2mit

CR-134001

E-2736

# INTEGRATED DIGITAL FLIGHT CONTROL SYSTEM FOR THE SPACE SHUTTLE ORBITER

Revision 1

July 1973

(NASA-CR-134001) INTEGRATED DIGITAL FLIGHT CONTROL SYSTEM FOR THE SPACE SHUTTLE ORBITER (Draper (Charles Stark) Lab., Inc.) 346 p HC $19.50 CSCL 22B

N73-30835

Unclas

G3/31 12473

Reproduced by
NATIONAL TECHNICAL INFORMATION SERVICE
US Department of Commerce
Springfield, VA. 22151

SEP 1973
RECEIVED
NASA STI FACILITY
INPUT BRANCH

**The Charles Stark Draper Laboratory, Inc.**
Cambridge, Massachusetts 02139

346

N O T I C E

THIS DOCUMENT HAS BEEN REPRODUCED FROM THE BEST COPY FURNISHED US BY THE SPONSORING AGENCY. ALTHOUGH IT IS RECOGNIZED THAT CERTAIN PORTIONS ARE ILLEGIBLE, IT IS BEING RELEASED IN THE INTEREST OF MAKING AVAILABLE AS MUCH INFORMATION AS POSSIBLE.

E-2736

# INTEGRATED DIGITAL FLIGHT CONTROL SYSTEM FOR THE SPACE SHUTTLE ORBITER

Revision 1

July 1973

THE CHARLES STARK DRAPER LABORATORY, INC.
CAMBRIDGE, MASSACHUSETTS 02139

APPROVED: Donald C Fraser Date: 13 July 1973
D. C. FRASER

APPROVED: David G. Hoag Date 13 Jul 73
D. G. HOAG

ACKNOWLEDGMENT

This report was prepared under DSR Project 55-47300, sponsored by the Johnson Spacecraft Center of the National Aeronautics and Space Administration through Contract NAS9-11764.

The individual authors are cited at the beginning of their respective sections. Technical coordination for the report was provided by Peter Weissman. Liaison between the Guidance and Control Systems Branch and the Draper Laboratory was carried out by Edwin Olsson.

The publication of this report does not constitute approval by the National Aeronautics and Space Administration of the findings therein. It is published only for the exchange and stimulation of ideas.

## ABSTRACT

The objective of the integrated Digital Flight Control System is to provide rotational and translational control of the Space Shuttle orbiter in all phases of flight: from launch ascent through orbit to entry and touchdown, and during powered horizontal flights. The program provides a versatile control system structure while maintaining uniform communications with other programs, sensors, and control effectors by using an executive routine/functional subroutine format. The program reads all external variables at a single point, copies them into its dedicated storage, and then calls the required subroutines in the proper sequence. As a result, the flight control program is largely independent of other programs in the GN&C computer complex and is equally insensitive to the characteristics of the processor configuration.

Section 1 of Part 1 describes the integrated structure of the control system and the DFCS executive routine which embodies that structure. Section 2 deals with the input and output including jet selection. The succeeding sections show specific estimation and control algorithms for the various mission phases: cruise (including horizontal powered flight), entry, on-orbit, and boost.

Part 2 describes attitude maneuver routines that interface with the DFCS.

# FOREWORD

This document presents the Space Shuttle digital flight control design base developed by the Draper Laboratory Control and Flight Dynamics Division and the JSC Guidance and Control Systems Branch. The flight control and attitude control of the Shuttle from launch through touchdown and for powered horizontal flight is addressed. The subject matter reflects different levels of design maturity and completeness of documentation for the several different mission phases. It is anticipated that this report will be utilized to establish an initial baseline for the digital flight control software design. It is essentially the same as the NASA document Digital Flight Control Software Design Requirements, Rev. A, JSC-07759, July 6, 1973.

This revision reflects more than just the additional design activities performed in the past half-year by the groups mentioned above. Honeywell Aerospace Division completed an extensive review of the original version, and some changes resulted from the constructive Honeywell review comments to NASA. Additionally, treatment of control modes in this revision reflects information exchanges with Rockwell and the Flight Crew personnel.

There are numerous important design areas which have not been covered. It is anticipated that they will be addressed as baseline additions at an appropriate time. These areas include the treatment of redundant sensors and redundant actuators. Interface with pilot's displays has not been considered in this revision. Manual modes have been included only for the Cruise phase. All design to date has assumed rigid bodies; the filter structure, however, has been so specified as to facilitate the handling of bending and slosh when details are available. The Roll-out control phase has not been addressed.

In this document, a number of areas have been addressed in considerable detail. The following paragraphs summarize the scope of the document and the design activity which it reports.

The unified structure which is presented for the Digital Flight Control System (DFCS) is a useful, flexible framework for the evolving Space Shuttle control system. The implementation of input-output processing, on the other hand, can be

PRECEDING PAGE BLANK NOT FILMED

expected to change as the Shuttle avionics becomes better defined.

Two designs for jet selection are included. The first, which is based upon linear programming, is flexible and powerful. The second, which is based upon a table look-up scheme, is more restricted. The latter will be considered the baseline design until the Shuttle requirements are found to go beyond its capabilities.

The design for the Cruise phase addresses the requirement for a variety of manual, semi-automatic and automatic control modes during the developmental horizontal flight tests and the ferry flights as well as during the final stages of operational vertical flights. The control gains have been calculated for the Rockwell 0089B vehicle, but only for a single flight condition; the scheduling of gains for a full range of conditions is currently in progress.

A number of features have been added to the design for the Entry phase since this document was originally issued. The applicable flight regime has been extended to subsonic, thus eliminating the separate Transition phase. A state estimator has been introduced. Rudder augmentation has been added to the control during the early part of entry. The gains have been calculated for the Rockwell 0089B vehicle. A capability for higher maneuver rates in response to large bank commands has been added. The elevator-trim algorithm during early entry has been changed from open-loop to closed-loop.

The design for control using the reaction control jets during the On-orbit mission phase has been developed and implemented, as has been the design for the attitude maneuver routine during the phase. The design does not reflect the availability of vernier jets or the unavailability of a clock for on-off commands other than once per minor cycle; these factors must be considered in redesigning firing policies. The design for control during major maneuvers in orbit (including Insertion) is in development; only a functional outline is presented in this document for that segment of the DFCS.

The control for the Boost phase, which was developed outside the DFCS framework, is now incorporated into it.

All the designs presented in this document have been set up for the Rockwell International 0089B Space Shuttle orbiter except that the ACPS thruster configuration is assumed to be that included in the Rockwell International ATP orbiter.

There is some lack of commonality and uniformity in the material presented in this document as a result of various design efforts not yet being fully integrated. For example, the phase plane developed for on-orbit ACPS control may well be usable in the Entry phase as well. Such areas need to be addressed.

A version of the DFCS which incorporates the designs for the Cruise, Entry and Boost phases has been implemented in FORTRAN for testing on the SSFS (Space Shuttle Functional Simulator). A second version has been programmed in HAL using structured programming techniques. It incorporates the designs for the On-orbit phase, and runs on the engineering simulator (FSIM) which is being developed at the C.S. Draper Laboratory for the testing of flight programs that are written in HAL.

TABLE OF CONTENTS

PART 1


| Section | | Page |
|---|---|---|
| 1 | INTEGRATED STRUCTURE; EXECUTIVE ROUTINE | |
| | 1.1 Integrated Structure | 1-1 |
| | 1.2 DFCS Executive Routine | 1-2 |
| 2 | DFCS INPUT AND OUTPUT | |
| | 2.1 Inputs Required by the DFCS | 2-1 |
| | 2.2 Outputs to Control Effectors | 2-12 |
| | 2.3 ACPS Jet Selection Subroutine | 2-16 |
| | 2.4 ACPS Jet Selection Logic for the ATP Configuration | 2-30 |
| 3 | STATE ESTIMATOR (ENTRY PHASE) | |
| | 3.1 Introduction | 3-1 |
| | 3.2 General Description | 3-1 |
| | 3.3 Implementation | 3-3 |
| | 3.4 Preliminary Performance Summary | 3-17 |
| 4 | CRUISE PHASE | |
| | 4.1 Introduction | 4-1 |
| | 4.2 Control Modes | 4-2 |
| | 4.3 Digital Filters and Their Implementation | 4-12 |
| | 4.4 Preflight Parameter Specifications | 4-19 |
| | 4.5 Input/Output Summaries | 4-19 |
| | 4.6 Functional Block Diagrams for Cruise Phase DFCS | 4-29 |
| | 4.7 Functional Flow Charts - Cruise Phase Initialization | 4-46 |
| | Appendix: Detailed Flow Charts for Cruise Phase DFCS | 4-52 |
| 5 | ENTRY PHASE | |
| | 5.1 Introduction | 5-1 |
| | 5.2 Overview | 5-1 |
| | 5.3 Description of Control Laws | 5-3 |
| | 5.4 Performance Summary | 5-31 |




PRECEDING PAGE BLANK NOT FILMED

TABLE OF CONTENTS (Cont)


| Section | PART 1 (Cont) | Page |
|---|---|---|
| 6 | ON-ORBIT PHASE | |
| | 6.1 Introduction | 6-1 |
| | 6.2 Engineering Description | 6-1 |
| | 6.3 On-Orbit TVC Phase | 6-24 |
| | 6.4 Digital Implementation | 6-27 |
| | References | 6-48 |
| 7 | BOOST PHASE | |
| | 7.1 Introduction | 7-1 |
| | 7.2 Functional Description | 7-1 |
| | 7.3 Logic Flow Description | 7-6 |
| | 7.4 Values of Control Gains | 7-6 |
| | References | 7-13 |
| 8 | GLOSSARY FOR DFCS | 8-1 |

PART 2

| | | |
|---|---|---|
| 1 | AUTOMATIC TRACKING | |
| | 1.1 Tracking Options | 1-1 |
| | 1.2 Option 0: Orient the Spacecraft Such that a Specified Body-Fixed Pointing Vector Aligns with the Estimated Line of Sight to the TV | 1-2 |
| | 1.3 Option 1: Orient the Spacecraft Such that SCAXIS Aligns with the LOS to a Specified Celestial Body | 1-4 |
| | 1.4 Option 2: Rotate the Spacecraft About SCAXIS | 1-4 |
| | 1.5 Option 4: Orient the Spacecraft as in Option 0, but Further Specify an Angle (OMICRON) of Rotation about SCAXIS | 1-5 |
| | 1.6 Option 5: Orient the Spacecraft as in Option 1, but Further Specify an Angle (OMICRON) of Rotation about SCAXIS | 1-7 |
| 2 | ON-ORBIT ATTITUDE MANEUVER ROUTINE | |
| | 2.1 Introduction | 2-1 |
| | 2.2 ATTITUDE_MANEUVER | 2-3 |
| | 2.3 VECPOINT | 2-3 |
| | 2.4 KALCMANU Maneuver - Parameters Calculation | 2-12 |
| | 2.5 Steering Through a Maneuver | 2-15 |
| | 2.6 STEERING | 2-17 |
| | References | 2-30 |

LIST OF FIGURES

PART 1


| Figure | | Page |
|---|---|---|
| 1-1 | Logic Flow of DFCS Executive | 1-3 |
| 1-2 | Detailed Logic Flow of DFCS Executive | 1-4 |
| 2-1 | Input Interface Routine Logic Flow | 2-9 |
| 2-2 | Computations in Air Data Routine | 2-13 |
| 2-3 | Sign Conventions for Aerodynamic Control Surfaces and Manual Controllers | 2-15 |
| 2-4 | Thruster Locations - NR ATP Baseline Orbiter | 2-17 |
| 2-5 | ACPS Jet Selection Functional Flow Diagram | 2-20 |
| 2-6a | Features of Linear Programming Subroutine, CCWLJS | 2-22 |
| 2-6b | Flow Chart of Linear Programming Subroutine, CCWLJS | 2-24 |
| 2-7a | Flow Chart Jet Selection Logic for ATP Configuration | 2-32 |
| 2-7b | Definition of Symbols for Jet Selection Logic | 2-54 |
| 3-1 | Measurement Incorporation for State Estimation | 3-5 |
| 3-2 | Computation of Estimator Gain Matrices | 3-6 |
| 3-3 | State Extrapolation for the Entry State Estimator | 3-9 |
| 3-4 | Computation of Estimator State Transition Matrices | 3-10 |
| 3-5a | $C_{\ell_{\delta_a}}'$ vs Mach Number | 3-13 |
| 3-5b | $Cn_{\delta_a}'$ vs Mach Number | 3-14 |
| 3-5c | $C_{\ell_\beta}'$ vs Mach Number | 3-15 |
| 3-5d | $C_{n_\beta}'$ vs Mach Number | 3-16 |
| 3-6a | Longitudinal and Lateral Responses with Noise Sources Simulated and FILTER = 0 (no estimator in loop) | 3-18 |
| 3-6b | Longitudinal and Lateral Responses with Noise Sources Simulated and FILTER = 1 (estimator in loop) | 3-19 |
| 4-1 | Simplified Functional Block Diagram of the Stability Augmentation System (also Backup Control System) Mode | 4-3 |
| 4-2 | Simplified Functional Block Diagram of Command Augmentation System Mode | 4-4 |
| 4-3 | Simplified Functional Block Diagram of Outer Loops for Semiautomatic Modes of Operation | 4-6 |

LIST OF FIGURES (Cont)

PART 1 (Cont)


| Figure | | Page |
|---|---|---|
| 4-4 | Simplified Functional Block Diagram of the Automatic Mode | 4-7 |
| 4-5 | Simplified Functional Block Diagrams - Speed Control | 4-9 |
| 4-6 | Functional Block Diagram of Body Flap | 4-11 |
| 4-7a | Mode-Select/Indicate Panels, Type 1, Cruise Phase DFCS | 4-13 |
| 4-7b | Mode-Select/Indicate Panels, Type 2 (Axis-by-Axis), Cruise Phase DFCS | 4-14 |
| 4-8 | $N^{th}$-Order Digital Filter | 4-15 |
| 4-9 | Digital Washout Filter | 4-17 |
| 4-10 | Digital Low Pass Filter | 4-18 |
| 4-11 | Digital Integrator Filters | 4-18 |
| 4-12 | Parameter Estimation Routine for Cruise Phase | 4-23 |
| 4-13a | Speed Control Input/Output Interface Variables | 4-24 |
| 4-13b | Elevator and Flap Input/Output Interface Variables | 4-25 |
| 4-13c | Aileron Input/Output Interface Variables | 4-26 |
| 4-13d | Rudder Input/Output Interface Variables | 4-27 |
| 4-13e | Backdrop Control System Input/Output Interface Variables | 4-28 |
| 4-14 | Simplifying Conventions for Block Diagrams | 4-30 |
| 4-15 | Block Diagram, Pilot Input Signal Processing | 4-34 |
| 4-16 | Block Diagram, Pilot Manual Trim Circuits | 4-35 |
| 4-17 | Block Diagram, Outer Loop Error Signal Generation | 4-36 |
| 4-18a | Speed Control Loops | 4-37 |
| 4-18b | TAEM Guidance Processing for Speed Control | 4-38 |
| 4-19 | Elevator Control Loops | 4-39 |
| 4-20 | Aileron Control Loops | 4-41 |
| 4-21 | Rudder Control Loops | 4-42 |
| 4-22 | Control Output Signals | 4-43 |
| 4-23 | Backup Control System | 4-44 |
| 4-24 | Functional Flow Diagram, Load Manipulation Routine | 4-47 |
| 4-25 | Functional Flow Diagram, Speed Initialization Routine | 4-48 |
| 4-26 | Functional Flow Diagram, Elevator Initialization Routine | 4-49 |
| 4-27 | Functional Flow Diagram, Aileron Initialization Routine | 4-50 |
| 4-28 | Functional Flow Diagram, Rudder Initialization | 4-51 |
| 4A-1 | Load Manipulator Routine Cruise Section | 4-53 |
| 4A-2 | Overview of Cruise Section Fortran Flow Charts | 4-54 |
| 4A-3 | Initialization Subroutine, Cruise Phase Section, Speed Subsection | 4-55 |
| 4A-4a | Initialization Subroutine, Cruise Phase Section, Elevator Subsection | 4-56 |
| 4A-4b | Initialization Subroutine, Cruise Phase Section, Elevator Trim Subsection | 4-57 |

LIST OF FIGURES (Cont)

PART 1 (Cont)


| Figure | | Page |
|---|---|---|
| 4A-5a | Initialization Subroutine, Cruise Phase Section, Aileron Subsection | 4-58 |
| 4A-5b | Initialization Subroutine, Cruise Phase Section, Aileron Trim Subsection | 4-59 |
| 4A-6a | Initialization Subroutine, Cruise Phase Section, Rudder Subsection | 4-60 |
| 4A-6b | Initialization Subroutine, Cruise Phase Section, Rudder Trim Subsection | 4-61 |
| 4A-7a | Filter Part I Subroutine, Cruise Phase Section Speed Subsection | 4-62 |
| 4A-7b | Filter Part I Subroutine, Cruise Phase Section Speed Subsection, (TAEM Guidance Routine) | 4-63 |
| 4A-8a | Filter Part I Subroutine, Cruise Phase Section, Elevator Subsection | 4-64 |
| 4A-8b | Filter Part I Subroutine, Cruise Phase, Elevator Subsection, Outer Loops | 4-65 |
| 4A-8c | Filter Part I Subroutine, Cruise Phase, Elevator Subsection, Feedback | 4-66 |
| 4A-9a | Filter Part I Subroutine, Cruise Phase Section, Aileron Subsection | 4-67 |
| 4A-9b | Filter Part I Subroutine, Cruise Phase Section, Aileron Subsection, Outer Loops | 4-68 |
| 4A-10 | Filter Part I Subroutine, Cruise Phase Section, Rudder Subsection | 4-69 |
| 4A-11 | Control Law Subroutine, Cruise Phase Section, Speed Subsection | 4-70 |
| 4A-12 | Control Law Subroutine, Cruise Phase Section, Elevator Subsection | 4-71 |
| 4A-13 | Control Law Subroutine, Cruise Phase Section, Aileron and Rudder Subsections | 4-72 |
| 4A-14 | Filter Part II Subroutine, Cruise Phase Section, Speed Subsection | 4-73 |
| 4A-15 | Filter Part II Subroutine, Cruise Phase Section, Elevator Subsection | 4-74 |
| 4A-16 | Filter Part II Subroutine, Cruise Phase Section, Aileron Subsection | 4-75 |
| 4A-17 | Filter Part II Subroutine, Cruise Phase Section, Rudder Subsection | 4-76 |
| 4A-18 | Filter Part II Subroutine, Cruise Phase Section, Gear/Integrating Moding | 4-77 |
| 4A-19a | Cruise Phase BCS Subroutine, Speed, Elevator, and Flap Subsection | 4-78 |
| 4A-19b | Cruise Phase BCS Subroutine, Aileron and Rudder Subsection | 4-79 |
| 4A-20 | Cruise Phase BCS Subroutine, Wrap up Subsection | 4-80 |

LIST OF FIGURES (Cont)

PART 1 (Cont)


| Figure | | Page |
|---|---|---|
| 5-1 | Overview of Entry Control | 5-2 |
| 5-2 | Controller Activity during Entry (Longitudinal) | 5-4 |
| 5-3a | Elevator Control Block Diagram | 5-6 |
| 5-3b | Elevator Deflection Limit Parameter | 5-6 |
| 5-4 | ACPS Switch Logic | 5-8 |
| 5-5 | Pitch ACPS Control Block Diagram | 5-8 |
| 5-6a | Logic Flow for the Longitudinal Control Channel | 5-10 |
| 5-6b | Logic Flow for Elevator Control | 5-11 |
| 5-6c | Logic Flow for Pitch ACPS Control | 5-13 |
| 5-7 | Controller Activity during Entry (Lateral) | 5-14 |
| 5-8 | Overview of Aileron/Rudder Control Logic | 5-17 |
| 5-9 | Control Submode Switch Parameter Definition | 5-17 |
| 5-10a | Aileron/Rudder Control Block Diagram (Reverse Mode $I_{Lat} = 0$) | 5-18 |
| 5-10b | Aileron/Deflection Limit Parameter | 5-19 |
| 5-11 | Aileron/Rudder Control Block Diagram (Proverse Mode, $I_{Lat} = 1$) | 5-21 |
| 5-12 | Roll Jet Control when $u_z \neq 0$ | 5-23 |
| 5-13 | Roll Jet Control when $u_z = 0$ ($\bar{q} \geq \bar{q}_3$) | 5-24 |
| 5-14 | Yaw and Roll ACPS Control Block Diagram | 5-25 |
| 5-15a | Logic Flow for the Lateral Channel | 5-26 |
| 5-15b | Logic Flow for Aileron/Rudder Control | 5-27 |
| 5-15c | Logic Flow for Yaw/Roll ACPS Control | 5-29 |
| 5-16 | Gain Schedule for $\xi_1$ | 5-35 |
| 5-17 | Gain Schedule for $\xi_2$ | 5-35 |
| 5-18 | Gain Schedule for $\xi_3$ | 5-36 |
| 5-19 | Gain Schedule for $\xi_4$ | 5-36 |
| 5-20 | Gain Schedule for $\xi_5$ | 5-37 |
| 5-21 | Gain Schedule for $\xi_6$ | 5-37 |
| 5-22 | Gain Schedule for $\nu_1$ | 5-38 |
| 5-23 | Gain Schedule for $\nu_2$ | 5-38 |
| 5-24 | Gain Schedule for $\gamma_{11}$ | 5-39 |
| 5-25 | Gain Schedule for $\gamma_{12}$ | 5-39 |
| 5-26 | Gain Schedule for $\gamma_{21}$ | 5-40 |
| 5-27 | Gain Schedule for $\gamma_{22}$ | 5-40 |
| 5-28 | Entry Load Manipulation | 5-41 |
| 5-29 | Controllability Regimes | 5-42 |

LIST OF FIGURES (Cont)

PART 1 (Cont)


| Figure | | Page |
|---|---|---|
| 6-1 | On-Orbit ACPS | 6-2 |
| 6-2 | u, v, w Control Axes | 6-5 |
| 6-3 | ACPS Jet Control Acceleration Model | 6-8 |
| 6-4 | Idealized Limit Cycle for a Positive Disturbing Acceleration | 6-14 |
| 6-5 | Limit Cycle in Presence of No Disturbing Accelerations | 6-17 |
| 6-6 | SMALL_ERROR_DISTURBANCE_LOGIC | 6-18 |
| 6-7 | Large Error Control Region | 6-22 |
| 6-8a | Functional Block Diagram of the TVC DAP | 6-25 |
| 6-8b | Block Diagram of the Thrust Misalignment Correction (TMC) Loop | 6-26 |
| 6-9 | Logic Flow of DFCS | 6-28 |
| 6-10 | MASS_CALC: Procedure | 6-29 |
| 6-11 | ORBIT_ACPS_FILTER1: Label | 6-31 |
| 6-12 | CONTROL: Procedure | 6-32 |
| 6-13 | ERROR: Procedure | 6-33 |
| 6-14a | PHASE_PLANE: FUNCTION ($\theta_e'$, $\omega_e'$, $\lvert U_C \rvert$, $U_D$, $X_{D\,old}$) | 6-34 |
| 6-14b | LARGE_ERROR_CONTROL_LOGIC: Procedure | 6-35 |
| 6-14c | SMALL_ERROR_CONTROL_LOGIC: Procedure | 6-36 |
| 6-14d | REGION_II_TARGETTING | 6-37 |
| 6-14e | BOUNDARY: FUNCTION (B_SIGN, ω, D, UC, UD, CD) | 6-37 |
| 6-15a | JET_CONTROL | 6-38 |
| 6-15b | UVWTIME | 6-39 |
| 6-15c | OFF1UJETS | 6-40 |
| 6-15d | OFF2UJETS: Procedure | 6-40 |
| 6-15e | OFF1VJETS | 6-41 |
| 6-15f | OFF2VJETS: Procedure | 6-41 |
| 6-15g | OFF1WJETS: Procedure | 6-42 |
| 6-15h | OFF2WJETS: Procedure | 6-42 |
| 6-16 | ORBIT_ACPS_FILTER2: Label | 6-43 |
| 7-1 | Baseline Launch Configuration Control System Software Roll Channel | 7-3 |
| 7-2 | Baseline Launch Configuration Control System Yaw Channel | 7-5 |
| 7-3 | Baseline Launch Configuration Control System Software Pitch Channel | 7-7 |
| 7-4 | Logic Flow | 7-8 |

LIST OF FIGURES (Cont)

PART 2


| Figure | | Page |
|---|---|---|
| 2-1 | ATTITUDE_MANEUVER: Structure | 2-2 |
| 2-2 | ATTITUDE_MANEUVER: Program | 2-4 |
| 2-3 | ANGLES_TO_DCM: Procedure ([$\theta$]) Assign ($\overset{*}{C}$) | 2-7 |
| 2-4 | DCM_TO_ANGLES: Procedure ($\overset{*}{C}$) Assign ([$\theta$]) | 2-8 |
| 2-5 | ROTATION_MATRIX | 2-10 |
| 2-6 | The Coordinate Frames and Transformations used in VECPOINT | 2-10 |
| 2-7 | VECPOINT: Procedure | 2-11 |
| 2-8 | KALCMANU: Procedure | 2-14 |
| 2-9 | Maneuver Phase-Plane Trajectories | 2-18 |
| 2-10 | STEERING: Procedure | 2-20 |
| 2-11 | STOPRATE: Procedure | 2-21 |
| 2-12 | HOLD_ATTITUDE: Procedure | 2-23 |

LIST OF TABLES

PART 1


| Table | | Page |
|---|---|---|
| 1-1 | DFCS Mission Phases | 1-6 |
| 1-2 | Inputs to DFCS Executive Routine | 1-7 |
| 1-3 | Outputs of DFCS Executive Routine | 1-7 |
| 2-1 | Inputs to the DFCS | 2-2 |
| 2-2 | Inputs to the Input Interface Routine | 2-5 |
| 2-3 | Outputs from Input Interface Routine | 2-7 |
| 2-4 | Discretes Governing Logic Flow in Input Interface Routine | 2-8 |
| 2-5 | Control Effectors Used by the DFCS | 2-14 |
| 2-6 | DFCS Limits for Commands to Control Effectors | 2-14 |
| 2-7 | Distribution of ACPS Thrusters NR ATP Baseline Orbiter | 2-18 |
| 2-8 | Parameters of Individual Jets | 2-19 |
| 3-1 | Longitudinal Estimator Gains Scheduled vs Mach Number | 3-7 |
| 3-2 | Lateral Estimator Gains Scheduled vs Mach Number | 3-7 |
| 3-3 | Parameter Values for State Extrapolation | 3-12 |
| 4-1a | Preflight Parameter Specifications for the Cruise Phase | 4-20 |
| 4-1b | Preflight Parameter Specifications for the Cruise Phase | 4-22 |
| 4-2 | Logic for Speed Control Flag Settings | 4-32 |
| 4-3 | Logic for Elevator Control Flag Settings | 4-33 |
| 4-4 | Logic for Aileron Control Flag Settings | 4-33 |
| 5-1 | Fixed Parameters | 5-32 |
| 6-1 | Definitions of Boundaries of Small Error Control Regions | 6-19 |
| 6-2 | Control Actions in Small-Error Control Regions | 6-20 |
| 6-3 | Definitions of Boundaries and Derived Parameters for the Large-Error Control Regions | 6-23 |
| 6-4 | Control Actions in Large-Error Control Regions | 6-23 |
| 6-5 | Glossary for On-Orbit Phase | 6-44 |
| 7-1 | Input/output Summary | 7-2 |
| 7-2 | Gains for Boost Phase | 7-12 |


PART 2


| | | |
|---|---|---|
| 2-1 | ATTITUDE_MANEUVER MODULE DICTIONARY | 2-24 |
| 2-2 | ON-ORBIT ATTITUDE MANEUVER VARIABLE-CONSTANT DICTIONARY | 2-25 |

# PART 1

# DIGITAL FLIGHT CONTROL SYSTEM

SECTION 1

# INTEGRATED STRUCTURE; EXECUTIVE ROUTINE

by

Peter S. Weissman

## 1.1 Integrated Structure

The DFCS program consists of an executive routine, several major (functional) subroutines, and a number of utility subroutines. The executive routine performs "housekeeping" which is required regardless of the mission phase and calls the major subroutines in the appropriate order. There is a major subroutine for each of the following functions: Backup Control System (BCS) implementation, "pad load" manipulation, input manipulation, filter and parameter initialization, filter update (or measurement incorporation), control logic, command transmission, filter pushdown, (or state extrapolation), and parameter estimation. Most of these routines are divided, at least in part, into separate segments for the various mission phases.

This structure of executive routine and functional subroutines gives modularity to the program; it allows complete flexibility in designing the state estimation and control algorithms for each of the mission phases, designating new sensors or control effectors, and indeed, redefining the mission phases themselves. At the same time, however, the unified structure which groups together the coding for similar functions encourages economy both in computer storage, and in engineering design and programming effort because it is so easy for different mission phases to use common coding whenever it is appropriate to do so. Another advantage to this structure is that the calls to the functional subroutines indicate the completion of one set of calculations and the beginning of another; hence, they are suitable break points for external interrupts, and they facilitate the testing of new coding. For example, if modification to the orbit thrust vector control law is contemplated while other segments of this phase's computations are unchanged, the new coding can be added as a totally separate subroutine, leaving the old coding intact. During testing, it is then possible to switch between old and new control laws following completion of the first segment of state filtering. This feature also could be of utmost importance for emergency or backup coding that is required to be functionally separated from the primary coding.

The unified DFCS structure has additional, managerial advantages that are valuable in an enterprise as complex as the Space Shuttle program. The use in different mission phases of the same assumptions about input/output, timing and other

interfaces is encouraged if not enforced. Furthermore, commonality in conventions, nomenclature and formats is encouraged, which simplifies and clarifies both testing and documentation.

Not all operations of a control system need to be performed with the same frequency. The current version makes allowance for 3 sampling intervals, the longer intervals being multiples of the shortest interval. In the current cruising flight phase, for example, the "fast" rate (10 sps) is used for angular control, the "medium" rate (2 sps) is used for velocity control, and the "slow" rate (1/2 sps) is used for altitude control and parameter updating. The medium and slow computations are offset, in order to prevent them from occurring on the same control cycle. Allowance is made for different sets of sampling intervals in different mission phases.

## 1.2 DFCS Executive Routine

An overview of the logic flow through the DFCS Executive Routine is shown in Fig. 1-1; Fig. 1-2 illustrates the implementation of this logic. The major subroutines of the DFCS are denoted in these figures by the hexagonal blocks. Although the executive serves all the mission phases, many of the algorithms within the subroutines are specifically designed for a single phase. Table 1-1 defines the mission phases for which the DFCS makes provision. The inputs to the Executive Routine are listed in Table 1-2; the outputs are listed in Table 1-3. The operations performed by the Executive Routine are discussed in the following paragraphs.

