NASA Technical Paper 1700

Development of the Reentry Flight Dynamics Simulator for Evaluation of Space Shuttle Orbiter Entry Systems

Lawrence F. Rowell, Richard W. Powell,
and Howard W. Stone, Jr.

OCTOBER 1980

NASA

NASA Technical Paper 1700

Development of the Reentry Flight Dynamics Simulator for Evaluation of Space Shuttle Orbiter Entry Systems

Lawrence F. Rowell, Richard W. Powell,
and Howard W. Stone, Jr.
Langley Research Center
Hampton, Virginia

National Aeronautics
and Space Administration

**Scientific and Technical
Information Branch**

1980

CONTENTS

SUMMARY

The reentry flight dynamics simulator (RFDS) developed by the NASA Langley Research Center is a nonlinear, six-degree-of-freedom, digital-computer simulation of a vehicle which has constant mass properties and whose attitudes are controlled by both aerodynamic surfaces and reaction-control-system thrusters. A rotating, oblate Earth model was used to describe the gravitational forces which affect long-duration Earth entry trajectories. This program can be executed in a nonreal-time mode or connected to a simulation cockpit to conduct piloted and autopilot studies. The RFDS is being used to evaluate the onboard guidance and control software (November 1976 version) used by the Space Shuttle Orbiter for its descent from approximately 121.9 km to touchdown on the runway.

INTRODUCTION

The reentry flight dynamics simulator (RFDS) developed by the NASA Langley Research Center (LaRC) is a nonlinear, six-degree-of-freedom, digital simulation of a vehicle which has constant mass properties (no main engine thrust) and whose attitudes are controlled by both aerodynamic surfaces and reaction control system (RCS) thrusters. A rotating, oblate Earth was modeled to describe accurately the gravitational forces which affect long-duration Earth entry trajectories, and provisions have been made for the application of steady-state winds. This program can be connected to simulation hardware using the LaRC real-time simulation subsystem (RTSS) to conduct autopilot or pilot-in-the-loop studies in a cockpit mockup. This program may also be used to conduct studies in a nonreal-time (batch) mode. The RFDS is an extended version of the program described in reference 1 which did not have the capability for piloted studies. In addition to the cockpit capability, this version also has substantially different control and guidance systems, a new aerodynamic data package, expanded data-analysis capabilities, and several new models such as those defining the actuators and the reaction control system.

The RFDS is being used in an independent assessment of the onboard guidance and control software used by the Space Shuttle Orbiter for its descent from the approximate atmospheric interface at 121.9 km (400 000 ft) altitude down to runway touchdown. The specific mission being studied is the entry from the first orbital flight of the Space Shuttle. The version of the onboard guidance and control systems software modeled in this program is known as the November 1976 version.

The objectives of this report are to describe both the Space Shuttle Orbiter systems modeled in the latest studies and the operational features of the RFDS. Appendix C of reference 1 contains a description of the equations of motion used in the RFDS. The following sections of the present report reference appendixes A to G which give details on particular models used in the program. It should also be noted that appendix G, by Lawrence E. Barker, Jr., and Lawrence F. Rowell of LaRC, is an extended version of the appendix D from reference 1 which contains a derivation of the convolution integration technique developed. References 2 to 4 give further descriptions of Space Shuttle missions and systems.

Use of trade names or names of manufacturers in this report does not constitute an official endorsement of such products or manufacturers, either expressed or implied, by the National Aeronautics and Space Administration.

FLIGHT-SYSTEMS MODELING

The RFDS is being used to conduct an independent assessment of the effectiveness of the orbiter's guidance and control systems software which will reside in the onboard computers. A simplified block diagram designed to show the various components of the orbiter modeled in the RFDS is given in figure 1. The appropriate vehicle physical properties for this flight are given in table I. This section of the report describes the modeling tasks required by these various components.

One of the significant modeling tasks was to simulate the interface between the onboard computers and the orbiter hardware. Since the control and guidance systems software receive inputs from the navigation system and sensors, the various sampling frequencies of this hardware had to be matched before the simulated onboard software could be properly updated with inputs from these sources. In addition, transport delays to this information and the analog prefilters on certain channels were modeled to incorporate the possible impact of these items on the entry systems.

The guidance software uses information from the navigation system and generates commands for both the cockpit displays and error needles (see section on cockpit description) as well as for the flight-control software when it is operated in the automatic mode. During the unpowered descent of the orbiter, there are three major guidance algorithms which calculate the commands necessary to perform (1) the 5500 km (3000 n. mi.) flight from deorbit to the terminal area, (2) the turn onto the runway approach, and (3) the flares and touchdown. The principal modeling task posed by this software was to properly duplicate the transition logic that must take place between successive algorithms. This logic not only sets flags to trigger initialization of the guidance software but also determines the occurrence of specific events which require reconfiguration of the control-system software as well. These three algorithms are computed at

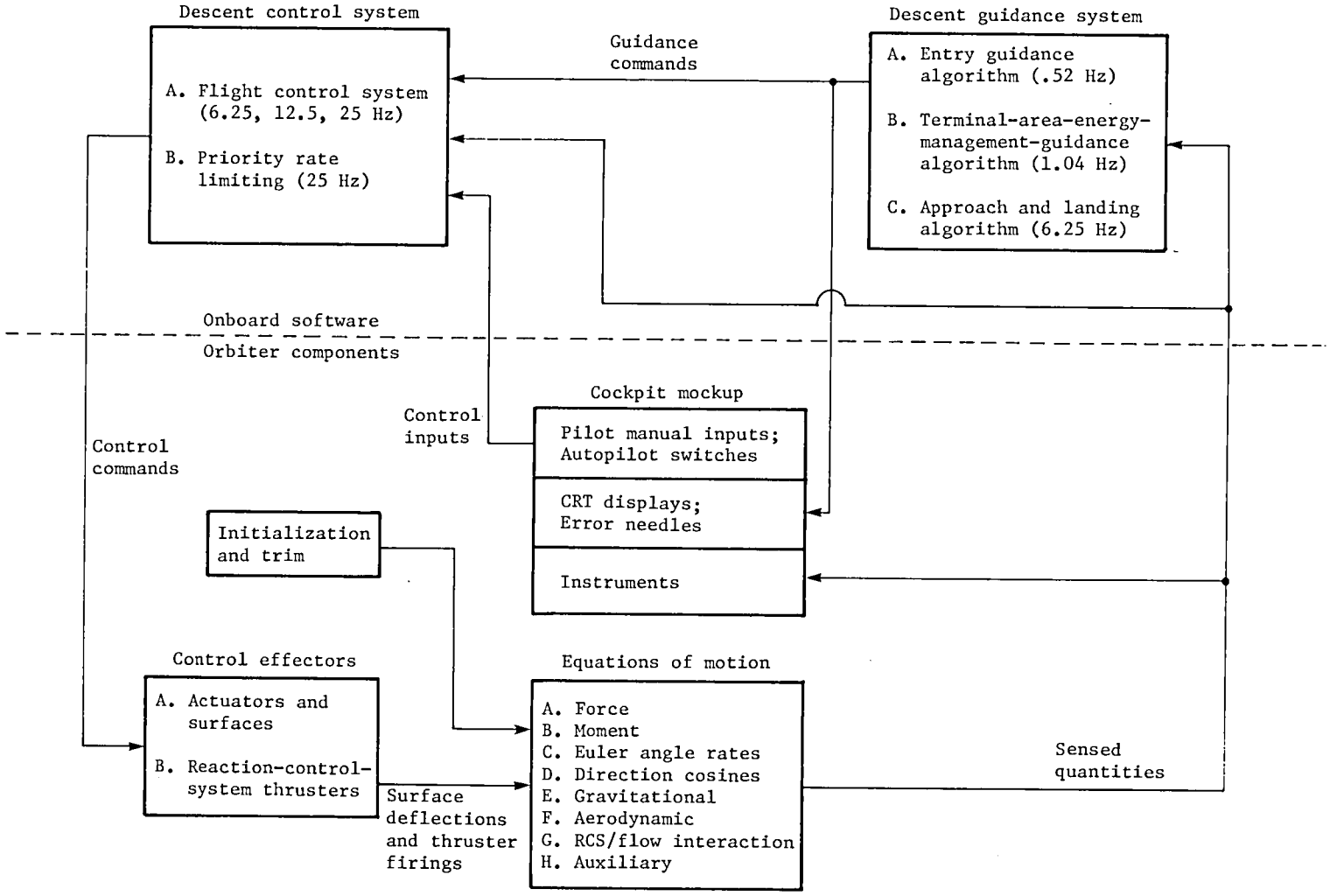


Figure 1.- Block diagram of Space Shuttle Orbiter components modeled in reentry flight dynamics simulator.

TABLE I.- SPACE SHUTTLE ORBITER PHYSICAL CHARACTERISTICS DURING ENTRY

Mass properties:	
Mass, kg (lb)	83 388 (183 840)
Moments and products of inertia:	
I_{XX}, kg-m^2 (slug-ft^2)	1 169 236 (862 384)
I_{YY}, kg-m^2 (slug-ft^2)	8 729 397 (6 438 473)
I_{ZZ}, kg-m^2 (slug-ft^2)	8 991 771 (6 631 990)
I_{XY}, kg-m^2 (slug-ft^2)	3868 (2852)
I_{XZ}, kg-m^2 (slug-ft^2)	-218 615 (-161 242)
I_{YZ}, kg-m^2 (slug-ft^2)	3441 (2537)
Nominal center-of-gravity offsets:	
x_{off}, m (ft)	-0.5004 (-1.6416)
y_{off}, m (ft)	0.0381 (0.125)
z_{off}, m (ft)	-0.0203 (-0.0667)
Surfaces:	
Wing:	
Reference area, m^2 (ft^2)	249.91 (2690)
Mean aerodynamic chord, m (ft)	12.06 (39.5675)
Span, m (ft)	23.79 (78.0567)
Elevon (per side):	
Reference area, m^2 (ft^2)	19.51 (210)
Mean aerodynamic chord, m (ft)	2.30 (7.558)
Rudder (per side panel):	
Reference area, m^2 (ft^2)	9.30 (100.15)
Mean aerodynamic chord, m (ft)	1.86 (6.100)
Body flap:	
Reference area, m^2 (ft^2)	12.54 (135)
Mean aerodynamic chord, m (ft)	2.06 (6.75)

different frequencies which are, respectively, 0.52 Hz, 1.04 Hz, and 6.25 Hz. A detailed description of the guidance software is given in appendix A.

The flight control software uses either the commands from the guidance software in the automatic mode or the pilot's control inputs in the manual model to generate the commands for the control-surface actuators and the reaction control system (RCS). Figure 2 shows the location of these control effectors. The commands are scaled by the software to provide the desired vehicle rates and turn coordination and also to provide stability augmentation. Since the various functions performed by the control software are conducted at several rates, the simulation program again had to control execution of these functions to duplicate onboard timing. The description of the flight control software given in appendix B includes both the automatic and manual configurations.

The evaluation of the effectiveness of the guidance and control software requires modeling the vehicle aerodynamics and control effectors, such as control surfaces or reaction control thrusters, over the full range of conditions encountered throughout entry to touchdown. All the longitudinal and lateral-directional performance, stability, control, and hinge-moment parameters are included in the RFDS and are derived from wind-tunnel measurements. Also, ground effects, landing-gear-down effects, and viscous effects due to high altitude flight are included. These data, which range from hypersonic to subsonic velocities, are discussed further in appendix C.

To facilitate studying the guidance and control system in a specific region of the entry, it is desirable to be able to initialize flight at various points along the trajectory. This requires the ability to calculate a trim condition for the vehicle, both longitudinally and directionally, at a given attitude; the equations developed for this purpose are described in appendix D.

During atmospheric flight, an interaction occurs between the plume from the RCS thrusters and the flow field in the vicinity of the thrusters. This interaction has been modeled from wind-tunnel-test results and is discussed along with the RCS model in appendix E.

The orbiter's control surfaces (fig. 2) consist of a body flap, a rudder that flares to provide speed brakes, and separate inboard and outboard elevon panels on each wing. These elevon panels are driven by individual actuators which are simulated by a nonlinear model representing the associated dead zones, gear slop, hydraulic system, and servo amplifiers. This nonlinear model, discussed in detail in appendix F, contains both first-order and second-order linear components. A special integration method used to solve the linear elements is described in appendix G; this appendix includes the equations to calculate all the constants needed to implement the method.

The simulation cockpit and the use of the RFDS are described in the following sections.

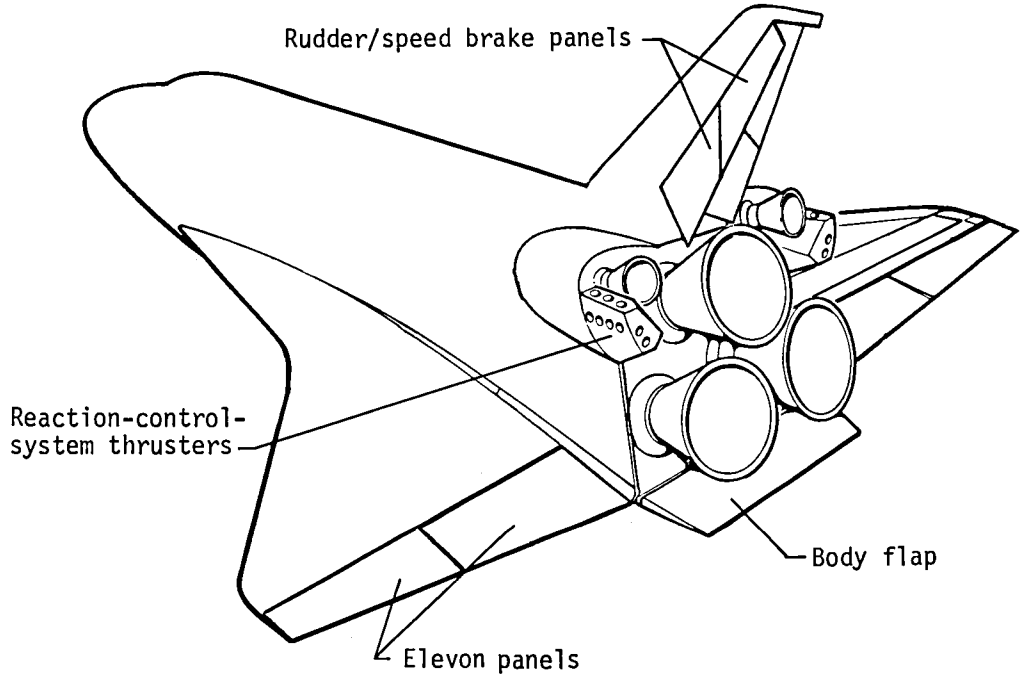


Figure 2.- Sketch of Space Shuttle Orbiter with control effectors identified.

SIMULATION-COCKPIT DESCRIPTION

In order to do a thorough evaluation of the guidance and control software, the ability to conduct pilot-in-the-loop simulations is required in addition to the capability for automatic flight. For this reason, the simulation cockpit shown in figure 3 was interfaced with the program to provide the attitude and rate indicators, displays, switches, and manual controls available to the orbiter pilot.

This cockpit is a single-seat unit that can be placed adjacent to the simulation control station where data are collected and test cases are selected. While the unit is not a replica of the Shuttle Orbiter cockpit, it does provide the necessary displays, controllers, and switches to "fly" the Shuttle entry mission. The cockpit has a three-axis hand controller (only the roll and pitch axes are used), rudder pedals, speed-brake controller, body-flap rate switch, and roll, pitch, and yaw trim switches. There are also push button indicator switches for selecting the autopilot or manual control modes and switches for possible flight-control-software modification by the pilot in flight referred to as downmoding.

The instruments provide altitude, altitude rate, equivalent airspeed, elevon and body-flap position, Mach number, speed-brake deflection, RCS active lights, angle of attack, sideslip, roll rate, and an attitude director indicator (ADI). The ADI has a pitch rate indicator and error needles driven by the guidance software.

Two cathode-ray tubes (CRT) are used to provide the guidance display information that will be available onboard the Shuttle Orbiter. These displays change automatically as the vehicle enters different flight regimes. A sample CRT display is shown in figure 4. These displays are generated by an Adage AGT 130 graphics terminal.

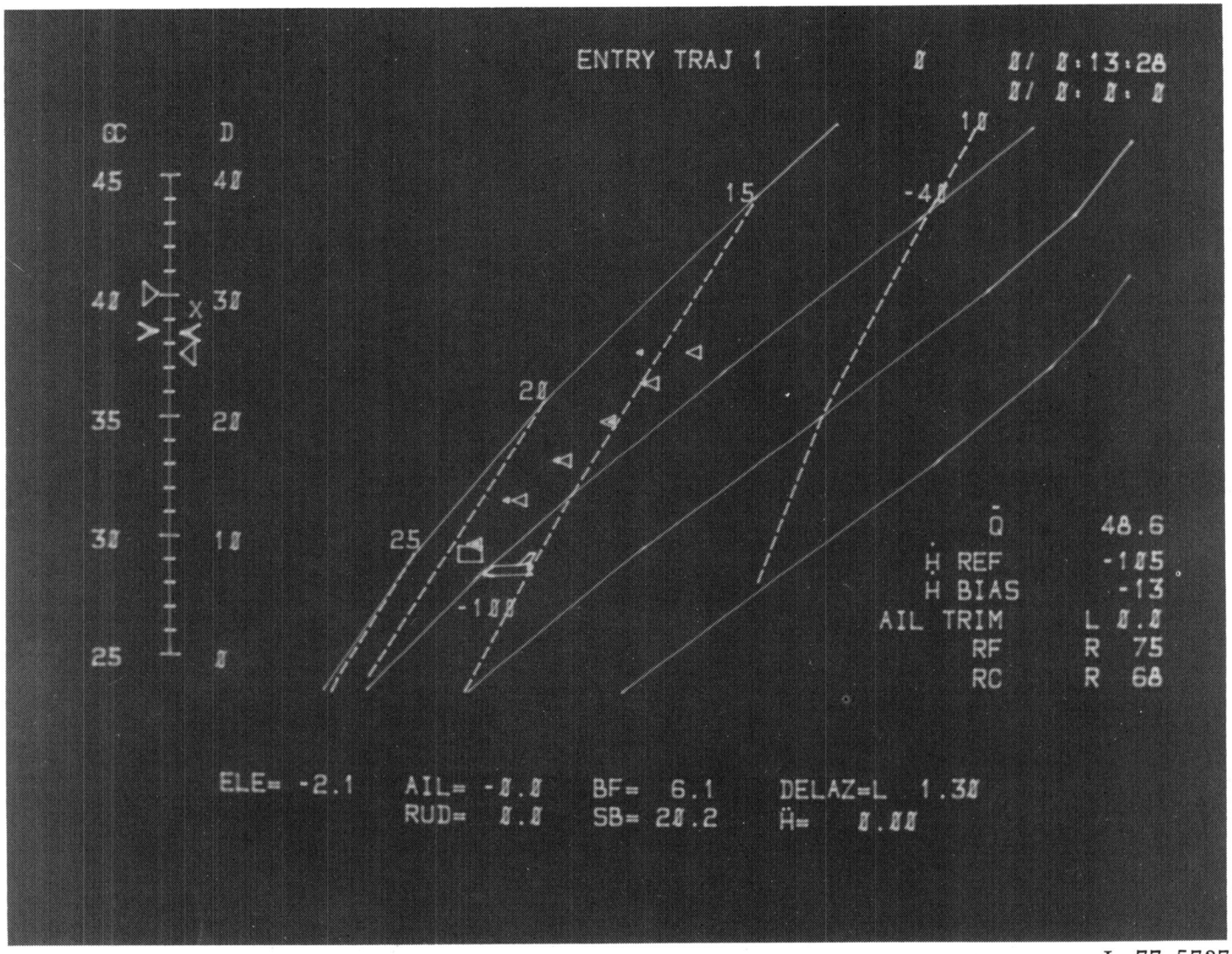
PROGRAM DESCRIPTION

The RFDS has been developed to use the RTSS for simulation of the Space Shuttle Orbiter entry flight. Since the main features of the RFDS and the major orbiter models have been described in previous sections, this section will serve to detail the program flow, the program options available, the integration schemes used, the parameters that can be changed, and the various ways in which the program can be executed.

The requirement for a man and/or hardware in the loop is the principal distinction between a real-time simulation program and a nonreal-time (batch) simulation program. Special software must be provided to handle communications between the hardware and the computer. Also, the iteration rates for this exchange must be high enough so that output appears continuous to the man in the loop while still providing enough intervening compute time to solve the program equations. The RFDS is computed at 25 iterations

L-77-5733

Figure 3.- Simulation cockpit.



L-77-5737

Figure 4.- Example of guidance system display provided to cockpit.

per second to provide both the appearance of continuity at the hardware and to duplicate the basic sampling frequency of the onboard software. The coordination of the hardware functions with the program execution is controlled by executive software that is incorporated into the FORTRAN simulation program. It controls such things as data collection on disc, analog/digital conversion, clock synchronization, graphics communications, and so forth. This executive software, referred to as the Supervisor, is a library of subroutines that becomes a part of the real-time application program. When both the RTSS hardware and Supervisor software have been properly integrated with the simulation program, this program (such as RFDS) can be operated from a control station and, if desired, connected to a simulation cockpit. The control station, its functions, and its associated output devices have been described in reference 1. The Supervisor is described in reference 5. A generalized flow chart of a real-time program and the major calls to Supervisor software are given in figure 5 and are discussed below. These programs are run on a Control Data CYBER 175 computer using the network operating system (NOS).

The FORTRAN program code must be divided functionally to utilize the three principal modes (reset, hold, and operate) that can be selected at the control console. The following discussion of figure 5 defines the purpose of each mode as well as each of the major calls to RTSS software shown in the figure.

The call to RTINIT initializes the Supervisor software. The first block containing "constants and initialization" represents any code that must be performed only once when the program is first submitted to the computer. The call to RTSRT begins clock synchronization of both the iterative computations and the conversion of analog and digital signals for input and output (I/O). From this point, all calculations must be performed fast enough to complete one integration step before the next clock pulse calls for I/O.

Selecting the RESET button at the control console will place the flow in a loop beginning with the block containing "initial conditions" and extending to the call to RTMODE. RTMODE checks the switch setting of the control console and directs the flow according to the mode switch that is depressed. In RESET, the value of the integration control variable, INT, is always zero. Selecting the HOLD button will place the flow in a loop extending from the block of "hold equations" to the call to RTMODE. The HOLD mode normally consists of any special equations that are to be performed before the simulation proceeds with the integration process.

