B(2)	–	.41156	±1					
	B(3)	–	–.25434	±1					
	B(4)	–	.08496	±1					
	B(5)	–	.023008	±1					
R_b	RRFC	ft	7500, 0	20000					
V_b	VRFC	ft/sec	400, 270	1000					
dvdr	DVDR	1/sec	.013, .017	.1					
V_{td}	VZERO	ft/sec	270	500					
V_d	VD	ft/sec	270, 400	500					
δr_{fc}	RFC, ORFC	rad	0, 1.5	2					
V_{es}	RFCDV	ft/sec	7	100					
h_{fb}	–	ft	50	200					

9.15.2.5

DATA SHEET

9.15.2.2 Final Approach Guidance (cont'd)

SYMBOL	MNEMONIC	UNITS	Nominal Value or Expected Range	Range ± number or ZERO to number
h	Y	ft	0, 2000 -	20000
y	CR	ft	± 200.	± 1000.
y_L	CRSWI	ft	50.	1000
K_y	$G^{Y}CR$	rad/ft	.0016, .0028	.005
σ	HEDING	rad	± .1	± 2.
K_σ	GKHED	rad/rad	6.	20.
ϕ_c	PHIC	rad	± .244	± 2.
ϕ_L	PHICL	rad	.175, .244	2.
K_{ϕ_c}	GKPHIC	none	.5, 1.	1.
K_{yI}	—	sec^{-1}	.0886	1.

Figure 6c Constants/Variable Summary

	NAME	DATE	LOCKHEED ELECTRONICS COMPANY HOUSTON AEROSPACE SYSTEMS DIVISION	Page 1	TEMP.	PERM
Prepared						
Checked			TITLE FIG 7a VARIABLE DESCRIPTION (1)	Model		
Approved				Report No.		

R Range to go to touchdown point, ft

h Vehicle altitude, ft

V Vehicle velocity, fps

γ Inertial flight path angle, rad

dt_g Guidance solution frequency, sec. Not used in calc.

ρ_p, ρ_f Ref traj. constants. Sets pullup and flare g-levels.

m_p, m_f Ref. traj. constants. Negative of steep minus shallow and shallow minus touchdown flight path angles.

m_{bp}, m_{bf} Ref traj. constants, Negative of ref flight path angles after pullup and at touchdown.

σ_p Function ERF calling argument, range to end of pullup normalized to ρ_p

	NAME	DATE	LOCKHEED ELECTRONICS COMPANY HOUSTON AEROSPACE SYSTEMS DIVISION	2 Page	TEMP.	PE 4
Prepared						
Checked			TITLE FIG 7b VARIABLE DESCRIPTION (2)	Model		
Approved				Report No.		

R_p, V_p Desired R, V at end of pullup

h_p Ref alt at R_p, V_p

cg Alt. of cg at touchdown

$\dot{h}_{td}$ Target sink rate at touchdown, (−2fps)
V_{td} Target inertial velocity magnitude at touchdown
γ_2 Ref shallow flight path angle, rad (−3.5°)

h_r Ref alt
γ_r Ref flight path angle
$\dot{\gamma}_r$ Ref flight path angle rate

V_d Ref. inertial vel. magnitude

V_b, R_b Target velocity, range in ref. vel. function

dvdr V_d slope fps/ft

V_{es} Vel. error to saturate speed brakes
h_{fb} Altitude to freeze brakes
δ_{rfc} Flared rudder brake Cmm'd (included angle)

K_γ, K_H Guidance error gains, 1/sec, 1/ft

K_{HV} Input gam constant, 1/sec

q_c Pitch rate cmm'd, rad/sec

q_{clim} Pitch rate cmm'd limit, rad/sec

q_v Constant pitch rate cmm'd during shallow approach & flare, rad/sec

9-15-78

9.15.2.2 <u>Final Approach Guidance</u> (cont'd)

h	Vehicle altitude, ft.
Y	Vehicle lateral displacement from runway center line, ft.
V	Vehicle velocity, fps.
σ	Velocity heading angle $\tan \sigma = - \frac{V_Y}{V_R}$, rad.
Y_L	Lateral displacement integral threshold, ft.
ϕ_c	Roll command, rad.
ϕ_L	Roll command limit, rad.
K_Y, K_{YI}, K_σ	Guidance gains, rad/ft, 1/sec, rad/rad.