The phase indication is changed from "entry" to "cruise" when the Mach number drops below 0.9. All other determination of phase is currently assumed to be done outside the DFCS.

When the Backup Control System (BCS) is requested, almost all the control system functions are performed using algorithms in a special subroutine that is separate from the rest of the DFCS. The executive transfers control to this subroutine if the appropriate flags have been set. The BCS mode has been designed only for the cruise phase at this time.

If there is an external restart request, or the phase or mode has changed since the previous pass, the executive sets flags to

1) call for a DFCS initialization pass
2) call for initialization of parameters for any modes that are new
3) cause the computations that are done at the slower sample rates to be performed this pass.

If the reason for initialization was an external restart request or a phase change, the Executive Routine establishes the counters and timers that reflect the desired

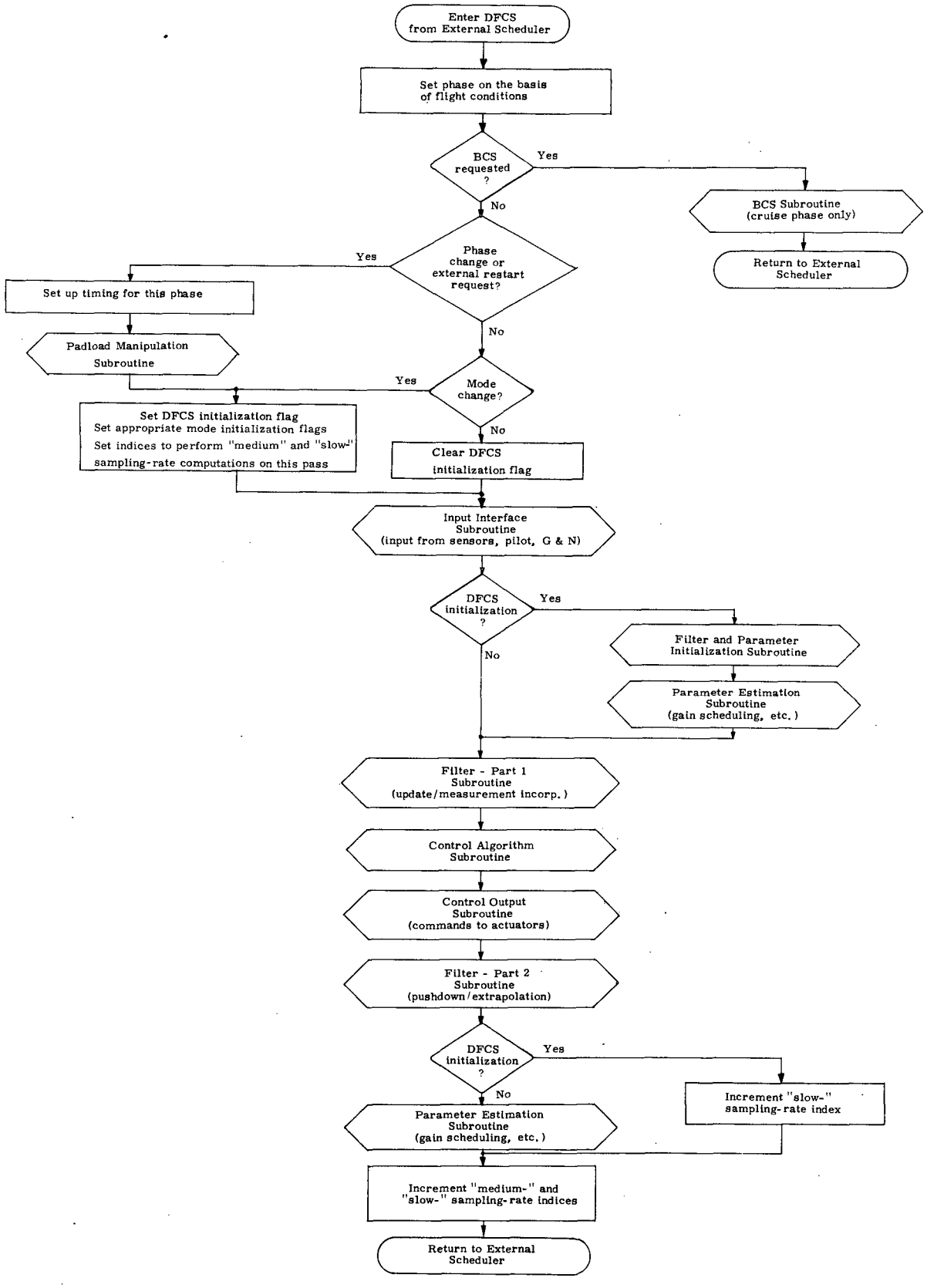



Fig. 1-1 Logic Flow of DFCS Executive

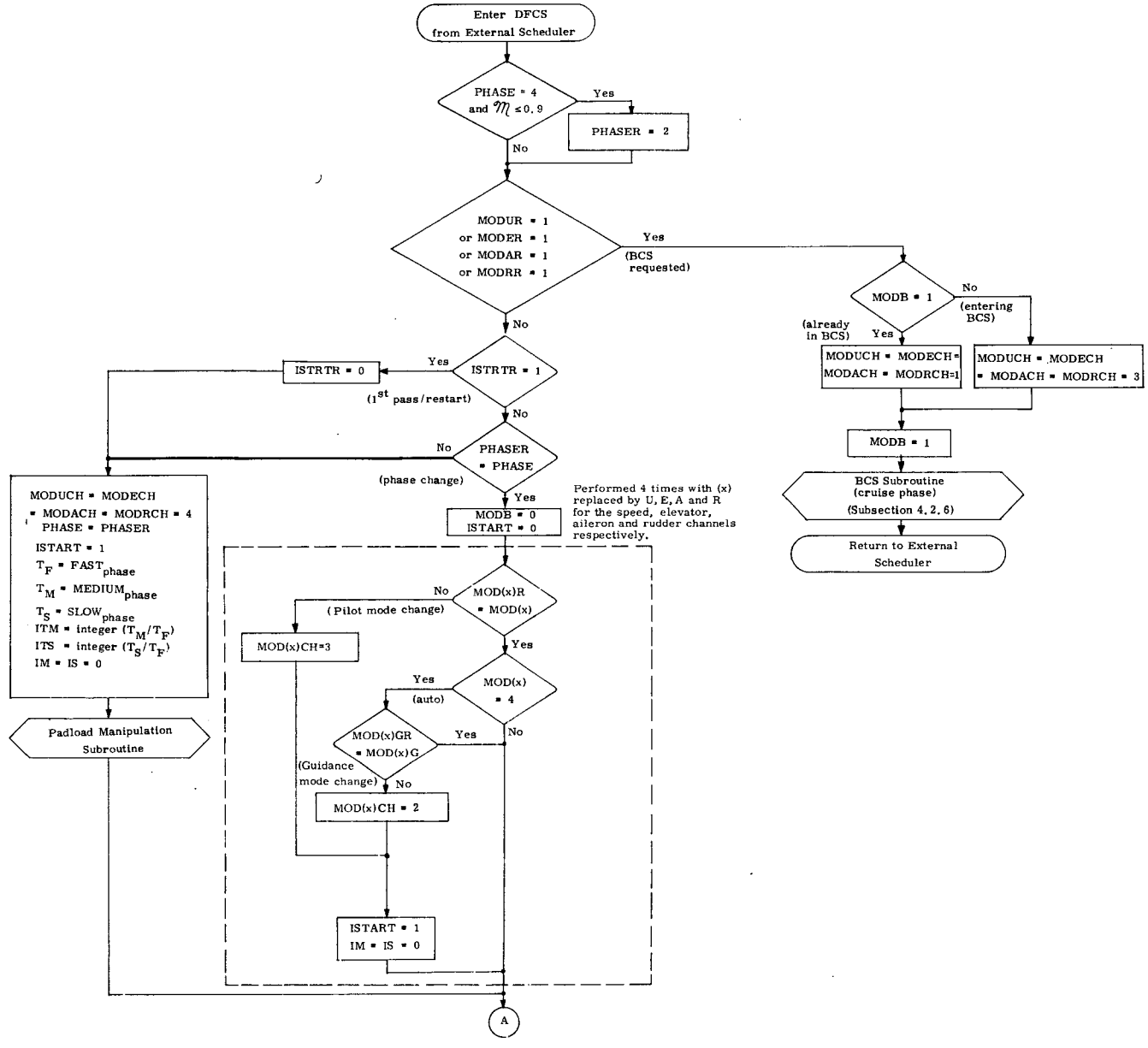
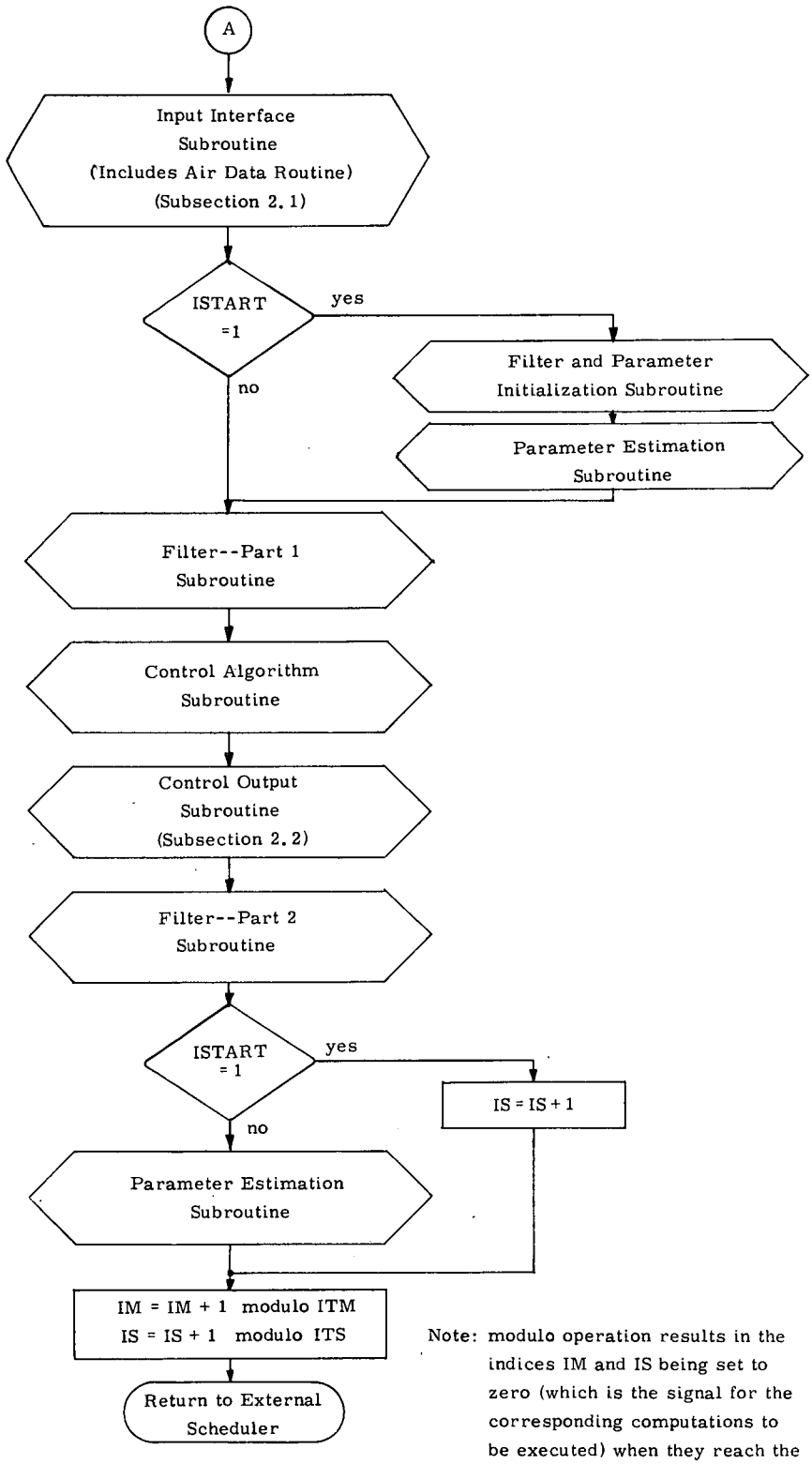



Fig. 1-2 Detailed Logic Flow of DFCS Executive




Note: modulo operation results in the indices IM and IS being set to zero (which is the signal for the corresponding computations to be executed) when they reach the values ITM and ITS respectively.

Fig. 1-2 Detailed Logic Flow of DFCS Executive (Cont)

Table 1-1 DFCS Mission Phases

| | | |
|---|---|---|
| 1 | Rollout | Transition from aerodynamic control to steering and braking with landing gear; after touchdown; not currently implemented. |
| 2 | Cruise | Approach and landing following entry phase (currently below Mach 0.9) and horizontal powered flight (take-off through landing). |
| 3 | — | |
| 4 | Entry | From after the deorbit maneuver to cruise phase; from total reliance on jets (ACPS), high angle of attack and hypersonic to primary control with the aerodynamic surfaces (ACS), low angle of attack and subsonic. |
| 5 | On-orbit, TVC | Thrust vector control during maneuvers with the Orbital Maneuvering System. |
| 6 | On-orbit, ACPS | Control with the jets during coasting flight. |
| 7 | Insertion | Insertion into orbit following the Boost phase; not currently implemented, but similar to on-orbit TVC control. |
| 8 | Boost | Flight from launch almost to insertion in orbit using the Solid Rocket Motors and/or the Main Propulsion System. |

Table 1-2 Inputs to DFCS Executive Routine

| | |
|---|---|
| ISTRTR | External request for DFCS initialization (1); none (0); cleared in Executive Routine |
| $\mathcal{M}$ | Mach number |
| MODA, MODE, MODR, MODU | Current control mode - aileron, elevator, rudder, speed loops |
| MODAG, MODEG, MODRG, MODUG | Current guidance mode - aileron, elevator, rudder, speed loops |
| MODAGR, MODEGR, MODRGR, MODUGR | Requested guidance mode - aileron, elevator, rudder, speed loops |
| MODAR, MODER, MODRR, MODUR | Requested control mode - aileron, elevator, rudder, speed loops |
| PHASER | Flight phase flag (external) |

Table 1-3 Outputs of DFCS Executive Routine

| | |
|---|---|
| *IM, IS | Medium- and slow- sampling-rate indices: 0 = perform calculations this pass |
| ISTART | Initialization pass (1), not initialization (0) |
| ITM, ITS | Number of fast passes per medium (slow) pass |
| *MODB | BCS mode (1), not BCS mode (0) |
| MODACH, MODECH, MODRCH, MODUCH | Indicate mode change - aileron, elevator, rudder, speed loops: 1 = no change; 2 = guidance mode change; 3 = control mode change; 4 = phase initialization |
| *PHASE | Flight phase flag (internal copy of PHASER) |
| $T_F$, $T_M$, $T_S$ | Fast-, medium- and slow- sampling-rate sample periods for current phase |

* Also an input to the next Executive Routine pass

sampling rates for the current phase, and calls the major subroutine which transfers the set of pad-loaded constants required for the current phase from the pad-load locations into working storage. The "pad-load" denotes those parameter values that are determined prior to a flight and are loaded into the flight computer on the launch "pad".

Regardless of whether or not it is an initialization pass, the Executive Routine calls the Input Interface Routine, which reads all the dynamic input required by the current phase and mode: sensor measurements, navigation estimates, guidance commands, and manual commands. All the dynamic data is read at the same point to ensure that all the values are registered with respect to the other values. The parameters are differenced, rotated into other coordinate systems and otherwise manipulated so as to be in the proper form for use in the other routines. During the appropriate phases, an Air Data Routine is called which manipulates some aerodynamic parameters. These two routines are described in Section 2.

At this point, the logic flow is detoured in an initialization pass to go through the Filter and Parameter Initialization Routine and the Parameter Estimation Routine. The former assigns values to those variables that need to be initialized when the phase (or mode) is entered. The latter determines the current values of parameters that will be regularly updated in the course of this phase (or mode).

The flow then returns to the normal path. The subroutines entitled Filter - Part 1, Control Algorithms, Control Output and Filter - Part 2 are called in sequence. These subroutines comprise the core of the control system. After they are executed, the Parameter Estimation Routine is called; it is not called at this point in an initialization pass because it was executed earlier in the pass. The state filters are divided into two parts in order to minimize the amount of computation between data input and control command. The first part is limited to updating the filter estimate according to the most current measurement. The second part, which occurs after the control is exercised, is used to pushdown data (in the case of classical digital filters) or to propagate the state (in the case of discrete-time modern control estimation). The Control Algorithm Routine scales and combines the error signals from the filter routine to form control commands for the aerodynamic surfaces, reaction control thrusters, etc. The Control Output Routine, which is described in Subsection 2.2, prepares the control commands for the DFCS interface with the control actuators. The Parameter Estimation Routine updates the gains required for estimation and control. The functions performed in this routine may include scheduled or adaptive gain adjustment. Much of the Parameter Estimation Routine is executed only at the slow sampling rate.

After these routines have been executed, the Executive Routine increments the indices that indicate the "medium-" and "slow-sampling-rate" DFCS passes. If it is

the end of an initialization pass, the "slow-sampling-rate" index is given an extra increment. This extra increment will smooth the load on the computer by offsetting future "slow" passes from "medium" passes, provided that one sampling rate is an integer multiple of the other. The Executive Routine then returns control to the external executive.

The algorithms incorporated in the routines pertaining to input and output are presented in the following section of this document. The contents of the other routines are presented phase by phase in the succeeding sections.

# SECTION 2

# DFCS INPUT AND OUTPUT

by

J. Edwin Jones, Peter S. Weissman, Craig C. Work,
Edward T. Kubiak (JSC/EJ3)

This section describes the interface between the DFCS and other components of the Space Shuttle avionics. Subsection 2.1 discusses the DFCS inputs. It lists all the inputs; i.e., state measurements, guidance commands, and pilot commands. It includes a description of a routine which processes some of these quantities prior to their use in the filtering and control algorithms. It also describes a routine that fills the function of an air data computer.

Subsection 2.2 lists all the outputs from the DFCS to the control actuators and describes the limiting performed on these commands. Finally, Subsections 2.3 and 2.4 present two candidate jet selection schemes; the first is powerful and flexible, but the second may suffice for the Space Shuttle requirements.

Much of what is presented in this section depends heavily on many features of the Shuttle avionics that have not yet been specified, much less modelled in a simulator. Therefore, in many cases convenient assumptions have been made for the sake of proceeding with the DFCS design. These assumptions include the following:

1) The inputs indicated in Subsection 2.1 are, in fact, available to the DFCS.
2) The actuators indicated in Subsection 2.2 are, in fact, available to the DFCS.
3) The DFCS does not deal with redundancy management of sensors or actuators.
4) The DFCS does not create outputs for pilot displays or telemetry.
5) The DFCS can schedule software tasks to turn ACPS jets off and a sufficiently fine clock granularity is available for such scheduling.

It is expected that each of these assumptions will, in time, be replaced by more realistic specifications, at which time these designs will have to be revised.

## 2.1 Inputs Required by the DFCS

Table 2-1 lists all inputs currently required by the DFCS in the various mission phases. They are presumed to be available from sensors (or sensor-interface modules), the guidance and navigation module, and the pilot (or command-interface modules). In this interface, and throughout the DFCS, English units (viz feet, pounds, and seconds) and radians are used.

Table 2-1 Inputs to the DFCS

| Phase | Mnemonic | Symbol in Text | Definition |
|---|---|---|---|
| 2,4 | ACCBOD | $\underline{\tilde{a}}^b$ | Linear accel measurement (body coord) |
| 2,4 | AG | $\alpha_G$ | Angle-of-attack guidance command |
| 8,6 | AIV(1) | $\phi$ | Platform outer gimbal angle |
| 8,6 | AIV(2) | $\theta$ | Platform inner gimbal angle |
| 8,6 | AIV(3) | $\psi$ | Platform middle gimbal angle |
| 8 | AIVC(1) | $\phi_C$ | Platform outer gimbal angle command |
| 8 | AIVC(2) | $\psi_C$ | Platform inner gimbal angle command |
| 8 | AIVC(3) | $\psi_C$ | Platform middle gimbal angle command |
| 2 | AYM | $\tilde{a}y$ | Measured lateral acceleration |
| 2 | AZM | $\tilde{a}z$ | Measured normal acceleration (+zbody) |
| 4 | BANKG | $\phi_{VG}$ | Bank guidance command |
| 2 | BG | $\beta_G$ | Sideslip guidance command |
| 2 | CG | $c_G$ | CAS guidance command (QG or CSTARG) |
| 2,4 | CRTB | $C_r^b$ | Reference-to-body direction cosine matrix |
| 2 | DAG | $\delta a_G$ | Aileron guidance command |
| 2 | DDG | $\dot{\delta}_G$ | TAEM guidance interface variable |
| 2 | DEG | $\delta e_G$ | Elevator guidance command |
| 8 | DELORB | $\delta_{ORB}$ | Prestored orbiter engine deflection command |
| 8 | DELSRM | $\delta_{SRM}$ | Prestored SRM engine deflection command |
| 2 | DRG | $\delta r_G$ | Rudder guidance command |
| 2 | DSG | $\delta s_G$ | Speed brake guidance command |
| 2 | DTG | $\delta t_G$ | Throttle guidance command |
| 2,4 | DYNP | $\tilde{\bar{q}}$ | Dynamic pressure estimate |
| 2 | HG | $H_G$ | Altitude guidance command |
| 2 | HM | $\tilde{H}$ | Measured altitude |
| 2 | IAGAIN | | AGAIN switch position (1 thru 4) |
| 2 | IATRIM | | Roll stick trim activity: 0,+1(rgt),-1(left) |

Table 2-1 Input to the DFCS (Cont)

| Phase | Mnemonic | Symbol in Text | Definition |
|---|---|---|---|
| 2 | IEGAIN | | EGAIN switch position (1 thru 4) |
| 2 | IETRIM | | Pitch stick trim activity: 0,+1(fwd),-1(aft) |
| 2 | IGEARX | | Extend gear (1), retract gear (0) |
| - | INGUID | INGUID | Flag specifies guidance command set |
| 2 | IPHM | | PHM valid (1), invalid (0) |
| 2 | IRCHAI | | Request change IAINT state (1), no request(0) |
| 2 | IRCHEI | | Request change IEINT state (1), no request(0) |
| 2 | IRGAIN | | RGAIN switch position (1 thru 4) |
| 2 | IRTRIM | | Pedal trim activity: 0,+1(lft),-1(rgt) |
| - | ISTRTR | ISTRTR | Request initialization(1), no request(0) |
| 2 | MODAGR | | Requested values for MODAG |
| 2 | MODAR | | Requested values for MODA |
| 2 | MODBR | | Requested values for MODB |
| 2 | MODEGR | | Requested values for MODEG |
| 2 | MODER | | Requested values for MODE |
| 2 | MODRGR | | Requested values for MODRG |
| 2 | MODRR | | Requested values for MODR |
| 2 | MODUGR | | Requested values for MODUG |
| 2 | MODUR | | Requested values for MODU |
| 2 | PG | $p_G$ | Roll rate guidance command |
| - | PHASER | PHASER | Flight phase flag (external):1=rollout phase<br>2=cruising phase<br>4=entry phase<br>5=orbit TVC phase<br>6=orbit RCS phase<br>7=insertion TVC phase<br>8=booster TVC phase |
| 2 | PHSEL | $\phi_{SEL}$ | Pilot-selected values for PHREF |
| 2,4 | PM | $\tilde{p}$ | Measured roll rate |

Table 2-1 Input to the DFCS (Cont)

| Phase | Mnemonic | Symbol in Text | Definition |
|---|---|---|---|
| 2 | PSG | $\psi_G$ | Heading guidance command |
| 2 | PSM | $\tilde{\psi}$ | Measured heading |
| 2 | PSSEL | $\psi_{SEL}$ | Pilot-selected values for PSREF |
| 2 | QG | $q_G$ | Pitch rate guidance command |
| 2,4 | QM | $\tilde{q}$ | Measured pitch rate |
| 2,4 | RM | $\tilde{r}$ | Measured yaw rate |
| 2,4 | ROLL | $\tilde{\phi}_b$ | Computed roll angle |
| 2 | ROLLG | $\phi_{bG}$ | Guidance commanded roll angle |
| 4 | RR | $\underline{R}_{rel}$ | Position vector in reference coordinates |
| 2,4 | TEMPRK | $TK_A$ | Input air temperature in deg. Kelvin |
| 2 | THG | $\theta_G$ | Theta guidance command |
| 2 | THM | $\tilde{\theta}$ | Measured theta |
| 2 | THSEL | $\theta_{SEL}$ | Pilot-selected values for THREF |
| 6,8 | TNOW | $t_n$ | Present time |
| 2 | UG | $U_G$ | Velocity guidance command |
| 2 | USEL | $u_{SEL}$ | Pilot-selected values for UREF |
| 2,4 | UXV,UYV,UZV | $\underline{UV}_x, \underline{UV}_y, \underline{UV}_z$ | Unit vectors along the velocity coordinate system, in reference coordinates |
| 4 | VBODY | | Air-relative velocity in body coordinates |
| 4 | VMACH1 | | Lateral switch parameter |
| 4 | VR | $\underline{V}_{rel}$ | Air-relative velocity in reference coordinates |
| 8 | WB(1,I) | $RRATE_i$ | Individual roll rate gyro signal |
| 8 | WB(2,I) | $PRATE_i$ | Individual pitch rate gyro signal |
| 8 | WB(3,I) | $YRATE_i$ | Individual yaw rate gyro signal |
| 2 | XAM | $\tilde{x}a$ | Measured roll stick deflection |
| 2 | XEM | $\tilde{x}e$ | Measured pitch stick deflection |
| 2 | XFM | $\tilde{x}f$ | Measured flap controller deflection |
| 2 | XRM | $\tilde{x}r$ | Measured pedal deflection |

Table 2-1 Input to the DFCS (Cont)

| Phase | Mnemonic | Symbol in Text | Definition |
|---|---|---|---|
| 2 | XSM | $\tilde{x}_s$ | Measured speed brake controller deflection |
| 2 | XTM | $\tilde{x}_t$ | Measured throttle controller deflection |
| 8 | YACIN(I) | $\ddot{Y}_i$ | Individual y accelerometer signal |
| 8 | ZACIN(I) | $\ddot{Z}_i$ | Individual z accelerometer signal |

Table 2-2 Inputs to the Input Interface Routine

| Phase | Mnemonic | Symbol in Text | Definition |
|---|---|---|---|
| 2,4 | ACCBOD | $\tilde{\underline{a}}^b$ | Linear accel measurement (body coord) |
| 2,4 | AG | $\alpha_G$ | Angle-of-attack guidance command |
| 4 | BANKDO(2) | $\Delta\dot{\phi}'_V$ | Bank rate filter extrapolated value |
| 4 | BANKG | $\phi_{VG}$ | Bank guidance command |
| 4 | CAT | $\cos\alpha_T$ | Cos (AMEAN) |
| 2,4 | CRTB | $C_r^b$ | Reference-to-body direction cosine matrix |
| 2,4 | DYNP | $\bar{q}$ | Dynamic pressure estimate |
| 2 | HM | $\tilde{H}$ | Measured altitude |
| 2,4 | PM | $\tilde{p}$ | Measured roll rate |
| 2 | QG | $q_G$ | Pitch rate guidance command |
| 2,4 | QM | $\tilde{q}$ | Measured pitch rate |
| 4 | RDOTC | $\dot{\phi}_C$ | Bank rate command |
| 2,4 | RM | $\tilde{r}$ | Measured yaw rate |
| 2,4 | ROLL | $\tilde{\phi}_b$ | Computed roll angle |
| 2 | ROLLG | $\phi_{bG}$ | Guidance commanded roll angle |
| 4 | RR | $\underline{R}_{rel}$ | Position vector in reference coordinates |
| 4 | SAT | $\sin\alpha_T$ | Sin (AMEAN) |
| 4 | VR | $\underline{V}_{rel}$ | Air-relative velocity in reference coordinates |

### 2.1.1 Input Interface Routine

The purpose of the Input Interface Routine is to preprocess measurements and compute state-estimator inputs. Inputs to the routine are listed in Table 2-2. Outputs are listed in Table 2-3. Interface-routine operations are governed by the discretes INGUID, ISTART, PHASE, and MODA in the manner summarized in Table 2-4. The remainder of this subsection is a description of the logic flow of the Input Interface Routine, which is illustrated in Fig. 2-1.

The routine is called by the DFCS executive. If the phase is not either entry or cruise, it returns immediately. If it is a DFCS initialization pass, some variables internal to this routine are set to nominal values.

The section of the routine for preprocessing of measurements is then entered. The angle of attack, the sideslip angle, and the air speed are computed using the air-relative velocity and the direction-cosine matrix. The Air Data Routine is called if the phase is in the aerodynamic regime. Some aerodynamic "measurements" are generated in the Air Data Routine, which is described in the next subsection. The computation of bank angle (under the indicated conditions) concludes the measurement preprocessing segment of the interface routine.

The next section of the routine computes some nominal state values. Nominal values are computed for those state components of the entry phase state estimator (Section 3) for which nominal values are not specified by guidance. Nominal body rates are computed to provide turn coordination in the entry phase. A nominal (commanded) bank rate is also computed for use in the entry phase control logic in the specified flight regime ($I_{LAT} = 0, \bar{q} \geq \bar{q}_3^*$) when the bank error is sufficiently large.

The filter input computation section is now entered. When in cruise phase, either the current roll angle or the bank angle measurement is put into the cell PHI for processing by the cruise phase filters. When in entry phase, the state estimator inputs (namely, errors in the angles of attack, sideslip and bank, and errors in the body rates) are computed. Additionally, estimator extrapolated values are adjusted for changes in the state-variable nominal values.

Finally, a check is made to ensure that INGUID is negative. This avoids reprocessing non-updated G&N inputs in subsequent DFCS passes. Control is then returned to the DFCS executive routine.

### 2.1.2 Air Data Routine

The Air Data Routine is called by the Input Interface Routine during the atmospheric flight phases. It computes several parameters required for aerodynamic control, serving the purpose of an air data computer.