The OPERATE mode extends from the block containing "inputs from cockpit and control station" to the call to RTCYCLE. This mode performs data storage, hardware I/O, and the integration of the equations of motion. The calculation of the outputs to hardware, as well as the other code bypassed when INT > 1, will only be executed once per integration step and not each pass of a multipass numerical integration scheme. The call to DISPLAY is made once per iteration during all three modes to update the variables

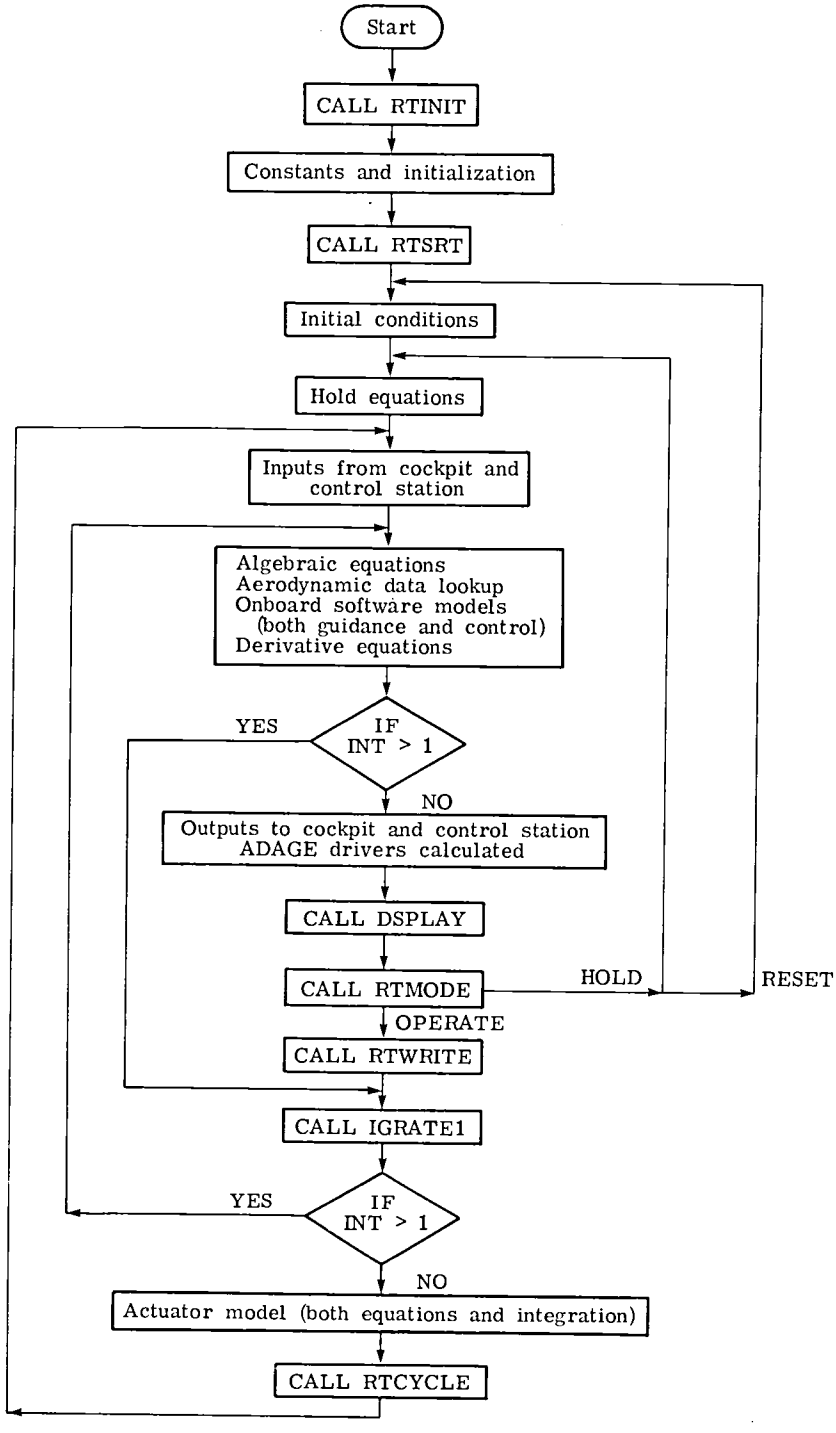


Figure 5.- Block diagram of real-time program flow.

being displayed at the control console. The subroutine RTWRITE is called only once per iteration in the OPERATE mode to place on disc the current values of any variables desired for printout or post processing. These must be specified by calls made to Supervisor subroutines in the initialization section.

The subroutine IGRATE1 performs the numerical integration of the nonlinear differential equations. INT is used to control the looping of the multipass schemes so that I/O is performed and data recorded only when all passes of the integration step are completed, indicated by INT = 1. The standard numerical schemes available in the IGRATE1 subroutine are as follows:

(1) Runge-Kutta, second order

(2) Runge-Kutta, fourth order

(3) Adams-Moulton, second order

(4) Adams-Moulton, fourth order

The predictor-corrector methods, items 3 and 4 above, can be reduced to predictor-only methods when desired.

There are two actuator models that can be selected in this simulation. One is a simplified linear first-order model, and the second is a nonlinear model. (See appendix F.) Both contain a special integration method, known as the convolution technique, developed for the solution of the linear portions of such models. This method needs only one pass – thus requiring that these routines be placed so as to coordinate with the multipass schemes of IGRATE1. The convolution integration technique is described in appendix G.

The call to RTCYCLE is made when all passes of the integration step are completed, and at this point the program will cease computations until a clock pulse signals for I/O. When this occurs, output is sent to hardware and the latest inputs are available for the next iteration in OPERATE. There are many other calls to RTSS software, but these are used primarily in the initialization of the program and will not be discussed here.

PROGRAM OPERATION

When the program if first brought up on the RTSS, it will perform all initialization tasks and then, typically, it will loop in the RESET mode where various options can be selected and parameters and initial conditions set for the desired studies of the day.

Some of the changes that might be made include altering the value of a program parameter, selecting options using the console function switches, reading data, trimming

the vehicle, or going to the HOLD or OPERATE modes. The following paragraphs describe some of these options.

From the control console, any variable in central memory can be changed or displayed through a digital decimal display and its associated keyboard. The parameters most often changed are placed in FORTRAN arrays for easy access referred to as "in-tables" addressing. The selection of all other variables is referred to as "no-tables" addressing. The parameters presently assigned to the "in-tables" arrays include orbiter physical properties (table I), initial vehicle altitude and position and rates, initial control-surface deflections, steady-state wind components, aerodynamic uncertainty gains, instrument inaccuracies, downmoding gains, onboard software sampling intervals, and printout frequency.

The control console also has 16 function sense switches (FSS) that serve as logical variables in the program. These have been used in the RFDS as follows:

FSS	Use
1	TRUE - Execute subroutines necessary to connect to the simulation cockpit and drive ADAGE displays.
	FALSE - Do not connect to cockpit.
3	TRUE - Execute the dynamic check subroutine.
	FALSE - Do not execute dynamic checks.
4	TRUE - To give mode control to the cockpit.
	FALSE - Do not remote mode control.
5	TRUE - To enter a short RESET loop to save central processor time.
	FALSE - Full RESET loop.
6	TRUE - Use the nonlinear actuator model.
	FALSE - Use simplified linear actuator.
8	TRUE - Take out any hardware biases in manual controls.
	FALSE - No effect.
9	TRUE - Simulate yaw jet manifold failure (only two yaw jets remaining).
	FALSE - No effect.
10	TRUE - Route printing to central complex.
	FALSE - Print in RTS area.

FSS	Use
11	TRUE - Simulate one auxiliary power unit failure.
	FALSE - No effect.
12	TRUE - Trim orbiter at initial conditions.
	FALSE - Do not trim.
13	TRUE - Make a data tape for post-plotting.
	FALSE - No effect.
14	TRUE - Print the ADAGE interface buffer.
	FALSE - No effect.
15	TRUE - Select desired ADAGE display from control console.
	FALSE - Select display from cockpit.
16	TRUE - In-tables addressing.
	FALSE - No-tables addressing.
2,7	Not used.

The control console has red and white lights which can be set by the program to indicate the occurrence of selected events. These have been used as follows:

Red Lights	Use
1	Signal the beginning of the TAEM guidance algorithm.
2	Signal the beginning of the approach and landing guidance algorithm.
3	Signal sea-level altitude.
8	Signal when wheels touch runway.

White Lights	Use
1 - 5	To indicate guidance subphases within each guidance algorithm.
21 - 26	To indicate RCS jet firings.

The "in-tables" addressing mode is used to set additional program logical variables in the LOGIC array to control program branching. These have been used as follows:

LOGIC	Use
1	TRUE - Perform aerodynamic data lookup only once per step, not each pass.
	FALSE - Calculate data during all passes.
2	TRUE - Use spherical Earth model.
	FALSE - Use oblate Earth model.
3	TRUE - Use with integration schemes 3 or 4 to get a predictor-only integration from IGRATE1.
	FALSE - No effect.
4	TRUE - Initialize the ADAGE software.
	FALSE - No effect.
5-8	Used by the dynamic check subroutine to select options.

SIMULATOR APPLICATION

The RFDS and the systems modeling described herein are being used to assess the Space Shuttle Orbiter entry system's sensitivity to off-nominal vehicle characteristics and flight conditions. The orbiter must be man-rated before it is launched, and a thorough analysis of the system must give consideration to the envelope of possible aerodynamic characteristics, the expected center-of-gravity range, system failures, and biases and quantization of system-derived parameters such as the feedback parameters. Reference 6 presents the results of the analysis in the Mach 10 to Mach 2.5 flight regime which addressed off-nominal aerodynamics, lateral center-of-gravity offsets, two RCS thrusters failed on each side, and sensed angle-of-attack errors. The flight control system exhibited extreme sensitivity to the angle-of-attack errors, poor lateral trim characteristics, and an overgained system under some conditions.

System modifications were developed and tested on the RFDS to help alleviate these problems. Future orbiter-system redesigns and refinements are expected to result in continued development and use of this simulation capability through the first few flights of the Space Shuttle.

CONCLUDING REMARKS

The studies for which the RFDS was designed include evaluation of the Space Shuttle Orbiter's entry guidance and control software in the presence of (1) off-nominal aerodynamics, (2) bias errors in sensed quantities, (3) actuator and RCS thruster failures, and

(4) off-nominal flight conditions and maneuvering. These can be accomplished with both autopilot and pilot-in-the-loop studies using the RFDS with a cockpit mockup or by nonreal-time studies using the RFDS in a batch or interactive manner.

To date, the RFDS has been used to suggest modifications to the control software to alleviate sensitivities to conditions such as center-of-gravity offsets and off-nominal aerodynamics and to discover problem areas in which further study is required.

Langley Research Center
National Aeronautics and Space Administration
Hampton, VA 23665
June 25, 1980

APPENDIX A

DESCRIPTION OF THE NOVEMBER 1976 GUIDANCE SYSTEM

The guidance system software, which resides in the orbiter's onboard computers, takes the navigation inputs and the vehicle attitudes and generates both the commands for the pilot displays and for the autopilot software. The guidance system consists of three sequential algorithms designed to guide the orbiter from just after the deorbit maneuver to touchdown. This version is known as the November 1976 guidance system and has been designed for the descent from the first orbital flight of the Space Shuttle Orbiter.

The entry guidance algorithm is designed to guide the orbiter from entry into the atmosphere to a narrow target which must be achieved at a velocity of 762 m/sec (2500 ft/sec) while keeping the orbiter within its aerodynamic heating constraints and reducing the cross-range error. The vehicle then switches to the terminal-area-energy-management (TAEM) algorithm in which the glide energy is controlled and the vehicle is turned onto and around a hypothetical heading alignment cylinder (h.a.c.) which is tangent to the runway center line and about 10 668 m (35 000 ft) from the runway threshold. The approach and landing guidance algorithm begins as the vehicle comes off the h.a.c. in line with the runway. In this phase the guidance system maintains a high energy level on a 24^{o} glide-slope approach, commands the vehicle to flare to reduce sink rate, and executes the touchdown maneuver. The mechanization of these algorithms is discussed in the following paragraphs.

Entry Algorithm

The entry guidance algorithm computes range to go using preselected drag deceleration-versus-velocity profiles which have been shaped for five major trajectory phases (indicated by the variable ISLECT). The differences among the phases are in the nature of the trajectory constraints; thus, software for only one phase will be executed at any given time. However, each phase must calculate reference drag and altitude rate. The drag deceleration is controlled by varying the vertical lift, i.e., banking the vehicle, and the cross-range errors are limited by bank-angle reversals. The angle of attack follows a preselected schedule which was designed to meet the aerodynamic heating constraints, the cross range required, and aerodynamic stability and control margins. A diagram of the major sections of the entry guidance algorithm is presented in figure A1 and shows the order in which they are executed. The section names used in the diagram correspond to those used in the program. The discussion below gives an explanation of the function of each of these sections.

APPENDIX A

The entry guidance executive (EGEXEC) performs the transition logic for progressing from one guidance phase to another and executes each section in the proper sequence. In this executive, there is a call to subroutine EGRT. This subroutine provides navigation parameters, heading error to the h.a.c., and ground range to the runway threshold. The entry-guidance altitude scale height section (EGSCALEHT) computes the atmospheric density exponential-approximation scale height as a function of relative velocity. This height is used in other sections to compute reference energy and reference altitude rate values.

The entry-guidance initialization section (EGINIT) is entered only on the first pass through the entry guidance and contains initial values for coefficients, parameters, and flags.

The next section entered by the executive is the entry-guidance common computation section (EGCOMN) which computes several variables that are used in the various guidance phases such as the navigation-derived lift-drag ratio (L/D) and the energy over mass.

If the entry guidance system is in its first phase, ISLECT = 1, the executive enters the pre-entry phase (EGPEP) which determines the command vertical L/D for use in computing the roll-angle command. The initial L/D value is a function of the initial entry bank angle which for this mission is zero. This phase is normally terminated when the normal acceleration reaches 0.176g. If the guidance system is in either the temperature control or equilibrium glide phases, ISLECT = 2 or ISLECT = 3, respectively, the executive enters the range prediction section, EGRP. This section determines the appropriate drag-velocity profile that will meet the range requirement at the target by using the current glide range in approximate relationships. This drag velocity profile is then used in the entry-guidance reference parameters section (EGREF) to calculate reference drag and altitude rate values for later use in determining the desired bank-angle command. Also calculations are performed here to insure smooth transitions between phases 2 and 3 and phases 3 and 4. When ISLECT = 4, the entry-guidance constant drag phase section (EGREF4) is used to calculate the reference altitude rate and the derivative of range with respect to drag.

At about Mach 10 the executive sets ISLECT = 5, and the transition phase section (EGTRAN) is used to compute both the range potential between the end of the previous phase and the target condition at the entry TAEM interface and the drag-energy profile required to null range errors. From this profile the reference drag and altitude rate parameters are determined.

With the reference drag and altitude rate parameters for the current guidance phase determined, the executive proceeds to the angle-of-attack command section, EGALPCMD, the first of the sections to determine commands used by the flight control system. These

sections are entered in all five phases. In the entry algorithm, the angle-of-attack command follows a schedule dependent on velocity. The next section entered is the gain selection section, EGGNSLCT. The gains in the calculation of the desired vertical L/D are shaped to drive the vehicle to the reference drag level without excessive phugoiding along the trajectory. With the proper gain values determined and the reference drag and altitude rate values available from the previous sections, the executive proceeds to the entry guidance lateral logic and vertical L/D command section, EGLODVCMD. In this section, an approximation to the aerodynamic drag coefficient profile is used to correct the reference altitude rate for the time rate of change of the drag coefficient. Also, since the altitude rate obtained from the navigation calculations may have a significant error, a correction term is calculated using a feedback of the difference between the actual drag and the reference value. With these corrections accomplished, the vertical L/D command required to drive the trajectory toward the reference drag profile is determined. Also, this section checks to see if the heading angle has exceeded the heading to the tangent to the runway heading alignment cylinder (h.a.c.) by more than a dead-band value which is a function of velocity. If the dead band has been exceeded, this section calls for a bank reversal.

The final section to be entered is the roll command section (EGROLCMD) which calculates the bank-angle command from the vertical L/D and the bank direction from the previous section. This command is then sent to the flight control system. When the executive finds that the entry TAEM interface conditions have been met, a flag is set and the next guidance algorithm, TAEM, is called.

Terminal-Area-Energy-Management Algorithm

In the terminal-area-energy-management algorithm, energy control is achieved by guiding the vehicle along reference altitude and dynamic pressure-versus-range profiles. The control is provided by generating a normal acceleration command to null out dynamic pressure errors. If considerable excess energy exists, an S-turn maneuver is executed. If energy is low, the vehicle is commanded to fly near maximum for range stretch maneuvers and, if necessary, a signal is sent to the crew to switch to a h.a.c. nearer the runway threshold. To generate these commands, the TAEM guidance algorithm follows the sequence of calculation sections shown in figure A2. On the first pass only, the TAEM guidance executive (TGEXEC) executes the initialization section, TGINIT. The executive then proceeds to the TGXHAC section to determine the location of the h.a.c. centers relative to the runway threshold. This distance is used in the ground track predictor section (GTP) to compute the ground track range from the present position to the runway threshold for all phases of the TAEM trajectory. The executive then proceeds to the TAEM references and dynamic pressure section (TGCOMP) which calculates the following

parameters: the reference and actual energy over weight, the reference altitude and altitude error, the reference flight-path-angle tangent, the reference dynamic pressure as a function of range to go, the filtered dynamic pressure, the dynamic pressure error, and the equivalent airspeed.

The TAEM guidance algorithm has four phases which include the S-turn maneuver, the acquisition of the heading alignment cylinder, the maneuver around the heading alignment cylinder, and maneuvering off the cylinder when the vehicle is lined up with the runway. The transitions among these phases are determined by the TGTRAN section of the code. This section checks the energy over weight and range to go against reference values to determine phase section. It also determines the following: the direction in which an S-turn will be made if required; when a switch to the minimum entry point heading alignment (the cylinder closer to the runway) is required; and when the vehicle has achieved the approach and landing envelope and can begin the next guidance algorithm.

The TAEM guidance executive then proceeds to the sections that compute the three commands to be sent to the flight control system. The normal acceleration command section (TGNZC) computes the incremental normal acceleration command for all phases from the altitude rate error using the reference flight-path angle and the altitude error. This command is filtered for smoothing and then limited based on energy over weight and dynamic pressure. Also, the command has a correction term applied when in the S-turn phase. The speed-brake command section (TGSBC) generates a command dependent on Mach number M. If $M \leqq 0.9$, the command is based on the dynamic pressure error computed in TGCOMP, while for $M > 0.9$, the command is 65° except during an S-turn phase when the speed brake is commanded to be fully open (98°). The bank command section (TGPHIC) computes the bank-angle command for flight control which is dependent on the TAEM guidance phase. In the S-turn phase, the bank command is 30° in the direction designated in TGTRAN. When acquiring the h.a.c. in the acquisition phase, the bank command is a function of the heading error to the h.a.c. tangency point but not more than 30° to keep sonic boom overpressures down. As the vehicle moves around the h.a.c. in the heading alignment phase, the bank angle is computed to keep the vehicle on the cylinder. When the vehicle approaches the tangency point of the runway center line, the bank-angle command is a function of the distance to the center line. For Mach numbers less than or equal to 0.9, the bank angle is limited to 50° in the acquisition phase, 60° in the heading alignment phase, and 30° as the vehicle lines up on the runway. When the section TGTRAN determines that the approach and landing interface envelope has been achieved, the TAEM guidance algorithm is terminated.

Approach and Landing Algorithm

In the approach and landing algorithm, the vehicle lateral alignment with the runway center line is maintained and the approach and landing glide slope tracked to touchdown.

APPENDIX A

There are four longitudinal guidance phases (indicated by IPMODE) in the algorithm. These are the trajectory capture phase, the steep glide slope phase, the flare and shallow glide slope phase, and the final flare phase. The sections that perform these four longitudinal phases as well as the lateral guidance functions are shown in figure A3.

The PITCH TRAJECTORY section performs the required initialization of variables, flags, and discretes on the first pass only. It also computes the filtered true airspeed, the range to the ground target point, and the altitude, altitude rate, and flight path errors to the updated reference trajectory on each guidance system pass. The TRAJECTORY CAPTURE phase determines when the flight-path-angle error and altitude errors are sufficiently small for the guidance to switch to the linear STEEP GLIDE SLOPE phase. In the TRAJECTORY CAPTURE phase, when the vehicle is not within the above-mentioned error bounds, the STEEP GLIDE SLOPE phase is commanded to compute a normal acceleration command which is based on the altitude error and altitude rate error. When the TRAJECTORY CAPTURE phase determines that the errors are small, the linear STEEP GLIDE SLOPE phase is entered which computes a normal acceleration command based on altitude error, altitude rate error, and an altitude error integral term. This phase also tests to see if altitude for the first flare has been reached. If this test is true, the FLARE AND SHALLOW GLIDE SLOPE phase is entered on the next guidance pass by the longitudinal guidance. Here, normal acceleration commands are generated in three subphases to provide a smooth transition from the steep glide slope (nominally 24^{o}) to the shallow glide slope (nominally 3^{o}). In the first subphase, the body-flap retraction to its landing position is begun, and a normal acceleration command which is proportional to the difference between the actual flight-path angle and -3^{o} is sent to the flight control system. When an altitude of 503 m (1650 ft) is reached, the second subphase is begun in which the normal acceleration is such that the trajectory approximates the segment of a circle along which the normal acceleration is constant. When the vehicle reaches a particular distance from the runway threshold, the exponential capture subphase is begun. In this subphase the normal acceleration command is dependent upon an exponentially decaying altitude difference between the actual altitude and the 3^{o} glide slope altitude. As the vehicle reaches an altitude envelope and the vertical velocity reaches an appropriate value, the FINAL FLARE phase is initiated. This generates a normal acceleration command proportional to a filtered altitude rate error when the reference altitude rate is a linear function of the altitude. This calculation is terminated when the main landing gear touches down.

The SPEED-BRAKE COMMAND section computes the speed-brake command based on the difference between a reference velocity and the filtered equivalent airspeed until the initiation of the FLARE AND SHALLOW GLIDE SLOPE subfunction. At these lower altitudes the speed brake is retracted at about 10^{o} per second.

APPENDIX A

The approach and landing guidance now calls the lateral guidance logic to provide bank-angle and yaw rate commands to the flight control system. The ROLL COMMAND section generates a command based on the lateral error from the runway center line and the rate at which this error is changing. A lateral error integral term is also included after the heading angle error and lateral error are within particular bounds for the automatic system. Touchdown is initiated when the altitude is less than 1.5 m (5 ft) by modifying the roll-angle command to a yaw rate command and zeroing the roll angle to keep the wings level near and on the ground.

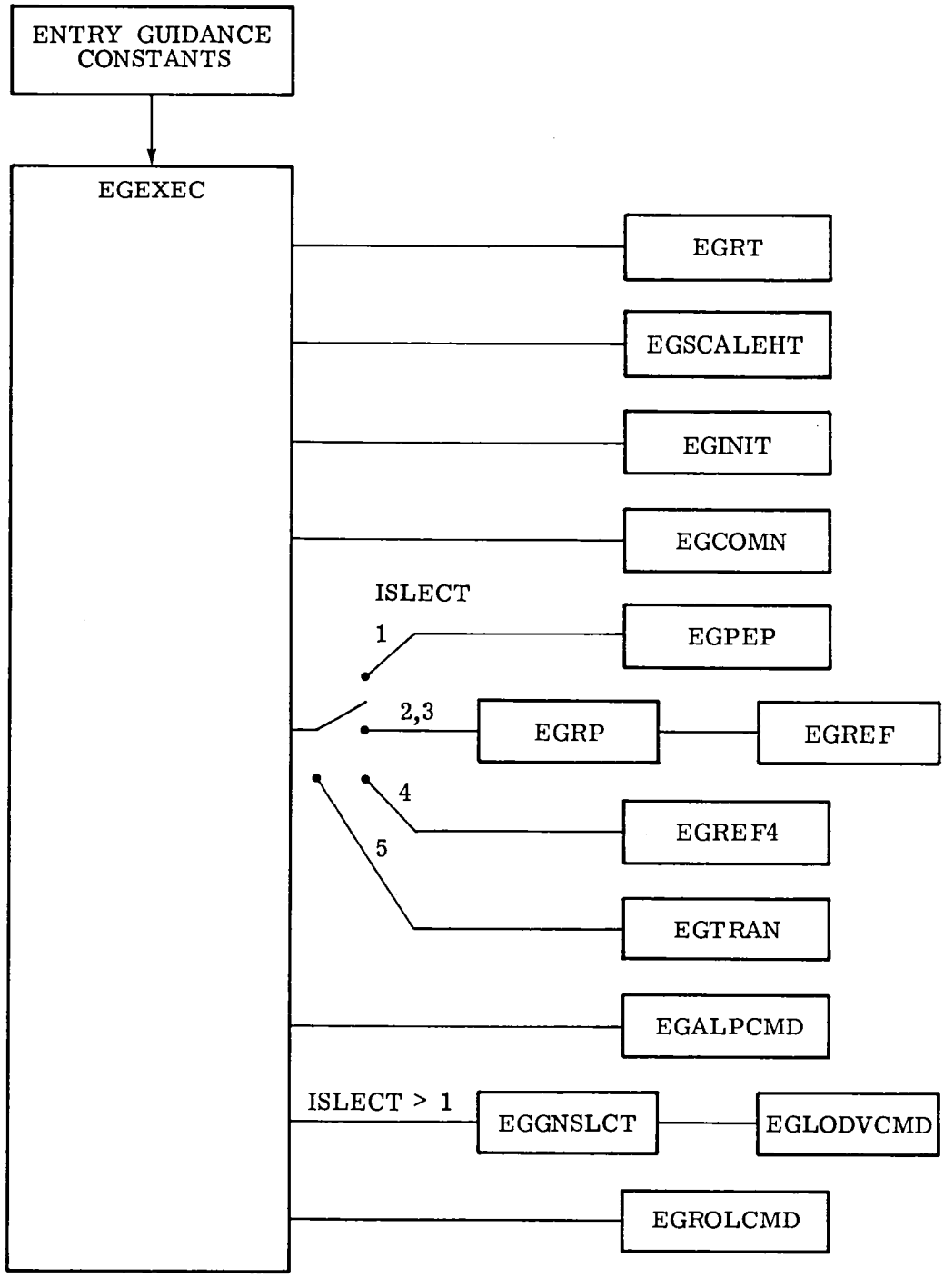


Figure A1.- Block diagram of entry guidance algorithm sections.