Figure 7c Variable Description

9.15.2.2 Final Approach Guidance (cont'd)

	NAME	DATE	LOCKHEED ELECTRONICS COMPANY HOUSTON AEROSPACE SYSTEMS DIVISION	Page	TEMP	PERM
Prepared						
Checked			TITLE FIG 8 INPUT VALUES - NOMINAL TRAJ.	Model		
Approved				Report No.		

Ref Traj

dt_g = .25 sec

V_{td} = 270 fps

γ_2 = -.061 rad (-3.5°)

$\dot{h}_{td}$ = - 3 fps

ρ_p = 1500 ft

ρ_f = 1000 ft

c_g = 18 ft.

R_p = 7500 ft.

V_p = 400 fps

Gains

g_{clim} = .088 rad/sec (5°/s)

V_{es} = 7 fps

h_{fb} = 50 ft

K_μ = 1 1/sec

K_{hV} = .25 1/sec

g_v = .0052 rad/sec

Vehicle Initial Conditions

R = 15000 ft

h = 2000 ft

V = 500 fps

γ = -.262 rad (-15°)

9.15.2.2 Final Approach Guidance (cont'd)

8. Testing Status

A touchdown dispersion study has been performed. Some results are shown in Figure 9. For tests beginning at 1740 ft. altitude and 15,000 ft. range, crossrange dispersion in ± 10 ft. and crossrange rate dispersion is ±2fps. Vehicle yaw at touchdown is determined essentially by the crosswind. A 20 -knot wind results in a 7.2 -degree yaw.

9. Derivation

Lateral guidance is basically an all-inertial version of Ref. 5.

10. Assumptions

NR 161C delta wing aero and mass properties (Ref. 4).
Unpowered landing
Error Models (per Ref. 3).

11. Further Activity

The integral-of-crossrange term in the lateral guidance limits steady state errors. Unfortunately, it also retards response to dynamic errors. Simulation runs indicate that improved performance may result from freezing(clamping) the integral at some point before touchdown.

Figure 9. Touchdown Dispersions

9.15.2.2 Final Approach ~~[illegible]~~ (cont'd)

Case	OFF NOMINAL INITIAL [1] CONDITIONS	WINDS [2]	CHANGES IN VEHICLE CHARACTERISTICS	Touchdown			
				y ft	$\dot{y}$ fps	yaw deg	roll deg
1	y = 1000 ft	—	—	-9.8	.7	.4	1.6
2	$\dot{y}$ = 53 fps	—	—	-12.3	.4	.1	.2
3		cross	—	4.6	-1.5	7.2	1.1
4	y = 1000 ft	—	.9 Lift	-10.2	.0	.5	2.2
5		cross	.5 $\dot{\delta}_{r max}$ [3]	-8.5	-.9	7.1	1.1
6		cross	.5 $\dot{\delta}_{e max}$ [4]	-8.9	-.9	7.1	1.1
7		cross	.9 Mass	4.3	-1.5	7.3	1.2

[1] The Nominal Initial Conditions are R = 15000 ft; Y = 0; h = 1740 ft; V = 500 fps; $\dot{y}$ = 0 fps

[2] 40 knot wind above 500 ft shearing to 20 knots at touchdown

[3] $\dot{\delta}_{r max}$ is the maximum rudder rate (25°/s)

[4] $\dot{\delta}_{e max}$ is the maximum elevon rate (28°/s)

9.15.2.2 Final Approach Guidance (cont'd)

REFERENCES

1. "High Crossrange Shuttle Autoland Guidance," LEC Memo 67542B/123503, K. Alder.

2. "An Analytical Approximation for erf(t)," MSC Memo EG2-70-145, J. H. Suddath.

3. Touchdown Performance, EG2 and Sperry Shuttle Autoland Guidance, MSC Internal Note (in preparation), D. A. Dyer.

4. "Delta-Wing Orbiter Vehicle Model for CRALS," LEC Memo SA3071-0020, J. Vanelli.

5. "Space Shuttle Data Flight Systems, Part II Orbiter," MDC E0334, 30 June 1971.