Table 2-3 Outputs from Input Interface Routine

| Phase | Mnemonic | Symbol in Text | Definition |
|---|---|---|---|
| 4 | AI(1) | $\Delta\tilde{\alpha}$ | Angle-of-attack filter input |
| 2 | AM | $\tilde{\alpha}$ | Measured angle-of-attack |
| 4 | AO(2) | $\Delta\hat{\alpha}'$ | Angle-of-attack filter extrapolated value |
| 4 | BANKI(1) | $\Delta\tilde{\phi}_V$ | Bank filter input |
| 4 | BI(1) | $\Delta\tilde{\beta}$ | Sideslip angle filter input |
| 2 | BM | $\tilde{\beta}$ | Measured sideslip |
| 2 | HM | $\tilde{H}$ | Measured altitude |
| 2 | PHG | $\phi_G$ | Bank angle guidance command |
| 2 | PHM | $\tilde{\phi}$ | Measured bank angle |
| 4 | PI(1) | $\Delta\tilde{p}$ | Roll rate filter input |
| 4 | QI(1) | $\Delta\tilde{q}$ | Pitch rate filter input |
| 4 | RI(1) | $\Delta\tilde{r}$ | Yaw rate filter input |
| 2 | UM | $\tilde{U}$ | Measured velocity |

Table 2-4 Discretes Governing Logic Flow in Input Interface Routine

| | |
|---|---|
| ISTART | DFCS initialization pass flag.<br>= 1: initialize variables this pass.<br>= 0: do not initialize this pass. |
| PHASE | Mission phase.<br>= 2: cruise phase.<br>= 4: entry phase.<br>(Routine not entered in other phases.) |
| INGUID | Flag from guidance.<br>= ±2: guidance commands are pitch rate and roll (about body x-axis); terminal part of cruise phase.<br>= ±1: guidance commands are angle-of-attack and bank (about velocity vector); entry and early cruise.<br>< 0: as above, but no new guidance commands to process this pass (set negative internally). |
| MODA | Mode flag for cruise phase aileron channels.<br>= 4: automatic mode; compute roll (or bank) filter inputs.<br>≠ 4: not automatic mode. |
| $I_{LAT}$ | Mode flag for entry phase lateral control.<br>= 0: early entry, compute desired bank rate.<br>= 1: late entry |

Note: The parameters $\bar{q}_3^*$, $r_{\phi 1}$, $r_{\phi 2}$, $r_{\phi 3}$ are used in the Input Interface Routine as limits. They are introduced in the discussion of the entry phase control logic in Section 5.

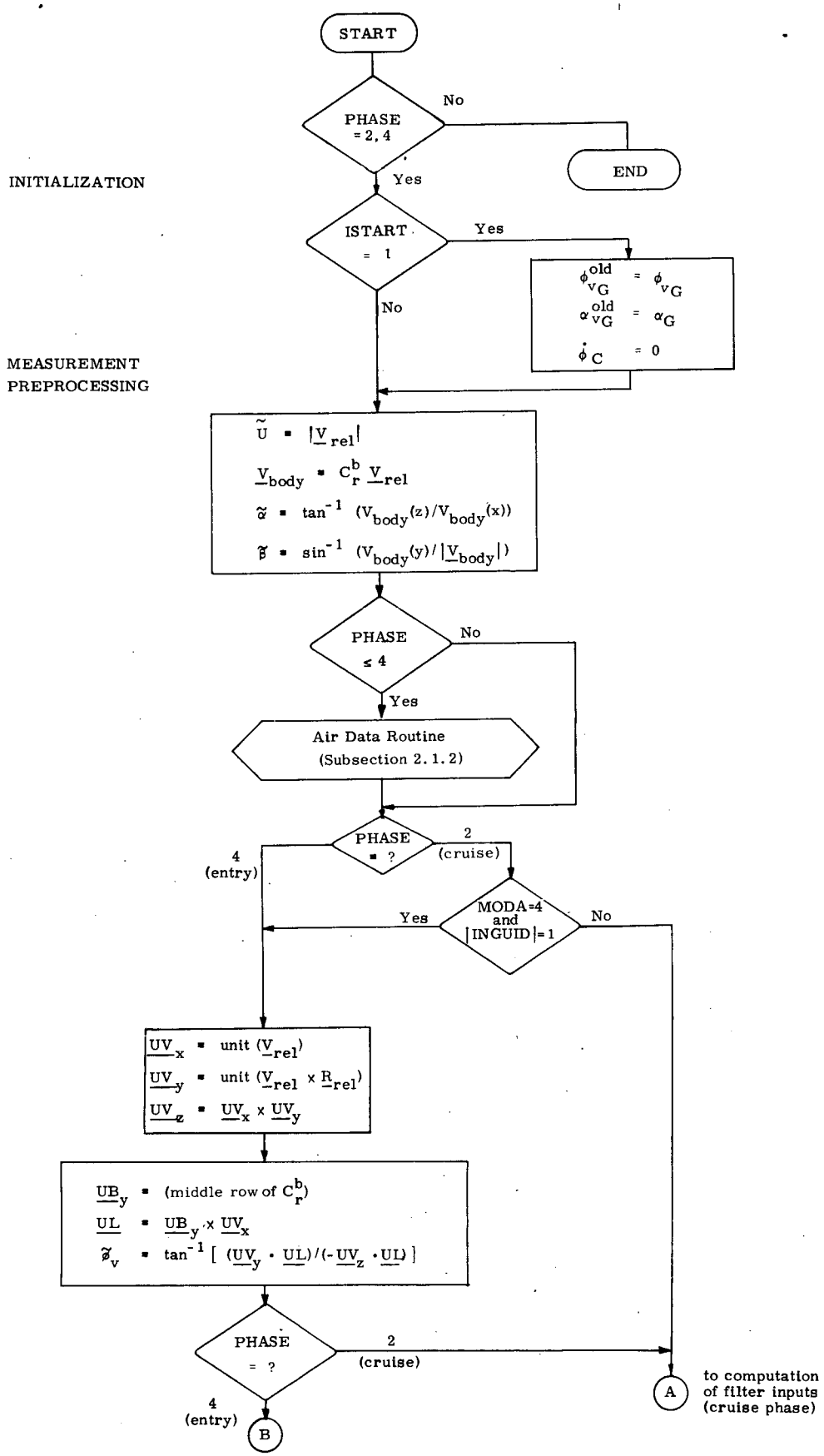



Fig. 2-1 Input Interface Routine Logic Flow

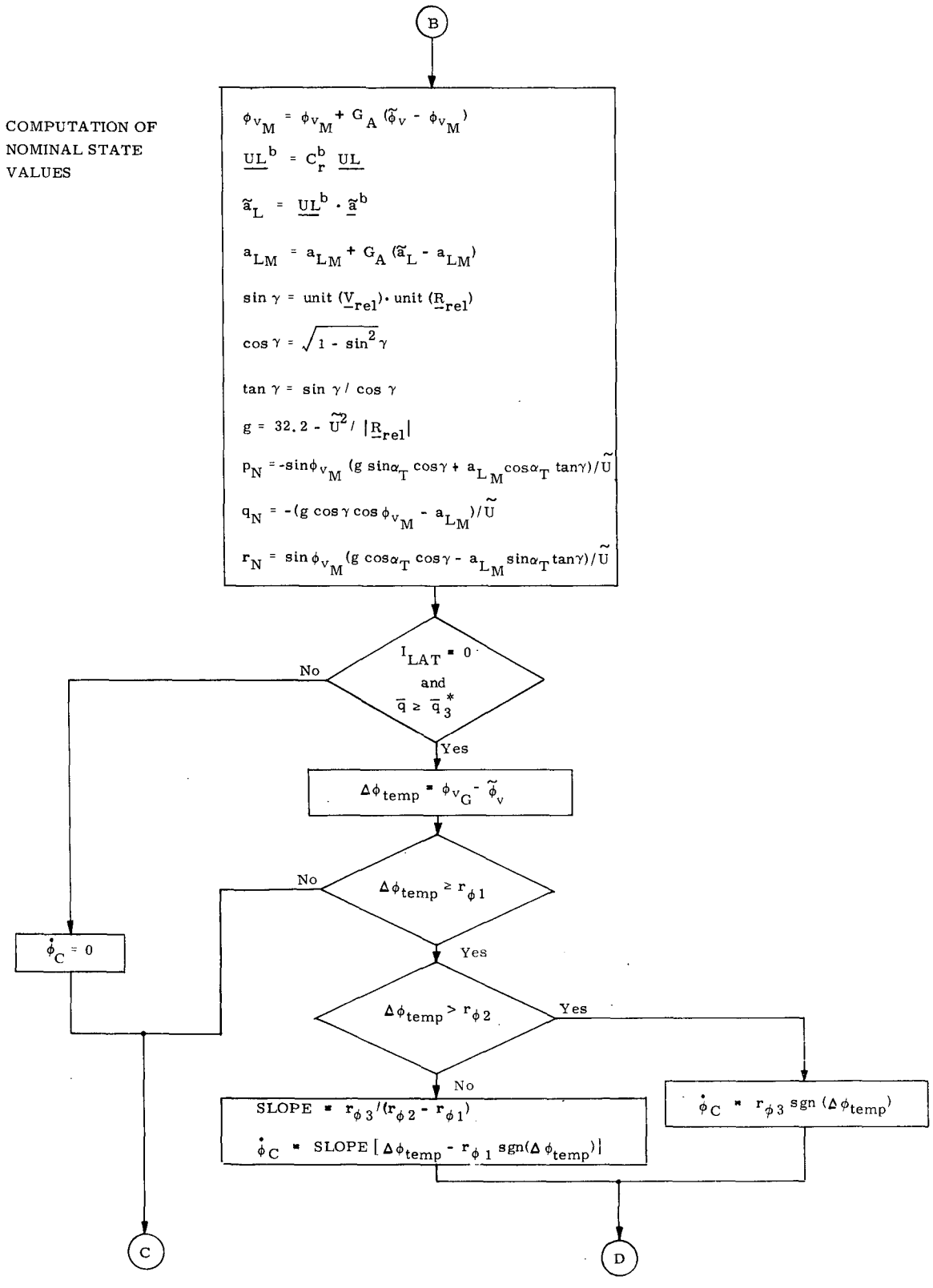



Fig. 2-1 Input Interface Routine Logic Flow (Cont)

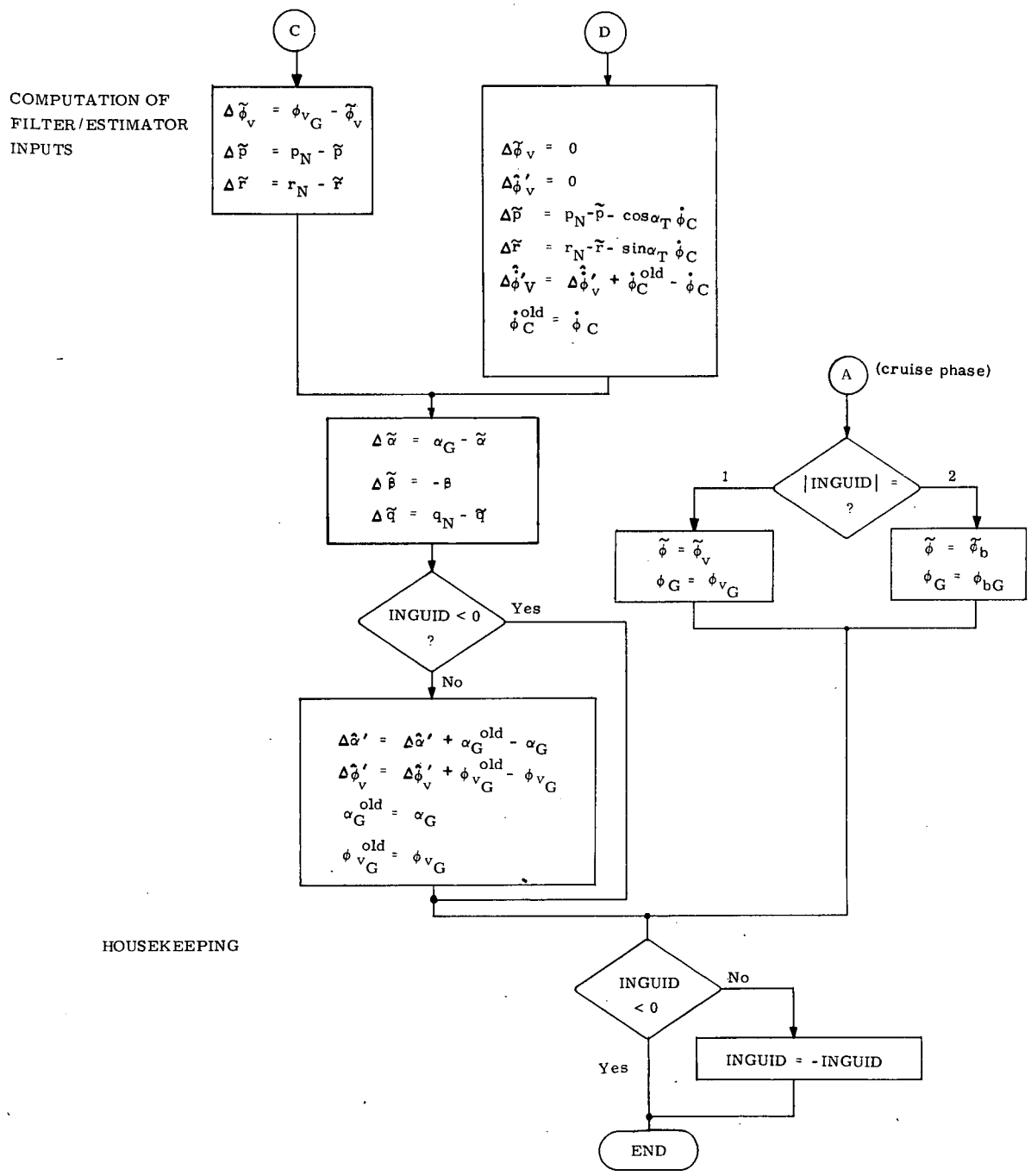



Fig. 2-1 Input Interface Routine Logic Flow (Cont)

The inputs to this routine are the following measurements which are obtained from outside the DFCS or are computed by the Input Interface Routine:

| | | |
|---|---|---|
| $TK_A$ | - | temperature in degrees Kelvin |
| $\tilde{\alpha}$ | - | angle-of-attack (radians) |
| $\tilde{H}$ | - | barometric altitude (ft) |
| $\tilde{U}$ | - | air-relative velocity (ft/sec) |

The outputs are:

| | | |
|---|---|---|
| $\alpha_T$ | - | mean (trim) angle-of-attack (radians) |
| $\sin\alpha_T$, $\cos\alpha_T$ | | sine (cosine) of $\alpha_T$ |
| $U_M$ | - | mean airspeed (ft/sec) |
| $\mathcal{M}$ | - | Mach number |

The means (which are the outputs of first-order filters with time constants equal to the slow sampling period) and the sine and cosine are updated every pass. The means are set equal to the values of their respective measured inputs in an initialization pass.

The other outputs are only computed on slow-sample-rate passes. The computations are shown in Fig. 2-2.

It is expected that, in the future, when more realistic models of the sensors are available, this routine will compute the dynamic pressure and perform other computations.

## 2.2 Outputs to Control Effectors

The DFCS computes commands for the control effectors listed in Table 2-5. The question of how the commands are transmitted to these effectors is considered to be outside the scope of this document; the following simple implementation is currently assumed:

1) Commands are transmitted by the DFCS to the actuators (or a software module that interacts with the actuators) at the end of each execution of the control routine.
2) The DFCS limits the magnitude of each output, but not its rate of change.
3) Timing of jet firings is obtained by means of a computer clock that can be set to call a DFCS routine which will turn jets off after the required time.

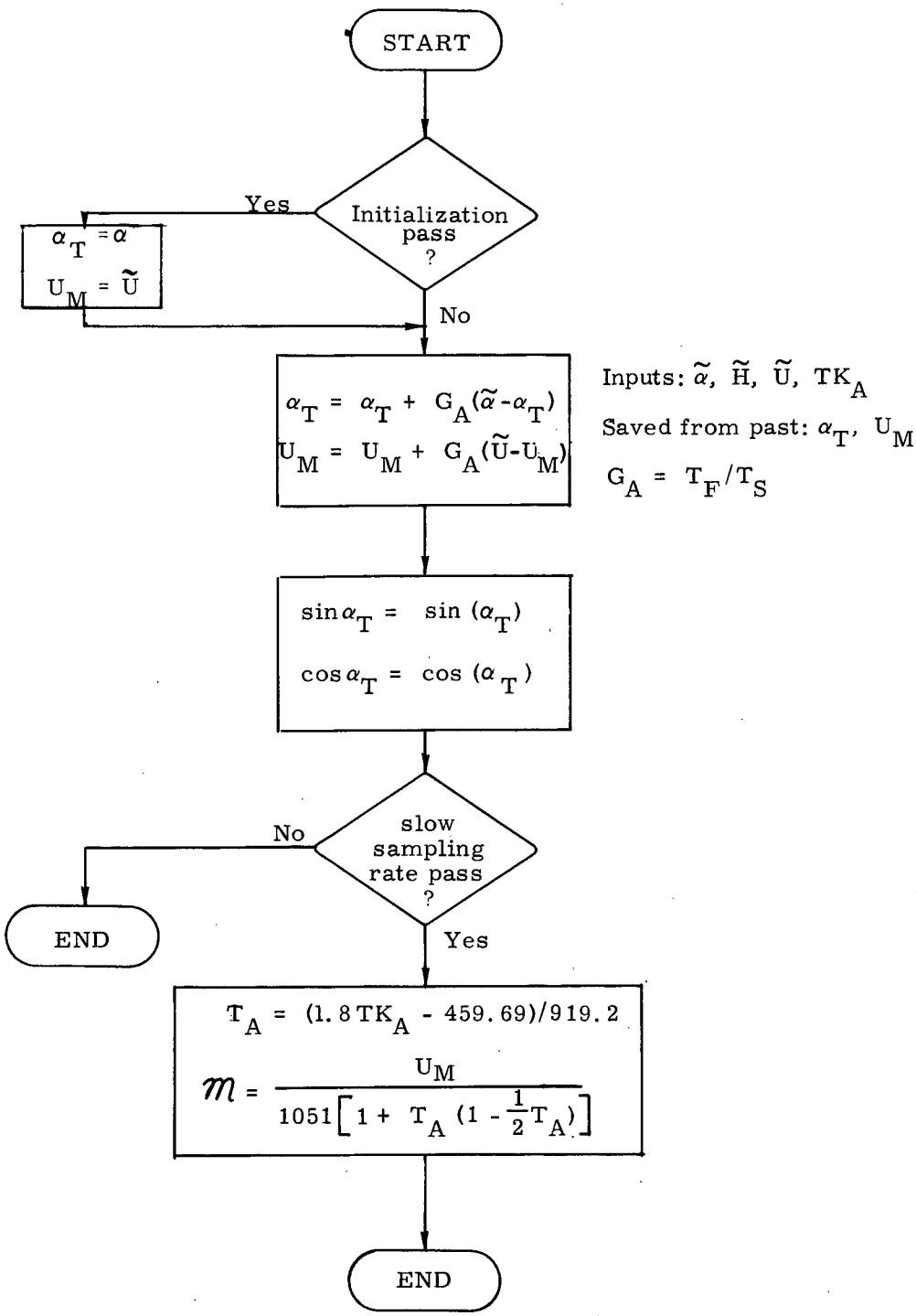



Fig. 2-2 Computations in Air Data Routine

Table 2-5 Control Effectors Used by the DFCS

| | | Phases in Which Used * |
|---|---|---|
| Aerodynamic Control System (ACS) | elevons, rudder, speed brake, flaps | 1, 2, 4, 8 |
| Attitude Control Propulsion System (ACPS) | 40 jet thrusters | 4, 5, 6, 7 |
| Air Breathing Engine System (ABES) | throttle | 1, 2 (horizontal flight only) |
| Solid Rocket Motors (SRM) | 2 engines; gimballed at $45^{\circ}$ from body axes | 8 |
| Main Propulsion System (MPS) | 3 engines; gimballed | 8 |
| Orbital Maneuvering System (OMS) | 2 engines; gimballed | 5, 7 |

* Phase designations defined in Table 1-1.

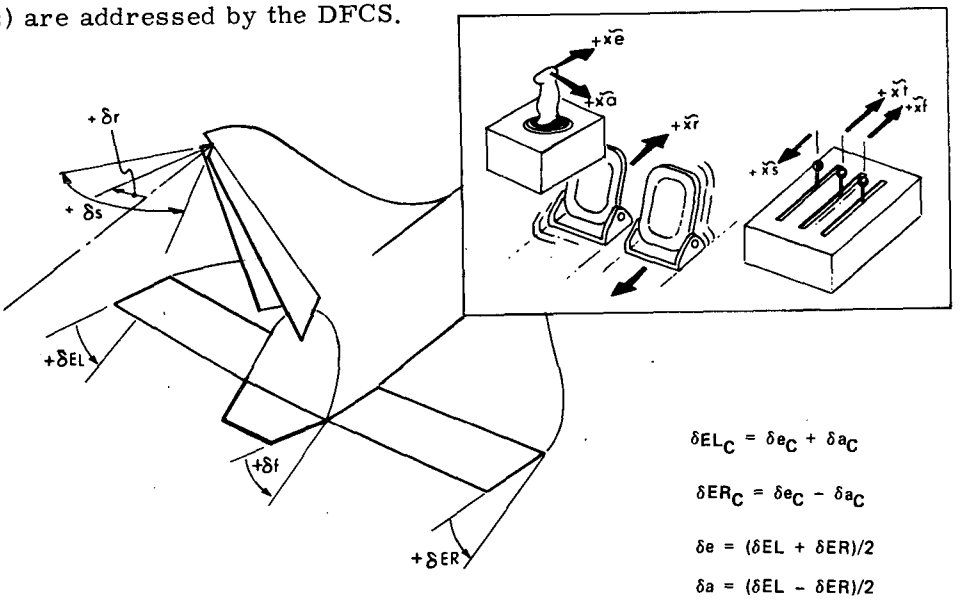
Table 2-6 DFCS Limits for Commands to Control Effectors

| Effector | Limit (deg ) |
|---|---|
| aileron (differential elevon) | $\delta a^{max} = \pm 15.0$ |
| elevator (tandem elevon) | $\delta e^{max} = +15.0$<br>$\delta e^{min} = -40.0$ |
| rudder | $\delta r^{max} = \pm 15.0$ |
| speed brake | $\delta s^{max} = +70.0$<br>$\delta s^{min} = -17.6$ |
| flap | $\delta f^{max} = +10.0$<br>$\delta f^{min} = -12.0$ |
| throttle | $\delta t^{max} = 18,000$ lb<br>$\delta t^{min} = 500$ lb |
| SRM and MPS gimbals | $\delta g^{max} = \pm 8.0$ |

The assumed sign conventions for the aerodynamic control surfaces and the associated pilot controller motions are illustrated in Fig. 2-3.

All the commands are limited within the DFCS prior to being issued. The limits magnitudes are shown in Table 2-6. The aileron and elevator commands are treated separately even though they apply to the same pair of actuators, the elevons, and are, therefore, interrelated; this question will be dealt with in the future.

The limiting and blending (for the elevons and MPS gimbals) of the actuator commands is performed in the Command Output Routine, which is called by the DFCS executive after the Control Routine. The Command Output Routine serves as the single point from which the actuators (or the software modules that interact with the actuators) are addressed by the DFCS.




$\delta EL_C = \delta e_C + \delta a_C$

$\delta ER_C = \delta e_C - \delta a_C$

$\delta e = (\delta EL + \delta ER)/2$

$\delta a = (\delta EL - \delta ER)/2$

| POSITIVE DEFLECTION OF | PRIMARY VEHICLE RESPONSE | PRIMARY FORCES AND MOMENTS | PILOT CONTROLLER MOTION |
|---|---|---|---|
| $\delta e$ | $-\theta$, $-\alpha$ | $-C_m$ | +xe |
| $\delta r$ | $+\beta$, $-\psi$ | $+C_y$, $-C_n$ | +xr |
| $\delta a$ | $+\phi$ | $+C_\ell$ | +xa |
| $\delta ER$ | $-\phi$ | $-C_\ell$ | |
| $\delta EL$ | $+\phi$ | $+C_\ell$ | |
| $\delta s$ | (- U) | $+C_D$ | + xs |
| $\delta f$ | $-\theta$, $-\alpha$ | $-C_M$ | + xf |

Fig. 2-3 Sign Conventions for Aerodynamic Control Surfaces and Manual Controllers

## 2.3 ACPS Jet Selection Subroutine

This section describes a scheme for converting angular-rate-change and translation requests into timed jet firing policies for the ACPS (Attitude Control Propulsion System). This routine has been tested with the DFCS program.

### 2.3.1 Thruster Geometry

The thruster configuration currently incorporated in the jet selection routine is the North American Rockwell ATP design.

As shown in Fig. 2-4, the thrusters for the ATP configuration are grouped in 16 locations. There are 40 thrusters in all, and these are distributed among the 16 locations as shown in Tables 2-7 and 2-8. There are 16 thrusters in the nose of the fuselage and 24 in the "tail".

### 2.3.2 Jet Selection Logic

The DFCS Control Routine determines a commanded velocity change and calls the jet selection subroutine, which handles angular velocity components and translational velocity components simultaneously with a single algorithm. A flag can be set instructing the jet selection subroutine to ignore translation, simplifying and speeding up the jet selection procedure. Furthermore, the jet selection logic can be instructed to deal with jet clusters or with the individual thrusters.

Figure 2-5 is a functional flow diagram of the jet selection logic. A discussion of this logic follows.

The ACPS jet control authority subroutine is only fully executed when the jet selection routine is called for the first time in a mission phase. The control authority subroutine calculates the ACPS control authorities on the basis of total vehicle mass, the vehicle inertia matrix, and the center of mass location. The control authorities are calculated for the jet clusters and for the individual thrusters. The control authority is the angular and translational acceleration in vehicle coordinates generated by firing a jet or cluster of jets. In addition, a cost per unit firing time is assigned to each jet. Currently, all these costs are equal. When a jet is disabled (or restored) and the jet-selection logic is operating on clusters, the jet control authority subroutine is partially executed, appropriately altering the corresponding acceleration vector and cost element.

The jet selection subroutine next checks its inputs to see whether or not there is a new velocity change request. If not, the jet selection subroutine call must be for the purpose of turning off jets which have completed their firing intervals. If the velocity change request is a new one, the linear programming subroutine is called, which expresses the velocity change vector as a linear combination of jet acceleration vectors, the linear coefficients being the firing times.

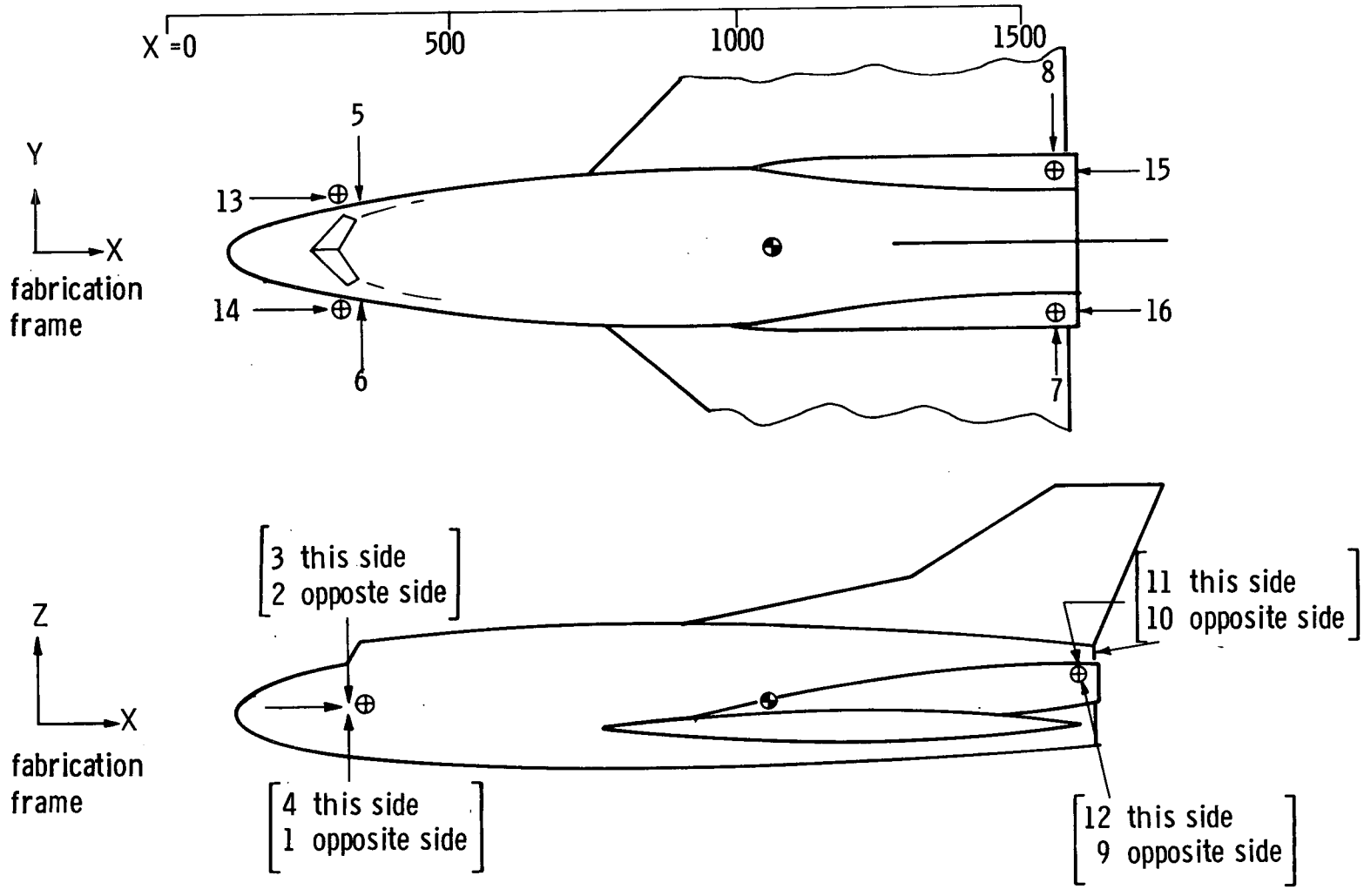



Fig. 2-4 Thruster Locations - NR ATP Baseline Orbiter

Table 2-7 Distribution of ACPS Thrusters

NR ATP Baseline Orbiter

| Thruster Location | Number of Thrusters | Thruster ID |
|---|---|---|
| 1 | 2 | 10, 12 |
| 2 | 2 | 14, 16 |
| 3 | 2 | 13, 15 |
| 4 | 2 | 9, 11 |
| 5 | 2 | 6, 8 |
| 6 | 2 | 5, 7 |
| 7 | 4 | 17, 19, 21, 23 |
| 8 | 4 | 18, 20, 22, 24 |
| 9 | 3 | 32, 34, 36 |
| 10 | 3 | 26, 28, 30 |
| 11 | 3 | 25, 27, 29 |
| 12 | 3 | 31, 33, 35 |
| 13 | 2 | 2, 4 |
| 14 | 2 | 1, 3 |
| 15 | 2 | 38, 40 |
| 16 | 2 | 37, 39 |
| | 40 total | |