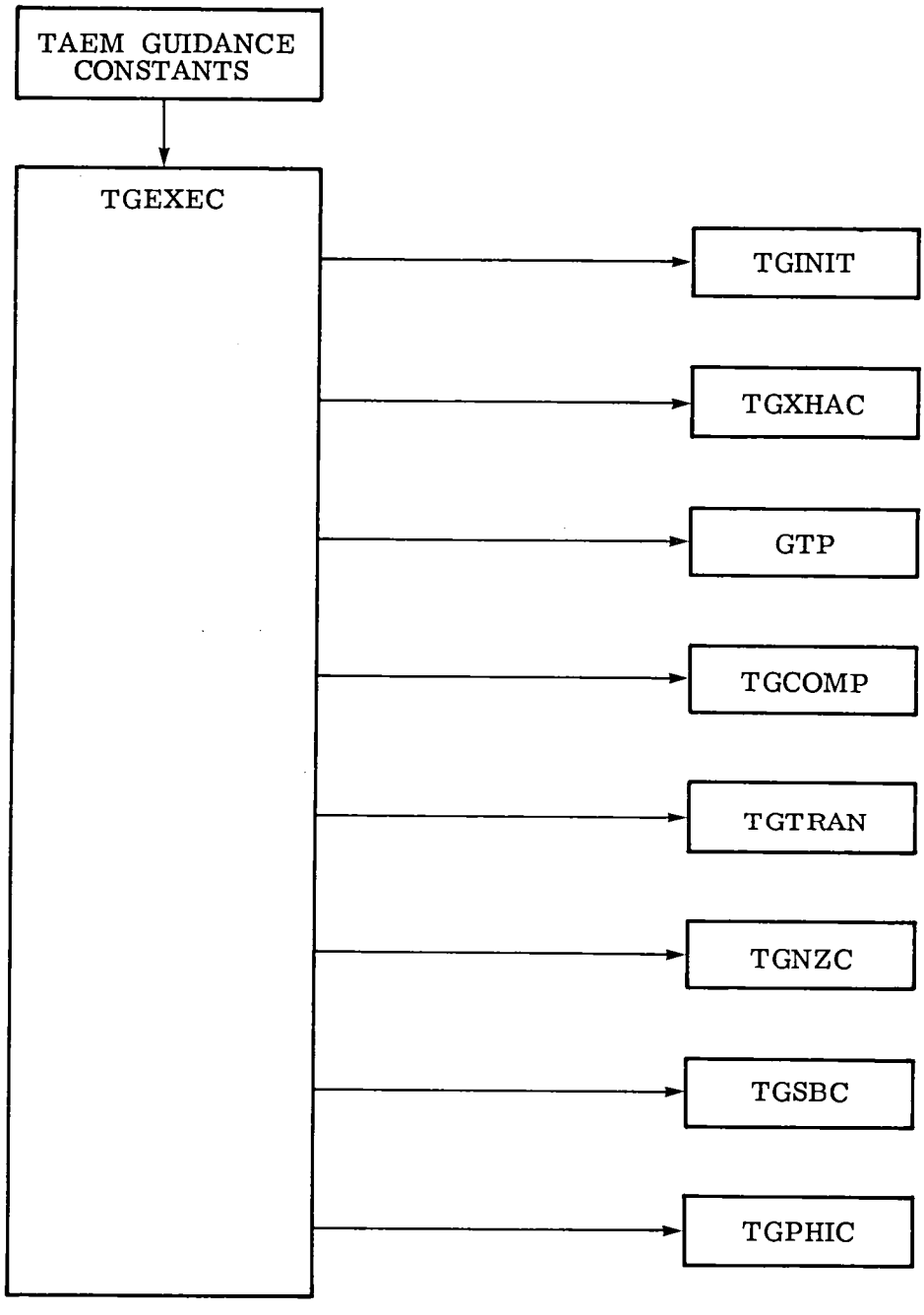


Figure A2.- Block diagram of TAEM guidance algorithm sections.

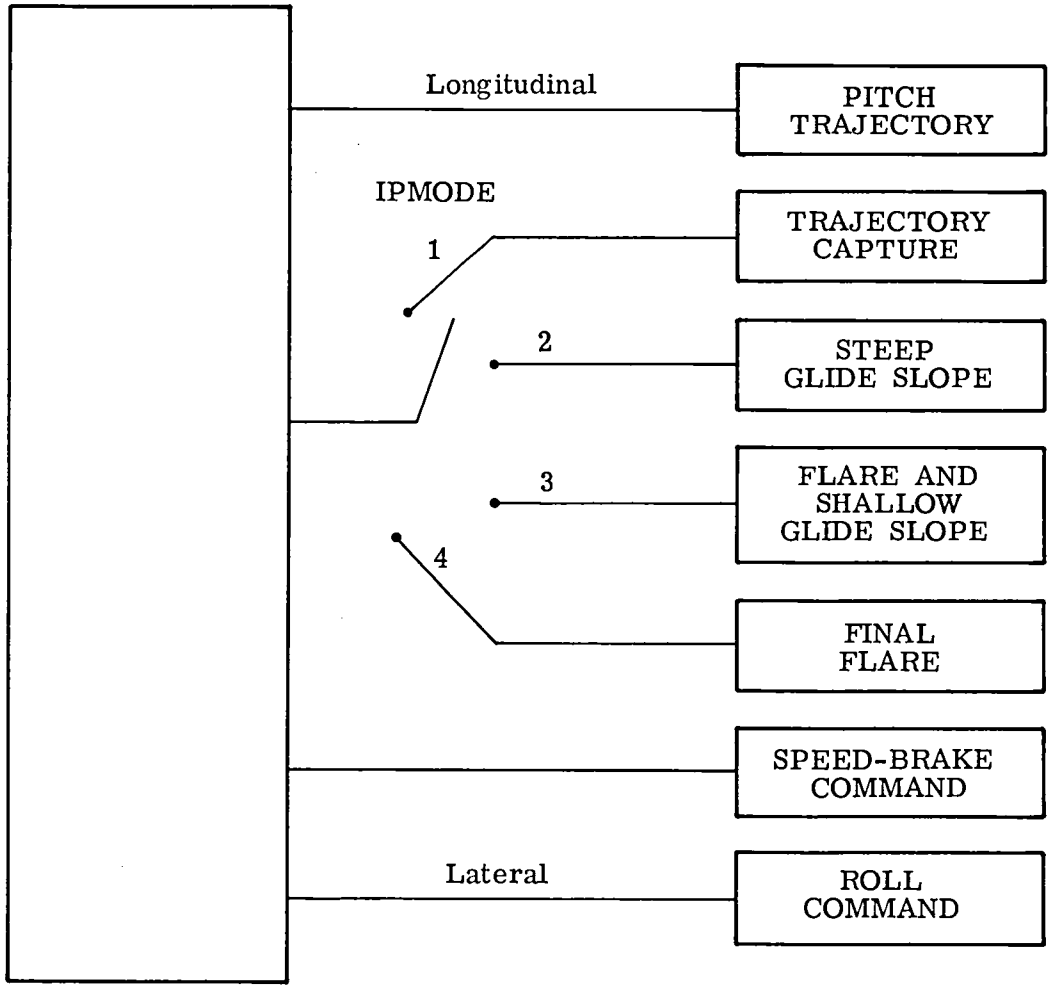


Figure A3.- Block diagram of approach and landing guidance algorithm.

APPENDIX B

DESCRIPTION OF THE NOVEMBER 1976 INTEGRATED DIGITAL AUTOPILOT FLIGHT CONTROL SYSTEM

The flight control system (known as the November 1976 Integrated Digital Autopilot) used in this simulation is designed to provide orbiter stability and control from after deorbit to touchdown. The system takes guidance system commands in the automatic mode and pilot commands in the manual mode and produces reaction-control-system (RCS) thruster commands and/or aerodynamic-control-surface deflection commands to the actuators. The pilot inputs include rotation-hand-controller deflections in pitch and roll, rudder-pedal command, speed-brake position command, body-flap rate command, panel-mounted rate trims (roll, pitch, and yaw), and system mode switches. Multiple RCS thrusters are located on pods at the base of the vertical tail (fig. 2) and provide roll, yaw, and pitching-moment capability. The aerodynamic surfaces include elevons, differentially deflected elevons (referred to as the aileron), rudder, speed brake, and body flap. This appendix describes the flight-control-system software in detail.

Notation

The flight control system was designed for measurements in the U.S. Customary Units. Therefore, units are given in both the SI and U.S. Customary Units.

ADI normal-acceleration error for flight director, g units

ADIF filtered normal-acceleration error for flight director, g units

AL approach and landing guidance

ALFERR angle-of-attack error, deg

ALFERRL limited angle-of-attack error, deg

ALPDG angle of attack, deg

ALPHACM entry-guidance angle-of-attack command, deg

ALPHCMS smoothed entry-guidance angle-of-attack command, deg

AUTO autopilot control mode

APPENDIX B

BANKERR	roll-angle error, deg
BANKYAW	roll-rate command, deg/sec
BCSL	filtered pitch-rate error, deg/sec
BETAF	filtered angle of sideslip, deg
BETDG	angle of sideslip, deg
BFT	commanded body-flap-deflection rate, deg/sec
BFTI	body-flap-deflection command, deg
BINC	increment used in description of FADER
CM	guidance-system command used in description of SMOOTHER
CMS	smoothed command used in description of SMOOTHER
COSALP	cosine of angle of attack
COSPHIL	cosine of limited sensed roll angle
COSTHE	cosine of pitch angle
COTALP	cotangent of angle of attack
DAM	roll rotation-hand-controller (RHC) command, deg
DAMS	shaped roll-stick command, deg
DAMSF	filtered roll-stick command, deg
DAMTR	roll-panel-trim command
DAMTRS	roll-panel-trim rate, deg/sec
DAT	aileron-trim rate, deg/sec

DATRIM aileron-trim command, deg

DAY lateral-acceleration error, g units

DAYF filtered lateral-acceleration error, g units

DBFDC body-flap-deflection-rate change

DBFPC body-flap-deflection command, deg

DBFRM manual body-flap command, deg/sec

DEC preliminary elevator-deflection command, deg

DECC preliminary elevator command, deg

DEL left-elevon-command rate, deg/sec

DELAC aileron-deflection command, deg

DELBF body-flap-position feedback, deg

DELBFRC commanded body-flap-deflection change, deg

DELEC elevator-deflection command, deg

DELEFB elevator-position feedback, deg

DELELC rate-limited left-elevon-deflection command, deg

DELELT left-elevon-deflection command, deg

DELERC rate-limited right-elevon-deflection command, deg

DELERT right-elevon-deflection command, deg

DELES preliminary pitch-trim command, deg

DELRC rate-limited rudder-deflection command, deg

DELRCP	past value of rate-limited rudder-deflection command, deg
DELSBC	rate-limited speed-brake-deflection command, deg
DELSBCP	past value of rate-limited speed-brake-deflection command, deg
DELSBE	speed-brake-increment cross feed, deg
DEM	pitch-rotation-hand-controller command, deg
DEMS	shaped-pitch-controller command, deg
DEMTR	panel-pitch trim, deg/sec
DEMTRS	trim rate due to panel-pitch trim, deg/sec
DER	right-elevon-command rate, deg/sec
DETRIM	pitch-trim command, deg
DNCAL	turn-compensated pitch rate, deg/sec
DPJET	pitch-rate error, deg/sec
DR	coordinated rudder command, deg
DRC	rudder-deflection command, deg
DRCRL	rudder-deflection-command rate, deg/sec
DRF	filtered rudder-deflection command, deg
DRFS	rudder-trim rate, deg/sec
DRFSI	rudder-trim command, deg
DRI	rudder command, deg
DRINCLM	rudder-command-rate limit, deg/sec

DRJET yaw-rate error, deg/sec

DRM rudder-pedal command, deg

DRMS shaped-rudder-pedal command, deg

DRPC rudder command, deg

DRRC yaw-rate error, deg/sec

DRT filtered rudder-deflection command, deg

DSBC speed-brake command, deg

DSBCM guidance speed-brake command, deg

DSBCOM entry-guidance speed-brake command schedule, deg

DSBM manual speed-brake command, deg

DSBNLM negative speed-brake-rate limit, deg/sec

DSBPLM positive speed-brake-rate limit, deg/sec

DSBRL speed-brake-deflection-command rate, deg/sec

EARLY flight-control-system subphase

ENTRY entry guidance

ERRNZ pitch rate due to normal-acceleration error, deg/sec

ERRNZF filtered pitch rate due to normal-acceleration error, deg/sec

FADER signal fading logic

FADOFF FADER logic flag

FLATURN flat-turn regime

GBFT body-flap-positive-deflection-limit schedule, deg

GDAC gain to convert roll-rate error into aileron command, deg/(deg/sec)

GDAM gain to convert roll-stick command to rate command, (deg/sec)/deg

GDAY gain to convert lateral-acceleration error to yaw-rate command, (deg/sec)/g units

GDBF gain to scale body-flap-deflection rate, deg/sec

GDEM gain to convert pitch-controller command into pitch-rate command, (deg/sec)/deg

GDQ gain to convert pitch-rate error into elevator command, deg/(deg/sec)

GDRE gain to convert yaw-rate error into rudder command, deg/(deg/sec)

GDRF gain to convert rudder command to rudder-trim rate, deg/(deg/sec)

GGDRC gain to convert yaw-rate error to rudder command, deg/(deg/sec)

GLIN linear coefficient in roll-stick shaping, deg/deg

GNY gain to convert rudder-pedal command to lateral-acceleration command, g units per degree

GNZ airspeed-variable gain

GPDAC scale factor

GPE gain to convert compensated roll-rate error into aileron command, deg/(deg/sec)

GPIT variable used to calculate GDQ, $\frac{\text{deg } (\text{N/m}^2)^{1/2}}{\text{deg/sec}} \left(\frac{\text{deg } (\text{lb/ft}^2)^{1/2}}{\text{deg/sec}} \right)$

GPPHI gain to convert roll-angle error to roll-rate command, (deg/sec)/deg

GQAL	gain to convert angle-of-attack error into pitch-rate command, (deg/sec)/deg
GRJ	gain to scale RCS thruster pulses into trim rate, deg/sec
GRPF	gain to scale filtered yaw rate
GSBB	scaling gain to convert trim increment into trim rate, (deg/sec)/deg
GSBP	gain to convert speed-brake increment into pitch-trim increment
GTEMB	gain to scale coordinating rudder command
GTRE	gain to convert preliminary pitch-trim command into pitch-trim rate, (deg/sec)/deg
GUIDDT	guidance-system step size, sec
GUY	gain to convert pitch RCS thruster command into pitch-trim rate, deg/sec
GXALR	gain to convert aileron command to coordinating rudder command
HA	altitude, m (ft)
HS	SMOOTHER step size, sec
H1	flight-control fast-cycle time, sec
IMAJ	SMOOTHER control flag
KGDRE	variable used for computation of GDRE
LATE	flight-control-system subphase
MACH	Mach number
MANBF	pilot-commanded body-flap mode
MANP	pilot-commanded pitch mode

MANRY	pilot-commanded roll and yaw modes
MANSB	pilot-commanded speed-brake mode
NUM	number of FADER steps remaining
NY	lateral-acceleration feedback, g units
NZ	normal-acceleration feedback, g units
NZCM	TAEM/AL guidance normal-acceleration command, g units
NZCMS	smoothed guidance normal-acceleration command, g units
PAR	coefficient of squared term in roll-stick shaping, deg/deg^2
PC	roll-rate command, deg/sec
PCLIM	roll-rate limit, deg/sec
PCP	roll-stick-rate command, deg/sec
PDAC	scaled aileron command, deg
PDACF	filtered aileron command, deg
PDG	sensed roll rate, deg/sec
PE	roll-rate error, deg/sec
PES	compensated roll-rate error, deg/sec
PHICM	guidance roll-angle command, deg
PHICMS	smoothed guidance roll-angle command, deg
PHIDG	sensed roll angle, deg
PSTABDG	stability-axis roll rate, deg/sec

QB dynamic pressure, N/m^2 (lb/ft^2)

QC pitch-rate command, deg/sec

QCAL pitch-rate error, deg/sec

QDG sensed pitch rate, deg/sec

QTR pitch-trim rate, deg/sec

QTRU unlimited pitch-trim rate, deg/sec

RDG sensed yaw rate, deg/sec

RJPULSE net RCS thruster pulses

RLIMR rudder-deflection limit, deg

RNZ = (GNZ)(NZCMS)

RP = (RDG) - $\left[(RTDG)(SINPHI)(COSTHE)\right]/V$, deg/sec

RSTAB scaled stability-axis yaw rate, deg/sec

RSTABG gain to scale yaw rate

RTDG = 57.3 g, deg-m/sec^2 (deg-ft/sec^2)

RTPHI = (RDG)(TANPHI), deg/sec

SINALP sine of angle of attack

SINPHI sine of roll angle

SMERR command increment used in each SMOOTHER step

SMOOTHER guidance-command smoothing logic

TAEM terminal-area-energy-management guidance

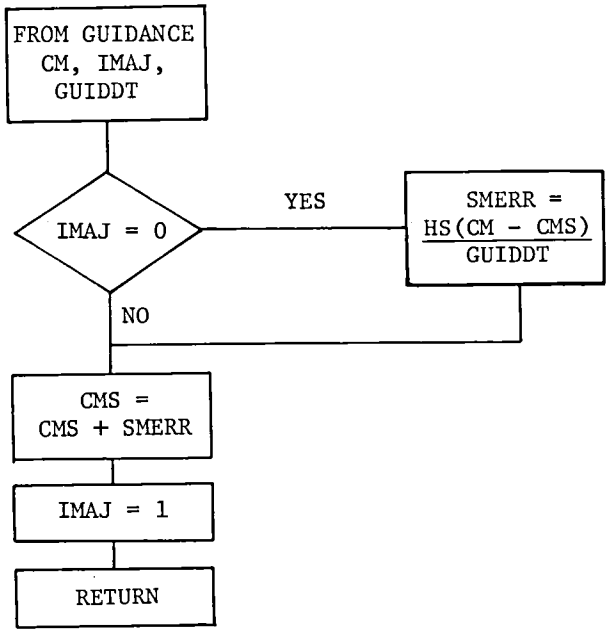
TANPHI	tangent of roll angle
TEMA	lateral-acceleration command due to rudder pedals, g units
TEMB	preliminary aileron command, deg
TEMD	guidance yaw-rate command, deg/sec
TEME	coordinating rudder command, deg
TEMF	elevator trim-deflection error, deg
TEMI	scaled coordinating rudder command, deg
THEPHI	= -(COSTHE)/(COSPHIL)
TRBF	scheduled elevator trim deflection, deg
UIN	combination of roll and yaw RCS thrusters commanded to fire on current pass
UINP	past value of UIN
UXC	number of roll RCS thrusters commanded to fire
UYC	number of pitch RCS thrusters commanded to fire
UZC	number of yaw RCS thrusters commanded to fire
V	airspeed, m/sec (ft/sec)
X	past value of signal to be faded
XNEW	current value of signal to be faded
YALCM	guidance yaw-rate command, deg/sec
YAWJET	yaw-rate error, deg/sec
Z	Z-transform variable

APPENDIX B

Description of Flight Control System

The block diagrams of the flight control system are presented in figures B1 to B10. The system was designed to minimize the time required to complete the flight-control calculations in the onboard digital computers. Thus, several computational frequencies exist among the various signal paths of the flight control system. The particular frequency for a path is indicated on the block diagram either in the figure legend or by the dashed boxes around the control-system signal paths.

Many of the guidance commands are computed at a significantly lower frequency than that used by the flight control system (PHICM in fig. B1(a)); thus, a smoothing function (SMOOTHER) is used to give the control system a more continuous command signal during that period when the RCS is active. The SMOOTHER logic is presented in sketch (a).

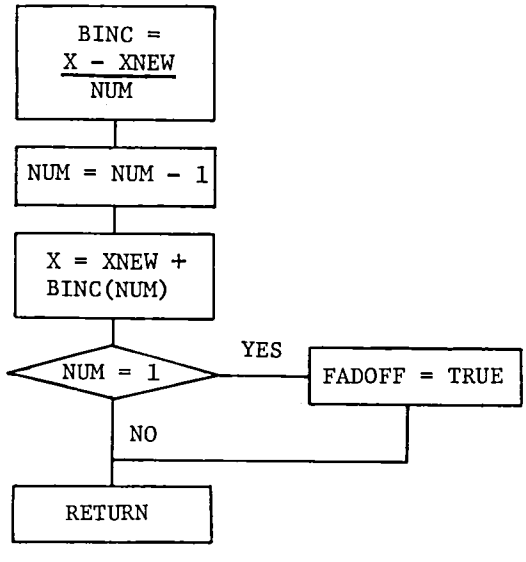


Sketch (a).

Each time the guidance algorithm is computed (low frequency), a new command (CM) is sent to the SMOOTHER and IMAJ is set to zero. Thus, SMERR is computed only when the guidance system is computed. Since HS is the flight-control-system sample time (higher frequency) for the SMOOTHER element and GUIDDT is the guidance-system sample time (lower frequency), SMERR is the fraction of the command change for each SMOOTHER step, which when added successively (i.e., CMS = CMS + SMERR for each flight-control-system step) will equal CM by the time the guidance system is computed again. Two observations should be noted: First, for the SMOOTHER to operate properly, the guidance-system sample time must be an integer multiple of the control-system

sample time; second, the value of CMS used in the calculation of SMERR is simply the output command of the SMOOTHER when the guidance command CM update occurs.

Since the vehicle angle of attack and the Mach number change considerably throughout the entry and, consequently, the aerodynamic characteristics change significantly, considerable control system modification is required. To attain proper stability augmentation and roll-angle control with the changing aerodynamics, the roll and yaw axes have two modes of operation. The first mode incurred during the entry; the EARLY, or spacecraft mode, uses the yaw thrusters to produce a yawing rate and a small sideslip angle which results in a rolling moment due to the effective dihedral of the orbiter. The yawing moment produced by the ailerons is used to provide turn coordination. Because the rudder is shielded from the free stream, it is not activated until the speed is reduced to Mach 5. At this point the rudder is used to augment the yaw RCS. Late in the entry after the orbiter pitches down to lower angles of attack and the rudder has become effective, the flight control system switches to the LATE or aircraft mode where the ailerons are used to produce a rolling rate and the rudder is deflected to coordinate the turn. This switch from EARLY to LATE nominally occurs at Mach 1.5, but the pilot can force this change anytime. In order to prevent undesirable switching transients from propagating through the system, signal fading logic (FADER) is used in both the aileron- and rudder-command channels. (See figs. B1(b) and B3(b).) This logic is illustrated in sketch (b).

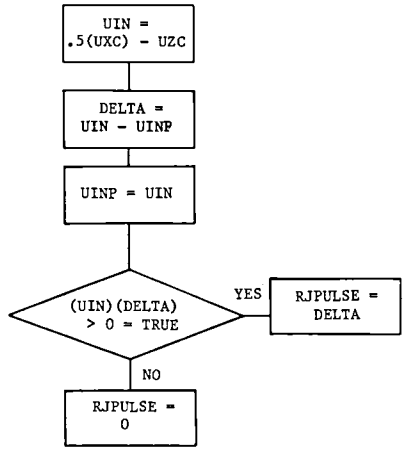


Sketch (b).

When a signal source is changed, the old signal must be faded out. FADOFF is set false in the main flight-control code which allows the FADER subroutine to be called. The parameter NUM is calculated on the first pass only by multiplying the flight-control-system

frequency by the fade time in seconds. In this system design, the fade time was 1 sec. The value of X used in the first pass to calculate the bias BINC is the value of the parameter being faded on the previous pass through the control system, and XNEW is the value of this parameter on the current pass. For recalculation of BINC, X is retained on each pass. Thus, the parameter being faded is successively changed from the initial value of X to XNEW as NUM approaches a value of 1. When NUM does reach 1, FADOFF is set TRUE and the FADER subroutine is bypassed.