Table 2-8 Parameters of Individual Jets

| ID | Jet Location — Body Station (inches) | | | Thrust Components | | |
|---|---|---|---|---|---|---|
| | Fabrication | Frame | | Vehicle | Frame | |
| 1 | 318.0 | - 96.0 | 374.0 | -1.0 | 0.0 | 0.0 |
| 2 | 318.0 | 96.0 | 374.0 | -1.0 | 0.0 | 0.0 |
| 3 | 318.0 | - 96.0 | 349.0 | -1.0 | 0.0 | 0.0 |
| 4 | 318.0 | 96.0 | 349.0 | -1.0 | 0.0 | 0.0 |
| 5 | 318.0 | -109.0 | 368.0 | 0.0 | 1.0 | 0.0 |
| 6 | 318.0 | 109.0 | 368.0 | 0.0 | -1.0 | 0.0 |
| 7 | 318.0 | -109.0 | 355.0 | 0.0 | 1.0 | 0.0 |
| 8 | 318.0 | 109.0 | 355.0 | 0.0 | -1.0 | 0.0 |
| 9 | 318.0 | -109.0 | 346.0 | 0.0 | 0.0 | -1.0 |
| 10 | 318.0 | 109.0 | 346.0 | 0.0 | 0.0 | -1.0 |
| 11 | 330.0 | -109.0 | 346.0 | 0.0 | 0.0 | -1.0 |
| 12 | 330.0 | 109.0 | 346.0 | 0.0 | 0.0 | -1.0 |
| 13 | 318.0 | - 80.0 | 367.0 | 0.0 | 0.0 | 1.0 |
| 14 | 318.0 | 80.0 | 367.0 | 0.0 | 0.0 | 1.0 |
| 15 | 330.0 | - 80.0 | 367.0 | 0.0 | 0.0 | 1.0 |
| 16 | 330.0 | 80.0 | 367.0 | 0.0 | 0.0 | 1.0 |
| 17 | 1536.5 | -158.0 | 425.0 | 0.0 | 1.0 | 0.0 |
| 18 | 1536.5 | 158.0 | 425.0 | 0.0 | -1.0 | 0.0 |
| 19 | 1532.5 | -158.0 | 438.0 | 0.0 | 1.0 | 0.0 |
| 20 | 1532.5 | 158.0 | 438.0 | 0.0 | -1.0 | 0.0 |
| 21 | 1528.5 | -158.0 | 450.0 | 0.0 | 1.0 | 0.0 |
| 22 | 1528.5 | 158.0 | 450.0 | 0.0 | -1.0 | 0.0 |
| 23 | 1525.5 | -158.0 | 464.0 | 0.0 | 1.0 | 0.0 |
| 24 | 1525.5 | 158.0 | 464.0 | 0.0 | -1.0 | 0.0 |
| 25 | 1541.5 | -130.0 | 480.0 | -0.234 | 0.0 | 0.980 |
| 26 | 1541.5 | 130.0 | 480.0 | -0.234 | 0.0 | 0.980 |
| 27 | 1555.5 | -130.0 | 484.0 | -0.234 | 0.0 | 0.980 |
| 28 | 1555.5 | 130.0 | 484.0 | -0.234 | 0.0 | 0.980 |
| 29 | 1567.5 | -130.0 | 487.0 | -0.234 | 0.0 | 0.980 |
| 30 | 1567.5 | 130.0 | 487.0 | -0.234 | 0.0 | 0.980 |
| 31 | 1546.5 | -130.0 | 416.0 | 0.407 | 0.0 | -0.915 |
| 32 | 1546.5 | 130.0 | 416.0 | 0.407 | 0.0 | -0.915 |
| 33 | 1559.5 | -130.0 | 420.0 | 0.407 | 0.0 | -0.915 |
| 34 | 1559.5 | 130.0 | 420.0 | 0.407 | 0.0 | -0.915 |
| 35 | 1572.5 | -130.0 | 428.0 | 0.407 | 0.0 | -0.915 |
| 36 | 1572.5 | 130.0 | 428.0 | 0.407 | 0.0 | -0.915 |
| 37 | 1579.5 | -149.9 | 468.0 | 0.980 | 0.0 | 0.234 |
| 38 | 1579.5 | 149.0 | 468.0 | 0.980 | 0.0 | 0.234 |
| 39 | 1579.5 | -149.0 | 455.0 | 0.980 | 0.0 | 0.234 |
| 40 | 1579.5 | 149.0 | 455.0 | 0.980 | 0.0 | 0.234 |
| c.g. | 1096.5 | 0.1 | 367.9 | | | |

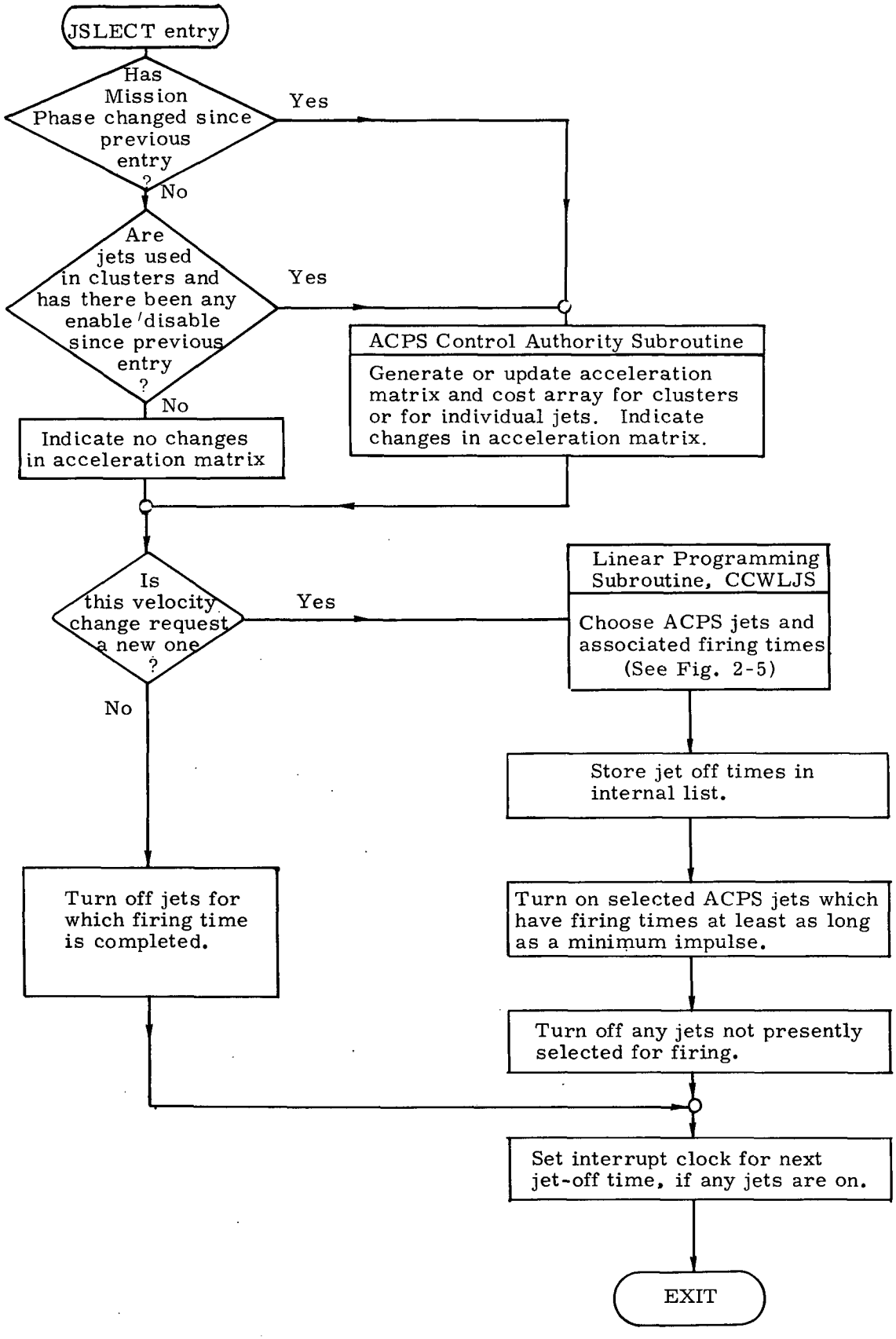



Fig. 2-5 ACPS Jet Selection Functional Flow Diagram

Generally, many different jet combinations could satisfy a given velocity change request; if there is more than one such combination, the total cost is minimized in the selection process. Any jet which is flagged because of failure or mission-related constraints is omitted from consideration.

The selected jets are turned on (if they are not already firing), and all others are turned off and removed from the internal jet-off list. The interrupt clock is set to return control to the jet selection subroutine when the next jet-off is due, and the "new velocity request" flag is zeroed.

The interface parameters for the jet selection package are:

INPUT:

a) Velocity change command vector.
b) Flag for new velocity vector.
c) ACPS jet exclusion list.
d) Translation inclusion flag.
e) Phase flag.

OUTPUT:

a) Jet-on/off commands.
b) Interrupt clock settings.
c) Zeroed flag for new request.

Figure 2-6a presents features of the linear programming subroutine, CCWLJS, and Fig. 2-6b describes its logic flow.

Linear Programming Subroutine

PURPOSE: Represents input vector W as a linear combination of linearly independent columns from matrix B, choosing columns from B to minimize cost of solution.

INPUTS:

LROW - Number of components in W. (Dimension of requests)

LCOL - Number of columns in B matrix. (Number of jets or clusters)

W - Vector (size LROW) to be analyzed as a linear combination of columns from B.

COST - Array (size LCOL) of strictly positive cost coefficients, one associated with each B column.

B - Matrix (LROW by LCOL). Each column represents the acceleration associated with a one-second firing of the corresponding jet in vehicle coordinates. Components may include rotational or translational acceleration elements, or both.

LALARM - Non-zero indicates that previous jet selection may not be taken as a starting basis, for one of the following reasons:

1) Jet exclusion list has been changed
2) $\overset{*}{B}$, LROW, or LCOL have been altered since the preceding pass
3) Preceding delta-V request was not successfully resolved into jet firings. Initialized to 1 on first pass.

JFAIL - Array (size LCOL) of flags, $J^{th}$ element non-zero to indicate $J^{th}$ jet failed.

$\overset{*}{BI}$, $\overline{CF}$, $\overline{LQ}$ - Can be inputs as well as outputs. See definitions below under OUTPUTS.

PARAMETERS - SAVING, MAXPAS, MAXITR, TOLRNC, URGNCY can be chosen to match the precision of the computation and the accuracy of the data.

Fig. 2-6a Features of Linear Programming Subroutine, CCWLJS

Linear Programming Subroutine (Cont)

OUTPUTS:

BI - Transformation matrix (LROW by LROW) going from vehicle coordinates to coordinates based on selected columns of B.

V - Solution vector (size LROW) of coefficients for selected columns of B matrix, satisfying: $\overset{*}{BI}\ \bar{W} = \bar{V}$

CF - Vector (size LROW) of cost coefficients selected from COST array, corresponding to chosen columns from B matrix.

Z - Cost of solution, satisfying: $Z = \bar{CF} \cdot \bar{V}$.

LQ - Array (size LROW) giving the indices and sequence of selected B columns.

MCOUNT - Number of basis element substitutions performed in current execution.

LALARM - Non-zero if $\bar{W}$ was not successfully resolved into jet firings before MCOUNT exceeded MAXITR.

ASSUMPTIONS:

1) Cost array entries are all strictly positive, i.e., greater than zero.

2) Negative LQLIST outputs and negative V components are to be excluded from any implementation of the solution.

3) COST array entries are small compared to URGNCY.

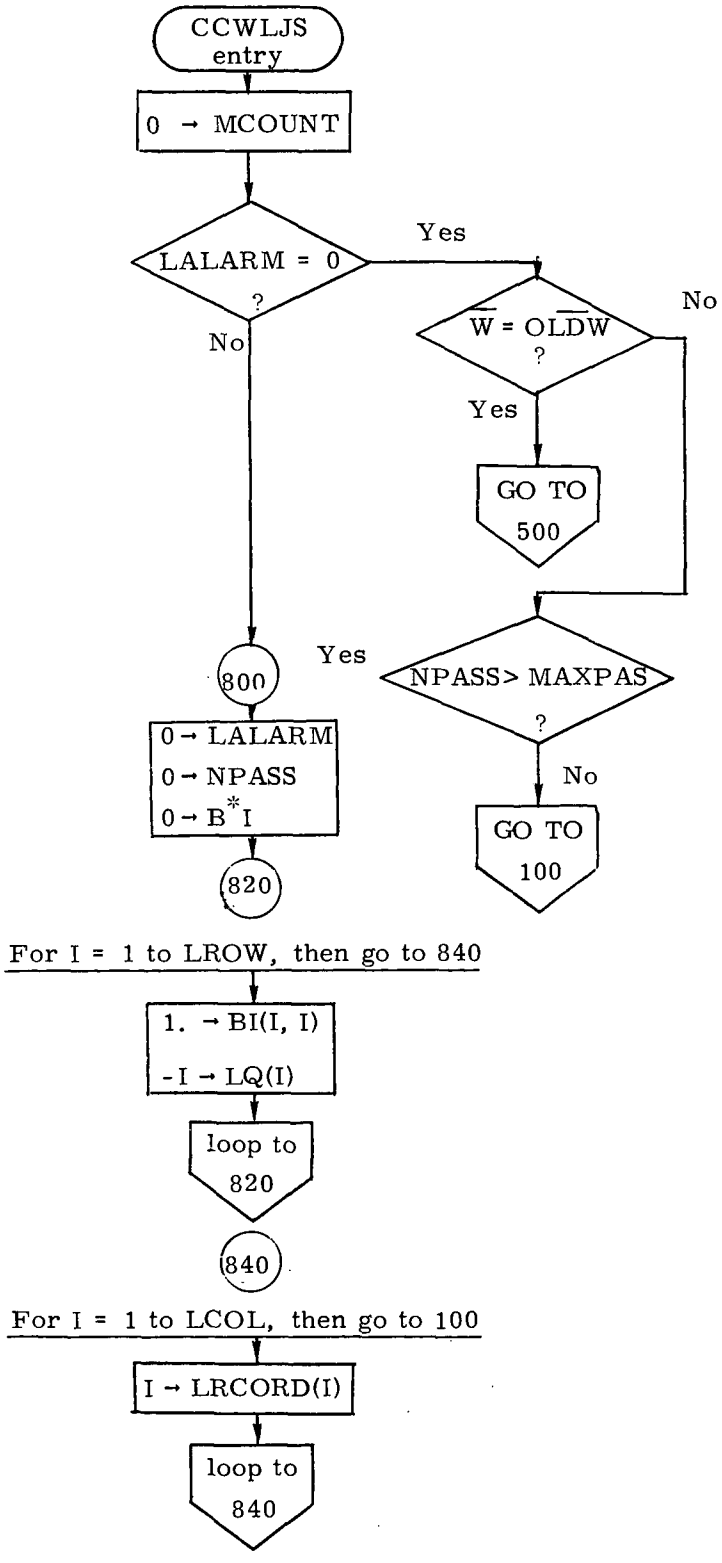
4) BI matrix is never altered by any other program.

5) LALARM will be input as non-zero whenever changes occur in LROW, LCOL, B, or the jet exclusion list.

ENTRY: Call CCWLJS (B, W)

EXIT: RETURN to caller.

Fig. 2-6a Features of Linear Programming Subroutine, CCWLJS (Cont)




MCOUNT counts iterations on this pass

If $\overset{*}{B}I$ cumulative roundoff is not bad, if previous solution was valid, and if $\overset{*}{B}$, LROW, LCOL are unchanged, then start with solution from previous pass. Otherwise initialize completely.

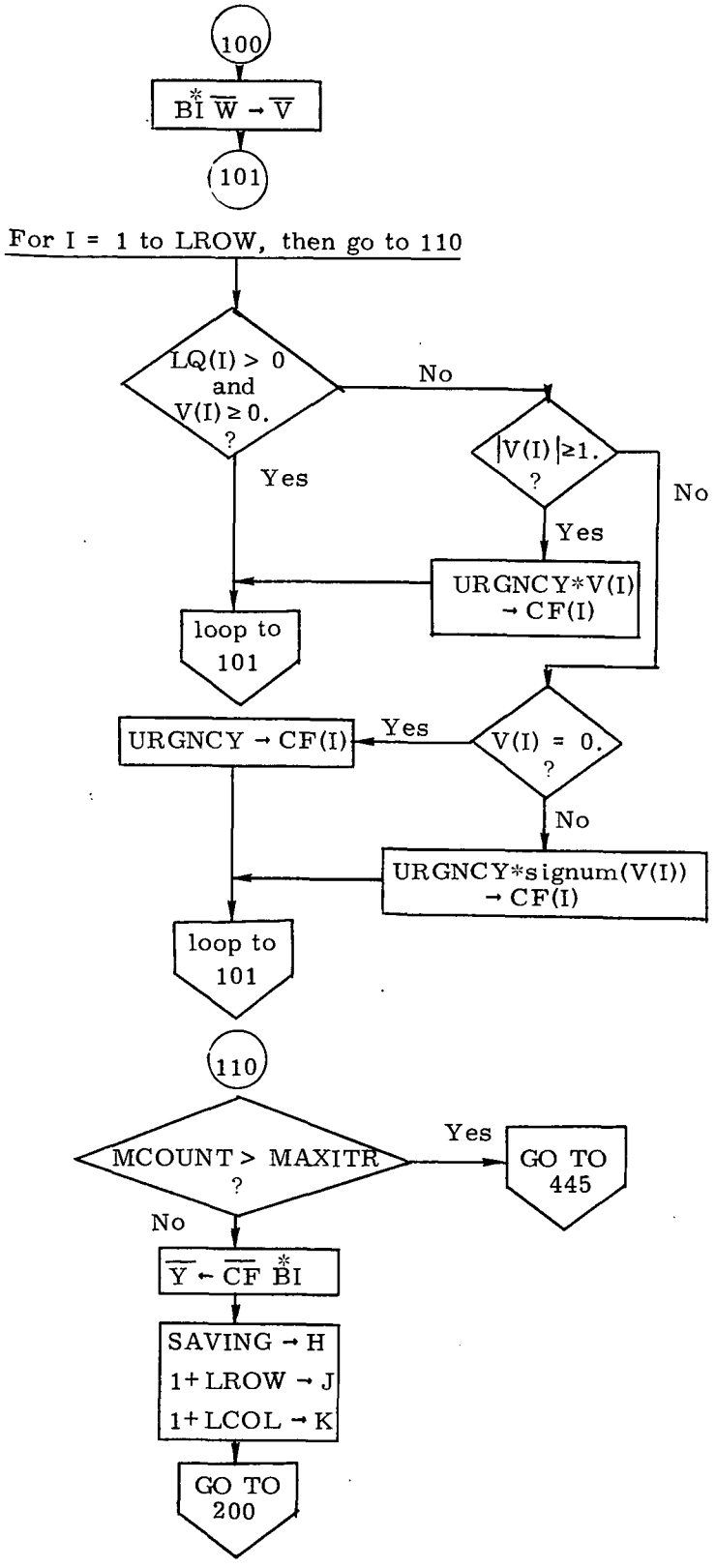
If $\overline{W}$ is unchanged from previous pass, use same solution.

NPASS counts iterations since last complete initialization.

Negative LQ values identify vehicle frame coordinates, which are non-jets.

Positive LRCORD values correspond to non-selected jets.

Fig. 2-6b Flow Chart of Linear Programming Subroutine, CCWLJS




$\overline{V}$ contains latest version of solution in terms of current basis.

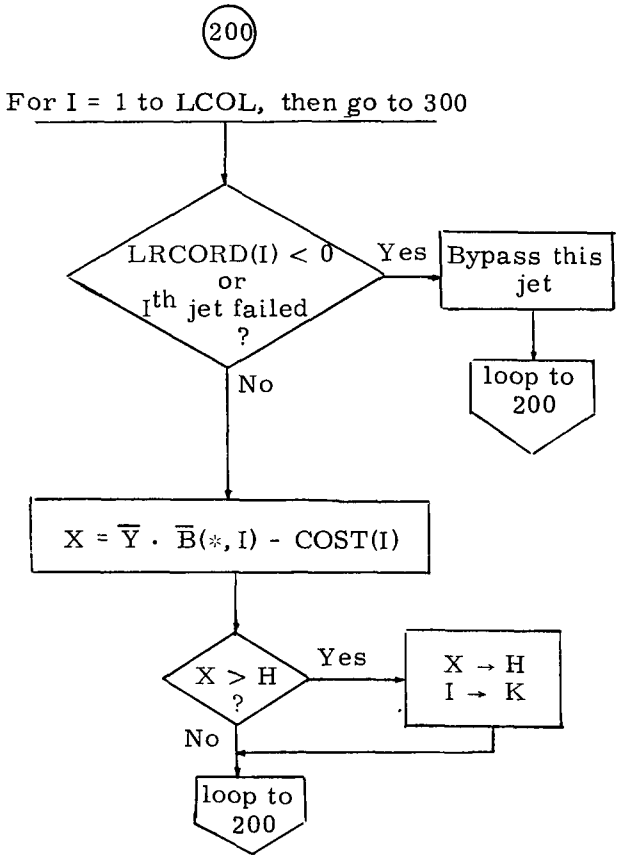
Cost target vector is initially a weighted version of $\overline{V}$ (or a signum ($\overline{V}$) vector).

URGNCY factor places premium on jets which approximate $\overline{W}$ or which fire in directions represented so far by non-jets.

Y(I) is weighted cost of generating the $I^{th}$ original (vehicle) basis vector as a linear combination of current basis vectors.

Initialize the search for the next jet to be included in basis.

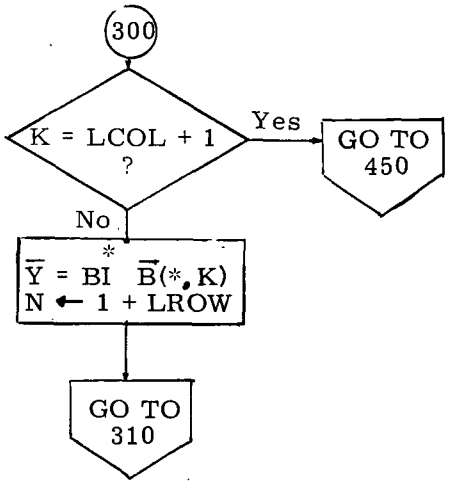
Fig. 2-6b Flow Chart of Linear Programming Subroutine, CCWLJS (Cont)




Search for next jet to be introduced to basis, bypassing any which are failed or already in basis.

X is the weighted price penalty for simulating a one-sec firing of $I^{th}$ jet, using the basis rather than the jet itself.

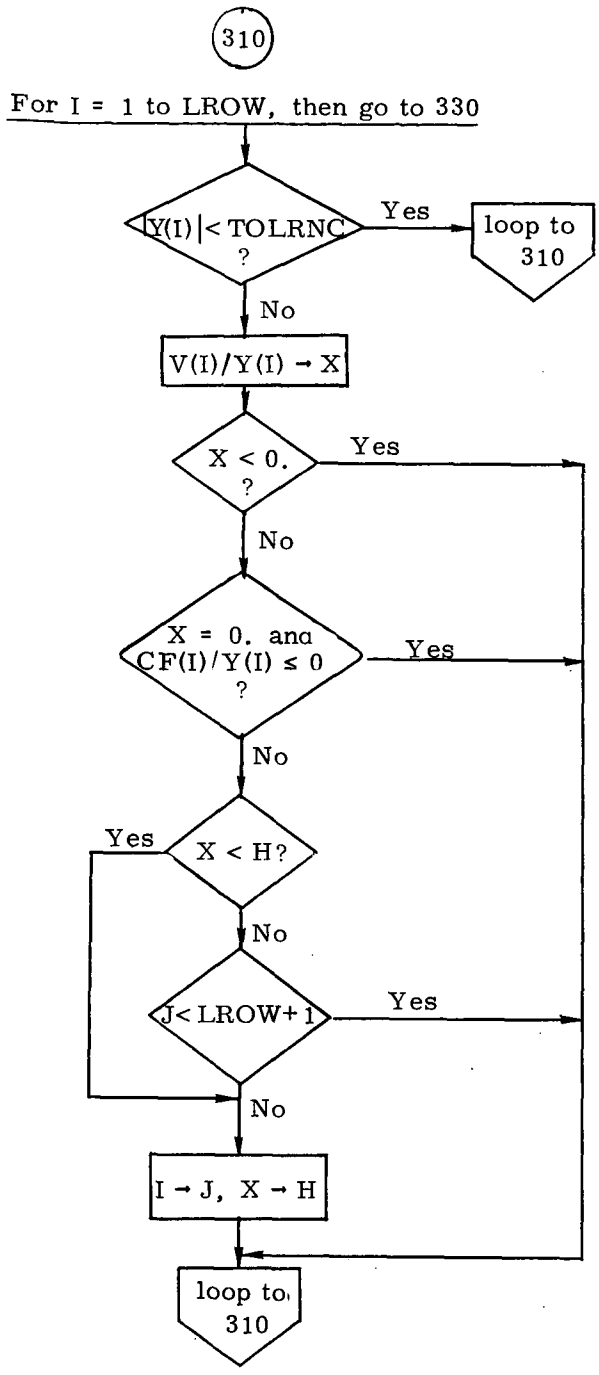
Greatest saving per second is H; corresponding jet is K.




If no jet was selected for inclusion, then exit.

$K^{th}$ jet is to be included. Y is $K^{th}$ jet acceleration vector described in current basis.

Fig. 2-6b Flow Chart of Linear Programming Subroutine, CCWLJS (Cont)




Choose $J^{th}$ basis member for replacement so that the new V(J) is ≥ 0., and no other V component changes sign.

Fig. 2-6b Flow Chart of Linear Programming Subroutine, CCWLJS (Cont)

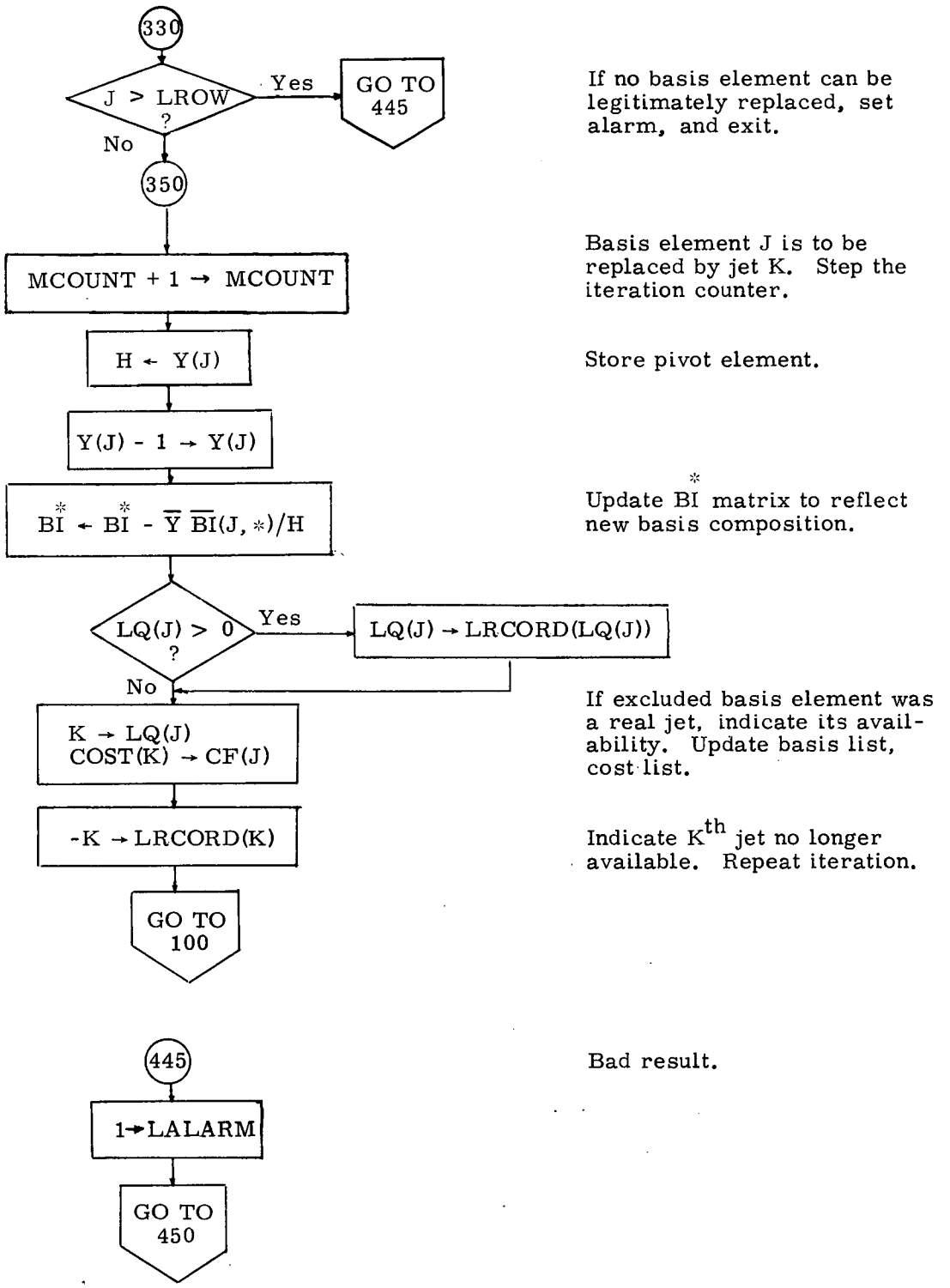
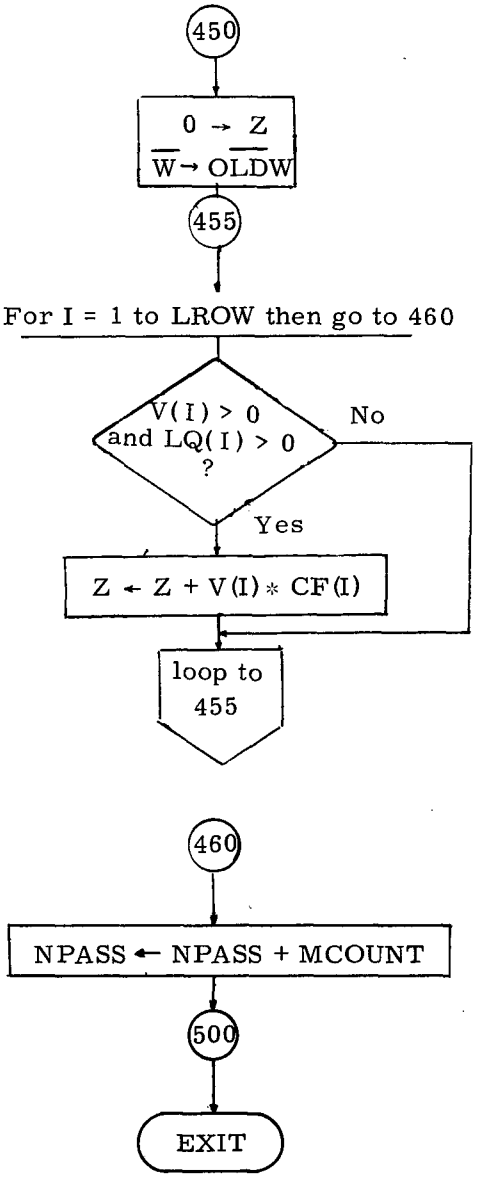



Fig. 2-6b Flow Chart of Linear Programming Subroutine, CCWLJS (Cont)




EXIT

Calculate cost of solution:
$Z = \overline{CF} \cdot \overline{V}$

Save $\overline{W}$ for comparison in next pass.