Roll axis.- The aileron command and roll RCS command control laws are presented in figures B1 and B2. Figure B1(a) shows how either the roll-stick command (DAM) or the smoothed roll-angle-guidance command (PHICMS), depending on pilot selection (MANRY) in the cockpit, is converted to roll-rate command (PC). Also based on MANRY, either the signal PCP derived from the stick input or the difference (BANKERR) between the guidance command PHICM and the actual roll angle PHIDG is sent to the yaw channel (fig. B4) to be used in the EARLY mode to execute the roll maneuver. The difference between the commanded roll rate PC and the actual stability-axis roll rate PSTABDG is calculated, converted to an aileron-deflection command, TEMB, and sent on to the aileron (fig. B1(b)) where it is used in the LATE mode. Figure B1(b) shows that the aileron-deflection command in both the EARLY and LATE modes is scaled by a function of dynamic pressure GPDAC and then filtered by a second-order bending filter to prevent any of the bending modes from driving the aileron command PDACF. The aileron-trim calculation DATRIM is a function of both EARLY/LATE mode and Mach number switches. For both of these switches in the trim network, the FADER is triggered to eliminate transients due to the switching. For Mach numbers greater than 3.5 and dynamic pressures QB greater than 96 Pa (2 lb/ft^2), the aileron trim is a function of RJPULSE, the output of a pulse counter diagramed in sketch (c).



Sketch (c).

The pulse counter takes the roll RCS (UXC from fig. B2) and yaw RCS (UZC from fig. B4) thruster commands and determines if the aileron-trim function should be changed. If the magnitude of the combination of roll and yaw RCS commands (UIN) increases from one step of the flight control system to the next and the sign doesn't change or if the signs of both UIN and DELTA change, the RJPULSE value will cause the integrated value DATRIM to change. When the control system switches to the LATE mode, the aileron-trim integrator is driven by the forward-loop signal PDACF and the panel trim DAMTR, as shown in figure B1(b).

The roll RCS command (UXC), shown in figure B2, is driven by the turn-coordination signal PE which passes through a hysteresis filter. The operation of these types of filters is described in appendix B of reference 1. The signal PE is also sent to the aileron (fig. B1(b)) in the EARLY mode and goes through a deadband to remove any low amplitude signal (sensor noise) in the channel.

Yaw axis.- The rudder and yaw RCS command diagrams are presented in figures B3 and B4. Again, the EARLY and LATE modes determine the signal source. In figure B3(a), a side-acceleration command (TEMA) from the rudder pedals (DRM) in the cockpit is compared to the side acceleration NY to generate a side-acceleration error which is filtered and converted to a yaw-rate command. This yaw-rate command is summed with the stability-axis yaw rate RSTAB as well as a guidance-system yaw-rate command YALCM when the approach and landing guidance AL is engaged. The resultant yaw-rate error DRRC is sent to both the rudder, figure B3(b), and the yaw thrusters, figure B4, where it is used in the LATE mode. Also in the LATE mode (fig. B3(b)), an aileron cross-feed term (TEMB from fig. B1(a)) is sampled to provide a coordinated rudder command (DR). This signal is filtered and limited to form the rudder-deflection command DRC. In figure B4, the roll-stick command (PCP from fig. B1) or the guidance-roll command (BANKERR from fig. B1) is used in the EARLY mode to generate a yaw-rate error YAWJET which is sent to the yaw thrusters DRJET and to the rudder, in figure B3(b), below Mach 5. The switching between EARLY and LATE triggers the FADER (fig. B3(b)) to eliminate switching transients. The yaw thrusters remain on until an altitude of 15 km (50 000 ft) is reached to provide sufficient control power in the yaw axis.

Pitch axis.- The elevator and pitch RCS command diagrams are presented in figures B5 and B6. Figure B5(a) shows that a pitch-rate command QC is generated either by converting a stick command DEM from the cockpit in the manual pitch mode (MANP), or by either an angle-of-attack error ALFERR or a normal-acceleration command error ERRNZF, depending on the guidance phase. In the approach and landing guidance phase, the NZCM signal is not smoothed since the guidance and flight control systems are being sampled at the same frequency. Also, a filtered normal-acceleration error ADIF is sent to the attitude-director-indicator error needles. The pitch-rate command QC is compared

to the actual pitch rate QDG as shown in figure B5(b). This error signal QCAL is also compensated for the pitch rate due to yawing at large roll angles RTPHI in the TAEM and AL guidance phases. The pitch-rate error QCAL is filtered, scaled to an elevator deflection, compared to the trim elevator setting DETRIM, filtered by a second-order bending filter, and limited to form the elevator-deflection command DELEC.

For dynamic pressures less than 958 Pa (20 lb/ft^2), a forward loop trim integrator (fig. B5(c)) with compensation for pitch RCS thruster firings (UYC from fig. B6) is used to determine the elevator trim value DETRIM. For higher dynamic pressures, the actual elevator position DELEFB and speed-brake command increment DELSBE are cross fed to the elevator trim. DEMTR is the pitch panel trim from the cockpit.

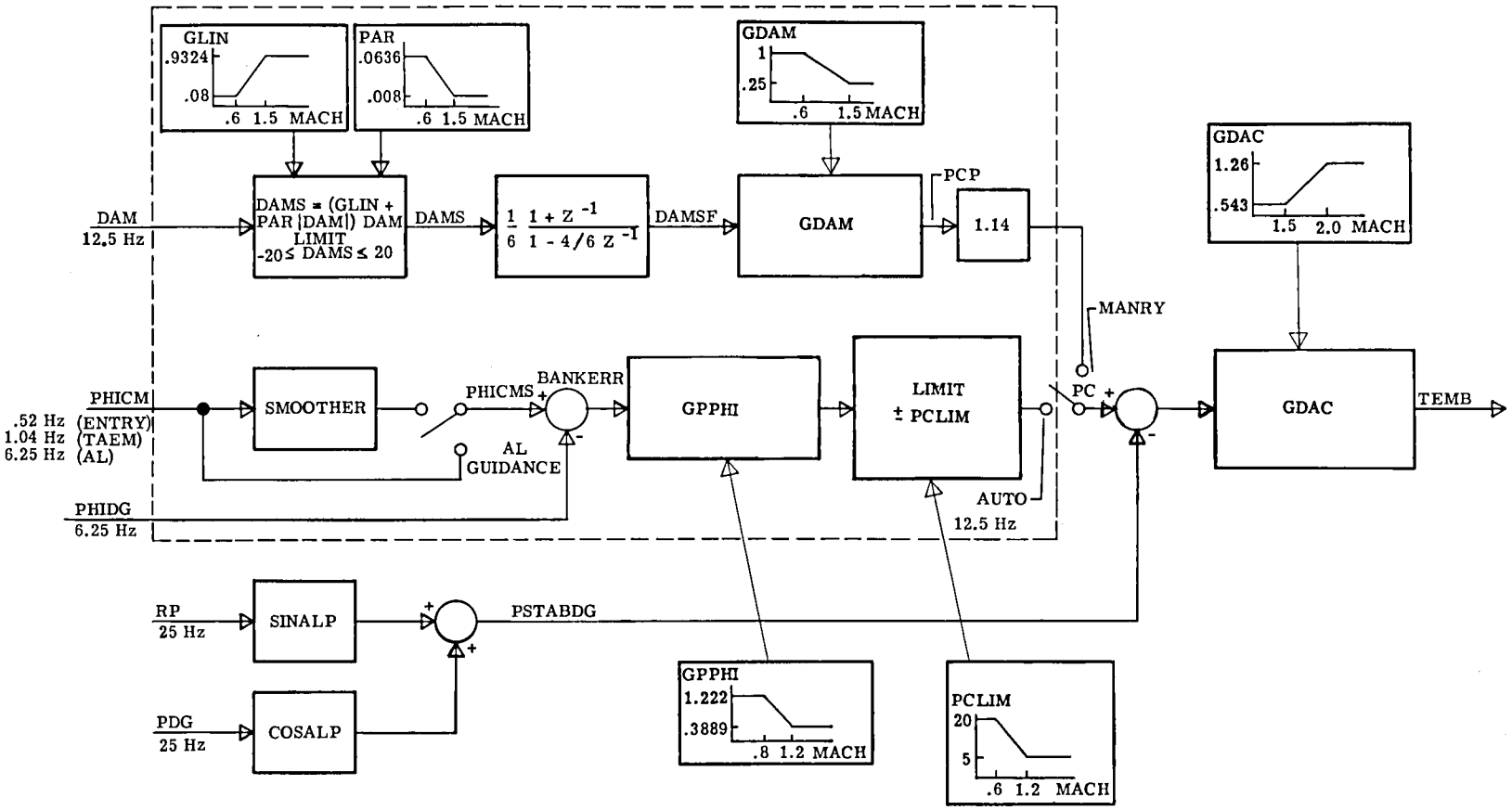
The pitch RCS thrusters (fig. B6) are turned off when QB reaches 958 Pa (20 lb/ft^2). In the AUTO mode the angle-of-attack error signal derived from the guidance command, ALFERRL, is compared to the pitch rate QDG to form a pitch-rate error DPJET which is used to command the thruster firings. In the manual mode, MANP, the pitch-rate command (QC from fig. B5(a)) derived from the hand controller is used to form the pitch-rate error DPJET.

Body-flap command.- The body flap responds to either a manual pilot command DBFRM or, in the automatic mode, to the difference between the elevon pitch trim DETRIM and the scheduled value TRBF. (See fig. B7.) This body-flap mechanization is designed to keep the elevator at the deflection required to obtain the desired yaw due to aileron. The body-flap-rate command BFT is integrated, limited, and then compared to the actual body-flap position DELBF. The command signal DELBFRC to the actuator scaling software program is a rate command. When the orbiter reaches the first flare maneuver, just prior to touchdown, the AL guidance sends a body-flap-retract signal, and the body flap is driven to its undeflected position (DBFPC = 0^o) for landing.

Speed-brake command.- The speed-brake command (fig. B8) is a function of the limited speed-brake deflection DSBM in the manual speed-brake mode. In the automatic mode, the command follows a schedule DSBCOM during the entry guidance phase and a guidance-system command during the TAEM and AL guidance phases. The command from the guidance system is 65^o until MACH = 0.9 where the speed brake becomes an active energy control device.

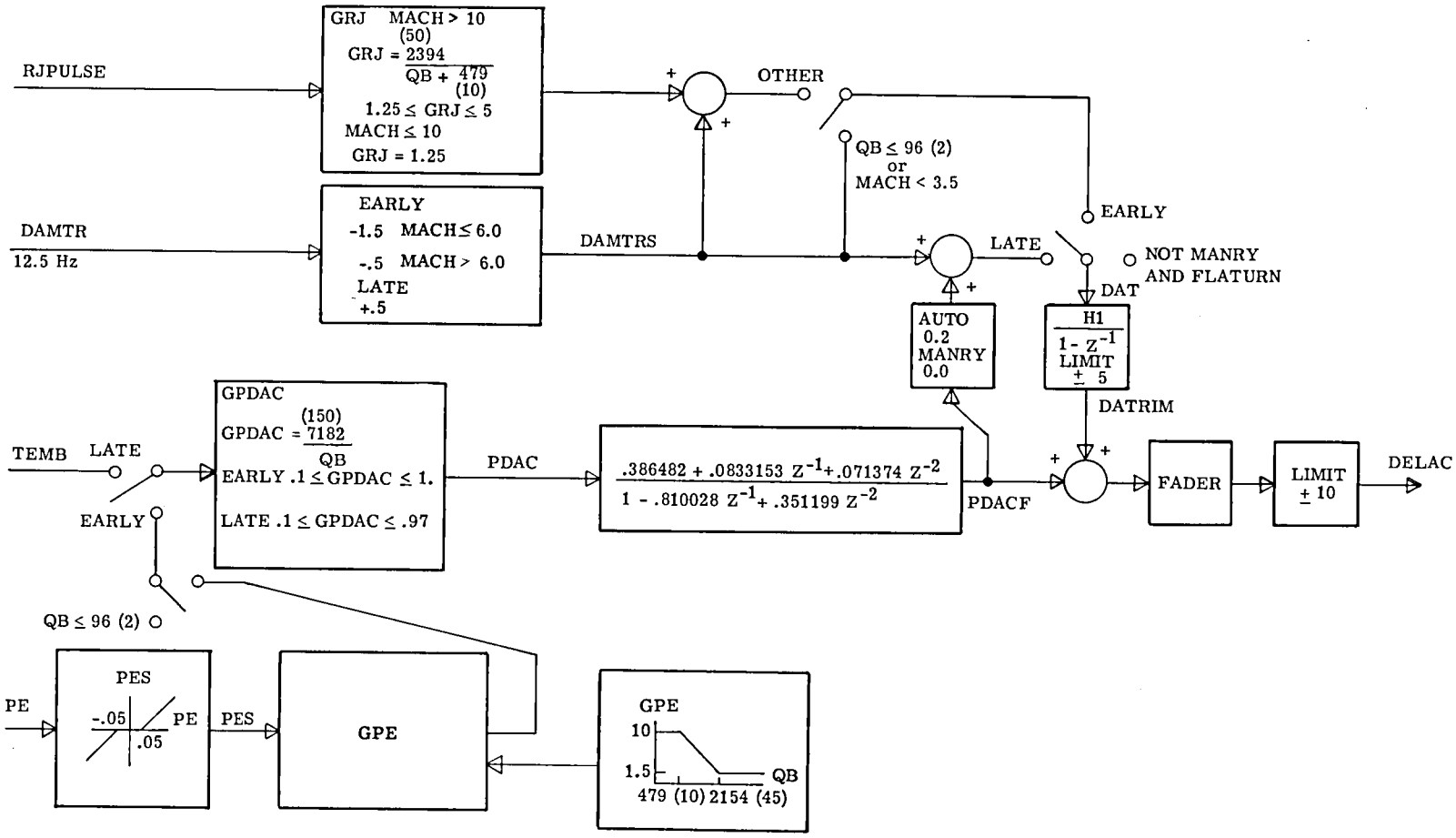
Control-surface rate limiting.- The elevator, aileron, rudder, and speed-brake commands (DELEC, DELAC, DRC, and DSBC, respectively) are subject to software rate-limiting logic which is diagramed in figures B9 and B10. This logic limits the step change in the aerodynamic-surface command to either the maximum surface rate or to a rate consistent with the hydraulic power available. The elevons which deflect due to elevator and aileron commands are rate limited only by the maximum actuator rate. The rudder and speed brake are limited, based on the elevon rates, the hydraulic fluid flow, and the

number of auxiliary power units operating. (DRINCLM, DSBNLM, and DSBPLM are functions of these conditions.) Also, the speed brake may be required to move, based on rudder demand.



(a) Part I.

Figure B1.- Aileron command.



(b) Part II (Frequency of execution = 25 Hz).

Figure B1.- Concluded.

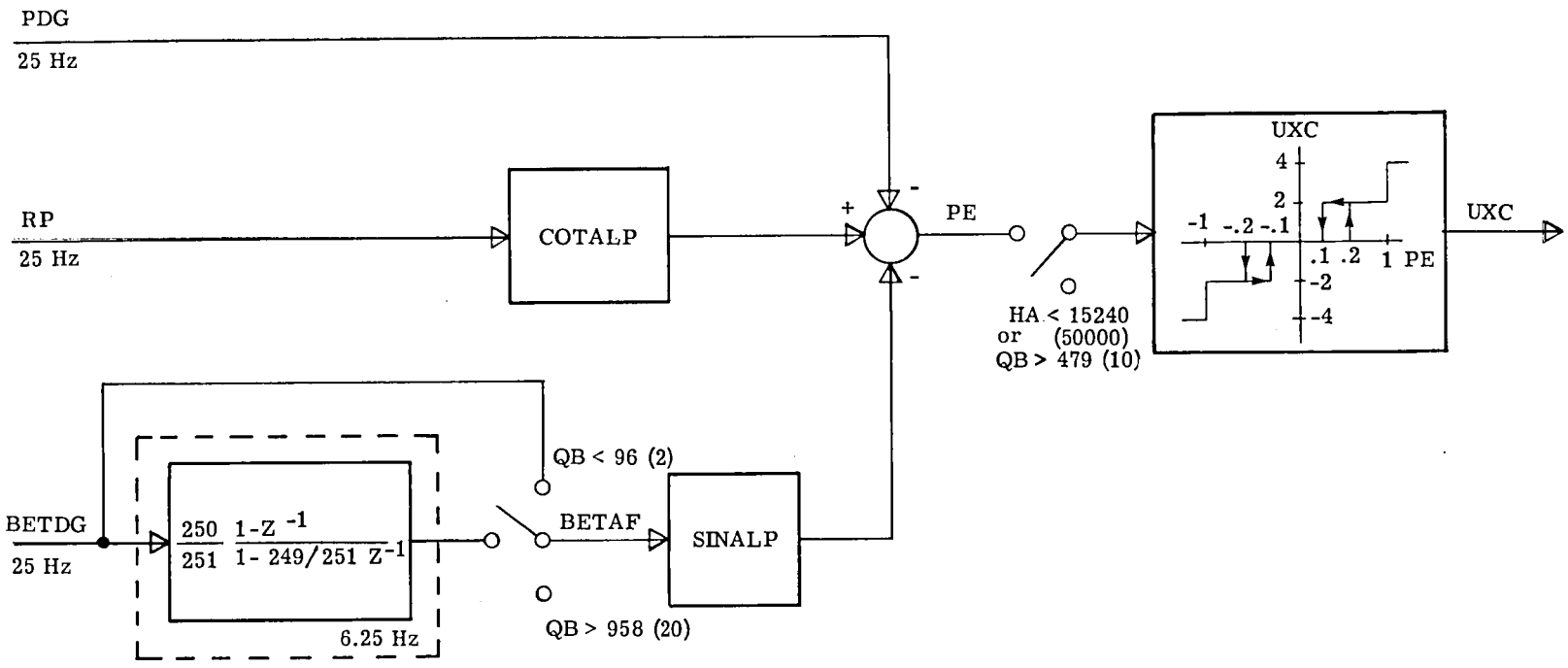
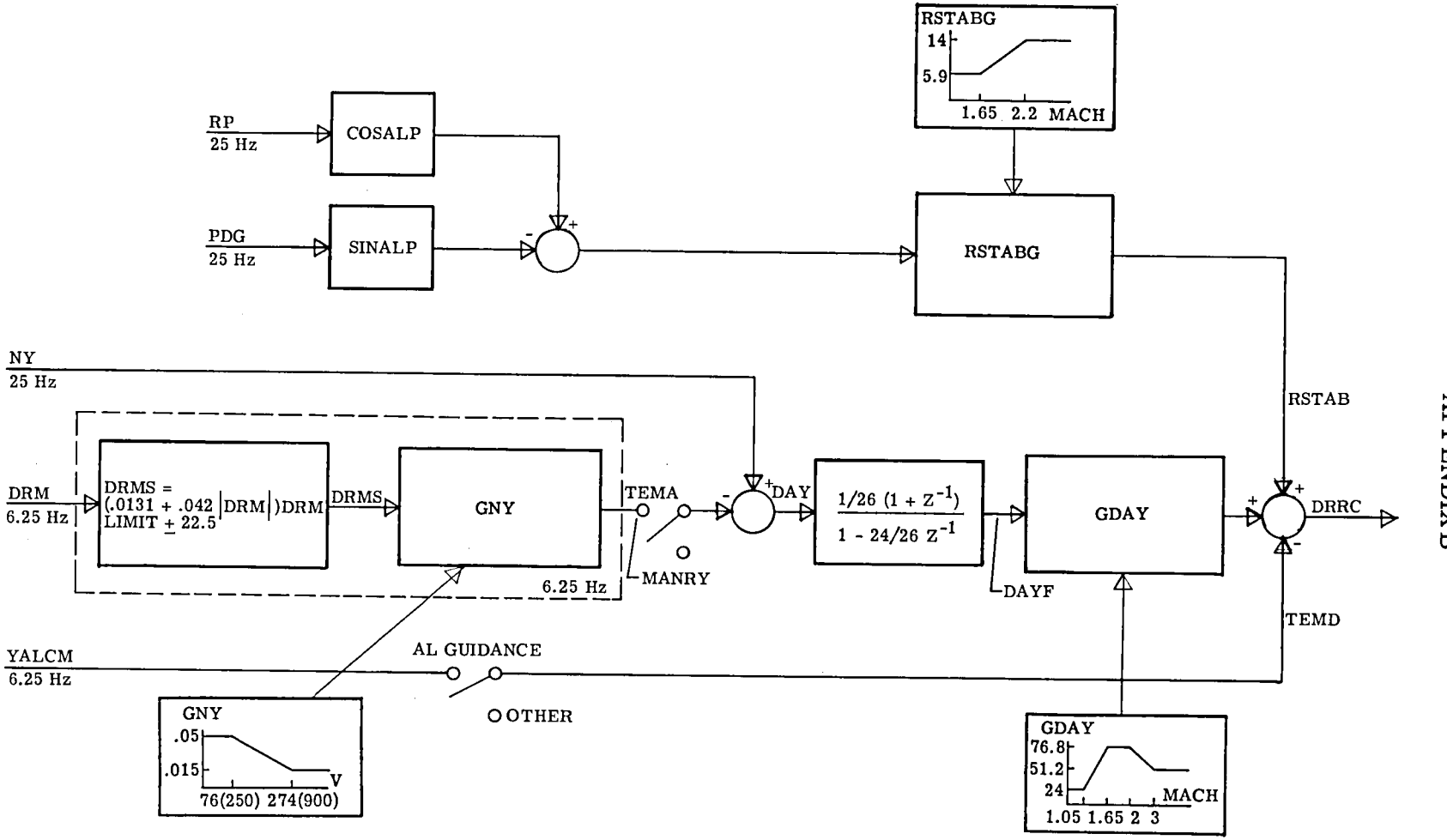
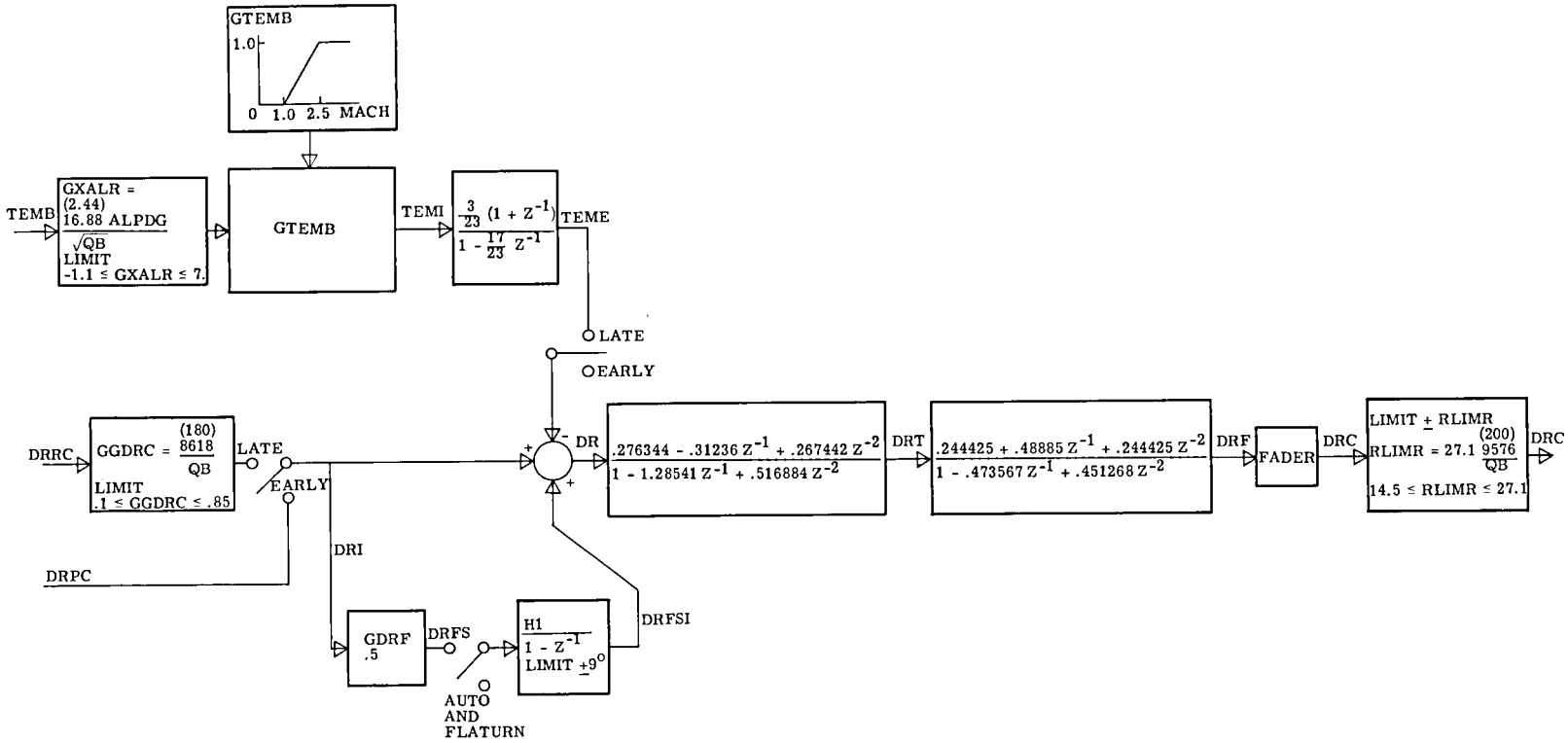


Figure B2.- Roll RCS (Frequency of execution = 25 Hz, unless otherwise noted).