Negative firing times represent requests which cannot be met with currently non-failed jets.

NPASS reflects quality of $BI^*$ matrix.

Return to caller.

Fig. 2-6b Flow Chart of Linear Programming Subroutine, CCWLJS (Cont)

## 2.4 ACPS Jet Selection Logic for the ATP Configuration

A jet selection routine is presented in this section that is more restricted than the design of the preceding section. It is essentially a table look-up; the firing policy for each possible command, as well as the individual jets and the control authority corresponding to each jet cluster are pre-determined and stored for access by this on-line routine. The North American Rockwell ATP design for the ACPS, which has been presented in Fig. 2-4 and Tables 2-7 and 2-8, was assumed in the off-line computations.

The flow chart of Fig. 2-7a describes in detail the jet selection logic for the ATP configuration. Figure 2-7b defines the flowchart symbols. A discussion of some of the features of the logic follows.

General - To provide attitude control with no translational disturbances, jet couples must be used. However, pure jet couples exist in only one axis, the roll axis. Jet couples to be used for pitch control (e.g. 3 & 12) and yaw control (e.g. 5 & 7), also effect roll. To achieve pure pitch or yaw two thruster couples must be fired to null the roll disturbance; for example, 3 and 1 12 with 2 and 10 for pitch and 5 and 7 with 3 and 1 for yaw. Hence, couple control can require as many as four thrusters to fire whereas non-couple control may require only one thruster firing (e.g. yaw control). Additionally, the multiple thruster firings required for couple control produces large minimum impulse rate changes and these cause fast limit cycles and inefficient propellent usage. This logic has the option of attitude control with either coupled or non-coupled jets.

Nose versus Tail Thrusters - for Non-coupled Jet Operation - The logic allows the crew to select either nose or tail thrusters for control of each axis. This serves two purposes:

1) Comparing nose thrusters to tail thrusters, the nose thrusters have shorter roll moment arms, but longer pitch and yaw moment arms. Hence, the crew can select the preferable moment arm length for each axis.

2) This is a simple means of accounting for thruster failures. Should all the thrusters in any single location fail, control can still be maintained (Note: Individual failures in any jet location are accounted for by placing another jet of the same location in "common" with the location, see page 16 of the flowcharts).

Phase Plane Jet Selection Logic Interaction - (Non-coupled Operation) - Thrusters in locations 1 through 4, and 9 through 12, produce both pitch and roll torques. Hence, it takes two thrusters to provide either pure roll or pure

pitch. However, smaller minimum impulse rate changes are possible by firing only one thruster. Therefore, even if there is a phase plane command from only one axis (roll or pitch), the logic will attempt to do the latter (fire only one thruster) if the resulting disturbance in the "non-command" axis reduces the rate error (called $RR_X$ or $RR_Y$ in the logic).

Use of the tail thrusters for yaw control causes a roll disturbance; should the roll disturbance exceed a rate limit, the logic will consider the disturbance part of the phase plane command and cause corrective roll thrusters to fire.

Translations - No combination of thrusters will produce Y or Z translations with negligible rotational disturbances. Thrusters must be modulated to null rotational disturbances. This modulation could be performed either open loop in the jet selection logic, or closed loop through the phase plane logic (the crossing of a switching line) and jet selection logic. The latter method was chosen because it will not produce unnecessary modulation (thruster firings) during small Y or Z translations (i. e., when rotational disturbance is not long enough to cause switching line crossing).

During +X-translations, it is possible to control yaw by modulating the +X-thrusters; this option is available in the logic.

Jet Coupling and Logic Flow - With respect to thruster control actions, there are two basic types of thrusters:

1) Those that produce Y translation and yaw.

2) Those that produce Z translation and roll and pitch.

Sequentially, the logic first calculates commands for Y translation and yaw and then issues commands for Z translation, roll, and pitch control.

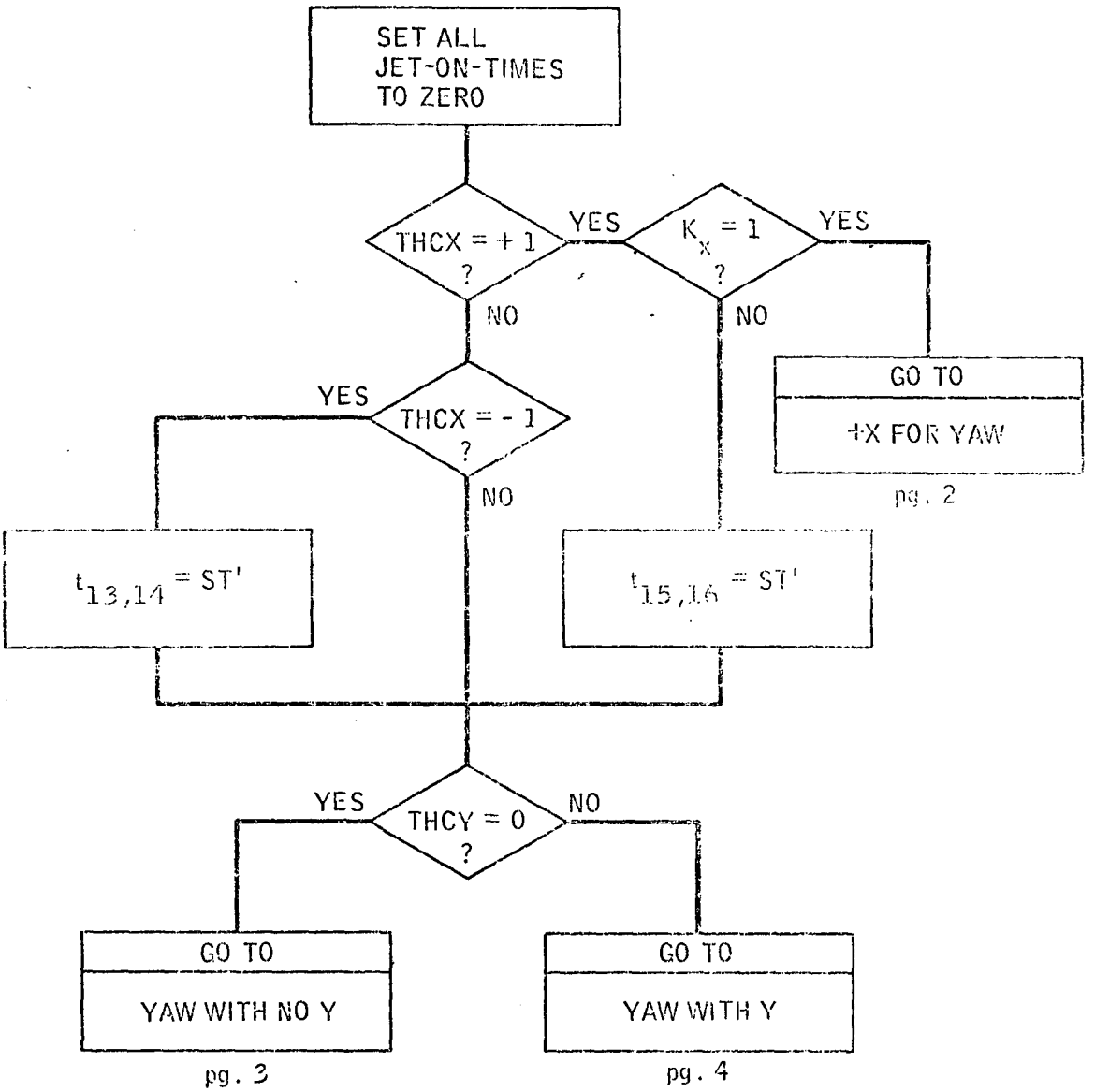
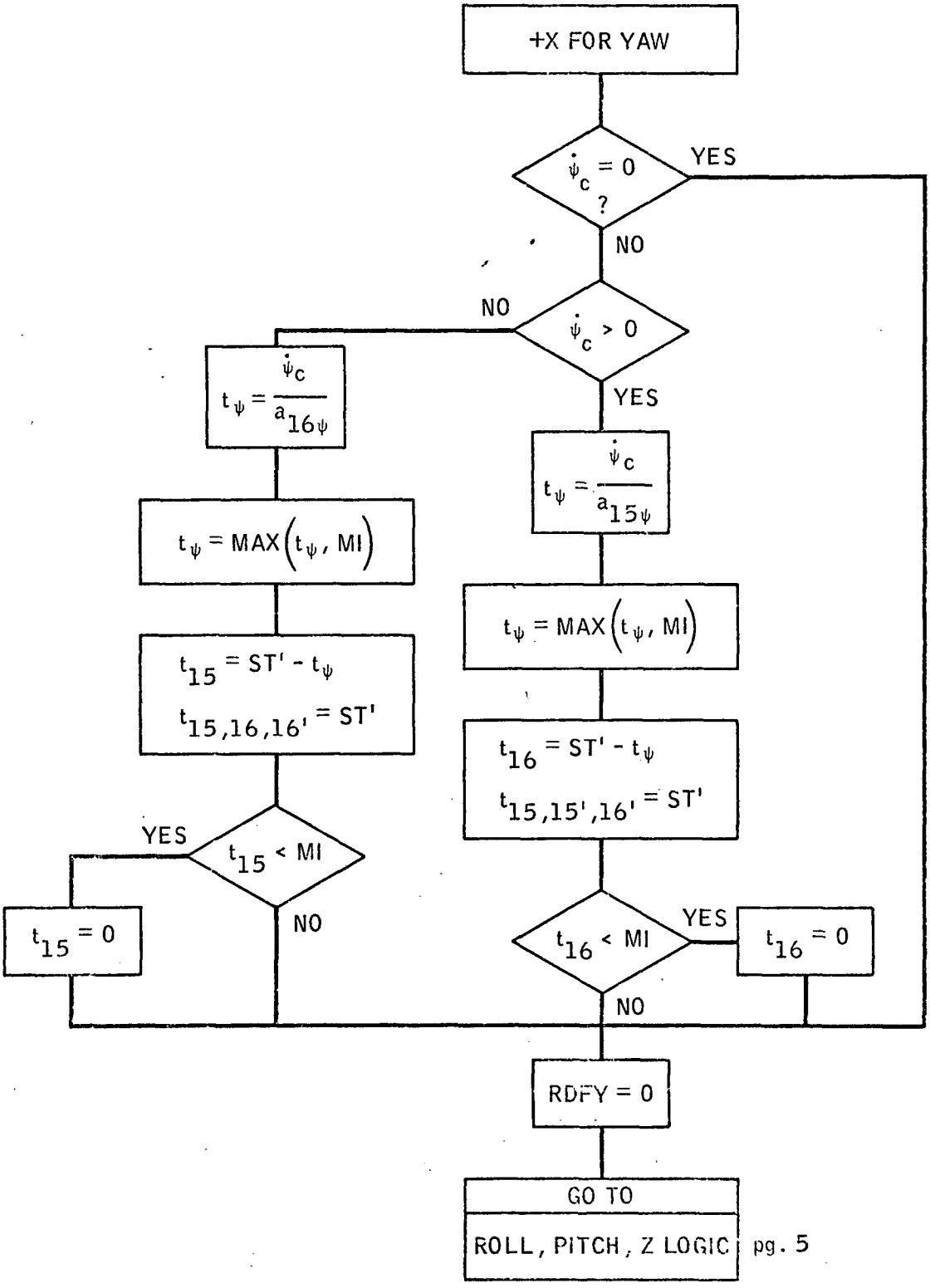
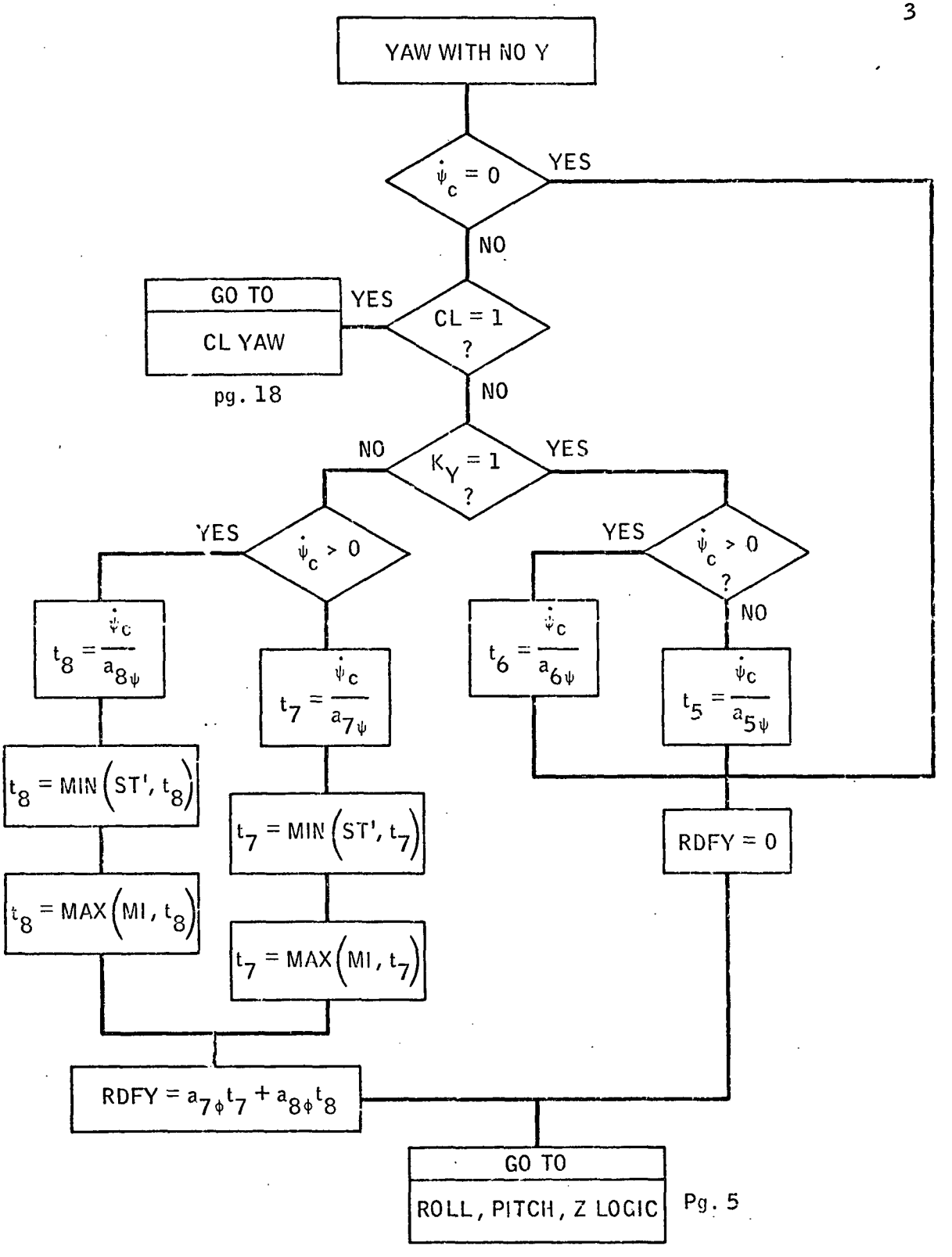
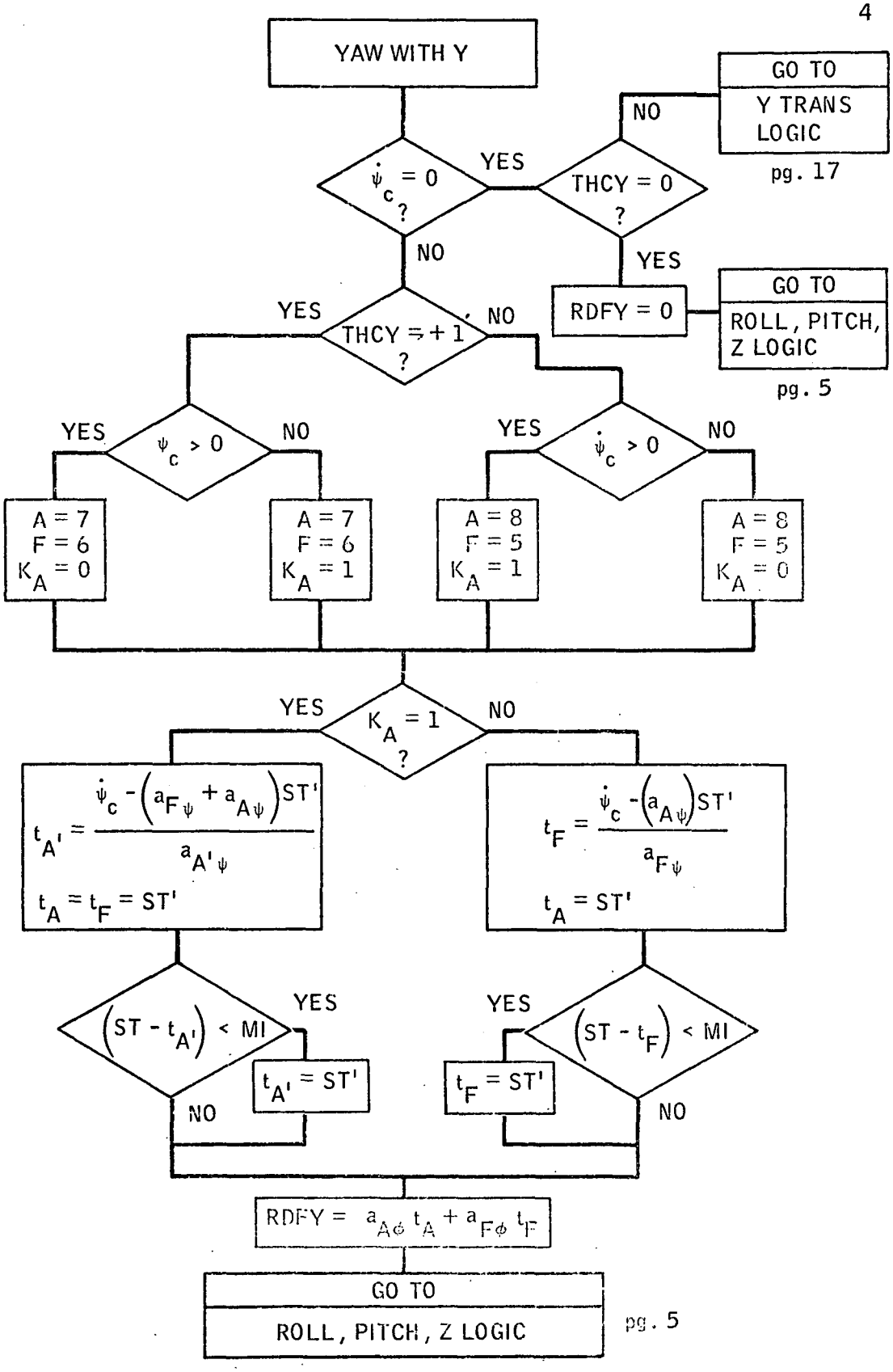
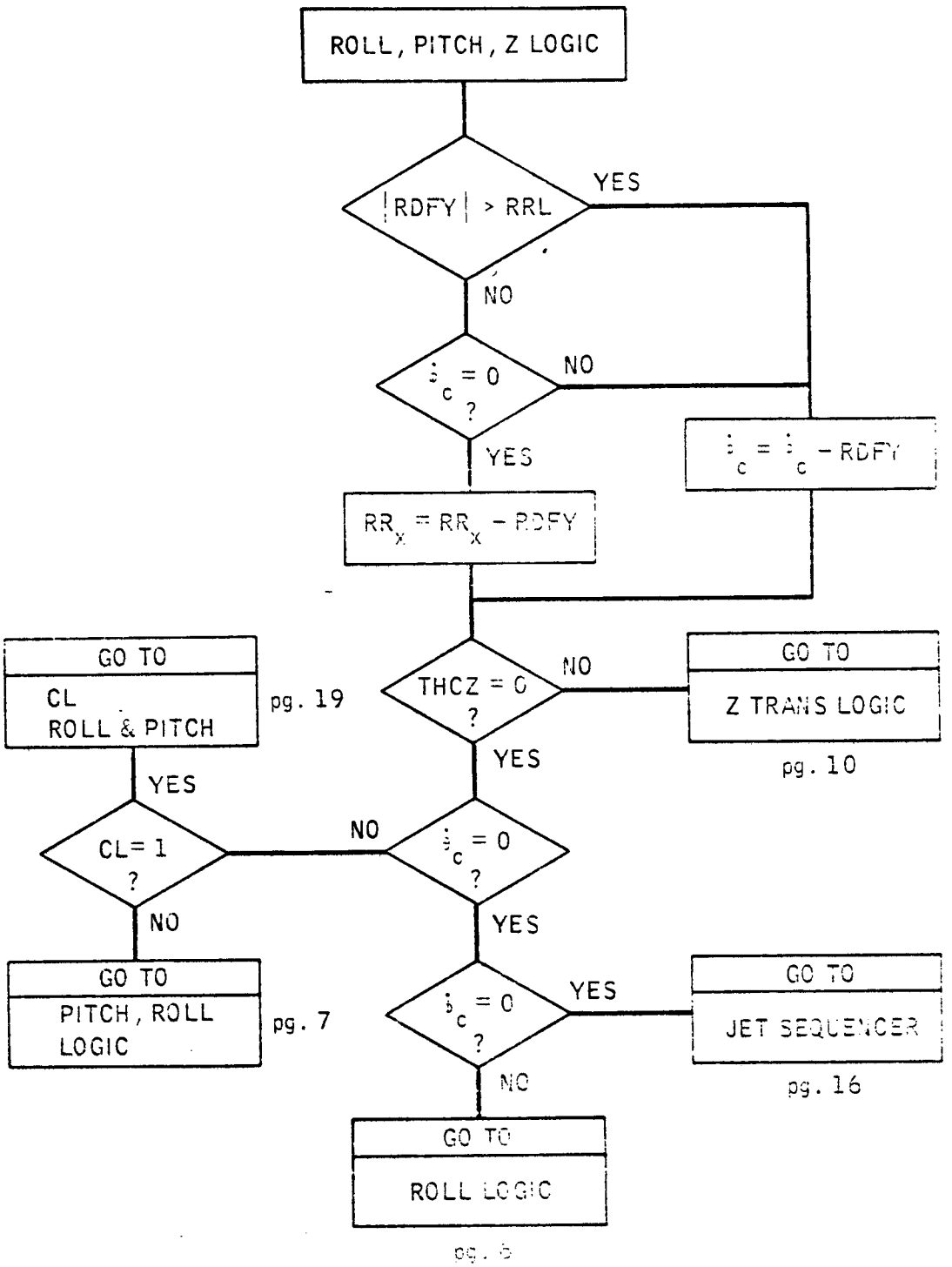


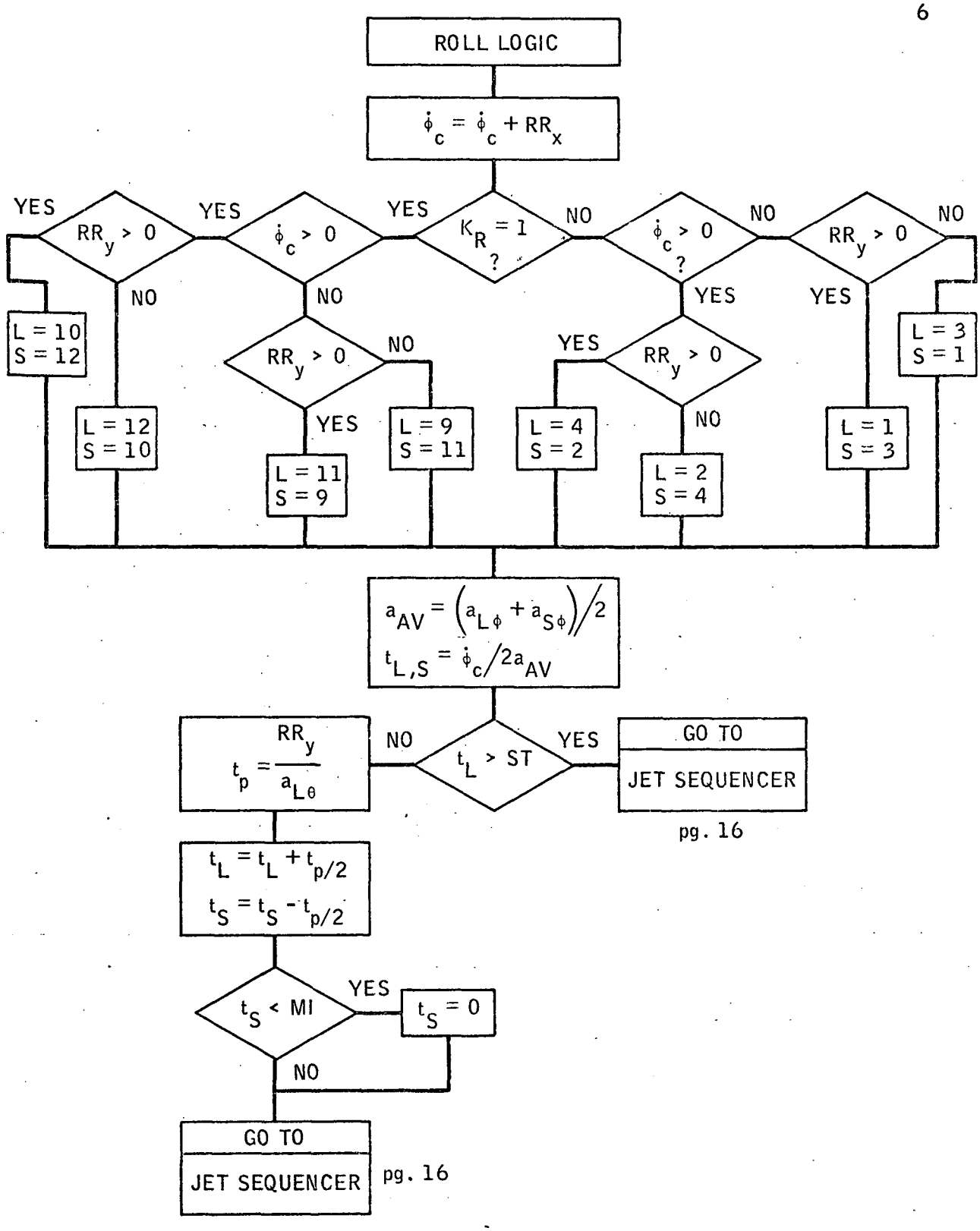

Fig. 2-7a Flow Chart Jet Selection Logic for ATP Configuration
(Flow Chart continues for 22 pages)

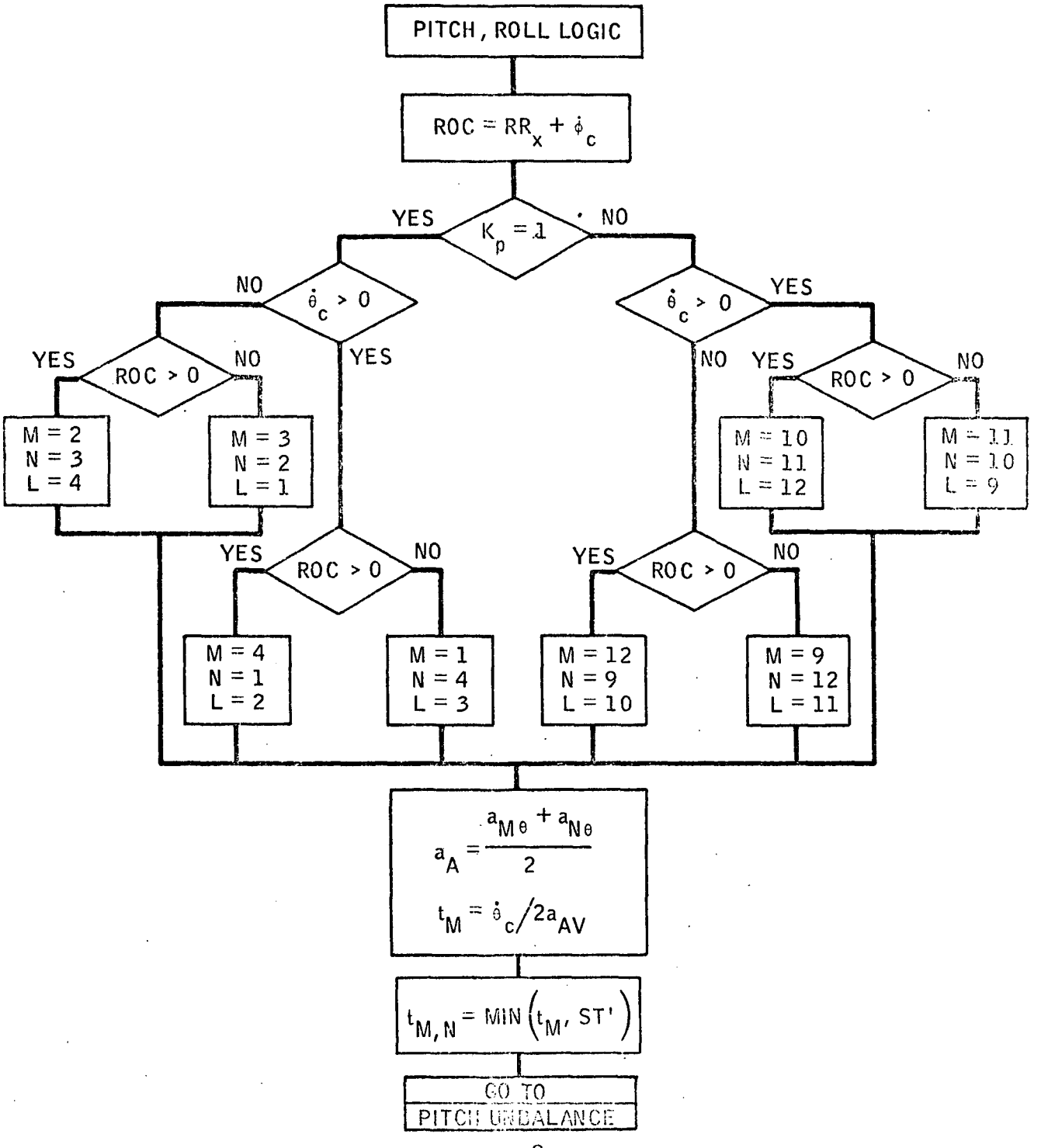

+X FOR YAW
$\dot{\psi}_c = 0$ ?
YES
NO
$\dot{\psi}_c > 0$
NO
YES
$t_\psi = \frac{\dot{\psi}_c}{a_{16\psi}}$
$t_\psi = \frac{\dot{\psi}_c}{a_{15\psi}}$
$t_\psi = MAX(t_\psi, MI)$
$t_\psi = MAX(t_\psi, MI)$
$t_{15} = ST' - t_\psi$
$t_{15,16,16'} = ST'$
$t_{16} = ST' - t_\psi$
$t_{15,15',16'} = ST'$
YES
$t_{15} < MI$
NO
$t_{15} = 0$
$t_{16} < MI$
YES
$t_{16} = 0$
NO
RDFY = 0
GO TO
ROLL, PITCH, Z LOGIC
pg. 5


YAW WITH NO Y
$\dot{\psi}_c = 0$
YES
NO
GO TO
CL YAW
pg. 18
YES
CL = 1 ?
NO
NO
$K_Y = 1$ ?
YES
YES
$\dot{\psi}_c > 0$
YES
$\dot{\psi}_c > 0$ ?
NO
$t_8 = \frac{\dot{\psi}_c}{a_{8\psi}}$
$t_7 = \frac{\dot{\psi}_c}{a_{7\psi}}$
$t_6 = \frac{\dot{\psi}_c}{a_{6\psi}}$
$t_5 = \frac{\dot{\psi}_c}{a_{5\psi}}$
$t_8 = MIN(ST', t_8)$
$t_7 = MIN(ST', t_7)$
RDFY = 0
$t_8 = MAX(MI, t_8)$
$t_7 = MAX(MI, t_7)$
$RDFY = a_{7\phi}t_7 + a_{8\phi}t_8$
GO TO
ROLL, PITCH, Z LOGIC
Pg. 5

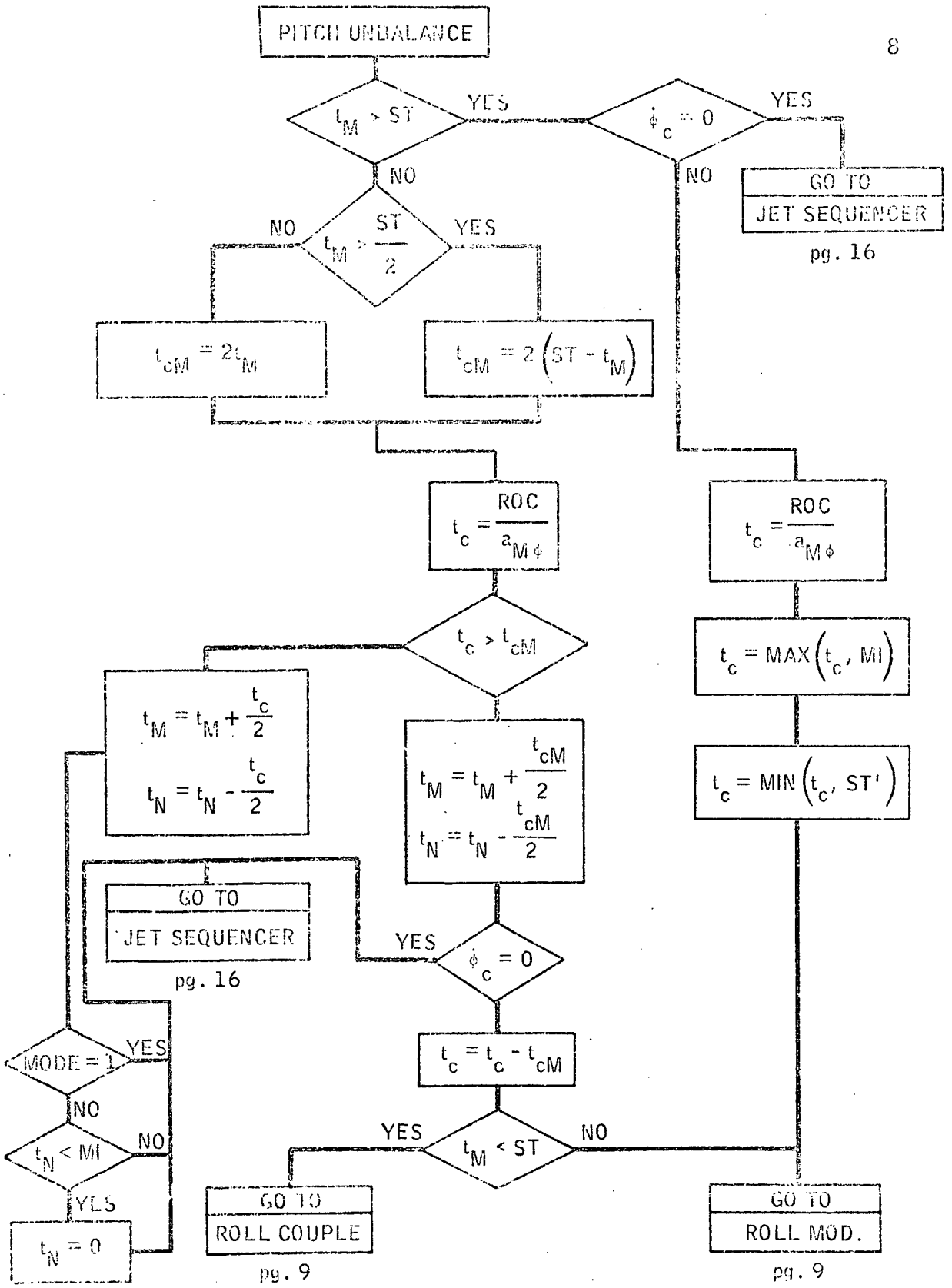

YAW WITH Y
$\dot{\psi}_c = 0$ ?
YES
NO
THCY = 0 ?
NO
GO TO
Y TRANS LOGIC
pg. 17
YES
RDFY = 0
GO TO
ROLL, PITCH, Z LOGIC
pg. 5
THCY = +1 ?
YES
NO
$\psi_c > 0$
YES
NO
$\dot{\psi}_c > 0$
YES
NO
A = 7
F = 6
$K_A = 0$
A = 7
F = 6
$K_A = 1$
A = 8
F = 5
$K_A = 1$
A = 8
F = 5
$K_A = 0$
$K_A = 1$ ?
YES
NO
$t_{A'} = \frac{\dot{\psi}_c - (a_{F\psi} + a_{A\psi})ST'}{a_{A'\psi}}$
$t_A = t_F = ST'$
$t_F = \frac{\dot{\psi}_c - (a_{A\psi})ST'}{a_{F\psi}}$
$t_A = ST'$
$(ST - t_{A'}) < MI$
YES
NO
$t_{A'} = ST'$
$(ST - t_F) < MI$
YES
NO
$t_F = ST'$
$RDFY = a_{A\phi} t_A + a_{F\phi} t_F$
GO TO
ROLL, PITCH, Z LOGIC
pg. 5

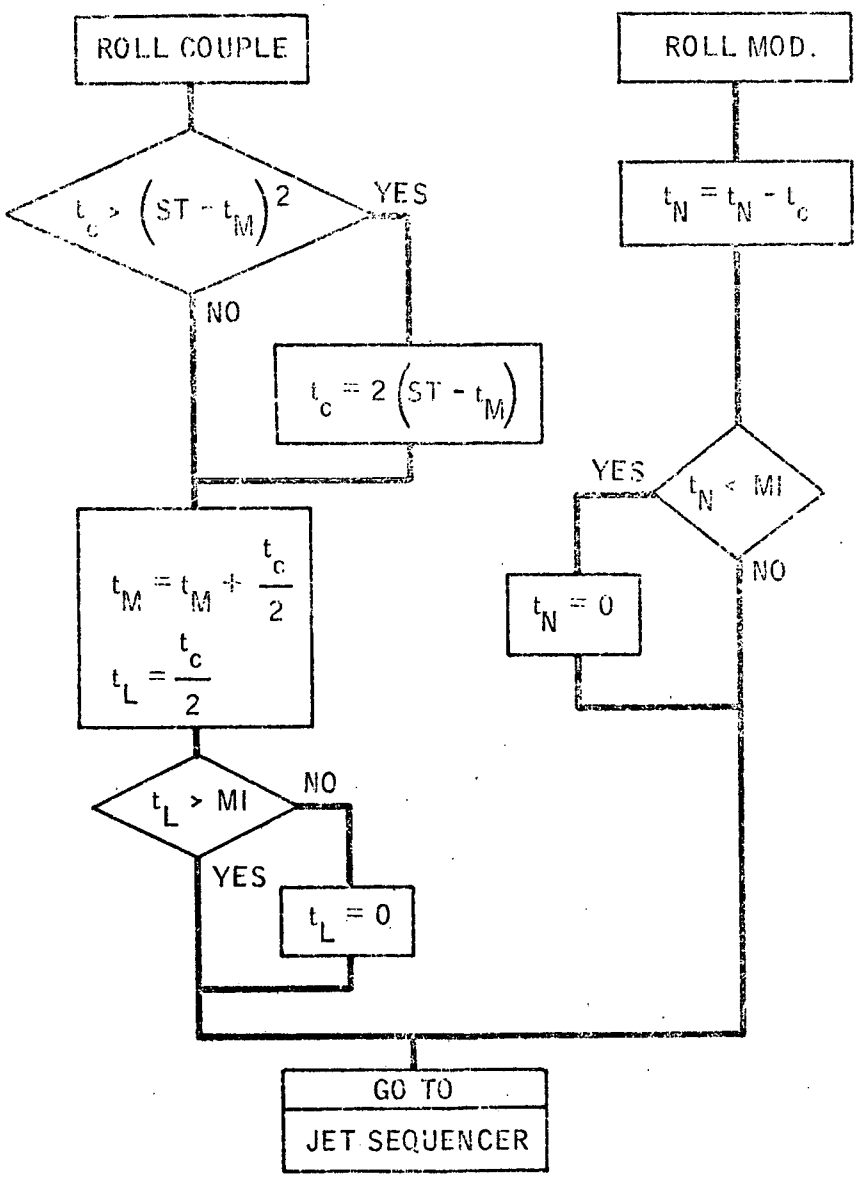

ROLL, PITCH, Z LOGIC
|RDFY| > RRL
YES
NO
$\dot{\theta}_c = 0$ ?
NO
YES
$\dot{\theta}_c = \dot{\theta}_c - RDFY$
$RR_x = RR_x - RDFY$
THCZ = 0 ?
NO
GO TO
Z TRANS LOGIC
pg. 10
YES
$\dot{\theta}_c = 0$ ?
NO
YES
GO TO
CL
ROLL & PITCH
pg. 19
YES
CL = 1 ?
NO
GO TO
PITCH, ROLL
LOGIC
pg. 7
$\dot{\theta}_c = 0$ ?
YES
GO TO
JET SEQUENCER
pg. 16
NO
GO TO
ROLL LOGIC
pg. 6


ROLL LOGIC
$\dot{\phi}_c = \dot{\phi}_c + RR_x$
$RR_y > 0$
$\dot{\phi}_c > 0$
$K_R = 1$ ?
$\dot{\phi}_c > 0$ ?
$RR_y > 0$
YES
NO
L = 10
S = 12
L = 12
S = 10
L = 11
S = 9
L = 9
S = 11
L = 4
S = 2
L = 2
S = 4
L = 1
S = 3
L = 3
S = 1
$a_{AV} = \left(a_{L\phi} + a_{S\phi}\right)/2$
$t_{L,S} = \dot{\phi}_c/2a_{AV}$
$t_p = \frac{RR_y}{a_{L\theta}}$
$t_L > ST$
GO TO
JET SEQUENCER
pg. 16
$t_L = t_L + t_{p/2}$
$t_S = t_S - t_{p/2}$
$t_S < MI$
$t_S = 0$
GO TO
JET SEQUENCER
pg. 16

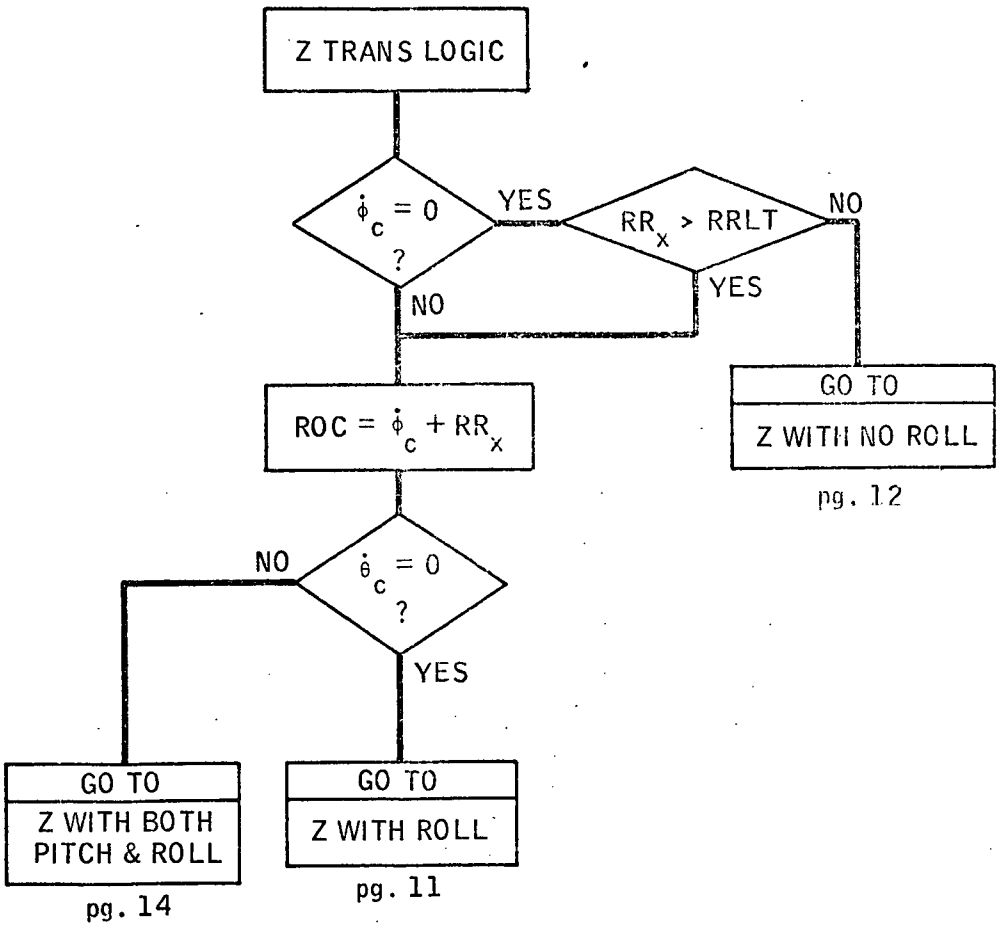

PITCH, ROLL LOGIC
ROC = RR_x + φ̇_c
K_p = 1
YES
NO
θ̇_c > 0
ROC > 0
M = 2
N = 3
L = 4
M = 3
N = 2
L = 1
M = 10
N = 11
L = 12
M = 11
N = 10
L = 9
M = 4
N = 1
L = 2
M = 1
N = 4
L = 3
M = 12
N = 9
L = 10
M = 9
N = 12
L = 11
a_A = (a_Mθ + a_Nθ)/2
t_M = θ̇_c/2a_AV
t_M,N = MIN (t_M, ST')
GO TO
PITCH UNBALANCE

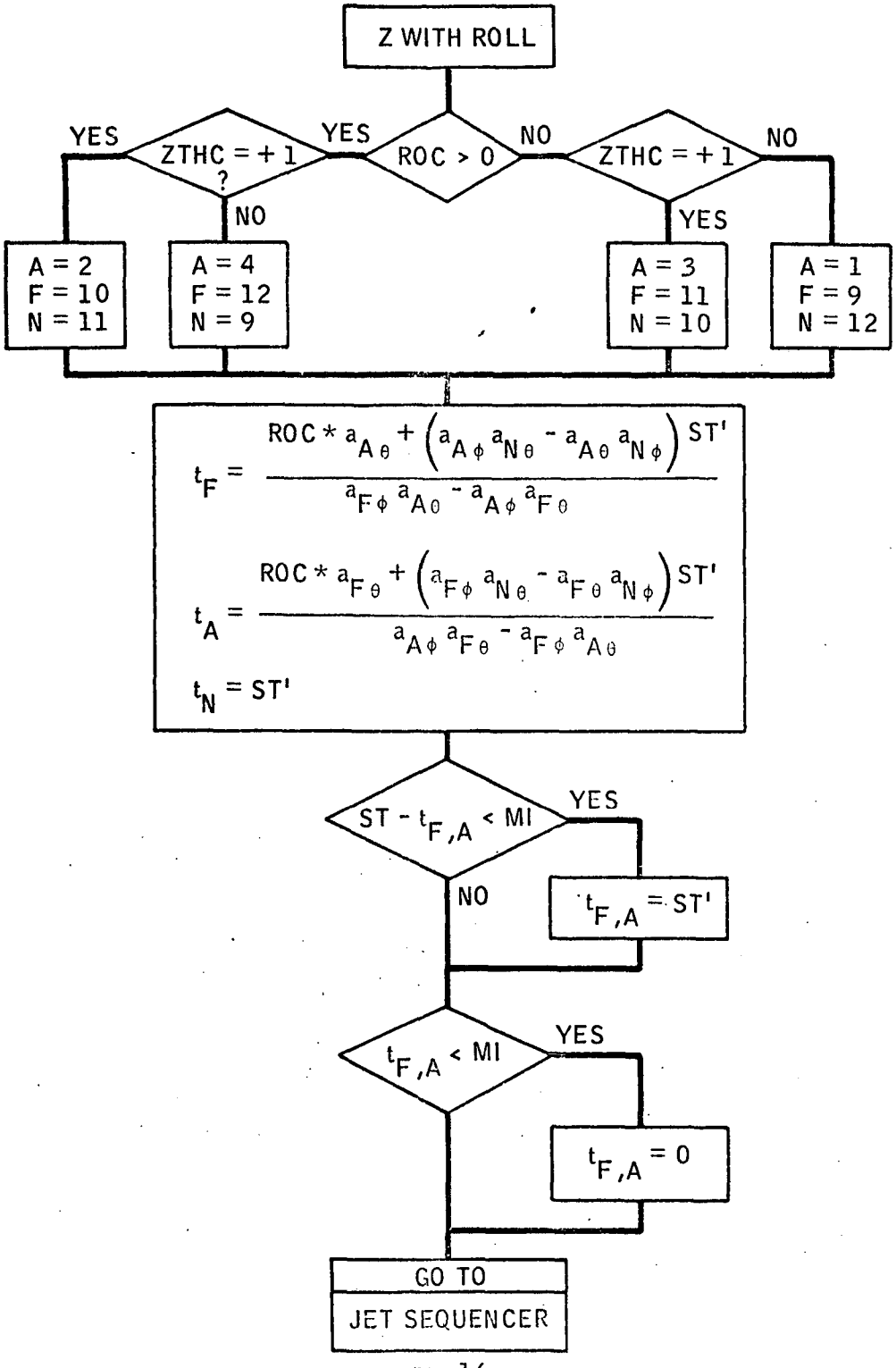
pg. 8


PITCH UNBALANCE
$t_M > ST$
YES
NO
$\dot{\phi}_c = 0$
YES
NO
GO TO
JET SEQUENCER
pg. 16
$t_M > \frac{ST}{2}$
NO
YES
$t_{cM} = 2t_M$
$t_{cM} = 2(ST - t_M)$
$t_c = \frac{ROC}{a_{M\phi}}$
$t_c = \frac{ROC}{a_{M\phi}}$
$t_c > t_{cM}$
$t_c = MAX(t_c, MI)$
$t_M = t_M + \frac{t_c}{2}$
$t_N = t_N - \frac{t_c}{2}$
$t_M = t_M + \frac{t_{cM}}{2}$
$t_N = t_N - \frac{t_{cM}}{2}$
$t_c = MIN(t_c, ST')$
GO TO
JET SEQUENCER
pg. 16
YES
$\dot{\phi}_c = 0$
MODE = 1
YES
NO
$t_c = t_c - t_{cM}$
$t_N < MI$
NO
YES
$t_M < ST$
YES
NO
$t_N = 0$
GO TO
ROLL COUPLE
pg. 9
GO TO
ROLL MOD.
pg. 9

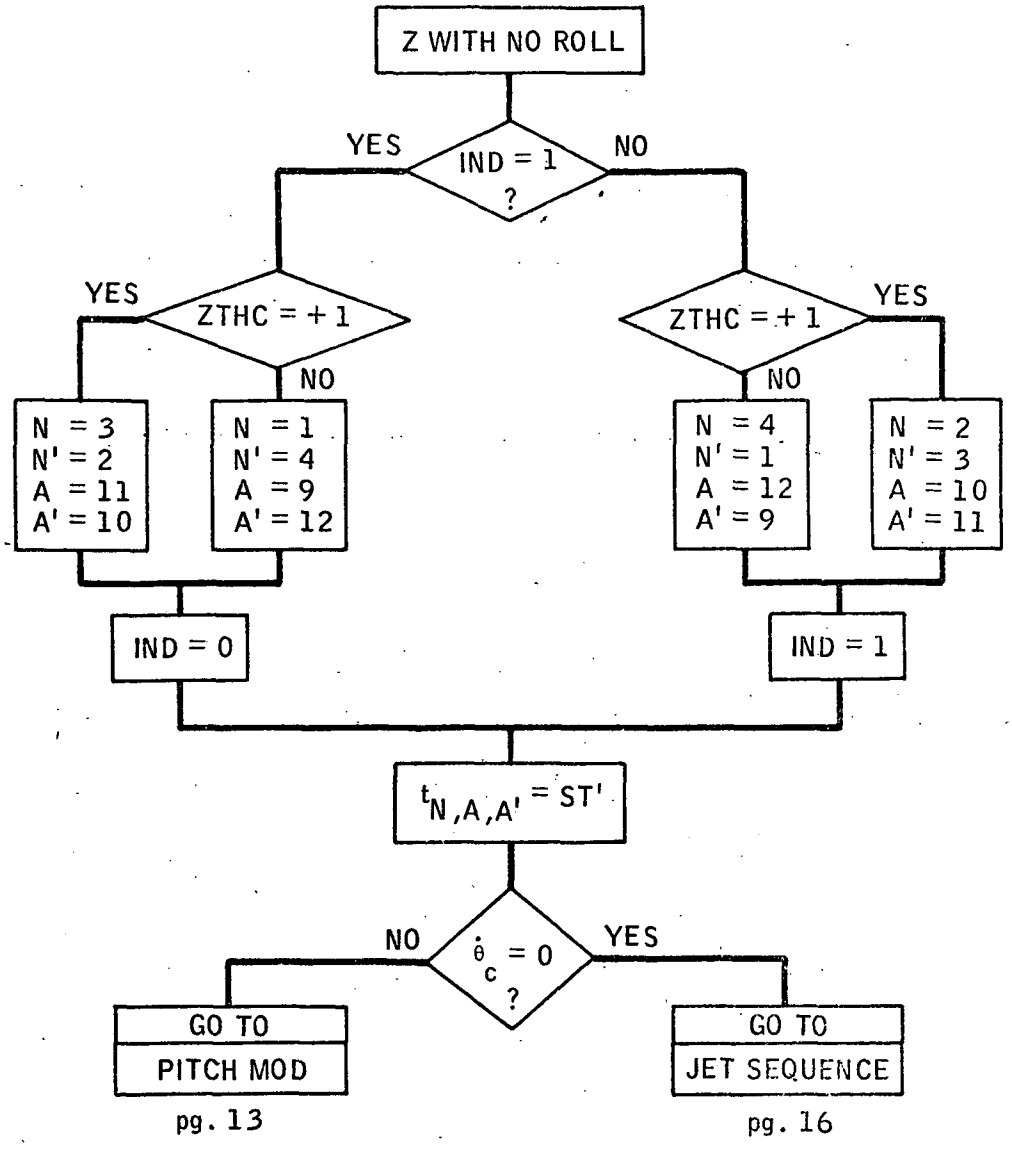

ROLL COUPLE
$t_c > (ST - t_M)^2$
YES
NO
$t_c = 2(ST - t_M)$
$t_M = t_M + \frac{t_c}{2}$
$t_L = \frac{t_c}{2}$
$t_L > MI$
NO
YES
$t_L = 0$
ROLL MOD.
$t_N = t_N - t_c$
$t_N < MI$
YES
NO
$t_N = 0$
GO TO
JET SEQUENCER

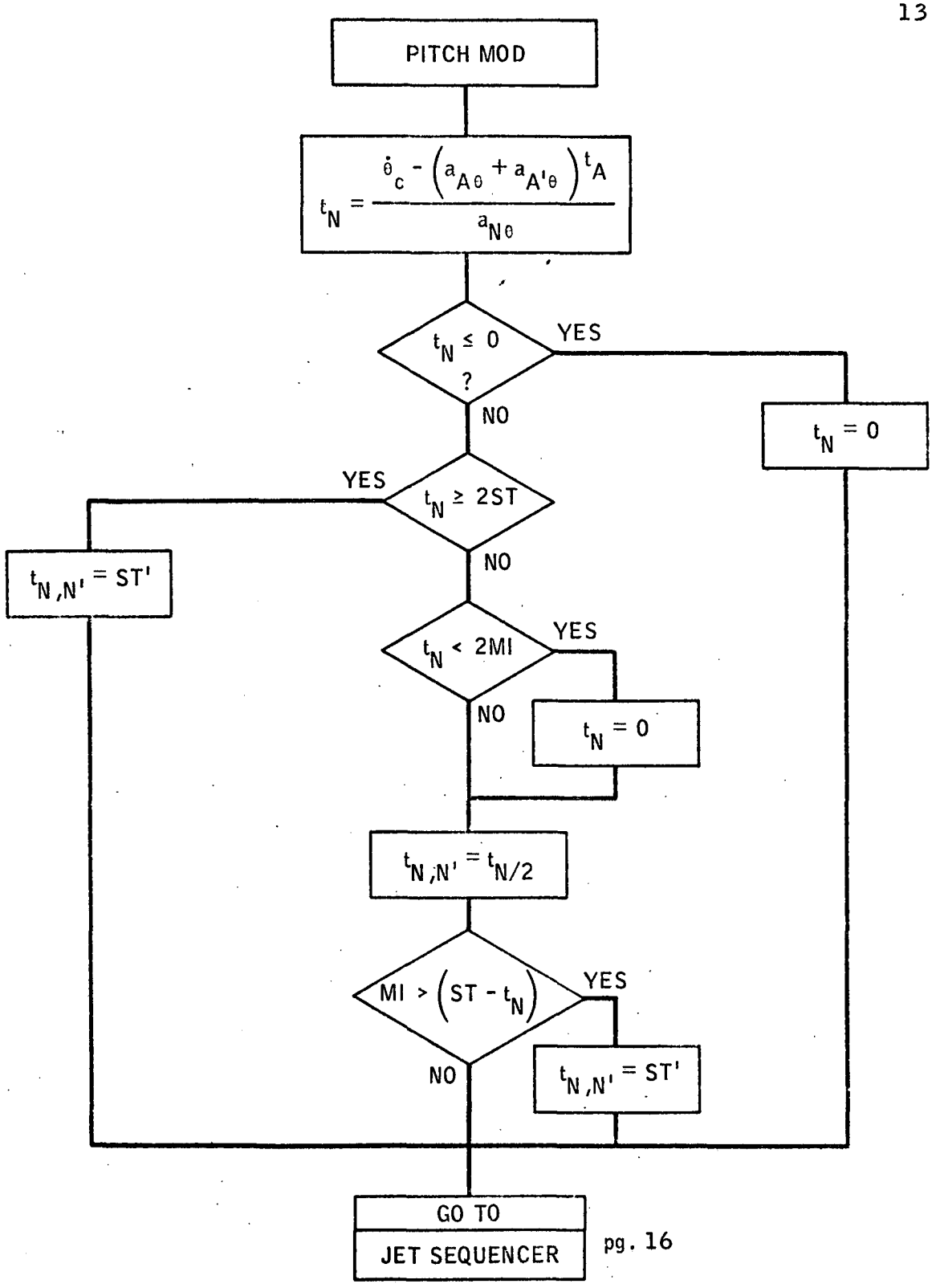
pg. 16


Z TRANS LOGIC
$\dot{\phi}_c = 0$ ?
YES
$RR_x > RRLT$
NO
YES
NO
GO TO
Z WITH NO ROLL
pg. 12
$ROC = \dot{\phi}_c + RR_x$
NO
$\dot{\theta}_c = 0$ ?
YES
GO TO
Z WITH BOTH PITCH & ROLL
pg. 14
GO TO
Z WITH ROLL
pg. 11