(a) Part I (Frequency of execution = 25 Hz, unless otherwise noted).

Figure B3.- Rudder command.



(b) Part II (Frequency of execution = 25 Hz).

Figure B3.- Concluded.

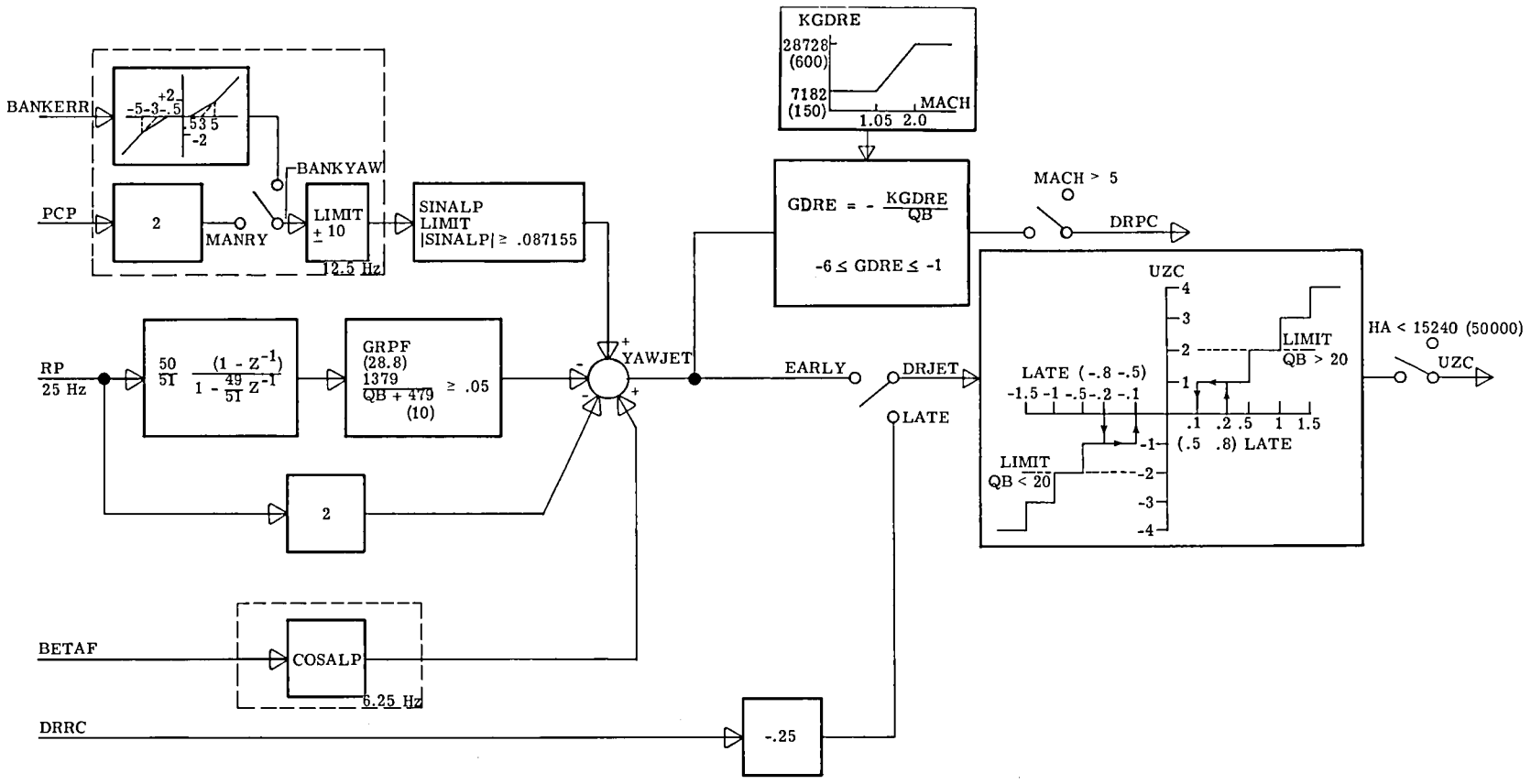
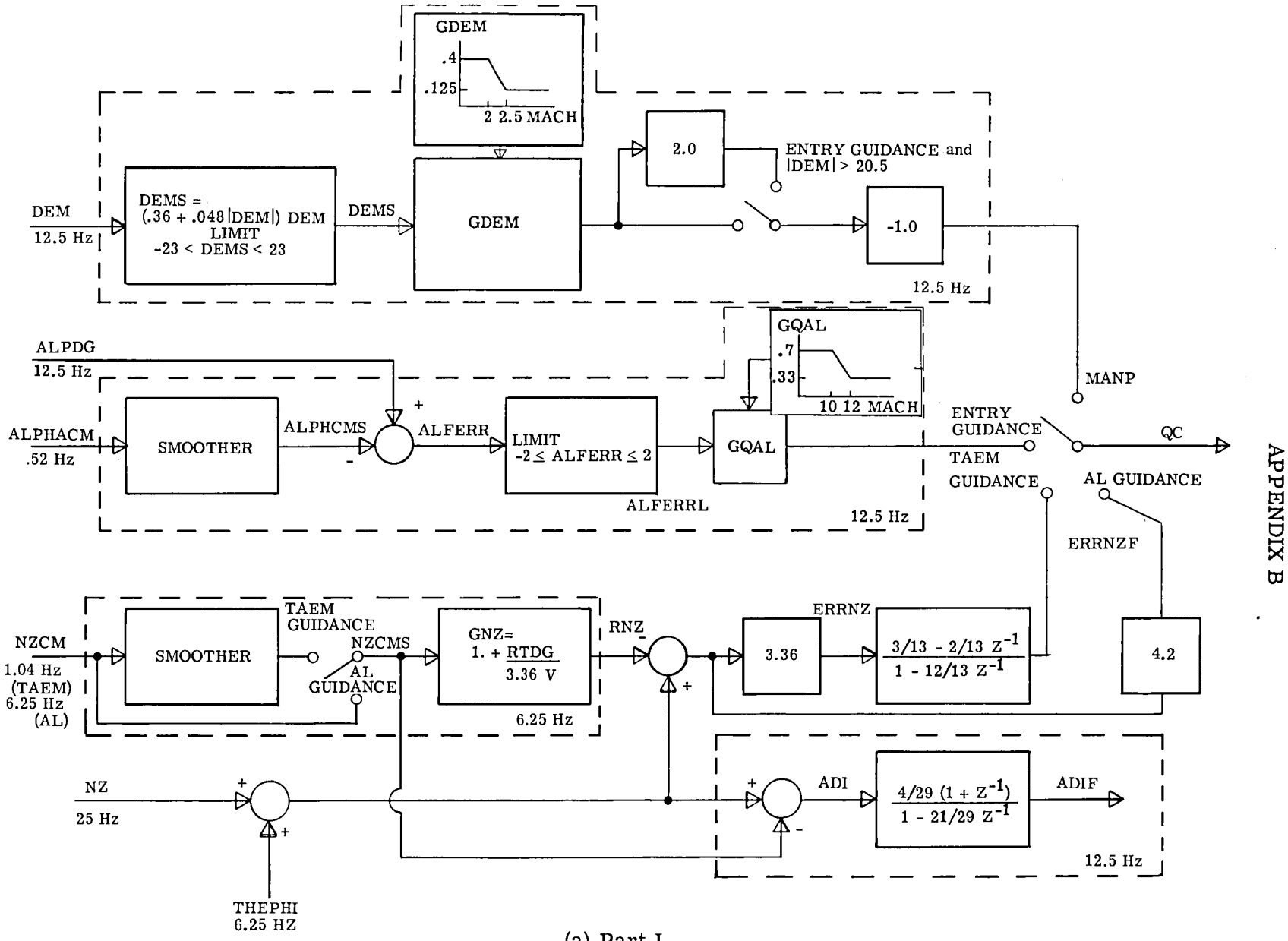
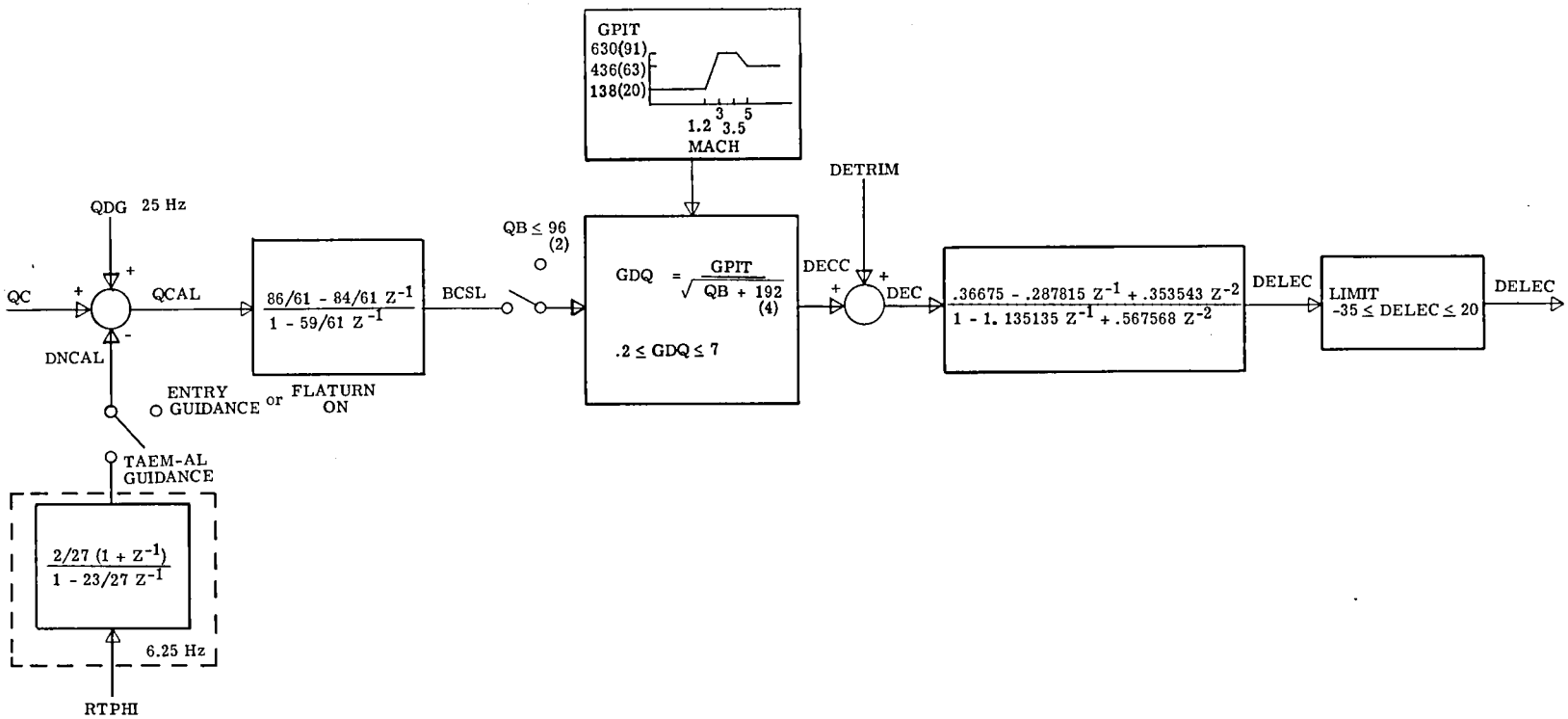


Figure B4.- Yaw RCS (Frequency of execution = 25 Hz, unless otherwise noted).



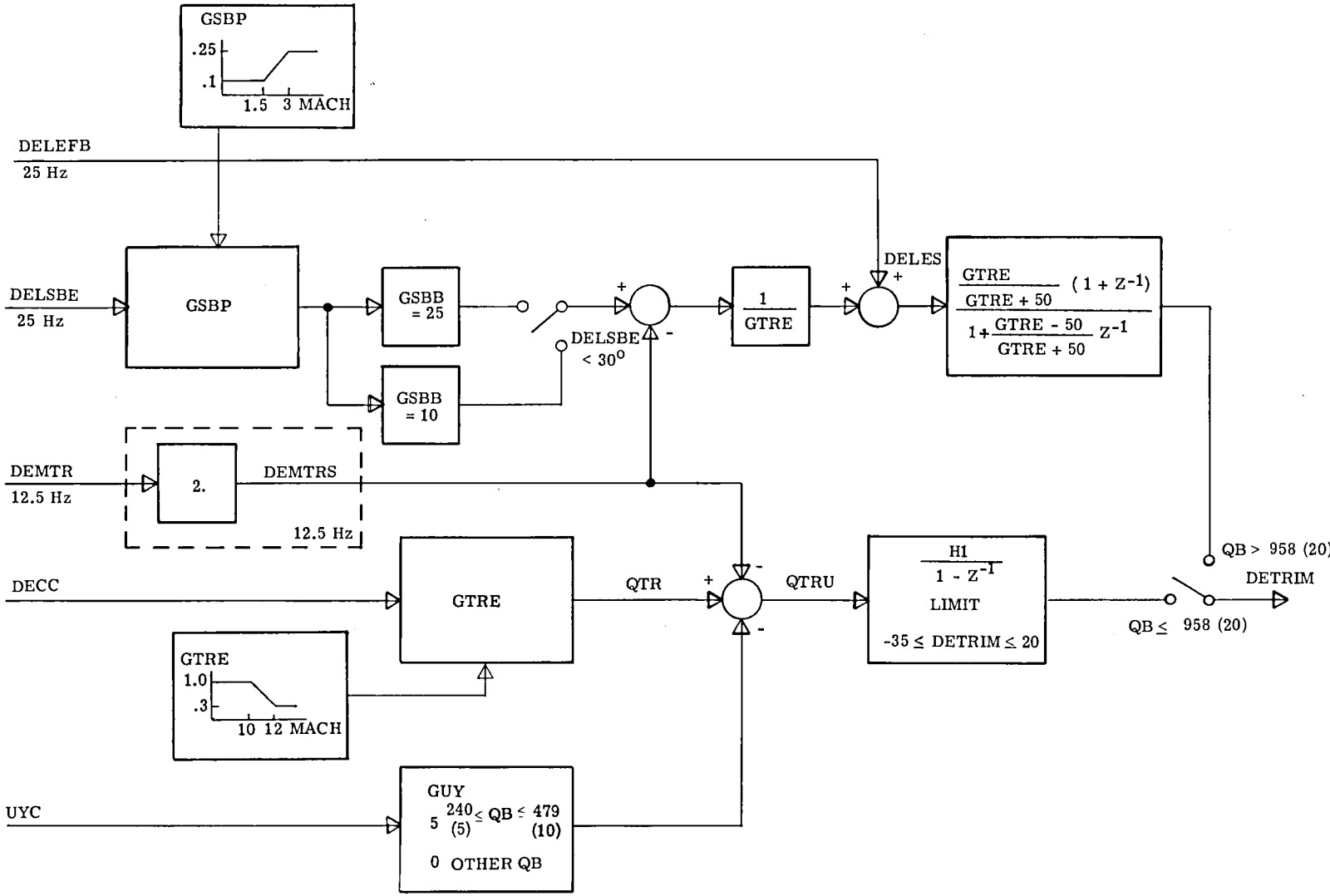
(a) Part I.

Figure B5.- Elevator command.



(b) Part II (Frequency of execution = 25 Hz, unless otherwise noted).

Figure B5.- Continued.



(c) Pitch trim (Frequency of execution = 25 Hz, unless otherwise noted).

Figure B5.- Concluded.

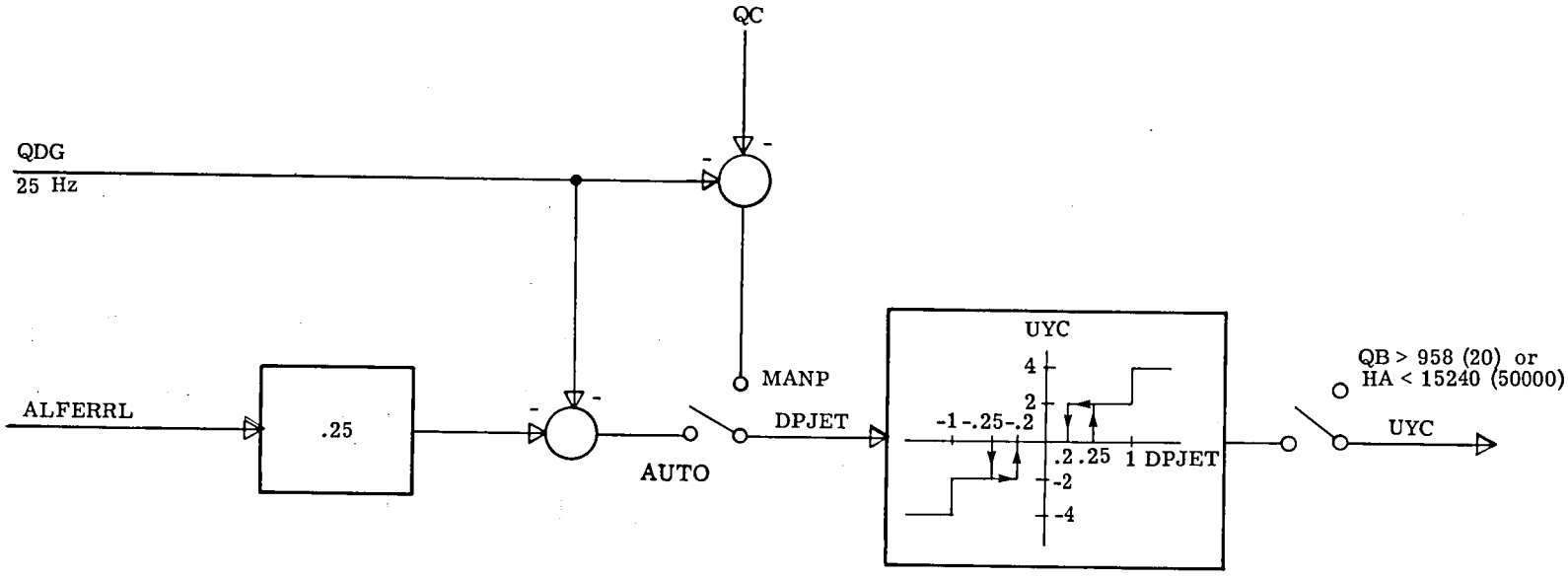


Figure B6.- Pitch RCS (Frequency of execution = 25 Hz).

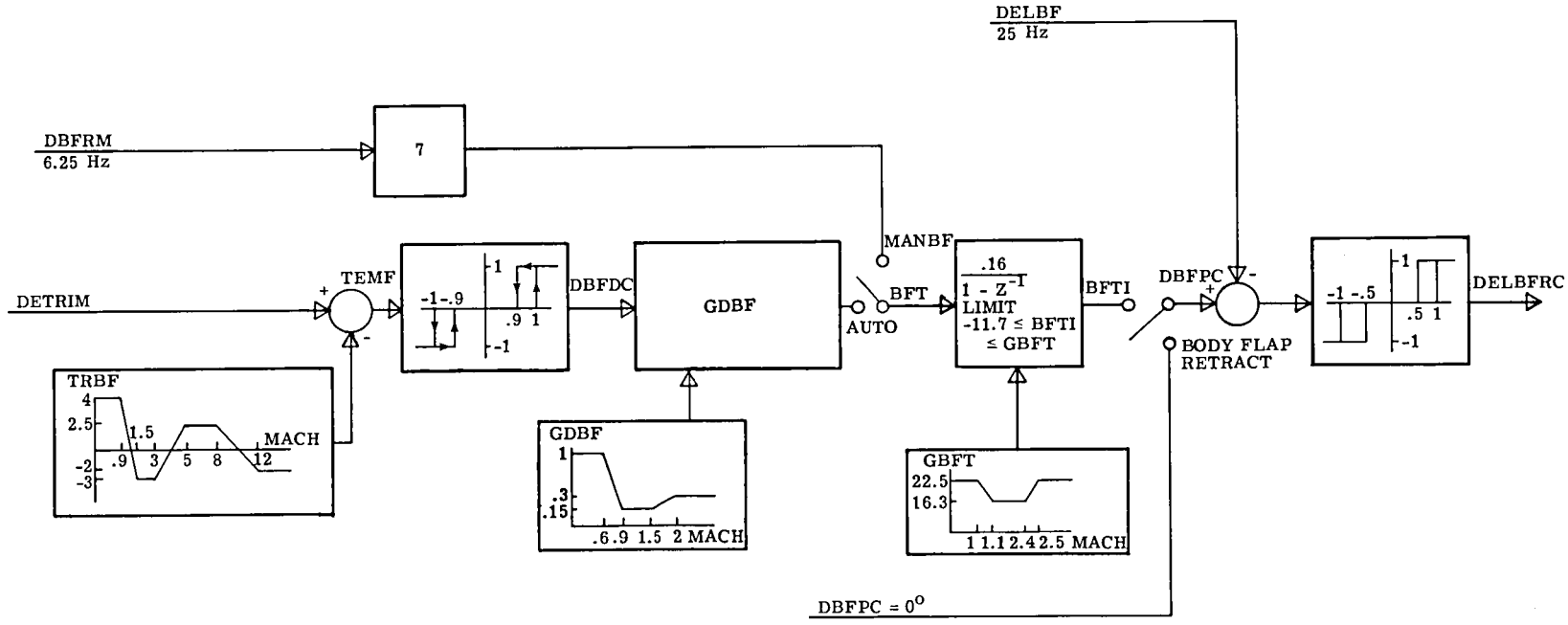


Figure B7.- Body-flap command (Frequency of execution = 6.25 Hz).

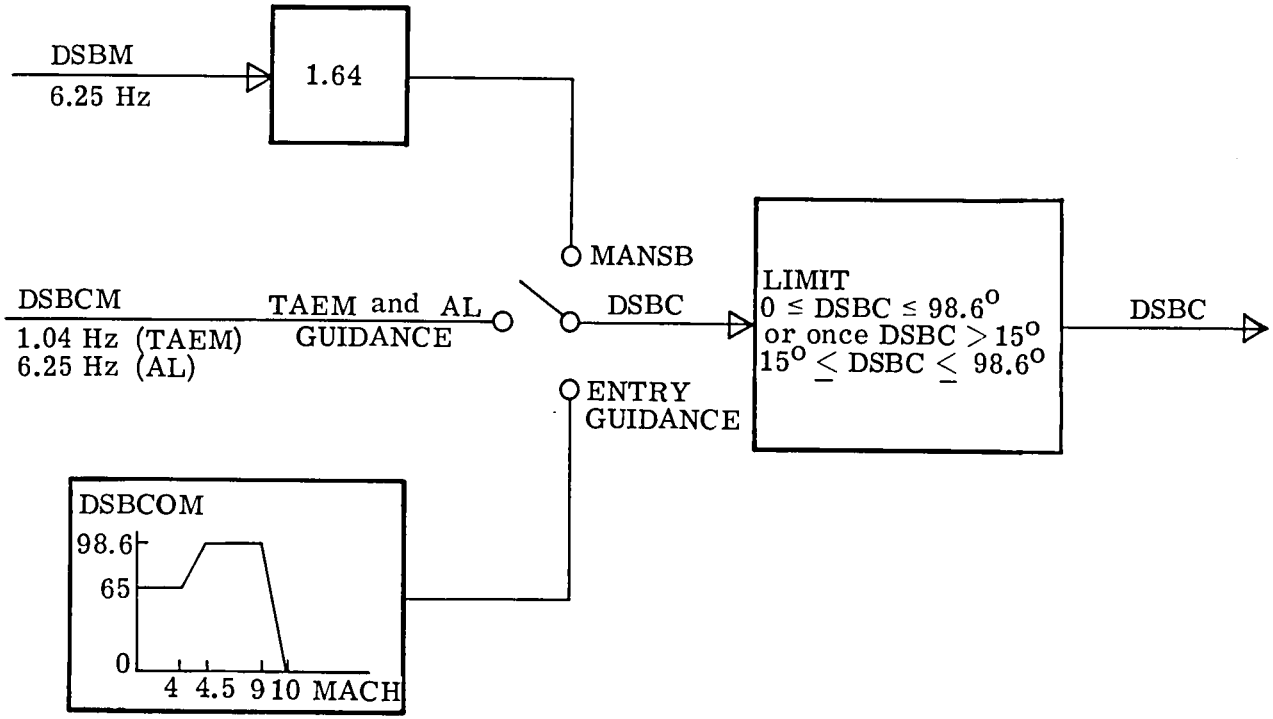


Figure B8.- Speed-brake command (Frequency of execution = 6.25 Hz).