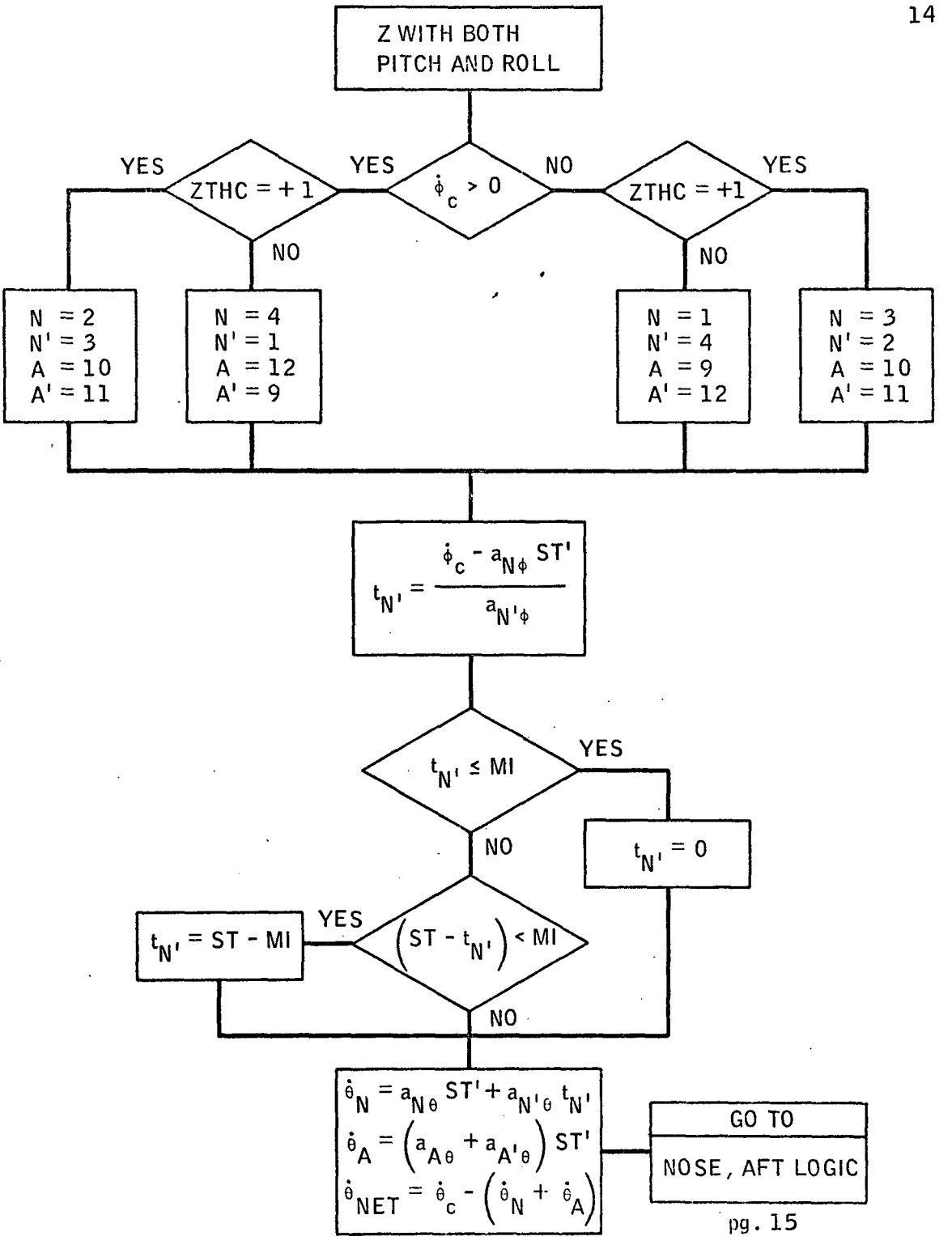

Z WITH ROLL
ROC > 0
YES
NO
ZTHC = + 1 ?
YES
NO
ZTHC = + 1
NO
YES
A = 2
F = 10
N = 11
A = 4
F = 12
N = 9
A = 3
F = 11
N = 10
A = 1
F = 9
N = 12
$t_F = \frac{ROC * a_{A\theta} + \left(a_{A\phi} a_{N\theta} - a_{A\theta} a_{N\phi}\right) ST'}{a_{F\phi} a_{A\theta} - a_{A\phi} a_{F\theta}}$
$t_A = \frac{ROC * a_{F\theta} + \left(a_{F\phi} a_{N\theta} - a_{F\theta} a_{N\phi}\right) ST'}{a_{A\phi} a_{F\theta} - a_{F\phi} a_{A\theta}}$
$t_N = ST'$
$ST - t_{F,A} < MI$
YES
NO
$t_{F,A} = ST'$
$t_{F,A} < MI$
YES
$t_{F,A} = 0$
GO TO
JET SEQUENCER

pg. 16

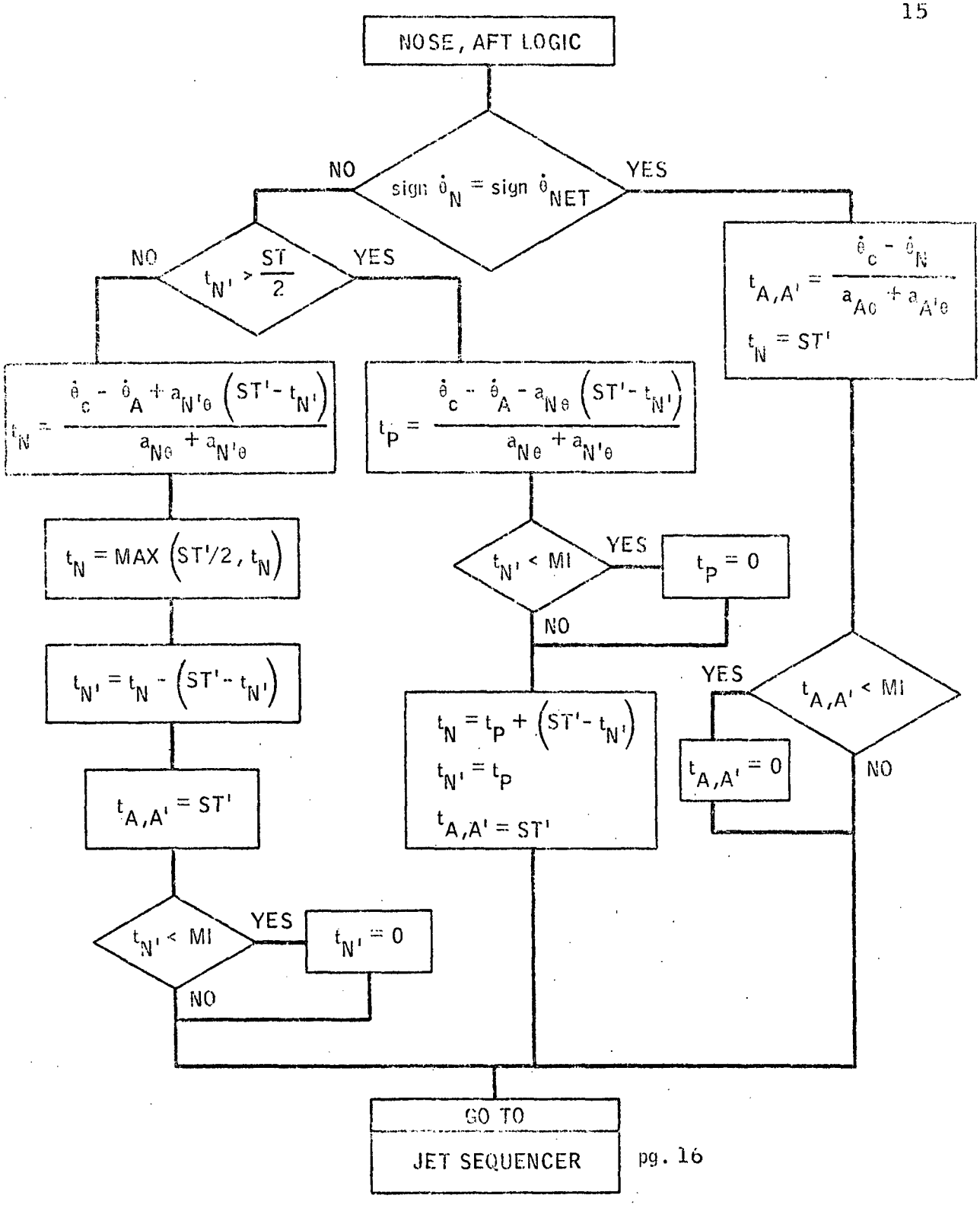

Z WITH NO ROLL
IND = 1 ?
YES
NO
ZTHC = + 1
YES
NO
ZTHC = + 1
YES
NO
N = 3
N' = 2
A = 11
A' = 10
N = 1
N' = 4
A = 9
A' = 12
N = 4
N' = 1
A = 12
A' = 9
N = 2
N' = 3
A = 10
A' = 11
IND = 0
IND = 1
$t_{N,A,A'} = ST'$
$\dot{\theta}_c = 0$ ?
NO
YES
GO TO
PITCH MOD
pg. 13
GO TO
JET SEQUENCE
pg. 16


PITCH MOD
$t_N = \frac{\dot{\theta}_c - \left(a_{A\theta} + a_{A'\theta}\right) t_A}{a_{N\theta}}$
$t_N \leq 0$ ?
YES
NO
$t_N = 0$
$t_N \geq 2ST$
YES
NO
$t_{N,N'} = ST'$
$t_N < 2MI$
YES
NO
$t_N = 0$
$t_{N,N'} = t_{N/2}$
$MI > \left(ST - t_N\right)$
YES
NO
$t_{N,N'} = ST'$
GO TO
JET SEQUENCER

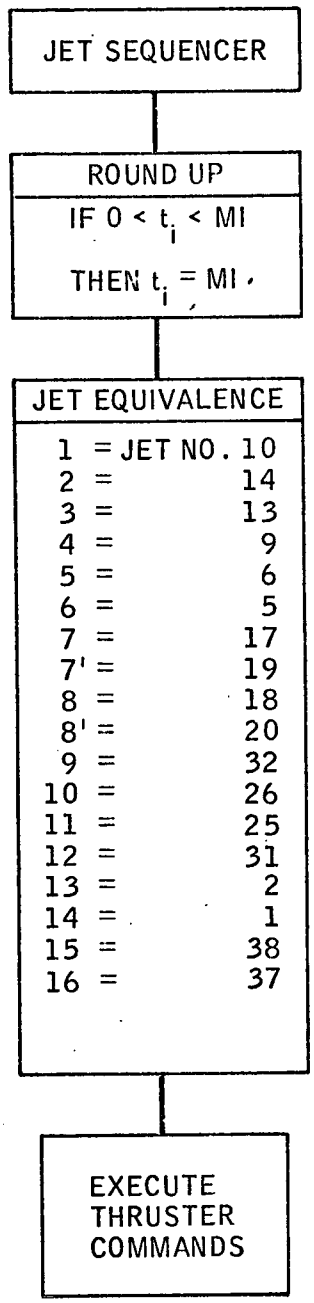
pg. 16

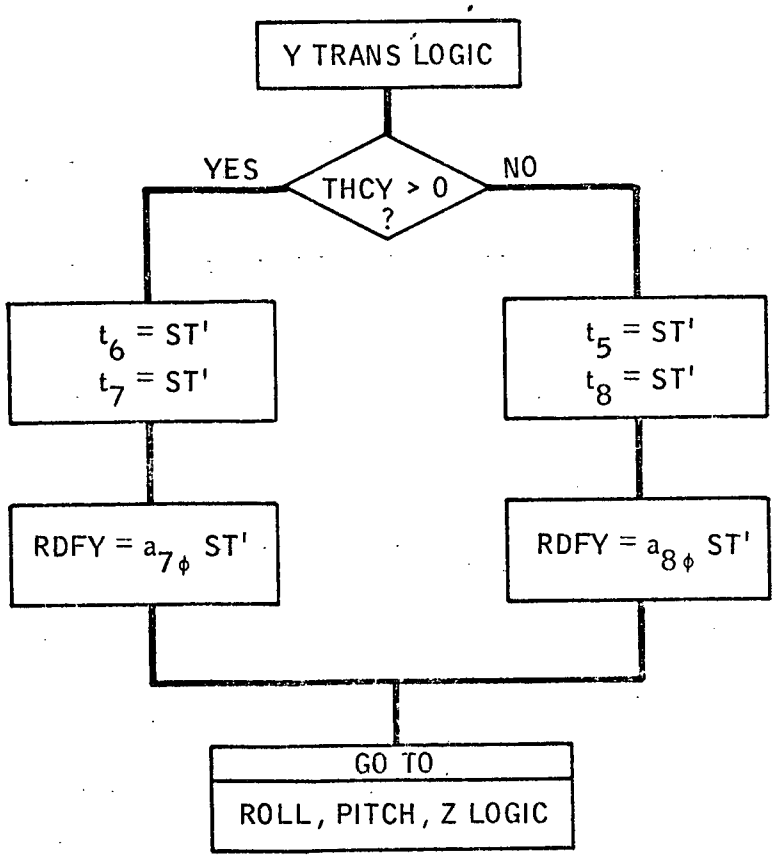

Z WITH BOTH PITCH AND ROLL
$\dot{\phi}_c > 0$
YES
NO
ZTHC = +1
ZTHC = +1
YES
NO
NO
YES
N = 2
N' = 3
A = 10
A' = 11
N = 4
N' = 1
A = 12
A' = 9
N = 1
N' = 4
A = 9
A' = 12
N = 3
N' = 2
A = 10
A' = 11
$t_{N'} = \frac{\dot{\phi}_c - a_{N\phi} ST'}{a_{N'\phi}}$
$t_{N'} \leq MI$
YES
NO
$t_{N'} = 0$
$\left(ST - t_{N'}\right) < MI$
YES
NO
$t_{N'} = ST - MI$
$\dot{\theta}_N = a_{N\theta} ST' + a_{N'\theta} t_{N'}$
$\dot{\theta}_A = \left(a_{A\theta} + a_{A'\theta}\right) ST'$
$\dot{\theta}_{NET} = \dot{\theta}_c - \left(\dot{\theta}_N + \dot{\theta}_A\right)$
GO TO
NOSE, AFT LOGIC

pg. 15

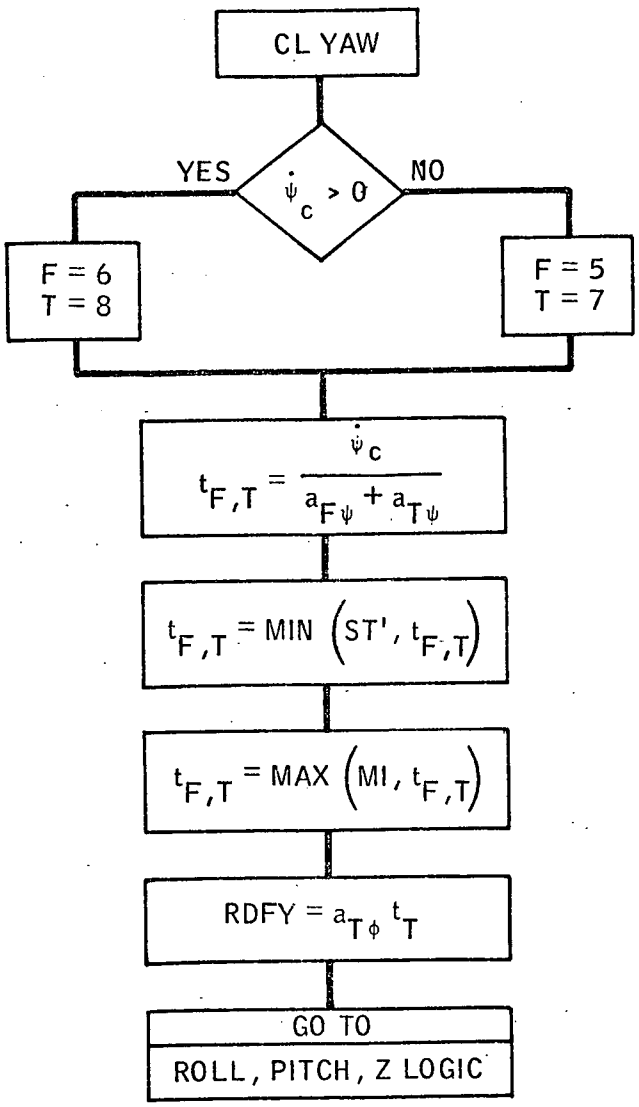

NOSE, AFT LOGIC
sign $\dot{\theta}_N$ = sign $\dot{\theta}_{NET}$
NO
YES
$t_{N'} > \frac{ST}{2}$
NO
YES
$t_{A,A'} = \frac{\dot{\theta}_c - \dot{\theta}_N}{a_{A\theta} + a_{A'\theta}}$
$t_N = ST'$
$t_N = \frac{\dot{\theta}_c - \dot{\theta}_A + a_{N'\theta}(ST' - t_{N'})}{a_{N\theta} + a_{N'\theta}}$
$t_P = \frac{\dot{\theta}_c - \dot{\theta}_A - a_{N\theta}(ST' - t_{N'})}{a_{N\theta} + a_{N'\theta}}$
$t_N = MAX(ST'/2, t_N)$
$t_{N'} < MI$
YES
$t_P = 0$
NO
$t_{N'} = t_N - (ST' - t_{N'})$
YES
$t_{A,A'} < MI$
$t_N = t_P + (ST' - t_{N'})$
$t_{N'} = t_P$
$t_{A,A'} = ST'$
$t_{A,A'} = 0$
NO
$t_{A,A'} = ST'$
$t_{N'} < MI$
YES
$t_{N'} = 0$
NO
GO TO
JET SEQUENCER

pg. 16

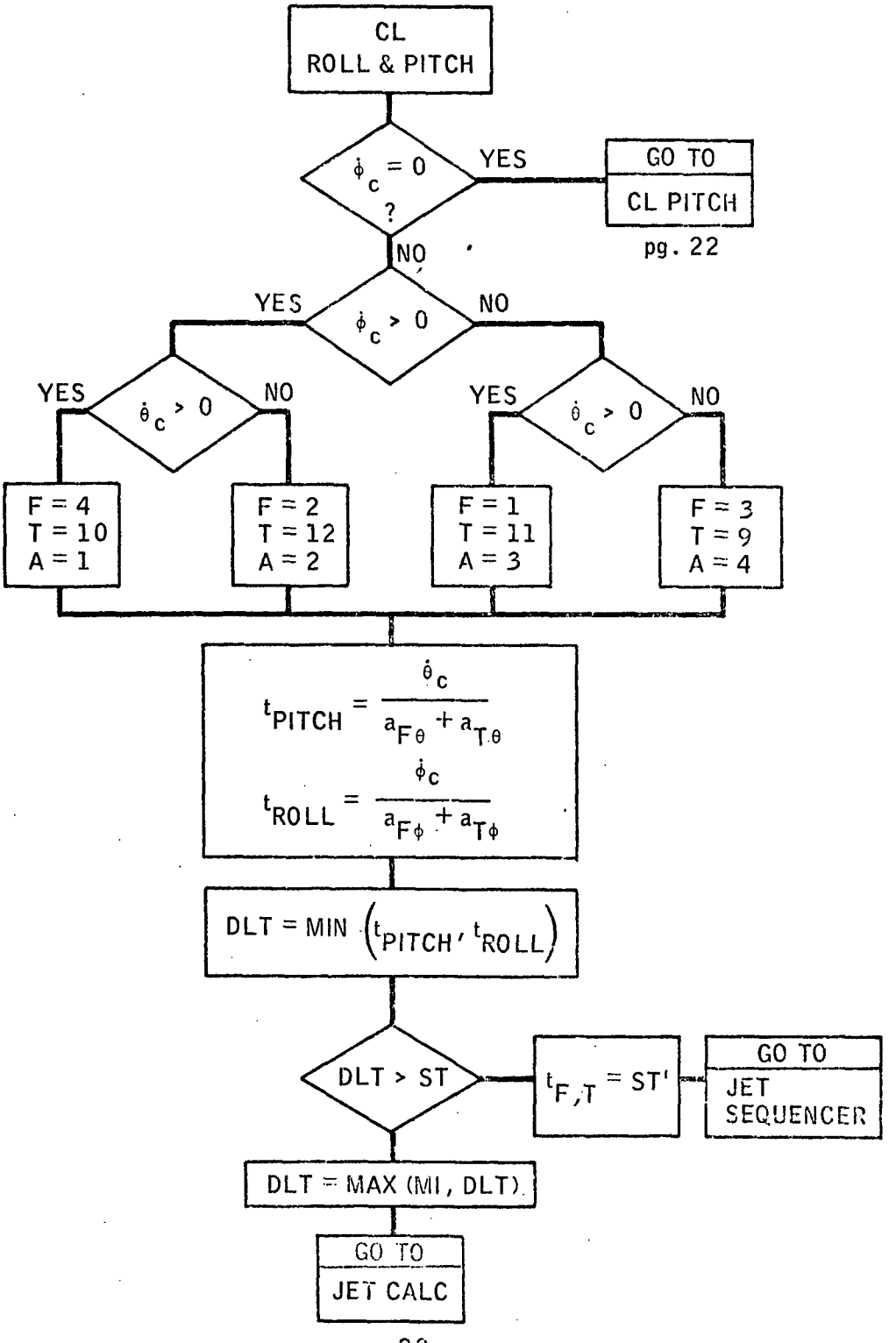

JET SEQUENCER
ROUND UP
IF 0 < $t_i$ < MI
THEN $t_i$ = MI
JET EQUIVALENCE
1 = JET NO. 10
2 = 14
3 = 13
4 = 9
5 = 6
6 = 5
7 = 17
7' = 19
8 = 18
8' = 20
9 = 32
10 = 26
11 = 25
12 = 31
13 = 2
14 = 1
15 = 38
16 = 37
EXECUTE THRUSTER COMMANDS

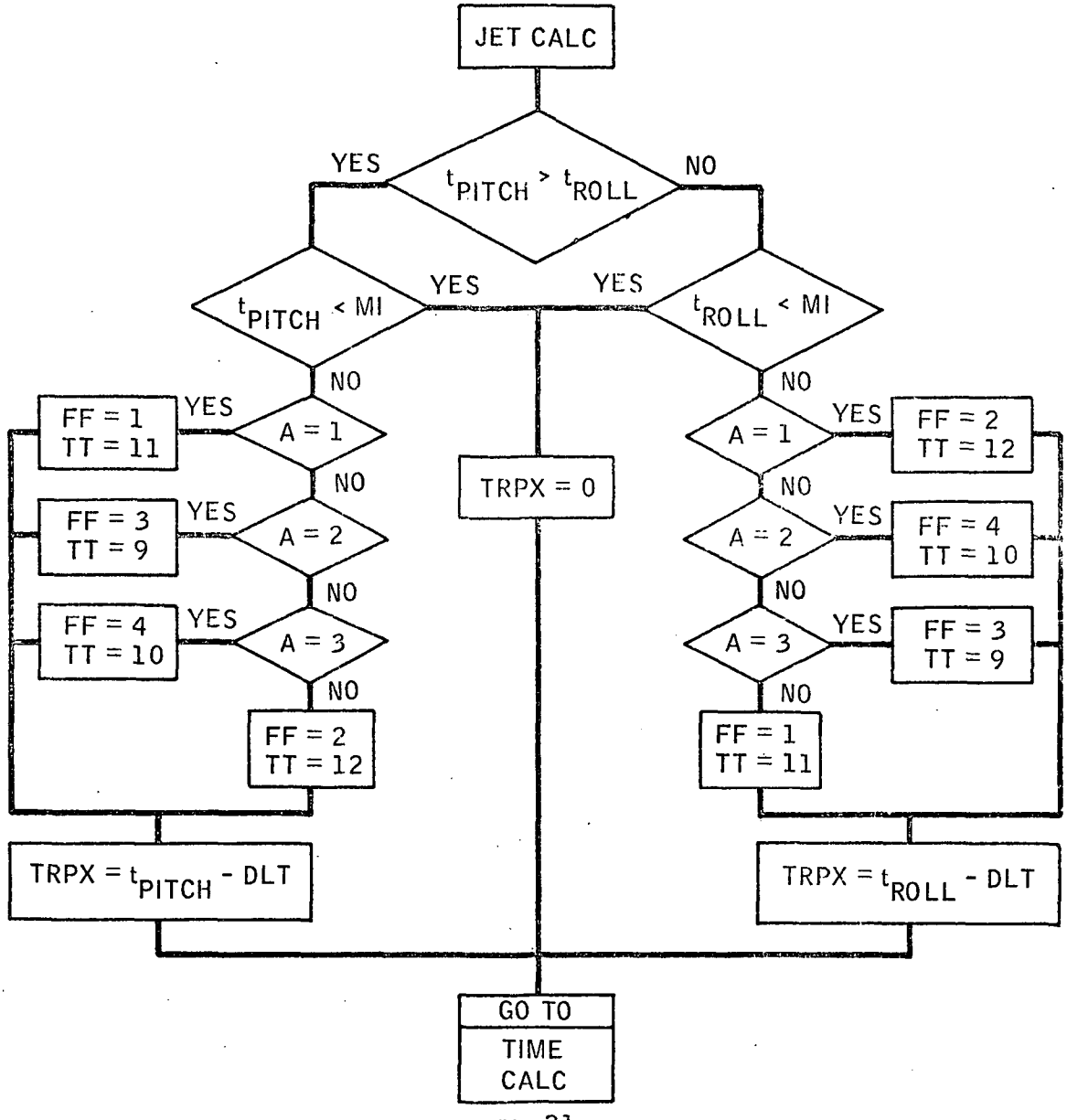

Y TRANS LOGIC
YES
THCY > 0 ?
NO
$t_6$ = ST'
$t_7$ = ST'
$t_5$ = ST'
$t_8$ = ST'
RDFY = $a_{7\phi}$ ST'
RDFY = $a_{8\phi}$ ST'
GO TO
ROLL, PITCH, Z LOGIC

pg. 5

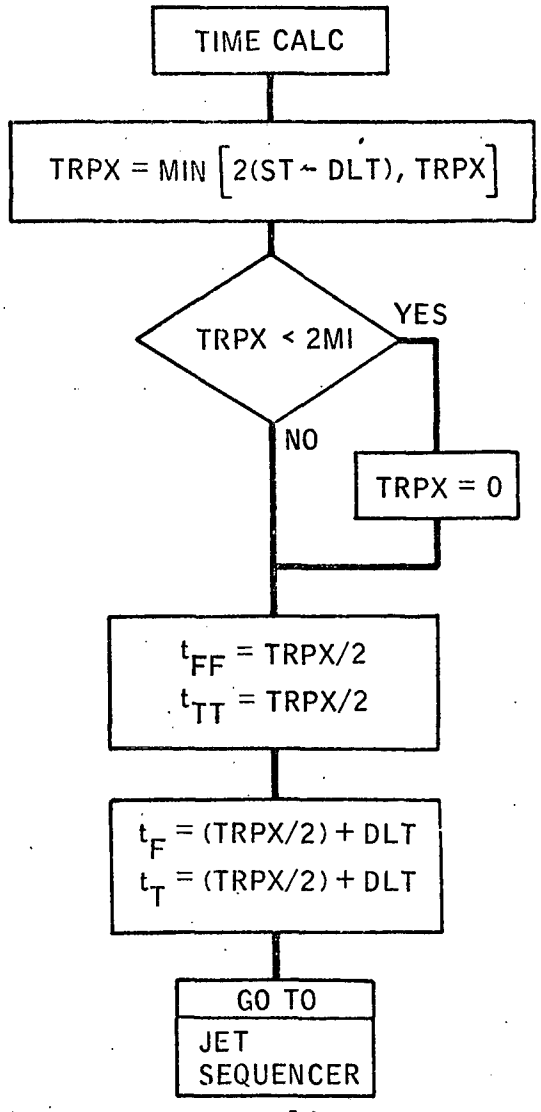

CL YAW
YES
$\dot{\psi}_c > 0$
NO
F = 6
T = 8
F = 5
T = 7
$t_{F,T} = \frac{\dot{\psi}_c}{a_{F\psi} + a_{T\psi}}$
$t_{F,T} = \text{MIN}\left(ST', t_{F,T}\right)$
$t_{F,T} = \text{MAX}\left(MI, t_{F,T}\right)$
$RDFY = a_{T\phi}\, t_T$
GO TO
ROLL, PITCH, Z LOGIC

pg. 5

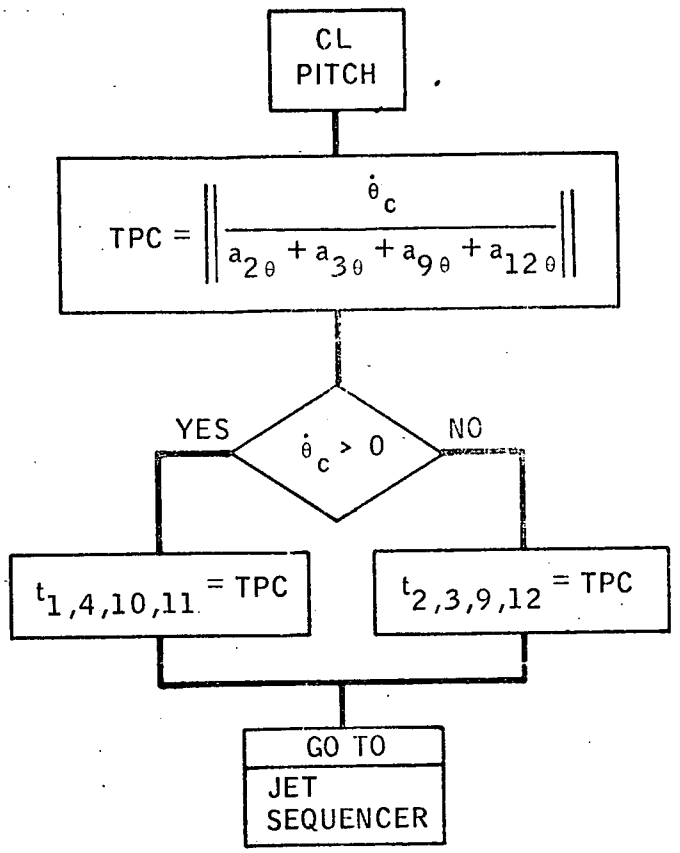

CL
ROLL & PITCH
$\dot{\phi}_c = 0$ ?
YES
GO TO
CL PITCH
pg. 22
NO
$\dot{\phi}_c > 0$
YES
NO
$\dot{\theta}_c > 0$
YES
NO
$\dot{\theta}_c > 0$
YES
NO
F = 4
T = 10
A = 1
F = 2
T = 12
A = 2
F = 1
T = 11
A = 3
F = 3
T = 9
A = 4
$t_{PITCH} = \frac{\dot{\theta}_c}{a_{F\theta} + a_{T\theta}}$
$t_{ROLL} = \frac{\dot{\phi}_c}{a_{F\phi} + a_{T\phi}}$
DLT = MIN $(t_{PITCH}, t_{ROLL})$
DLT > ST
$t_{F,T}$ = ST'
GO TO
JET
SEQUENCER
DLT = MAX (MI, DLT)
GO TO
JET CALC
pg. 20


JET CALC
$t_{PITCH} > t_{ROLL}$
YES
NO
$t_{PITCH} < MI$
YES
$t_{ROLL} < MI$
YES
NO
NO
FF = 1
TT = 11
YES
A = 1
NO
FF = 3
TT = 9
YES
A = 2
NO
FF = 4
TT = 10
YES
A = 3
NO
FF = 2
TT = 12
TRPX = 0
A = 1
YES
FF = 2
TT = 12
NO
A = 2
YES
FF = 4
TT = 10
NO
A = 3
YES
FF = 3
TT = 9
NO
FF = 1
TT = 11
TRPX = $t_{PITCH}$ - DLT
TRPX = $t_{ROLL}$ - DLT
GO TO
TIME
CALC

pg. 21


TIME CALC
TRPX = MIN [2(ST - DLT), TRPX]
TRPX < 2MI
YES
NO
TRPX = 0
$t_{FF}$ = TRPX/2
$t_{TT}$ = TRPX/2
$t_F$ = (TRPX/2) + DLT
$t_T$ = (TRPX/2) + DLT
GO TO
JET SEQUENCER

pg. 16


CL
PITCH
$TPC = \left\| \frac{\dot{\theta}_c}{a_{2\theta} + a_{3\theta} + a_{9\theta} + a_{12\theta}} \right\|$
YES
$\dot{\theta}_c > 0$
NO
$t_{1,4,10,11} = TPC$
$t_{2,3,9,12} = TPC$
GO TO
JET
SEQUENCER

pg. 16

| | |
|---|---|
| THCX, Y, Z | Commands for translation, either positive, none, or negative, depending on +1, 0, or -1 |
| $\dot{\phi}_C, \dot{\theta}_C, \dot{\psi}_C$ | Commanded rate changes from switching logic. |
| $a_i$, $\phi, \theta$, or $\psi$ | Angular Acceleration for $i^{TH}$ jet for either roll, pitch, or yaw. |
| $K_R, K_P, K_Y$ | Value of 1 (not 0) signifies request for long moment arm thrusters |
| J, K | Indices for jets |
| MI | Minimum impulse time |
| RDFY | Roll rate disturbance for yaw and/or Y translation firings |
| RRL | Roll rate limit before RDFY is added directly to $\dot{\phi}_C$ |
| $RR_Y$ | Roll rate error from phase plane logic when $\dot{\phi}_C$ is zero |
| PI | Index indicating presence (1) or absence (0) of roll command |
| ST | Sample time |
| M, N | Jet indices |
| $t_{CM}$ | Maximum roll compensation time available by unbalancing pitch firing times |
| E | Small number to account for round off in comparisions (say $10^{-4}$) |
| $t_C$ | One jet roll correction time |
| ROC | Net roll command, i.e., RR + $\dot{\phi}_C$ |
| RRLT | Roll rate limit before $RR_X$ is added to $\dot{\phi}_C$ during translation |
| $ST^1$ | Sample time plus E |
| $RR_Y$ | Pitch rate error from phase plane logic when $\dot{\theta}_C$ is zero |
| CL | Couple option |

Fig. 2-7b Definition of Symbols for Jet Selection Logic

SECTION 3

# STATE ESTIMATOR (ENTRY PHASE)

by

J. Edwin Jones

## 3.1 Introduction

The function of the state estimator is to obtain estimates of controller state variables in an uncertain environment. This uncertainty arises from noise in the measurement process and noise driving the states directly, e.g., due to wind gusts.