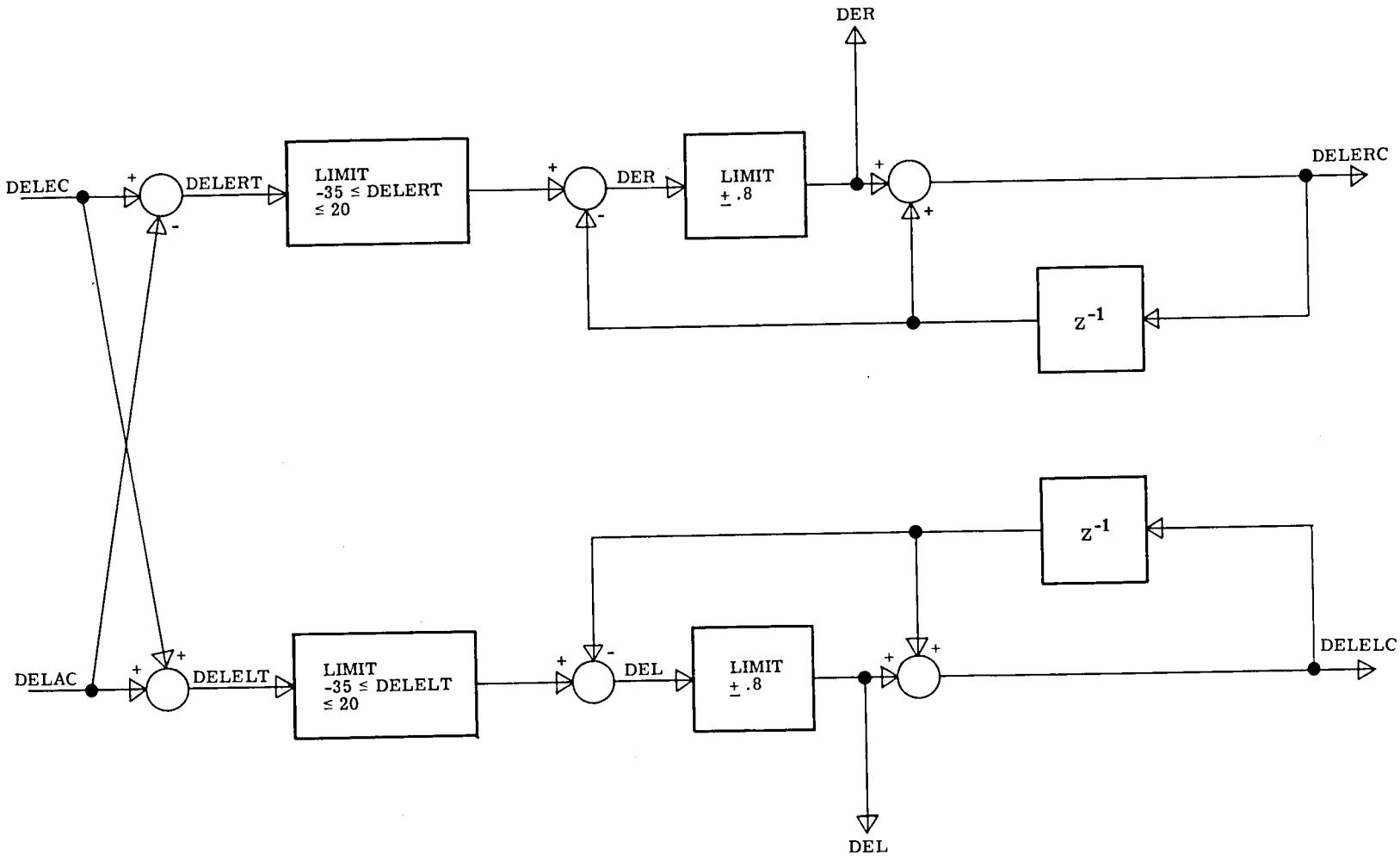


Figure B9.- Elevon-command rate limiting (Frequency of execution = 25 Hz).

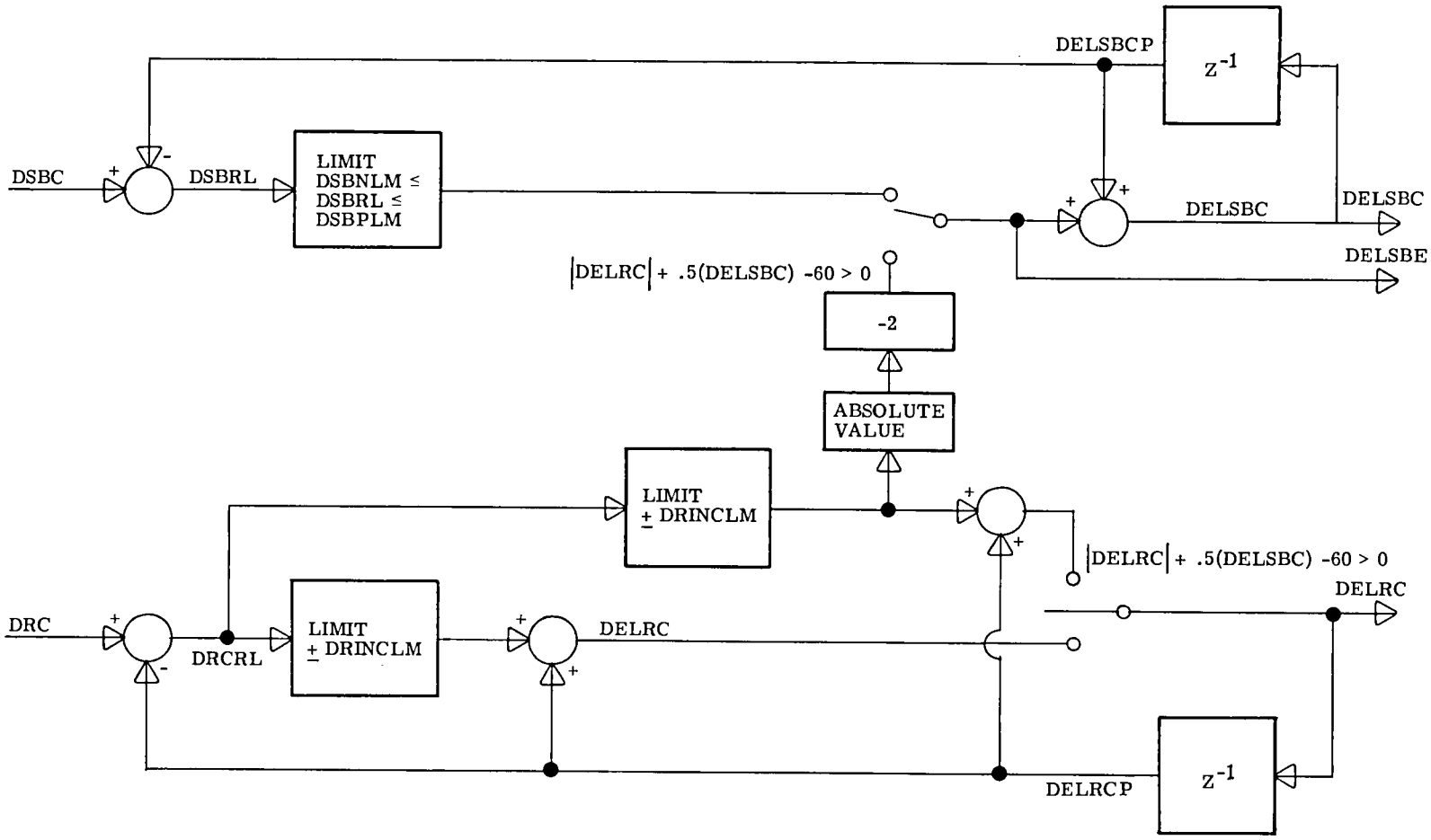


Figure B10.- Rudder-speed-brake-command rate limiting (Frequency of execution = 25 Hz).

APPENDIX C

AERODYNAMIC-DATA DESCRIPTION

The aerodynamic data used in the reentry flight dynamics simulator (RFDS) were obtained from the NASA Johnson Space Center. This data base is known as the December 1975 aerodynamic data base with June 1976 revisions, and the data are fully described and plotted in reference 7. This appendix briefly discusses the characteristics of these data.

The data contain all the aerodynamic coefficients from the low-dynamic-pressure environment of high altitude flight to the aerodynamics at runway touchdown. The effects of all the control surfaces are taken into account as are the viscous effects at low dynamic pressure. Dynamic derivatives are included as well as the effects of landing-gear deployment and the ground effects encountered below certain altitudes. Control-surface hinge moments are provided, and the sideslip derivatives are implemented as a function of sideslip creating nonlinearities in some flight regimes.

In addition, estimates of possible differences between wind-tunnel data and flight data, called variations, have been provided as have measures of scatter in the wind-tunnel data, called tolerances. These uncertainties have been determined for the static aerodynamic coefficients, the control surface derivatives, and the hinge-moment data. The RFDS has been used to apply these uncertainties to the aerodynamic data in several studies to determine their effects on the flight-control-system effectiveness.

Since this data base includes more than 46 000 data points, data had to be packed two data words per computer word in order to save computer memory. Linear interpolation routines designed to use this packed data were programed to provide very fast table look-up to meet the real-time simulation compute-time requirements.

APPENDIX D

TRIM EQUATIONS

In order to facilitate studies of the guidance and control system in an efficient manner, flights are usually initialized just prior to the onset of the conditions under study. Most of these initialization points are just before guidance algorithm switching or major control system switches. Currently, nine initialization points are preprogramed, and others may be specified at the control console. For any initialization point the orbiter must be trimmed to eliminate unrealistic transients at the beginning of the flight. This appendix describes the equations used for this purpose.

Symbols

b	wing span, m (ft)
$\bar{c}$	mean aerodynamic chord, m (ft)
C_A	axial-force coefficient
C_l	rolling-moment coefficient
C_{l_r}	$= \partial C_l \Big/ \partial \frac{rb}{2V}$, rad^{-1}
C_{l_β}	$= \partial C_l / \partial \beta$, deg^{-1}
$C_{l_{\delta_a}}$	$= \partial C_l / \partial \delta_a$, deg^{-1}
$C_{l_{\delta_r}}$	$= \partial C_l / \partial \delta_r$, deg^{-1}
C_m	pitching-moment coefficient
C_n	yawing-moment coefficient
C_N	normal-force coefficient

C_{n_r} $= \partial C_n / \partial \frac{rb}{2V}$, rad^{-1}

C_{n_β} $= \partial C_n / \partial \beta$

$C_{n_{\delta_a}}$ $= \partial C_n / \partial \delta_a$, deg^{-1}

$C_{n_{\delta_r}}$ $= \partial C_n / \partial \delta_r$, deg^{-1}

C_Y side-force coefficient

C_{Y_β} $= \partial C_Y / \partial \beta$, deg^{-1}

$C_{Y_{\delta_a}}$ $= \partial C_Y / \partial \delta_a$, deg^{-1}

$C_{Y_{\delta_r}}$ $= \partial C_Y / \partial \delta_r$, deg^{-1}

I moment-of-inertia matrix

I_{XX}, I_{YY}, I_{ZZ} moments of inertia about body axes, kg-m^2 (slug-ft^2)

I_{XY}, I_{XZ}, I_{YZ} products of inertia about body axes, kg-m^2 (slug-ft^2)

L total rolling moment, N-m (ft-lb)

M total pitching moment, N-m (ft-lb)

N total yawing moment, N-m (ft-lb)

p roll rate about body axes, rad/sec

q pitch rate about body axes, rad/sec

$\bar{q}$ dynamic pressure, Pa (lb/ft^2)

r yaw rate about body axes, rad/sec

S	reference area, m^2 (ft^2)
x_{off}, y_{off}, z_{off}	center-of-gravity offsets, m (ft)
V	velocity, m/sec (ft/sec)
β	sideslip angle, deg
δ_a	aileron deflection, deg
δ_{BF}	body-flap deflection, deg
δ_e	elevator deflection, deg
δ_r	rudder deflection, deg
δ_{SB}	speed-brake deflection, deg
ϕ	body roll angle, deg
ϕ_c	commanded body roll angle, deg

Subscripts:

GE	due to ground effects
LG	due to landing gear
T	total
u	due to uncertainties
v	due to viscous effects

A dot over a quantity indicates the derivative of that quantity with respect to time. A delta (Δ) indicates an increment.

APPENDIX D

Derivation

The trim equations that were developed for RFDS are designed to force $\dot{p}$, $\dot{q}$, and $\dot{r}$ to zero. The pertinent matrix equation is (assuming the moments of inertia are constant)

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = [I]^{-1} \left\{ \begin{bmatrix} L \\ M \\ N \end{bmatrix} - \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times [I] \begin{bmatrix} p \\ q \\ r \end{bmatrix} \right\}$$

where the moment-of-inertia matrix I is

$$[I] = \begin{bmatrix} I_{XX} & -I_{XY} & -I_{XZ} \\ -I_{YX} & I_{YY} & -I_{YZ} \\ -I_{ZX} & -I_{ZY} & I_{ZZ} \end{bmatrix}$$

These reduce (assuming p and q are zero) to

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = [I]^{-1} \begin{bmatrix} L - \left(I_{YZ}\right) r^2 \\ M - \left(I_{XZ}\right) r^2 \\ N \end{bmatrix}$$

Therefore, the requirement to set $\dot{p}$, $\dot{q}$, and $\dot{r}$ to zero reduces to

$$L - \left(I_{YZ}\right) r^2 = 0 \tag{1}$$

$$M - \left(I_{XZ}\right) r^2 = 0 \tag{2}$$

$$N = 0 \tag{3}$$

The r chosen is that required for a coordinated turn which is dependent on ϕ. The ϕ chosen is the initial ϕ_c issued by the guidance routines.

Equations (1) to (3) can be expanded as follows:

$$\bar{q}Sb\left\{\left[\left(C_{l_\beta}\right)_T + \frac{z_{off}}{b}\left(C_{Y_\beta}\right)_T\right]\beta + \left[\left(C_{l_{\delta_a}}\right)_T + \frac{z_{off}}{b}\left(C_{Y_{\delta_a}}\right)\right]\delta_a + \left[\left(C_{l_{\delta_r}}\right)_T + \frac{z_{off}}{b}\left(C_{Y_{\delta_r}}\right)_T\right]\delta_r + C_{l_r}\frac{rb}{2V} + \frac{y_{off}}{b}\left(C_N\right)_T\right\} - \left(I_{YZ}\right)r^2 = 0 \quad (4)$$

$$\bar{q}S\bar{c}\left[\left(C_m\right)_T - \frac{x_{off}}{\bar{c}}\left(C_N\right)_T + \frac{z_{off}}{\bar{c}}\left(C_A\right)_T\right] - \left(I_{XZ}\right)r^2 = 0 \quad (5)$$

$$\bar{q}Sb\left\{\left[\left(C_{n_\beta}\right)_T - \frac{x_{off}}{b}\left(C_{Y_\beta}\right)_T\right]\beta + \left[\left(C_{n_{\delta_a}}\right)_T - \frac{x_{off}}{b}\left(C_{Y_{\delta_a}}\right)\right]\delta_a + \left[\left(C_{n_{\delta_r}}\right)_T - \frac{x_{off}}{b}\left(C_{Y_{\delta_r}}\right)_T\right]\delta_r + C_{n_r}\frac{rb}{2V} - \frac{y_{off}}{b}\left(C_A\right)_T\right\} = 0 \quad (6)$$

where

$$\left(C_{l_\beta}\right)_T = C_{l_\beta} + \left(\Delta C_{l_\beta}\right)_{\delta_{SB}} + \left(\Delta C_{l_\beta}\right)_{\delta_e} + \left(\Delta C_{l_\beta}\right)_u$$

$$\left(C_{l_{\delta_a}}\right)_T = C_{l_{\delta_a}} + \left(\Delta C_{l_{\delta_a}}\right)_v + \left(\Delta C_{l_{\delta_a}}\right)_u$$

$$\left(C_{l_{\delta_r}}\right)_T = C_{l_{\delta_r}} + \left(\Delta C_{l_{\delta_r}}\right)_u$$

$$\left(C_{Y_\beta}\right)_T = C_{Y_\beta} + \left(\Delta C_{Y_\beta}\right)_{\delta_{SB}} + \left(\Delta C_{Y_\beta}\right)_{\delta_e} + \left(\Delta C_{Y_\beta}\right)_u$$

$$\left(C_{Y_{\delta_r}}\right)_T = C_{Y_{\delta_r}} + \left(\Delta C_{Y_{\delta_r}}\right)_u$$

$$\left(C_N\right)_T = C_N + \left(\Delta C_N\right)_{\delta_e} + \left(\Delta C_N\right)_{\delta_{SB}} + \left(\Delta C_N\right)_{\delta_{BF}} + \left(\Delta C_N\right)_{LG} + \left(\Delta C_N\right)_{GE} + \left(\Delta C_N\right)_u$$

$$\left(C_m\right)_T = C_m + \left(\Delta C_m\right)_{\delta_e} + \left(\Delta C_m\right)_{\delta_{SB}} + \left(\Delta C_m\right)_{\delta_{BF}} + \left(\Delta C_m\right)_{LG} + \left(\Delta C_m\right)_{GE} + \left(\Delta C_m\right)_u$$

$$\left(C_A\right)_T = C_A + \left(\Delta C_A\right)_{\delta_e} + \left(\Delta C_A\right)_{\delta_{SB}} + \left(\Delta C_A\right)_{\delta_{BF}} + \left(\Delta C_A\right)_{LG} + \left(\Delta C_A\right)_{GE} + \left(\Delta C_A\right)_u$$

$$\left(C_{n_\beta}\right)_T = C_{n_\beta} + \left(\Delta C_{n_\beta}\right)_{\delta_{SB}} + \left(\Delta C_{n_\beta}\right)_{\delta_e} + \left(\Delta C_{n_\beta}\right)_u$$

$$\left(C_{n_{\delta_a}}\right)_T = C_{n_{\delta_a}} + \left(\Delta C_{n_{\delta_a}}\right)_v + \left(\Delta C_{n_{\delta_a}}\right)_u$$

$$\left(C_{n_{\delta_r}}\right)_T = C_{n_{\delta_r}} + \left(\Delta C_{n_{\delta_r}}\right)_u$$

The speed brake and landing gear position (up or down) must be known before solving the trim equations. Also, the elevator deflection δ_e is set initially to the optimum value as determined by the control system. If equation (5) is then solved by varing the body flap deflection δ_{BF}, then δ_e and δ_{BF} have been determined. If δ_{BF} is unable to produce trim, it is set on its appropriate limit and δ_e moves for trim. Since

both δ_{BF} and δ_e have nonlinear control characteristics, a numerical search routine is used to solve equation (5).

Equations (1) and (3) are then solved simultaneously for the required aileron deflection δ_a, rudder deflection δ_r, and sideslip angle β. Since there are only two equations, one of the unknowns must be set before the equation can be solved. In addition, for there to be a closed form solution, the δ_a, δ_r, and β derivatives are assumed to be constant. The choice of the two variables depends on the control philosophy for the flight regime and has been implemented as follows:

(1) Mach > 4	δ_a	and	β	used with δ_r	set to 0.
(2) 1.26 ≦ Mach ≦ 4	δ_r	and	β	used with δ_a	set to 0.
(3) Mach < 1.26	δ_a	and	δ_r	used with β	set to 0.

Because the Space Shuttle Orbiter uses elevons to act as both elevators and ailerons, any aileron deflection will affect the elevator effectiveness. Because of this, the equations are iterated five times to get the final solution.

APPENDIX E

REACTION-CONTROL-SYSTEM MODEL INCLUDING AERODYNAMIC FLOW INTERACTION

The reaction-control-system (RCS) thrusters used during the entry flight are located above and at the trailing edge of the wing as is shown in figure 2. When the thrusters are fired during the atmospheric flight, the plumes interact with the flow field over the wing and along the side of the vertical tail. Wind-tunnel tests have shown that the interaction effects from plume impingement and flow field alteration due to the presence of the plume are significant. The model described in this appendix was derived from wind-tunnel testing and accounts for these interaction effects.

Symbols

H_{min} altitude below which minimum on-time for reaction control system (RCS) thrusters is increased, m (ft)

L_{LHDF} coefficient in L_{RCS} equation that shows left-hand downfiring thrusters contribution

L_{LHSF} coefficient in L_{RCS} equation that shows left-hand sidefiring thrusters contribution

L_{LHUF} coefficient in L_{RCS} equation that shows left-hand upfiring thrusters contribution

L_{RCS} total rolling moment produced by RCS, N-m (ft-lb)

L_{RHDF} coefficient in L_{RCS} equation that shows right-hand downfiring thrusters contribution

L_{RHSF} coefficient in L_{RCS} equation that shows right-hand sidefiring thrusters contribution

L_{RHUF} coefficient in L_{RCS} equation that shows right-hand upfiring thrusters contribution

M_{LHDF} coefficient in M_{RCS} equation that shows left-hand downfiring thrusters contribution

M_{LHSF} coefficient in M_{RCS} equation that shows left-hand sidefiring thrusters contribution

M_{LHUF} coefficient in M_{RCS} equation that shows left-hand upfiring thrusters contribution

M_{RCS} total pitching moment produced by RCS, N-m (ft-lb)

M_{RHDF} coefficient in M_{RCS} equation that shows right-hand downfiring thrusters contribution

M_{RHSF} coefficient in M_{RCS} equation that shows right-hand sidefiring thrusters contribution

M_{RHUF} coefficient in M_{RCS} equation that shows right-hand upfiring thrusters contribution

N_{LHDF} coefficient in N_{RCS} equation that shows left-hand downfiring thrusters contribution

N_{LHSF} coefficient in N_{RCS} equation that shows left-hand sidefiring thrusters contribution

N_{LHUF} coefficient in N_{RCS} equation that shows left-hand upfiring thrusters contribution

N_{RCS} total yawing moment produced by RCS, N-m (ft-lb)

N_{RHDF} coefficient in N_{RCS} equation that shows right-hand downfiring thrusters contribution

N_{RHSF} coefficient in N_{RCS} equation that shows right-hand sidefiring thrusters contribution

N_{RHUF} coefficient in N_{RCS} equation that shows right-hand upfiring thrusters contribution

$\bar{q}$ dynamic pressure, Pa (lb/ft^2)

T_{LHDF} total thrust of left-hand downfiring thrusters, N (lb)

T_{LHSF} total thrust of left-hand sidefiring thrusters, N (lb)

T_{LHUF} total thrust of left-hand upfiring thrusters, N (lb)

T_{RHDF} total thrust of right-hand downfiring thrusters, N (lb)

T_{RHSF} total thrust of right-hand sidefiring thrusters, N (lb)

T_{RHUF} total thrust of right-hand upfiring thrusters, N (lb)

$\left(X_{cg}\right)_m$ average distance between center of gravity and RCS thrusters used for pitch control, m (ft)

$\left(X_{cg}\right)_n$ average distance between center of gravity and RCS thrusters used for yaw control, m (ft)

$\left(Z_{cg}\right)_\ell$ average distance between center line of orbiter and RCS thrusters used for roll control, m (ft)

Subscripts:

A coefficient is ratio of actual interference-free moment (including cant angle effects) to design moment

I coefficient includes aerodynamic interference effects

Description of Reaction Control System

The reaction control system (RCS) is used to provide pitch, roll, and yaw control moments. The pitch RCS is used to augment the elevator and body flap and is active for $\bar{q}$ less than 958 Pa (20 lb/ft^2). The roll RCS is used to augment the aileron and is active for $\bar{q}$ less than 479 Pa (10 lb/ft^2). The yaw RCS is active until the Space Shuttle Orbiter's altitude drops below 15 240 m (50 000 ft) and is used in two ways. Until the rudder is activated, the yaw RCS is used for roll attitude control. After the rudder is activated, the yaw RCS augments the rudder. Block diagrams of the software implementation of the RCS commands are given in appendix B.

The RCS model used in RFDS consists of four subroutines called FIRETAB, EM, ECONF, and MNJETON. FIRETAB and EM are called by the main program at intervals

equal to the minimum on-time for the RCS. FIRETAB determines if the RCS thrusters are to be turned on by converting the commands from the control system to right or left, up, down, or sidefiring thrusters. At this point, ECONF is called to insure that both downfiring and upfiring thrusters on the same side are not commanded to be on. This condition is caused when both the pitch thrusters and the roll thrusters are commanded on. If this condition exists, ECONF turns off the excess thrusters. If the Space Shuttle Orbiter drops below H_{min}, the minimum on-time for the RCS must be increased from 0.04 to 0.08 sec. Thus, MNJETON is called if the altitude is below H_{min} + 304 m (1000 ft) to determine if any thrusters can be turned off by FIRETAB. MNJETON is called before H_{min} is reached for initialization purposes. Next, FIRETAB calculates the total thrust (without aerodynamic interference) for the upfiring, downfiring, and sidefiring thrusters. EM then calculates the resultant forces and moments of the RCS and considers both aerodynamic interaction effects and cant angle effects. The equations* used are:

$$
\begin{aligned}
L_{RCS} = {} & \left[\left(L_{LHDF}\right)_I + \left(L_{LHDF}\right)_A\right]\left(Z_{cg}\right)_\ell \, T_{LHDF} \\
& + \left[\left(L_{LHSF}\right)_I + \left(L_{LHSF}\right)_A\right]\left(Z_{cg}\right)_\ell \, T_{LHSF} \\
& + \left[\left(L_{LHUF}\right)_I + \left(L_{LHUF}\right)_A\right]\left(Z_{cg}\right)_\ell \, T_{LHUF} \\
& + \left[\left(L_{RHDF}\right)_I + \left(L_{RHDF}\right)_A\right]\left(Z_{cg}\right)_\ell \, T_{RHDF} \\
& + \left[\left(L_{RHSF}\right)_I + \left(L_{RHSF}\right)_A\right]\left(Z_{cg}\right)_\ell \, T_{RHSF} \\
& + \left[\left(L_{RHUF}\right)_I + \left(L_{RHUF}\right)_A\right]\left(Z_{cg}\right)_\ell \, T_{RHUF}
\end{aligned}
$$

*The interaction and amplification coefficients, subscripted with I and A, respectively, are defined to be referenced to a nominal moment about each axis. Thus, the coefficients that determine roll due to sidefiring thrusters are referenced to the nominal roll moment.

$$M_{RCS} = \left[\left(M_{LHDF}\right)_I + \left(M_{LHDF}\right)_A\right]\left(X_{cg}\right)_m T_{LHDF}$$
$$+ \left[\left(M_{LHUF}\right)_I + \left(M_{LHUF}\right)_A\right]\left(X_{cg}\right)_m T_{LHUF}$$
$$+ \left[\left(M_{LHSF}\right)_I + \left(M_{LHSF}\right)_A\right]\left(X_{cg}\right)_m T_{LHSF}$$
$$+ \left[\left(M_{RHDF}\right)_I + \left(M_{RHDF}\right)_A\right]\left(X_{cg}\right)_m T_{RHDF}$$
$$+ \left[\left(M_{RHUF}\right)_I + \left(M_{RHUF}\right)_A\right]\left(X_{cg}\right)_m T_{RHUF}$$
$$+ \left[\left(M_{RHSF}\right)_I + \left(M_{RHSF}\right)_A\right]\left(X_{cg}\right)_m T_{RHSF}$$

$$N_{RCS} = \left[\left(N_{LHDF}\right)_I + \left(N_{LHDF}\right)_A\right]\left(X_{cg}\right)_n T_{LHDF}$$
$$+ \left[\left(N_{LHUF}\right)_I + \left(N_{LHUF}\right)_A\right]\left(X_{cg}\right)_n T_{LHUF}$$
$$+ \left[\left(N_{LHSF}\right)_I + \left(N_{LHSF}\right)_A\right]\left(X_{cg}\right)_n T_{LHSF}$$
$$+ \left[\left(N_{RHDF}\right)_I + \left(N_{RHDF}\right)_A\right]\left(X_{cg}\right)_n T_{RHDF}$$
$$+ \left[\left(N_{RHUF}\right)_I + \left(N_{RHUF}\right)_A\right]\left(X_{cg}\right)_n T_{RHUF}$$
$$+ \left[\left(N_{RHSF}\right)_I + \left(N_{RHSF}\right)_A\right]\left(X_{cg}\right)_n T_{RHSF}$$

APPENDIX F

NONLINEAR ACTUATOR MODEL

The RFDS provides console selection of either a linear or a nonlinear, first-order actuator model. The linear model uses the sum of the inner and outer elevon panel hinge moments to treat the elevons as a single panel rather than as having separate inboard and outboard components. The linear model was described in appendix B of reference 1. The nonlinear model, described in detail in this appendix F, takes into account the deflections for both the inner and the outer surfaces for the elevon as well as the separate speed-brake panels which serve also as the rudder.

Notation

The models were designed for measurements in U.S. Customary Units. Therefore, values are given in both SI and U.S. Customary Units.

AR ram piston area, m^2 (in^2)

CEICD elevon servo output, mA

CEICDH current within elevon torque motor, mA

CEICDL limited elevon servo output, mA

CILIM servo current limit, mA

CRICD rudder servo output, mA

CRICDH current within rudder torque motor, mA

CRICDL limited rudder servo output, mA

D volumetric displacement of rudder actuator, m^3/rad (in^3/rad)

DEF filtered elevon command, deg

DELEL elevon surface position, deg

DELELC input elevon command, deg

DELELFB elevon position feedback, deg

DELR rudder surface position, deg

DELRC input rudder command, deg

DELSB speed-brake surface position, deg

DL = (DELRC - DELR) + (DSBCOM - DELSB)/2, deg

DLPAN left surface position, deg

DLPP = DRP + DSBP, deg

DR = (DELRC - DELR) - (DSBCOM - DELSB)/2, deg

DRDOT rudder surface rate, rad/sec

DRF rudder surface position without compliance, deg

DRP integral of rudder rate, deg

DRPAN right surface position, deg

DRPP = DRP - DSBP, deg

DSBCOM input speed-brake command, deg

DSBP integral of speed-brake rate, deg

DSRTMN minimum speed-brake rate, deg/sec

DSRTMX maximum speed-brake rate, deg/sec

DZ rudder gear dead zone, deg

EFF rudder gear efficiency

EXFB elevon actuator position feedback, m (in.)

GR	rudder gear ratio
HMET	elevon hinge moment, N-m (lb-ft)
HMRL	left-rudder panel hinge moment, N-m (lb-ft)
HMRR	right-rudder panel hinge moment, N-m (lb-ft)
HMSB	speed-brake hinge moment, N-m (lb-ft)
HMTOF	elevon system loading, N (lb)
HMTOFS	deflection due to elevon system stiffness external to actuator, m (in.)
HMTOFT	deflection due to total elevon actuator system stiffness, m (in.)
HYST	elevon torque motor and secondary actuator hysteresis, mA
HYSTF	hysteresis due to stiffness, elevon seal, and primary actuator ram friction, m (in.)
HYSTR	rudder torque motor and secondary actuator hysteresis, mA
KA	elevon servo gain, mA/V
KARR	rudder servo gain, mA/deg
KCMD	elevon command voltage scaling, V/deg
KFB	elevon position linear variable differential transformer gain, V/in.
KHR	rudder compliance, N-m/rad (in-lb/rad)
KQP	rudder power spool flow gain, $m^3/sec/kg^{1/2}$ ($in^3/sec/lb^{1/2}$)
KQPS	elevon power spool flow gain, $m^3/sec/kg^{1/2}$ ($in^3/sec/lb^{1/2}$)
KS	elevon system stiffness external and actuator, N/m (lb/in.)

KT total elevon actuation system stiffness, N/m (lb/in.)

KV elevon secondary actuator gain, m/mA (in/mA)

KVR rudder secondary actuator gain, m/mA (in/mA)

K2 elevon switching valve pressure drop coefficient, $(N/m^2)/(m^3/sec)^2$ $((lb/in^2)/(in^3/sec)^2)$

K2R rudder switching valve pressure drop coefficient, $(N/m^2)/(m^3/sec)^2$ $((lb/in^2)/(in^3/sec)^2)$

OLAP elevon valve overlap, m (in.)

OLAPR rudder power spool overlap, m (in.)

PL ram pressure due to aerodynamic loading of elevon, N/m^2 (lb/in^2)

PLR rudder motor loading due to aerodynamic loading of rudder surfaces, N/m^2 (lb/in^2)

PSS hydraulic system supply pressure, N/m^2 (lb/in^2)

PSSR hydraulic system supply pressure, N/m^2 (lb/in^2)

PV pressure differential across primary elevon actuator minus losses, N/m^2 (lb/in^2)

PVR pressure differential across primary rudder actuator minus losses, N/m^2 (lb/in^2)

PV1 supply pressure minus pressure due to aerodynamic loading on elevon, N/m^2 (lb/in^2)

PVR1 supply pressure minus pressure due to aerodynamic loading on rudder, N/m^2 (lb/in^2)

QL volume flow rate for elevon actuator, m^3/sec (in^3/sec)

QLR	volume flow rate for rudder actuator, m^3/sec (in^3/sec)
R	elevon control horn length, m (in.)
RTD	= $180/\pi$, deg/rad
s	Laplace operator
TLP	torque on gear due to aerodynamic forces on left rudder panel, N-m (in-lb)
TLPP	torque due to loading on left rudder panel, N-m (in-lb)
TRP	torque on gear due to aerodynamic forces on right rudder panel, N-m (in-lb)
TRPP	torque due to loading on right rudder panel, N-m (in-lb)
X	input to rudder motor equations, m (in.)
XPS	input to elevon power valve equations, m (in.)
XR	ideal elevon ram displacement, m (in.)
XRH	ideal elevon ram displacement minus hysteresis, m (in.)
XRS	actual elevon ram displacement minus hysteresis, m (in.)
ZP	elevon power spool inlet opening, m (in.)
ZPR	rudder power spool inlet opening, m (in.)
ζ_F	elevon command filter damping ratio
ζ_R	rudder command filter damping ratio
ω_F	elevon command filter break frequency, rad/sec
ω_R	rudder command filter break frequency, rad/sec

APPENDIX F

Actuator Equations

The nonlinear, first-order actuator models described in this appendix are designed to account for the following: the gain and transfer characteristics of the actuator servoamplifiers, the hysteresis and gain of the torque motors and secondary hydraulic actuators, the hysteresis, ram friction, stiffness, and power valve characteristics of the primary actuators, and the gear efficiency and dead zones of the rudder and speed brake. The models are diagramed in figures F1 and F2.

The diagrams show transfer functions in the s-plane form for the elevon and rudder input filters and for the speed-brake model. These functions were integrated by in-line code using coefficients that can be derived using the convolution integration technique described in appendix G.

The power valve equations for the elevon are as follows:

$$ZP = XPS - (OLAP)\ \text{sign}\ (XPS)$$

$$PV1 = PSS - (PL)\ \text{sign}\ (XPS)$$

$$PV = PV1 - K2\left[\frac{(KQPS)^2\ (ZP)^2\ (PV1)}{1 + (KQPS)^2\ (ZP)^2\ (K2)}\right]$$

$$QL = \left[\frac{(KQPS)^2\ (ZP)^2\ (PV1)}{1 + (KQPS)^2\ (ZP)^2\ (K2)}\right]^{1/2} \left[\text{sign}\ (PV \cdot XPS)\right]$$

where sign ( ) denotes the algebraic sign of the parenthetical quantity.

Since there are four elevon surfaces involved, the model is sequenced for each surface; and to meet accuracy requirements, the model is sampled in four steps for every commanded input. Thus, the program must contain looping structure to meet these requirements.

The rudder motor equations are as follows:

$$ZPR = X - (OLAPR)\left[\text{sign}\ (X)\right]$$

$$PVR1 = PSSR - (PLR)\left[\text{sign}\ (X)\right]$$

$$PVR = PVR1 - K2R\left[\frac{(KQP)^2\,(ZPR)^2\,(PVR1)}{1 + (KQP)^2\,(ZPR)^2\,(K2R)}\right]$$

$$QLR = \left[\frac{(KQP)^2\,(ZPR)^2\,(PVR1)}{1 + (KQP)^2\,(ZPR)^2\,(K2R)}\right]^{1/2}\left[\text{sign}\,(PVR \cdot X)\right]$$

$$DRDOT = 3.0\,\frac{QLR}{(D)(GR)}$$

These equations account for gear efficiency as well as the hydraulic actuator characteristics; the gear dead-zone effects are included later in the model. The integration of the rudder actuator model is also performed in four steps per command input for accuracy.

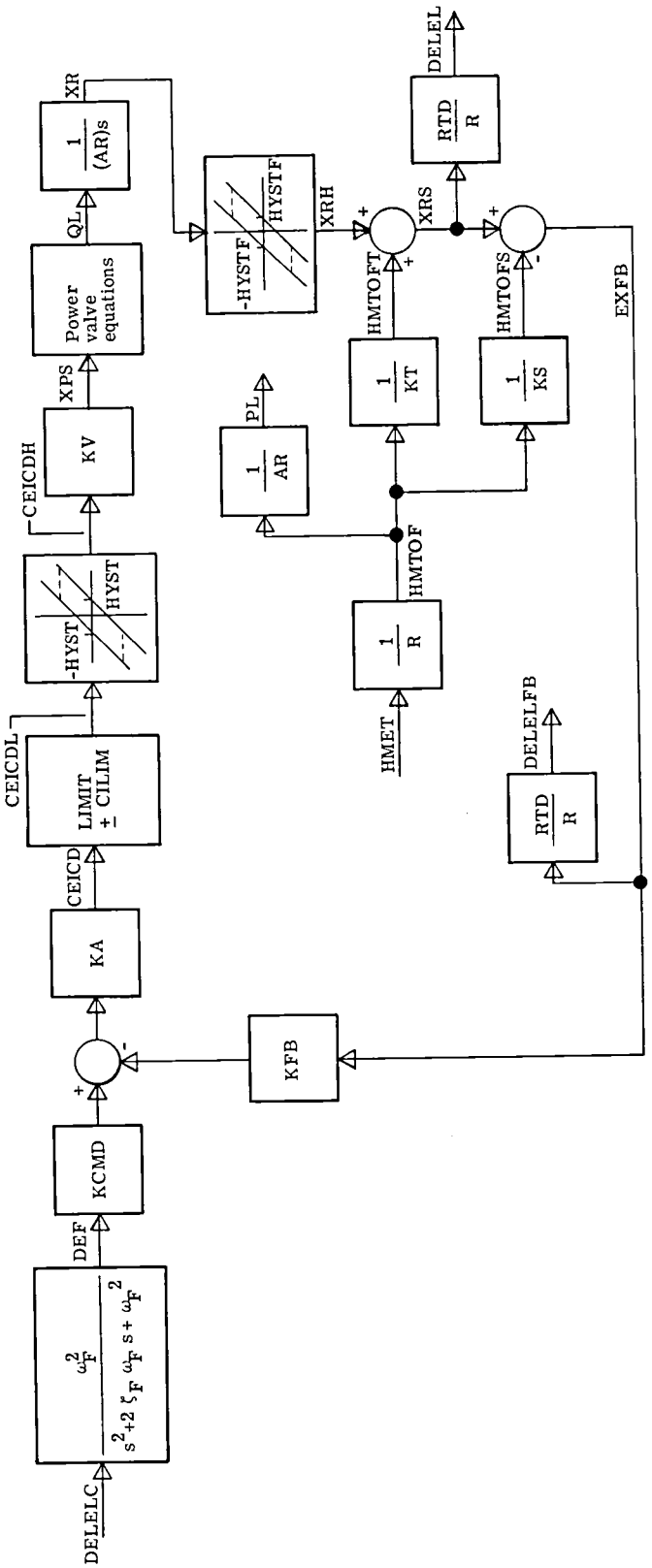


Figure F1.- Elevon actuator model.

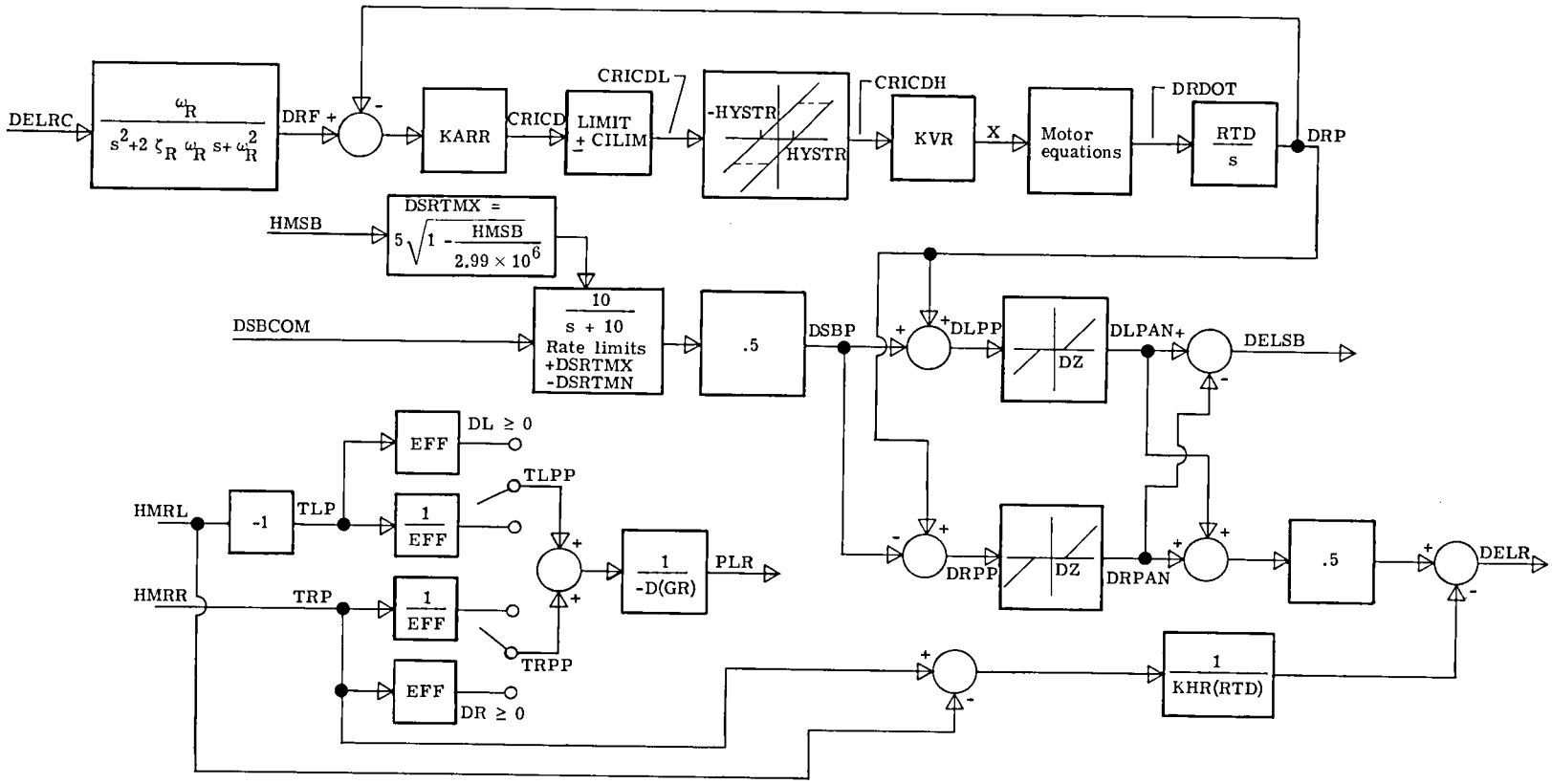


Figure F2.- Rudder and speed-brake actuator models.

APPENDIX G

CONVOLUTION INTEGRATION TECHNIQUE

Lawrence E. Barker, Jr. and Lawrence F. Rowell
Langley Research Center

The integration technique described is for the solution of linear differential equations with constant coefficients. The primary advantages of this technique are (1) the stability of the solution, (2) the high degree of accuracy for large system roots, (3) the variety of difference equations possible from this method of implementation, and (4) the ease in programing solutions to transfer functions when using the available subroutines. Appendix G is an expanded version of appendix D in reference 1. This version includes explicitly the constants required for second-order systems and has been rewritten for clarity in some places. Although this technique is general, its application to transfer functions is restrictive in that a forcing function is allowed only in the highest derivative.

Symbols

A	$n \times n$ matrix of system state coefficients
B	$n \times r$ matrix of forcing function coefficients
$\bar{b}$	nth-order vector of forcing function coefficients
C	constant gain matrix
D,E	coefficients of states when coefficient of $\ddot{x}$ is one (for second-order system)
e	base for natural logarithms
G	constant gain matrix
h	integration interval or step size, sec
I	identity matrix
K	constant gain matrix

k	number of discrete steps of size h taken
n	number of states in system
$P(h)$	constant matrix e^{Ah}
$p_{11}, p_{12}, p_{21}, p_{22}$	elements of matrix $P(h)$ for a second-order system
$Q(h)$	$n \times 3$ matrix of coefficients for three-term Taylor series
q_1, q_2, q_3	elements of $Q(h)$ for a first-order system
q_{11}, q_{12}, q_{13} q_{21}, q_{22}, q_{23}	elements of $Q(h)$ for a second-order system
$\bar{q}_i$	column vectors of $Q(h)$, where $i = 1, 2, 3$
r	number of forcing functions into system
r_1, r_2	roots of a second-order system
s	Laplace operator
t	time, sec
t_k	discrete sample time, $t = kh$ where $k = 0, 1, 2, \ldots$
$u(t)$	scalar forcing function input to a transfer function
$u(t_k)$	value of forcing function at sample time t_k
$\bar{u}(t)$	rth-order vector representing forcing functions
$\bar{u}^*(t_k)$	3×1 vector of terms resulting from Taylor series
W	single element of matrix A for a first-order system ($W = -A$)
$x(t)$	scalar solution state for a first-order system

$\bar{x}(t)$	nth-order vector representing system states of a dynamic system
$y(t)$	scalar output that results from $\bar{y}(t)$ for a single output transfer function
$\bar{y}(t)$	nth-order vector of multiple outputs of a transfer function
Z	Z-transform operator
γ	imaginary part of second-order roots
ζ	damping coefficient
σ	real part of second-order roots
τ	dummy integration variable
ω_n	natural frequency

A dot over a symbol indicates the derivative of the quantity with respect to time t. A double dot represents the second derivative with respect to time t.

Discussion of Convolution Approach

The behavior of a linear system can be described by a set of first-order differential equations of the form

$$\dot{\bar{x}}(t) = A\bar{x}(t) + B\bar{u}(t) \tag{G1}$$

where $\bar{x}(t)$ is the nth-order vector describing the system states, $\bar{u}(t)$ is an rth-order vector representing the forcing functions, A is an $n \times n$ matrix, B is an $n \times r$ matrix, and $\dot{\bar{x}}(t)$ is the time derivative of $\bar{x}(t)$. The well-known solution (ref. 8) to equation (G1) is

$$\bar{x}(t) = e^{At}\bar{x}(0) + \int_0^t e^{A(t-\tau)}B\bar{u}(\tau)\, d\tau \tag{G2}$$

where e^{At} is called the transition matrix and τ is a dummy integration variable. For digital (discrete) systems, the sample time is represented by t_k and equation (G2) becomes

$$\bar{x}\left(t_k + h\right) = P(h)\bar{x}\left(t_k\right) + \int_0^h P(h - \tau)B\bar{u}\left(t_k + \tau\right) d\tau \tag{G3}$$

where h is the integration step size and $P(h) = e^{Ah}$. Equation (G3) provides an exact solution; however, the mathematical expression shown as the integral term cannot be expressed in closed form and must be solved by some numeric process. The solution of equation (G3) can be obtained by several approaches. This appendix describes the convolution integration technique which is a numerical-analytical method derived by approximating the system forcing function $\bar{u}(t)$ with a Taylor series as it appears under the convolution integral in equation (G3). The approximation will be limited to three terms, but difference equations suitable for many applications can result from this approach. This technique is a special case of the local linearization described in reference 9.

Since systems defined in transfer function form will include a forcing function in only the highest-order derivative, $\bar{u}(t)$ can be treated as a scalar input $u(t)$, and B becomes

$$\bar{b} = \begin{bmatrix} 0 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \\ 1 \end{bmatrix}_{n\times 1}$$

Therefore, equation (G1) becomes

$$\dot{\bar{x}}(t) = A\bar{x}(t) + \bar{b}u(t) \tag{G4}$$

and equation (G3) becomes

$$\bar{x}\left(t_k + h\right) = P(h)\bar{x}\left(t_k\right) + \int_0^h P(h - \tau)\bar{b}u\left(t_k + \tau\right) d\tau \tag{G5}$$

By retaining three terms of the Taylor series representing $u\left(t_k + \tau\right)$, the resulting difference equation is

$$\bar{x}\left(t_k + h\right) = P(h)\bar{x}\left(t_k\right) + Q(h)\bar{u}^*\left(t_k\right) \tag{G6}$$

where

$$P(h) = e^{Ah}$$

$$Q(h) = \bar{q}_1(h), \bar{q}_2(h), \bar{q}_3(h)$$

and

$$\bar{u}^*\left(t_k\right) = \begin{bmatrix} u\left(t_k\right) \\ \dot{u}\left(t_k\right) \\ \ddot{u}\left(t_k\right) \end{bmatrix}$$

The vectors $\bar{q}_i$, where $i = 1, 2,$ and 3, are defined by

$$\bar{q}_i(h) = \frac{1}{(i-1)!}\int_0^h P(h-\tau)\tau^{i-1}\bar{b}\, d\tau \tag{G7}$$

The change in the state space vector, $\bar{x}(t)$, is the linear sum of $P(h)\bar{x}\left(t_k\right)$, $\bar{q}_1(h)u\left(t_k\right)$, $\bar{q}_2(h)\dot{u}\left(t_k\right)$, and $\bar{q}_3(h)\ddot{u}\left(t_k\right)$. The only approximation in equation (G6) (ref. 10) is the representation of the forcing function between sample points. Equation (G6) can be obtained also as a specific case from a more general algorithm derived by the method of local linearization applied to a nonlinear system of equations (ref. 9). The expressions for q_1, q_2, and q_3 that result from equation (G7) can be rewritten in matrix form to simplify the calculations as follows:

$$\bar{q}_1 = \int_0^h e^{A(h-\tau)}\bar{b}\, d\tau = A^{-1}\left[e^{Ah} - I\right]\bar{b} \tag{G8a}$$

$$\bar{q}_2 = \int_0^h e^{A(h-\tau)}\tau\bar{b}\, d\tau = A^{-1}\left[\bar{q}_1 - h\bar{b}\right] \tag{G8b}$$

$$\bar{q}_3 = \frac{1}{2}\int_0^h e^{A(h-\tau)}\tau^2\bar{b}\, d\tau = A^{-1}\left[\bar{q}_2 - \frac{h^2}{2}\bar{b}\right] \tag{G8c}$$

These equations are for a general system matrix A of any order.

APPENDIX G

First-Order Systems

Linear, first-order systems are described by

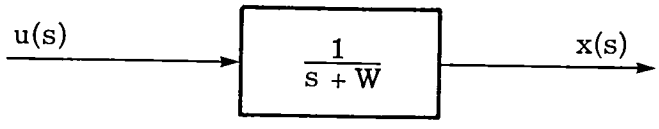
$$\dot{x}(t) = Ax(t) + u(t)$$

or

$$\dot{x}(t) - Ax(t) = u(t)$$

where $u(t)$ is the forcing function and $x(t)$ is the solution state. For first-order systems, $A = -W$ where W is the negative of the system root. The transfer function representation of the first-order system is as follows:

$$(s + W)x(s) = u(s)$$



From equations (G8) the convolution constants are

$$P(h) = e^{-Wh}$$

$$\bar{q}_1 = q_1 = (1 - P)/W$$

$$\bar{q}_2 = q_2 = \left(h - q_1\right)\Big/W$$

$$\bar{q}_3 = q_3 = \left(\frac{h^2}{2} - q_2\right)\Big/W$$

Thus,

$$x_1\left(t_k + h\right) = P(h)x_1\left(t_k\right) + q_1u\left(t_k\right) + q_2\dot{u}\left(t_k\right) + q_3\ddot{u}\left(t_k\right)$$

The derivative $\dot{x}(t)$ can be calculated as follows if velocity is required as an output or for limiting purposes:

$$\dot{x}\left(t_k + h\right) = u\left(t_k + h\right) - Wx_1\left(t_k + h\right)$$

APPENDIX G

Second-Order Systems

Linear, second-order systems are described by

$$\ddot{x}(t) + 2\zeta\omega_n\dot{x}(t) + \omega_n{}^2x(t) = u(t)$$

or

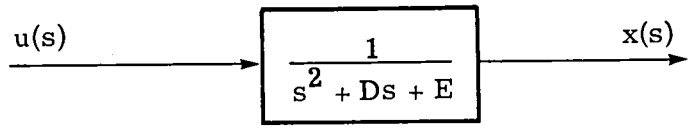
$$\ddot{x}(t) + D\dot{x}(t) + Ex(t) = u(t)$$

where ζ is the damping coefficient, ω_n is the natural frequency, $D = 2\zeta\omega_n$, and $E = \omega_n{}^2$. (D and E are the coefficients of the states when the coefficient of $\ddot{x}(t)$ is one.) For second-order systems,

$$A = \begin{bmatrix} 0 & 1 \\ -E & -D \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\omega_n{}^2 & -2\zeta\omega_n \end{bmatrix}$$

The transfer function representation of the second-order system is as follows:

$$(s^2 + Ds + E)\,x(s) = u(s)$$



The characteristics of the two roots r_1 and r_2 that result from the solution of a second-order system depend on ζ. In fact, three distinct cases can be described. From the quadratic formula,

$$r_1, r_2 = -\zeta\omega_n \pm \sqrt{\omega_n{}^2\left(\zeta^2 - 1\right)}$$

From equations (G8) the convolution constants may now be derived separately for each case as follows:

Real equal roots ($\zeta = 1$)

$$r_1 = r_2 = \sigma \qquad 2\zeta\omega_n = -2\sigma \qquad \omega_n{}^2 = \sigma^2$$

$$P(h) = e^{Ah} = \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix} = \begin{bmatrix} e^{\sigma h} - \sigma p_{12} & e^{\sigma h} h \\ -\sigma^2 p_{12} & e^{\sigma h} + \sigma p_{12} \end{bmatrix}$$

$$\bar{q}_1 = \begin{bmatrix} q_{11} \\ q_{21} \end{bmatrix} = \begin{bmatrix} -\frac{2\zeta}{\omega_n} p_{12} + \frac{1}{\omega_n^2}(1 - p_{22}) \\ p_{12} \end{bmatrix} = \begin{bmatrix} \frac{1}{\sigma^2}(2\sigma p_{12} + 1 - p_{22}) \\ p_{12} \end{bmatrix}$$

$$\bar{q}_2 = \begin{bmatrix} q_{12} \\ q_{22} \end{bmatrix} = \begin{bmatrix} \frac{h}{\omega_n^2} - \frac{2\zeta}{\omega_n} q_{11} - \frac{1}{\omega_n^2} q_{21} \\ q_{11} \end{bmatrix} = \begin{bmatrix} \frac{1}{\sigma^2}(h + 2\sigma q_{11} - q_{21}) \\ q_{11} \end{bmatrix}$$

$$\bar{q}_3 = \begin{bmatrix} q_{13} \\ q_{23} \end{bmatrix} = \begin{bmatrix} \frac{h^2}{2\omega_n^2} - \frac{2\zeta}{\omega_n} q_{12} - \frac{1}{\omega_n^2} q_{22} \\ q_{12} \end{bmatrix} = \begin{bmatrix} \frac{1}{\sigma^2}\left(\frac{h^2}{2} + 2\sigma q_{12} - q_{22}\right) \\ q_{12} \end{bmatrix}$$

The intermediate step in the preceding equations for $\bar{q}_i$ is of the most general form for all second-order cases. The remaining two cases are:

Real distinct roots ($\zeta > 1$)

$$r_1 = \sigma_1 \qquad r_2 = \sigma_2 \qquad 2\zeta\omega_n = -\sigma_1 - \sigma_2 \qquad \omega_n^2 = \sigma_1\sigma_2$$

$$P(h) = \begin{bmatrix} \dfrac{\left(\sigma_1 e^{\sigma_2 h} - \sigma_2 e^{\sigma_1 h}\right)}{\sigma_1 - \sigma_2} & \dfrac{e^{\sigma_1 h} - e^{\sigma_2 h}}{\sigma_1 - \sigma_2} \\ -\sigma_1\sigma_2 p_{12} & \dfrac{\left(\sigma_1 e^{\sigma_1 h} - \sigma_2 e^{\sigma_2 h}\right)}{\sigma_1 - \sigma_2} \end{bmatrix}$$

$$\bar{q}_1 = \begin{bmatrix} q_{11} \\ q_{21} \end{bmatrix} = \begin{bmatrix} \dfrac{1}{\sigma_1\sigma_2}\left[p_{12}\left(\sigma_1 + \sigma_2\right) + 1 - p_{22}\right] \\ p_{12} \end{bmatrix}$$

$$\bar{q}_2 = \begin{bmatrix} q_{12} \\ q_{22} \end{bmatrix} = \begin{bmatrix} \dfrac{1}{\sigma_1\sigma_2}\left[h + \left(\sigma_1 + \sigma_2\right) q_{11} - q_{21}\right] \\ q_{11} \end{bmatrix}$$

$$\bar{q}_3 = \begin{bmatrix} q_{13} \\ q_{23} \end{bmatrix} = \begin{bmatrix} \dfrac{1}{\sigma_1\sigma_2}\left[\dfrac{h^2}{2} + \left(\sigma_1 + \sigma_2\right) q_{12} - q_{22}\right] \\ q_{12} \end{bmatrix}$$

Complex conjugate roots ($\zeta < 1$)

$$r_1 = \sigma + \gamma i \qquad r_2 = \sigma - \gamma i \qquad 2\zeta\omega_n = -2\sigma \qquad \omega_n^2 = \sigma^2 + \gamma^2$$

$$P(h) = \begin{bmatrix} \left(-\sigma p_{12} + e^{\sigma h}\cos\gamma h\right) & \dfrac{e^{\sigma h}}{\gamma}\sin\gamma h \\ -\left(\sigma^2 + \gamma^2\right) p_{12} & \sigma p_{12} + e^{\sigma h}\cos\gamma h \end{bmatrix}$$

$$\bar{q}_1 = \begin{bmatrix} q_{11} \\ q_{21} \end{bmatrix} = \begin{bmatrix} \frac{1}{\sigma^2 + \gamma^2}\left(2\sigma p_{12} + 1 - p_{22}\right) \\ p_{12} \end{bmatrix}$$

$$\bar{q}_2 = \begin{bmatrix} q_{12} \\ q_{22} \end{bmatrix} = \begin{bmatrix} \frac{1}{\sigma^2 + \gamma^2}\left(h + 2\sigma q_{11} - q_{21}\right) \\ q_{11} \end{bmatrix}$$

$$\bar{q}_3 = \begin{bmatrix} q_{13} \\ q_{23} \end{bmatrix} = \begin{bmatrix} \frac{1}{\sigma^2 + \gamma^2}\left(\frac{h^2}{2} + 2\sigma q_{12} - q_{22}\right) \\ q_{12} \end{bmatrix}$$

For all three cases,

$$x_1\left(t_k + h\right) = p_{11}x_1\left(t_k\right) + p_{12}x_2\left(t_k\right) + q_{11}u\left(t_k\right) + q_{12}\dot{u}\left(t_k\right) + q_{13}\ddot{u}\left(t_k\right)$$

and

$$x_2\left(t_k + h\right) = p_{21}x_1\left(t_k\right) + p_{22}x_2\left(t_k\right) + q_{21}u\left(t_k\right) + q_{22}\dot{u}\left(t_k\right) + q_{23}\ddot{u}\left(t_k\right)$$

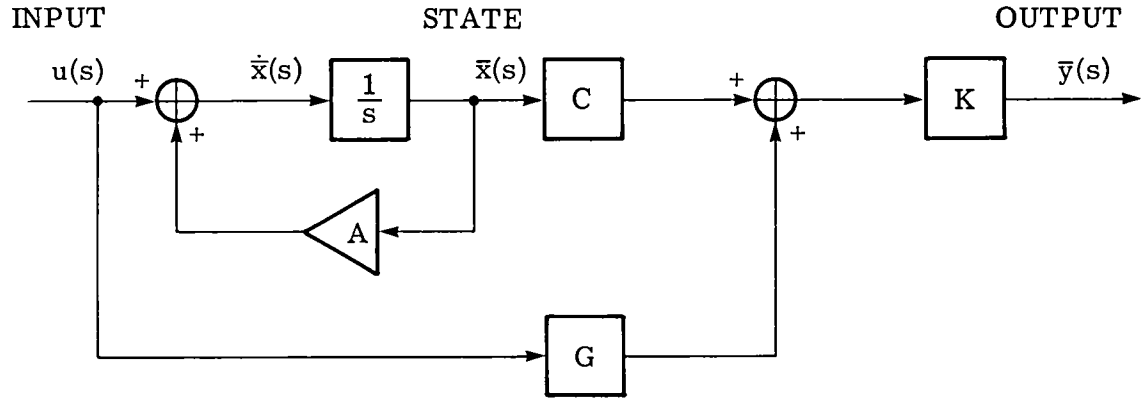
The derivative $\ddot{x}(t)$ can be calculated as follows if acceleration is required as an output:

$$\ddot{x}\left(t_k + h\right) = u\left(t_k + h\right) - Dx_2\left(t_k + h\right) - Ex_1\left(t_k + h\right)$$

Applications

The constants $P(h)$ and $Q(h)$ are functions of the system roots and the integration step size, and their analytic expressions have been programed into subroutines for first- and second-order systems. The constants, $P(h)$ and $Q(h)$, ensure correct placement of the continuous system roots, and their calculations need be performed only

once before the integration is begun. A representation of the state variable formulation for a single input, multiple output system is as follows:



Given the system described by equation (G4), the output equation is

$$\bar{y}(t) = K\left[C\bar{x}(t) + Gu(t)\right] \tag{G9}$$

where C, G, and K are gain matrices.

The programing concept described here differs from numerical integration in that information on the forcing function is supplied to the integration routine, and the solution from this routine is used in an algebraic "output equation." Numerical techniques require calculation of an algebraic derivative equation, and the result is passed to the integration routine to be used with past information in calculating the solution. When solving a transfer function, the result of integrating by the convolution technique is as though the corresponding transfer function had been solved for a numerator of 1. The output equation is used to account for the "numerator dynamics" and thus complete the solution.

Table GI and table GII give a reference to the output equations for some common first- and second-order transfer functions, respectively. The derivative equation shown following the state equations of each table is just one of the possible output equations.

By varying the method of computing $u(t)$, $\dot{u}(t)$, and $\ddot{u}(t)$ (that is, there is no restriction to use the values at time t_k), many different algorithms can be developed for digitally representing a filter transfer function when the filter specification is in analog form. The choice should depend on the application, the method of computing, or the system to be simulated.

For example, to model the first-order system $W/(s + W)$, three of the many possible algorithms that can be developed are as follows:

Case (a)

$$x(t_k + h) = e^{-Wh}x(t_k) + (1 - e^{-Wh})u(t_k) \qquad (u = u(t_k);\ \dot{u} = \ddot{u} = 0)$$

This difference equation corresponds to the Z-transform $(1 - e^{-Wh})/(Z - e^{-Wh})$.

Case (b)

$$x(t_k + h) = e^{-Wh}x(t_k) + (1 - e^{-Wh})u(t_k + h) \qquad (u = u(t_k + h);\ \dot{u} = \ddot{u} = 0)$$

This equation corresponds to the Z-transform $(1 - e^{-Wh})Z/(Z - e^{-Wh})$.

Case (c)

$$x(t_k + h) = e^{-Wh}x(t_k) + (1 - e^{-Wh})\frac{u(t_k) + u(t_k + h)}{2}$$

$$\left(u = \frac{u(t_k) + u(t_k + h)}{2};\ \dot{u} = \ddot{u} = 0\right)$$

This equation corresponds to the Z-transform $(1 - e^{-Wh})(Z + 1)/2(Z - e^{-Wh})$ where Z is the Z-transform operator.

For each of these cases (second-order cases are analogous), the roots of the continuous system are matched. The differences are in the reconstruction of the forcing function u(t) which is treated separately from the stability of the process. In these examples, $\dot{u}$ and $\ddot{u}$ were not used. In many applications, $\dot{u}$ is available within the program or can be approximated with a difference equation resulting in many more possible algorithms and improved accuracy.

There are other methods for deriving algorithms for modeling these transfer functions. There are substitution methods for the Laplace operator s (Tustin, for example, which is the equivalent to case (c) above with a Padé approximation made for e^{-Wh}). Also, a numerical integration scheme such as rectangular or Euler could be used. These methods give an approximation on the placement of the continuous system roots; but for $hW << 1$ (first-order case) or $h\sqrt{\sigma^2 + \gamma^2} << 1$ (second-order case), the results from all of these would be similar.

Equation (G6) has been programed for first- and second-order systems in subroutine form to make the use of the convolution technique quick and flexible. Thus, a programer can solve most linear transfer functions of first- and second-order blocks by

using subroutines to calculate the constants and then using the integration subroutines and output equations to calculate the solution of the system.

TABLE GI.- CONVERSION OF TRANSFER FUNCTIONS TO OUTPUT EQUATIONS FOR FIRST-ORDER SYSTEMS

$\left[\text{Forcing function is } u\left(t_k\right)\right]$

State equation:

$$x = x_1\left(t_k + h\right) = P(h)x\left(t_k\right) + q_1(h)u\left(t_k\right) + q_2(h)\dot{u}\left(t_k\right) + q_3(h)\ddot{u}\left(t_k\right)$$

First derivative:

$$\dot{x} = u\left(t_k + h\right) - Wx_1\left(t_k + h\right)$$

Transfer function	Output equation
$\frac{1}{s + W}$	$y\left(t_k + h\right) = x_1\left(t_k + h\right)$
$\frac{K}{s + W}$	$y\left(t_k + h\right) = Kx_1\left(t_k + h\right)$
$\frac{Ks}{s + W}$	$y\left(t_k + h\right) = Kx_2\left(t_k + h\right) = K\left[u\left(t_k + h\right) - Wx_1\left(t_k + h\right)\right]$
$\frac{K(As + B)}{s + W}$	$y\left(t_k + h\right) = KAx_2\left(t_k + h\right) + KBx_1\left(t_k + h\right) = K\left[(B - AW)x_1\left(t_k + h\right) + Au\left(t_k + h\right)\right]$

TABLE GII.- CONVERSION OF TRANSFER FUNCTIONS TO OUTPUT EQUATIONS FOR SECOND-ORDER SYSTEMS

State equations:

$$x = x_1(t_k + h) = p_{11}x_1(t_k) + p_{12}x_2(t_k) + q_{11}u(t_k) + q_{12}\dot{u}(t_k) + q_{13}\ddot{u}(t_k)$$

$$\dot{x} = x_2(t_k + h) = p_{21}x_1(t_k) + p_{22}x_2(t_k) + q_{21}u(t_k) + q_{22}\dot{u}(t_k) + q_{23}\ddot{u}(t_k)$$

Second derivative:

$$\ddot{x} = u(t_k + h) - Dx_2(t_k + h) - Ex_1(t_k + h)$$

Transfer function	Output equation
$\frac{1}{s^2 + Ds + E}$	$y(t_k + h) = x_1(t_k + h)$
$\frac{K}{s^2 + Ds + E}$	$y(t_k + h) = Kx_1(t_k + h)$
$\frac{Ks}{s^2 + Ds + E}$	$y(t_k + h) = Kx_2(t_k + h)$
$\frac{Ks^2}{s^2 + Ds + E}$	$y(t_k + h) = Kx_3(t_k + h)$ $= K\left[u(t_k + h) - Dx_2(t_k + h) - Ex_1(t_k + h)\right]$
$\frac{K(s + A)}{s^2 + Ds + E}$	$y(t_k + h) = K\left[Ax_1(t_k + h) + x_2(t_k + h)\right]$
$\frac{K(As^2 + Bs + C)}{s^2 + Ds + E}$	$y(t_k + h) = K\left[Ax_3(t_k + h) + Bx_2(t_k + h) + Cx_1(t_k + h)\right]$ $= K\left[Au(t_k + h) + (B - AD)x_2(t_k + h)\right.$ $\left. + (C - AE)x_1(t_k + h)\right]$

REFERENCES

1. Kaylor, Jack T.; Rowell, Lawrence F.; and Powell, Richard W.: A Real-Time Digital Computer Program for the Simulation of Automatic Spacecraft Reentries. NASA TM X-3496, 1977.

2. Space Shuttle. NASA SP-407, 1976.

3. Carrier, L. M.; and Minor, R. G.: Space Shuttle Orbiter Avionics. A Collection of Technical Papers – AIAA 2nd Digital Avionics Systems Conference, Nov. 1977, pp. 146-156. (Available as AIAA Paper 77-1501.)

4. Brand, Vance D.: Return to Earth in the Space Shuttle. 1977 Report to the Aerospace Profession, Tech. Rev., vol. 13, no. 4, Soc. Exp. Test Pilots, c.1978, pp. 223-231.

5. Cleveland, Jeff I., II; Crawford, Daniel J.; and Rowell, Lawrence F.: Reference Manual for the Langley Research Center Flight Simulation Computing System. NASA TM-78757, 1978.

6. Powell, Richard W.; and Stone, Howard W.: Analysis of Space Shuttle Orbiter Entry Dynamics From Mach 10 to Mach 2.5 With the November 1976 Flight Control System. NASA TP-1667, 1980.

7. Aerodynamic Design Data Book. Volume I: Orbiter Vehicle. NASA CR-160386, 1978.

8. Gupta, Someshwar C.; and Hasdorff, Lawrence: Fundamentals of Automatic Control. John Wiley & Sons, Inc., c.1970.

9. Barker, Lawrence E., Jr.; Bowles, Roland L.; and Williams, Louise H.: Development and Application of a Local Linearization Algorithm for the Integration of Quaternion Rate Equations in Real-Time Flight Simulation Problems. NASA TN D-7347, 1973.

10. Giese, Clarence: State Variable Difference Methods for Digital Simulation. Simulation, vol. 8, no. 5, May 1967, pp. 263-271.

1. Report No. NASA TP-1700	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle DEVELOPMENT OF THE REENTRY FLIGHT DYNAMICS SIMULATOR FOR EVALUATION OF SPACE SHUTTLE ORBITER ENTRY SYSTEMS		5. Report Date October 1980 6. Performing Organization Code
7. Author(s) Lawrence F. Rowell, Richard W. Powell, and Howard W. Stone, Jr.		8. Performing Organization Report No. L-13662
9. Performing Organization Name and Address NASA Langley Research Center Hampton, VA 23665		10. Work Unit No. 506-63-13-01 11. Contract or Grant No.
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, DC 20546		13. Type of Report and Period Covered Technical Paper 14. Sponsoring Agency Code

15. Supplementary Notes

Appendix G by Lawrence E. Barker, Jr., and Lawrence F. Rowell

16. Abstract

The reentry flight dynamics simulator (RFDS) developed by the NASA Langley Research Center is a nonlinear, six-degree-of-freedom, digital-computer simulation of a vehicle which has constant mass properties and whose attitudes are controlled by both aerodynamic surfaces and reaction control system thrusters. A rotating, oblate Earth model was used to describe the gravitational forces which affect long-duration Earth entry trajectories. This program can be executed in a nonreal-time mode or connected to a simulation cockpit to conduct piloted and autopilot studies. The RFDS is being used to evaluate the onboard guidance and control software (November 1976 version) used by the Space Shuttle Orbiter for its descent from approximately 121.9 km to touchdown on the runway.

17. Key Words (Suggested by Author(s))	18. Distribution Statement		
Space Shuttle Orbiter Digital simulation models Entry guidance and control	Unclassified – Unlimited Subject Category 18		
19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 95	22. Price A05

For sale by the National Technical Information Service, Springfield, Virginia 22161

NASA-Langley, 1980