The state estimator design must meet certain prescribed requirements in the face of externally-imposed constraints. These requirements, many of which are not yet specified quantitatively, pertain to the degree of accuracy and smoothness in the estimated state variables, the time delay between reading the sensors and issuing computed control commands, and the effective phase lag associated with filter dynamics. External constraints are imposed by fixed sensor hardware and configuration characteristics and computer timing and sizing limitations.

Meeting all requirements while satisfying all constraints may necessitate design compromises. The design should be comprehensive to ensure prescribed accuracy by making best use of available information, but design sophistication is limited by computer constraints. Noisy inputs may require heavy attenuation to ensure prescribed smoothness, but the low gains associated with heavy attenuation must not cause excessive phase lag or excessive sensitivity to modeling errors. These compromises are central in the development of the design presented below.

The current state estimator is designed specifically for the entry flight phase (and some of the parameters used by the estimator are defined in the entry control description, Section 5). The structure used, however, is applicable to any flight phase. Extension of the state estimator to other flight phases is under study.

## 3.2 General Description

The design presented for the state estimator is linear and therefore implies approximation, namely linearization of the state dynamics about a set of operating points and linear modeling of control effector and sensor operation. The linear assumptions facilitate both design and analysis. Nonlinear effects will be noted as development continues and possibly included.

The design is cast in the state-space formulation and uses more than one measurement to produce each state-variable estimate (multi-input, multi-output). It is comprehensive in the sense that it uses all pertinent information. However, approximations are made to decrease computer requirements; namely, only a reduced number of state-variables are estimated, and decoupling of longitudinal and lateral dynamics is assumed. An important feature of the design is that control effects are fed into the estimator's state-extrapolation step, keeping the estimator response fast even while noisy inputs are being heavily attenuated.

Time delay between reading sensors and issuing control commands is minimized in the sense that it provides for incorporating the current measurements and extrapolating the state[†] in two separate steps with the latter performed after the commands are generated. Furthermore, the slow-varying filter parameters are updated separately at a slower repetition rate. Details of these computer timing considerations are given in Implementation, Subsection 3.3.

The estimator gains are constant with respect to time at any particular flight condition; however, their on-board computation is dependent on the current estimates of dynamic pressure and Mach number. When the dynamic pressure is below a prescribed level,[††] the estimator gains are set equal to predetermined fixed values. When dynamic pressure is above this level, the gains are interpolated from predetermined tabulated values with Mach number as the sole independent variable.

The fixed gain values used at very low dynamic pressure were determined classically on the basis of desired filter bandwidth and damping ratio.[†††] The scheduled gain values used for high (above $\bar{q}_1^*$ or $\bar{q}_2^*$) dynamic pressure were found by analogy to the steady-state Kalman (Wiener) filter for the reduced, decoupled, linearized state assuming locally stationary noise statistics at each operating point. Measurement noise strengths were chosen to reflect relative sensor quality; driving noise strengths were chosen to keep the gains at a selected level, sufficiently low to achieve desired smoothness but sufficiently high to exclude extreme sensitivity to modeling errors.

---

[†] Analogous to "filter update" and "filter pushdown" in classical filters.

[††] The prescribed dynamic pressure level is $\bar{q}_1^*$ for the longitudinal channel and $\bar{q}_2^*$ for the lateral channel. Current values for $\bar{q}_1^*$ and $\bar{q}_2^*$ are 20 and 30 lb/ft$^2$, respectively.

[†††] Currently, the low-dynamic-pressure estimator bandwidth is 5 rps and the damping ratio is 0.7 sec$^{-1}$.

## 3.3 Implementation

### 3.3.1 Input-Output

Estimator state variables (outputs) are linearized variations about nominal or commanded states. These state variables are the error angle-of-attack $\Delta\hat{\alpha}$ and its time derivative $\Delta\hat{\dot{\alpha}}$, the error sideslip angle $\Delta\hat{\beta}$ and its time derivative $\Delta\hat{\dot{\beta}}$, and the error bank angle $\Delta\hat{\phi}_v$ and its time derivative $\Delta\hat{\dot{\phi}}_v$. (The symbol [^] signifies an estimated quantity.)

To obtain variational outputs, the measurements must be differenced with their respective desired values to form variational inputs. These variational inputs are the measured errors in angle-of-attack $\Delta\tilde{\alpha}$, body pitch rate $\Delta\tilde{q}$, sideslip angle $\Delta\tilde{\beta}$, bank angle $\Delta\tilde{\phi}_v$, body roll rate $\Delta\tilde{p}$, and body yaw rate $\Delta\tilde{r}$. (The tilda [~] symbolizes a measured quantity.) The measurements are either read directly from sensors or computed (preprocessed) from the sensor data. The body rate measurements are assumed to be directly accessible from body-mounted rate gyros, but the angle measurements are derived from IMU (Inertial Measurement Unit) and body-mounted accelerometer information. Commanded values corresponding to each of the measurements are either received from the guidance subsystem or computed internally. A detailed description of estimator input computation is presented in Subsection 2.1.

### 3.3.2 Measurement Incorporation

Current measurements are incorporated into the state estimate immediately after they are obtained and preprocessed. This is immediately before the control subroutine is called, in order that control commands computed using the most recent information may be issued as soon as possible after the DFCS is called. Subsequently, the state estimate is extrapolated in preparation for the next DFCS pass, as shown in Subsection 3.3.3 below.

Measurements are incorporated according to the equations

$$\begin{Bmatrix} \Delta\hat{\alpha} \\ \Delta\hat{\dot{\alpha}} \end{Bmatrix}_k = \begin{Bmatrix} \Delta\hat{\alpha}' \\ \Delta\hat{\dot{\alpha}}' \end{Bmatrix}_k + K_{lon} \begin{Bmatrix} (\Delta\tilde{\alpha} - \Delta\hat{\alpha}') \\ (\Delta\tilde{q} - \Delta\hat{\dot{\alpha}}') \end{Bmatrix}_k \tag{3-1}$$

$$\begin{Bmatrix} \Delta\hat{\beta} \\ \Delta\hat{\dot{\beta}} \\ \Delta\hat{\phi}_v \\ \Delta\hat{\dot{\phi}}_v \end{Bmatrix}_k = \begin{Bmatrix} \Delta\hat{\beta}' \\ \Delta\hat{\dot{\beta}}' \\ \Delta\hat{\phi}_v' \\ \Delta\hat{\dot{\phi}}_v' \end{Bmatrix}_k + K_{lat} \begin{Bmatrix} (\Delta\tilde{\beta} - \Delta\hat{\beta}') \\ (\Delta\tilde{\phi}_v - \Delta\hat{\phi}_v') \\ (\Delta\tilde{p} - \Delta\dot{\beta}' \sin\alpha_T - \Delta\hat{\dot{\phi}}_v' \cos\alpha_T) \\ (\Delta\tilde{r} + \Delta\hat{\dot{\beta}}' \cos\alpha_T - \Delta\hat{\dot{\phi}}_v' \sin\alpha_T) \end{Bmatrix}_k \tag{3-2}$$

where the prime (′) indicates a state variable propagated from the previous sample instant, $K_{lon}$ is the 2x2 longitudinal gain matrix, and $K_{lat}$ is the 4x4 lateral gain matrix. The gain matrices are updated at the slow repetition rate in the parameter estimation routine, which is performed after the state extrapolation step in the normal DFCS cycle.

The flow charts in Fig. 3-1 and 3-2 show the implementation in the DFCS program of the measurement incorporation equations and the computation of the estimator gain matrices. When dynamic pressure is below $\bar{q}_1^*$ for the longitudinal channel and $\bar{q}_2^*$ for the lateral channel, the constant values KC1 and KC2 are loaded into the estimator gain matrices as shown; the current values for these parameters are 0.45 and 2.5 respectively. For dynamic pressure greater than $\bar{q}_1^*$ and $\bar{q}_2^*$ for longitudinal and lateral channels respectively, the estimator gain matrices are found by linear interpolation on Mach number; the values are given in Tables 3-1 and 3-2.

### 3.3.3 State Extrapolation

After control commands are computed using the current state estimate (from Eq (3-1) and (3-2)) and issued, the estimator state is extrapolated to the next sample according to the equations:

$$\begin{Bmatrix} \Delta\hat{\alpha}' \\ \Delta\hat{\dot{\alpha}}' \end{Bmatrix}_{k+1} = \Phi_{lon}^{CL} \begin{Bmatrix} \Delta\hat{\alpha} \\ \Delta\hat{\dot{\alpha}} \end{Bmatrix}_k + \begin{Bmatrix} 0 \\ \Delta q^j \end{Bmatrix}_k \tag{3-3}$$

$$\begin{Bmatrix} \Delta\hat{\beta}' \\ \Delta\hat{\dot{\beta}}' \\ \Delta\hat{\phi}_v' \\ \Delta\hat{\dot{\phi}}_v' \end{Bmatrix}_{k+1} = \Phi_{lat}^{CL} \begin{Bmatrix} \Delta\hat{\beta} \\ \Delta\hat{\dot{\beta}} \\ \Delta\hat{\phi}_v \\ \Delta\hat{\dot{\phi}}_v \end{Bmatrix}_k + \begin{bmatrix} 0 & 0 \\ (\sin\alpha_T) & (-\cos\alpha_T) \\ 0 & 0 \\ (\cos\alpha_T) & (\sin\alpha_T) \end{bmatrix} \begin{Bmatrix} \Delta p^j \\ \Delta r^j \end{Bmatrix}_k \tag{3-4}$$

where $\Phi_{lon}^{CL}$ and $\Phi_{lat}^{CL}$ are the longitudinal (2x2) and lateral (4x4) state transition matrices†; $\Delta p^j$, $\Delta q^j$, $\Delta r^j$ are the average changes in body angular rates about the x, y and z axes due to ACPS jet activity over the previous sample interval, and $\alpha_T$ is the mean angle-of-attack.

---

†The transition matrices are designated $\Phi^{CL}$ (i.e., $\Phi$ closed-loop) to indicate that closure of the aerodynamic control loop is taken into account in their derivation:

$$\Phi^{CL} = \Phi^{OL} + \Gamma^{aero} C^{aero}$$

where $\Phi^{OL}$ is the open-loop state transition matrix, $\Gamma^{aero}$ is the aerodynamic-control-to-state matrix, and $C^{aero}$ are the linear-feedback aerodynamic control gains.

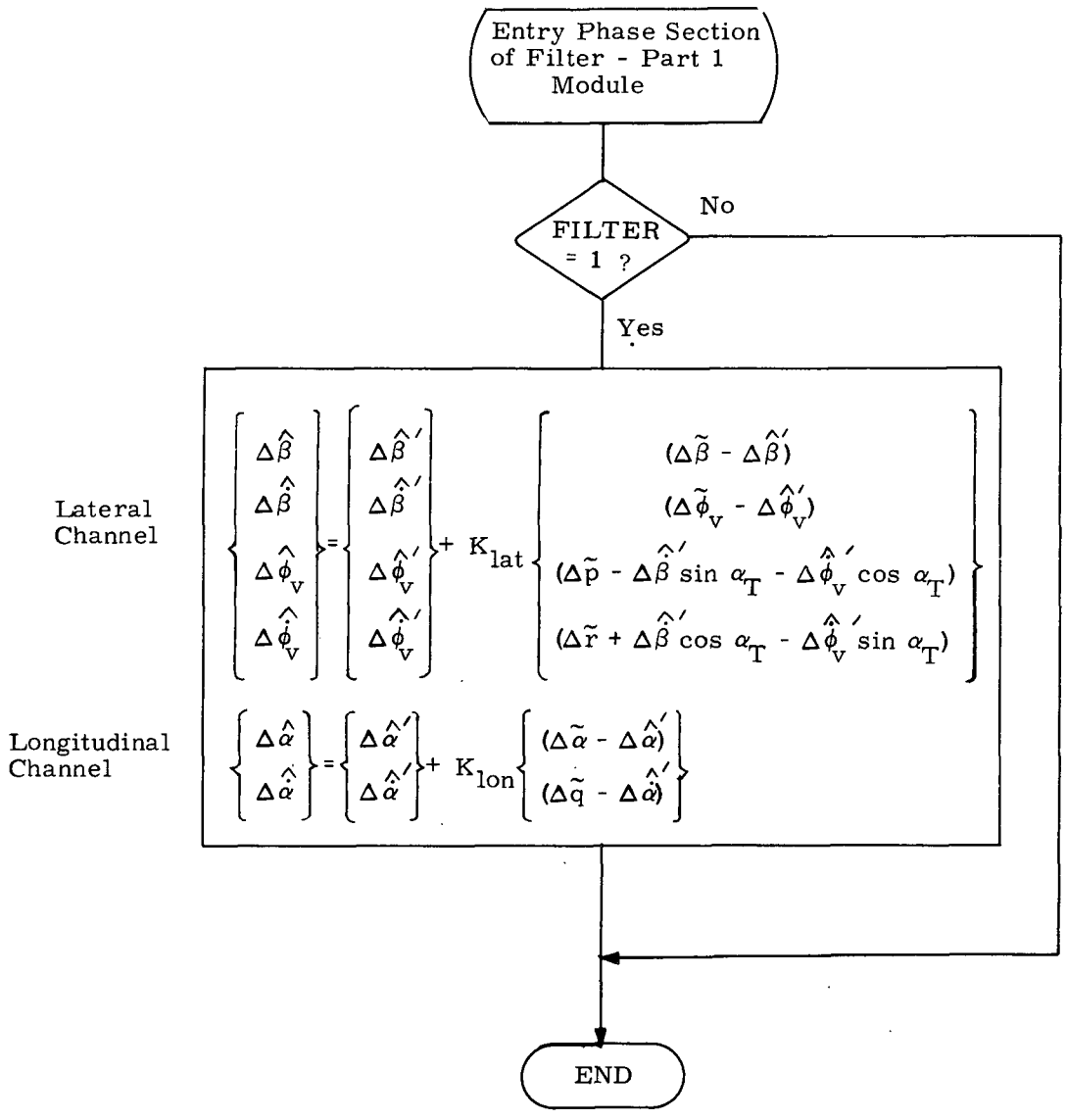



Fig. 3-1 Measurement Incorporation for State Estimation

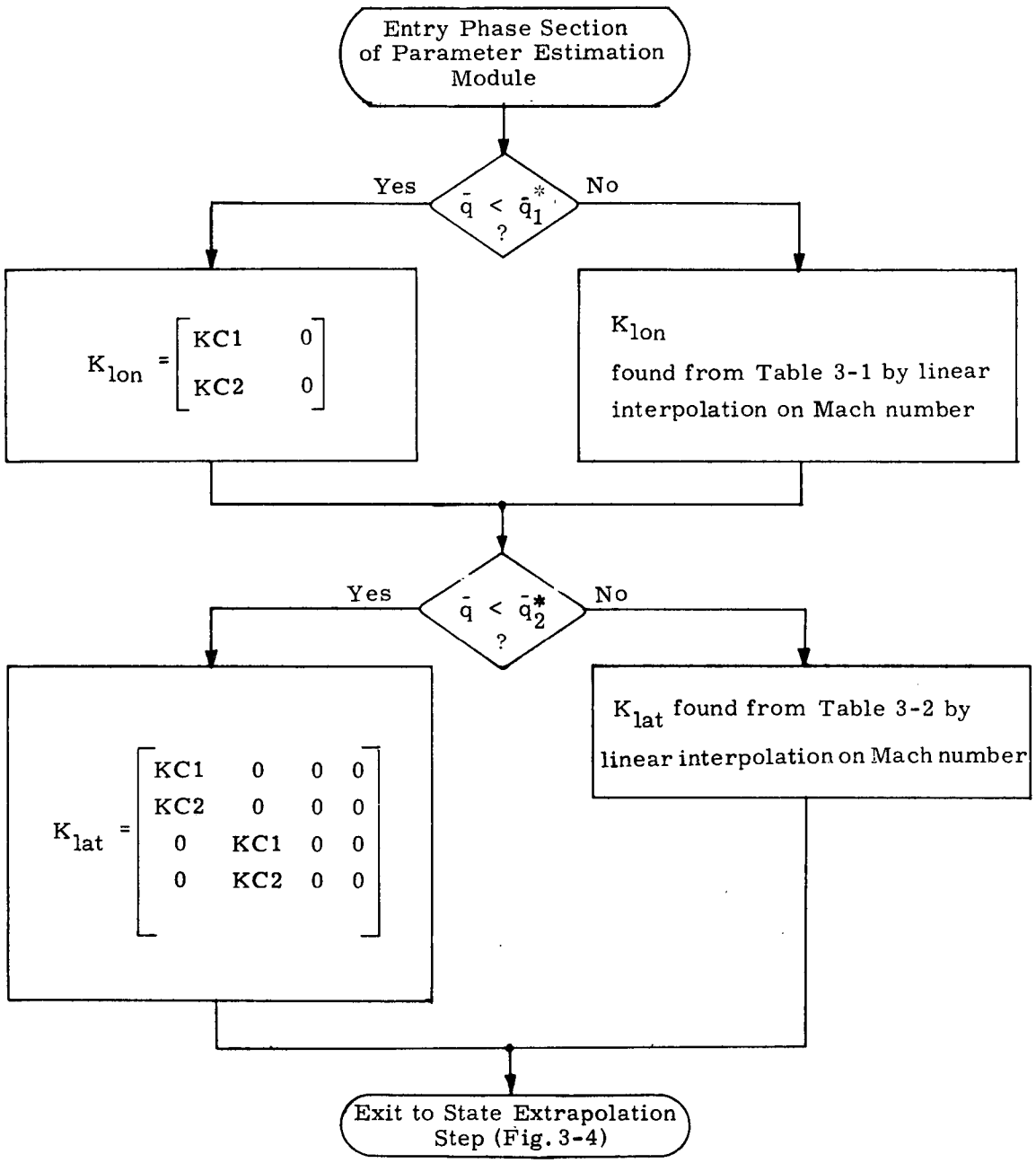



Fig. 3-2 Computation of Estimator Gain Matrices

Table 3-1 Longitudinal Estimator Gains Scheduled vs Mach Number

| | | Mach = 0.9 | Mach = 2 | Mach = 5 | Mach = 10 | Mach = 20 |
|---|---|---|---|---|---|---|
| $K_{lon}$ | (1, 1) | 0.0731 | 0.0769 | 0.0680 | 0.0568 | 0.0401 |
| | (1, 2) | 0.0408 | 0.0411 | 0.0554 | 0.0523 | 0.0586 |
| | (2, 1) | 0.0538 | 0.0353 | 0.0329 | 0.0207 | 0.0116 |
| | (2, 2) | 0.4320 | 0.4289 | 0.2351 | 0.2616 | 0.1909 |

Table 3-2 Lateral Estimator Gains Scheduled vs Mach Number

| | | Mach = 0.9 | Mach = 2 | Mach = 5 | Mach = 10 | Mach = 20 |
|---|---|---|---|---|---|---|
| $K_{lat}$ | (1, 1) | 0.2119 | 0.0578 | 0.0427 | 0.0314 | 0.0222 |
| | (1, 2) | -0.0297 | -0.0794 | -0.1495 | -0.1098 | -0.0942 |
| | (1, 3) | -0.0115 | -0.0282 | -0.0234 | -0.0210 | -0.0203 |
| | (1, 4) | -0.0337 | -0.0456 | -0.0637 | -0.0605 | -0.0659 |
| | (2, 1) | 0.2410 | 0.0238 | 0.0214 | 0.0115 | 0.0065 |
| | (2, 2) | 0.0257 | 0.0155 | 0.0016 | 0.0177 | 0.0075 |
| | (2, 3) | 0.0298 | 0.0696 | 0.0701 | 0.1256 | 0.0900 |
| | (2, 4) | -0.4127 | -0.4213 | -0.2223 | -0.2261 | -0.1650 |
| | (3, 1) | -0.0027 | -0.0008 | -0.0015 | -0.0011 | -0.0009 |
| | (3, 2) | 0.4924 | 0.4372 | 0.3569 | 0.3213 | 0.2440 |
| | (3, 3) | 0.0148 | 0.0187 | 0.0198 | 0.0224 | 0.0256 |
| | (3, 4) | 0.0010 | 0.0030 | 0.0072 | 0.0122 | 0.0142 |
| | (4, 1) | -0.1164 | -0.0212 | -0.0181 | -0.0133 | -0.0070 |
| | (4, 2) | 1.355 | 1.130 | 0.8698 | 0.6984 | 0.4032 |
| | (4, 3) | 0.3703 | 0.3846 | 0.1853 | 0.2043 | 0.1473 |
| | (4, 4) | 0.0387 | 0.0725 | 0.0783 | 0.1229 | 0.0903 |

The state transition matrices are computed to be the discrete versions of the respective matrices which describe the desired continuous closed-loop aerodynamics:

$$\Phi^{CL} \approx I + F^{CL} \Delta T \tag{3-5}$$

where I is the identity matrix, $\Delta T$ is the sample interval, and $F^{CL}$ is the system dynamics matrix; i.e.,

$$\Delta \dot{\underline{x}} = F^{CL} \Delta \underline{x} \tag{3-6}$$

where $\Delta \underline{x}$ is the variational state. Equation (3-5) is the first order power series expansion to the solution of Eq (3-6). A more accurate representation of $\Phi^{CL}$ would probably be necessary only if $\Delta T$ were not small with respect to the natural time constants of $F^{CL}$.

The longitudinal state transition matrix is found by Eq (3-5) from $F^{CL}_{lon}$, which takes on either of two forms:

$$F^{CL}_{lon} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \tag{3-7}$$

or

$$F^{CL}_{lon} = \begin{bmatrix} 0 & 1 \\ -\omega_d^2 & -2\zeta_d \omega_d \end{bmatrix} \tag{3-8}$$

where the parameters $\omega_d$ and $\zeta_d$ specify the desired longitudinal response. (See Subsection 5.3.1.1). Equation (3-7) is used for dynamic pressure below $\bar{q}_1^*$, when the elevator is essentially used only for trim. Equation (3-8) is used for dynamic pressure above $\bar{q}_1^*$.

The lateral state transition matrix is found by Eq (3-5) from $F^{CL}_{lat}$, which takes on either of two forms:

$$F^{CL}_{lat} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{3-9}$$

or

$$F^{CL}_{lat} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -b_{11} & -a_{11} & -b_{12} & -a_{12} \\ 0 & 0 & 0 & 1 \\ -b_{21} & -a_{21} & -b_{22} & -a_{22} \end{bmatrix} \tag{3-10}$$

where the parameters $a_{11}$ through $b_{22}$ specify the desired lateral response (see Subsection 5.3.2.1.2). Equation (3-9) is used for dynamic pressure below $\bar{q}_2^*$, when the aileron is essentially not used. Equation (3-10) is used for dynamic pressure above $\bar{q}_2^*$.

The flow charts in Figs. 3-3 and 3-4 show the implementation in the DFCS program of the state extrapolation equations and the computation of the state transition matrices. The numerical values of the parameters used in the state extrapolation are indicated in Table 3-3. The control gains $C'_{\ell_{\delta a}}$, $C'_{n_{\delta a}}$, $C'_{\ell_\beta}$, $C'_{n_\beta}$ will be obtained from the entry control module in the future. Currently, however, they are independently scheduled as functions of Mach number, as shown in Fig. 3-5.

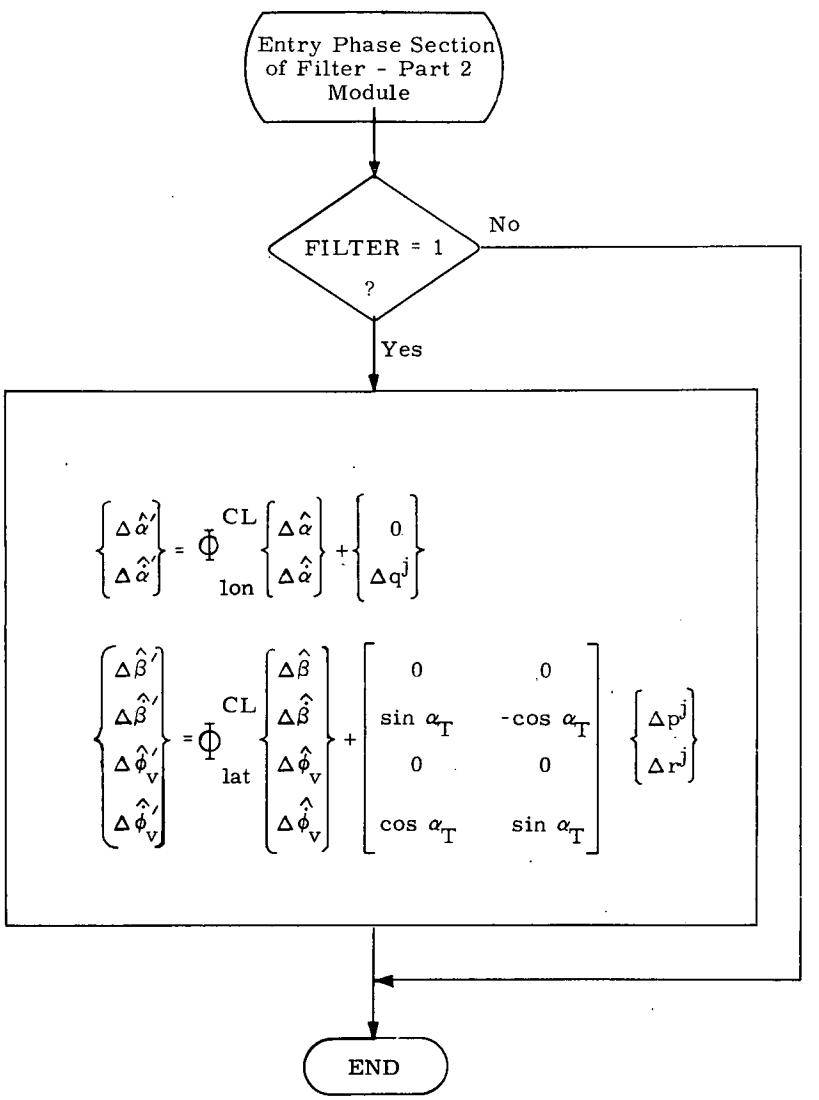



Fig. 3-3 State Extrapolation for the Entry State Estimator

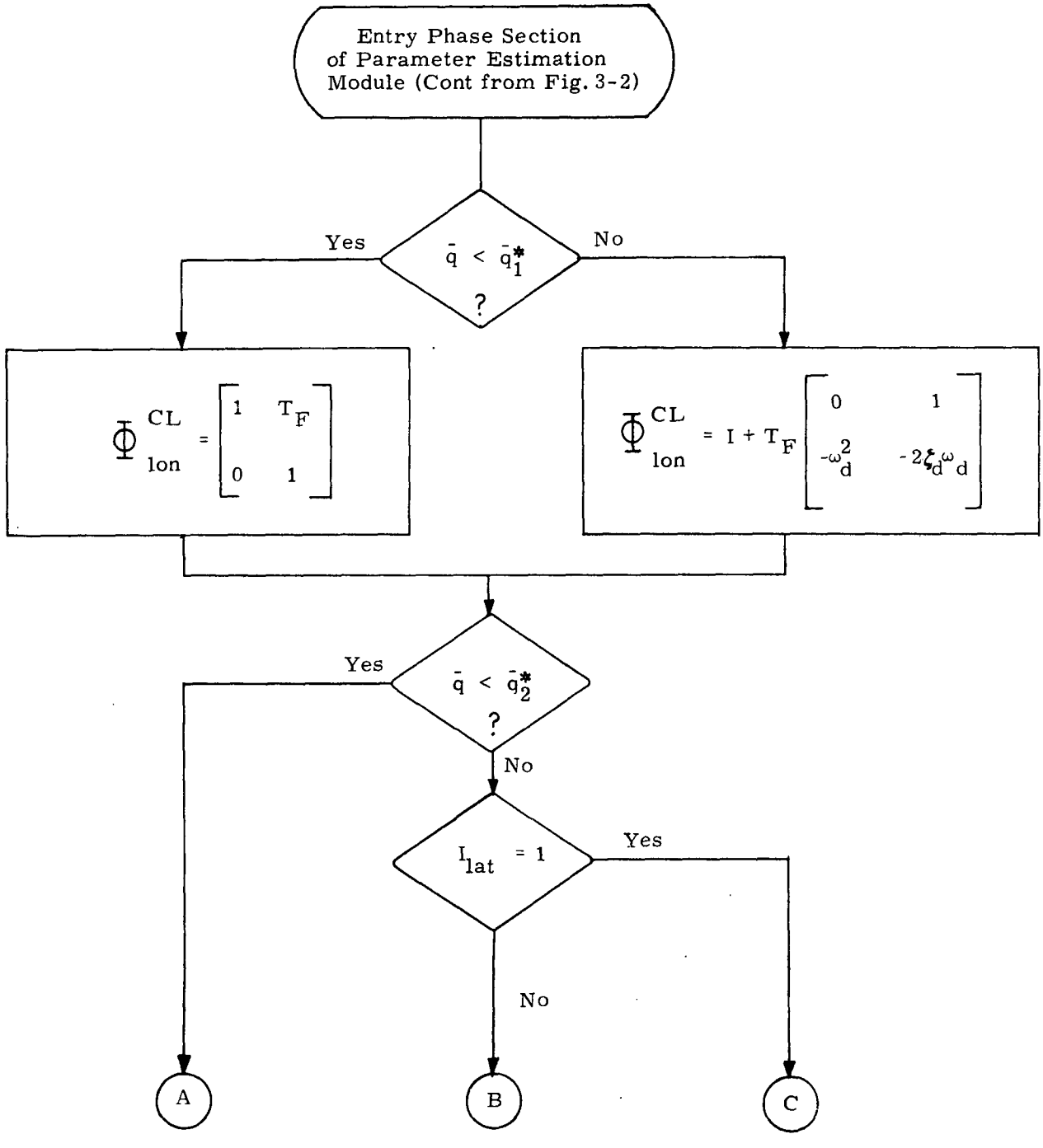



Fig. 3-4 Computation of Estimator State Transition Matrices

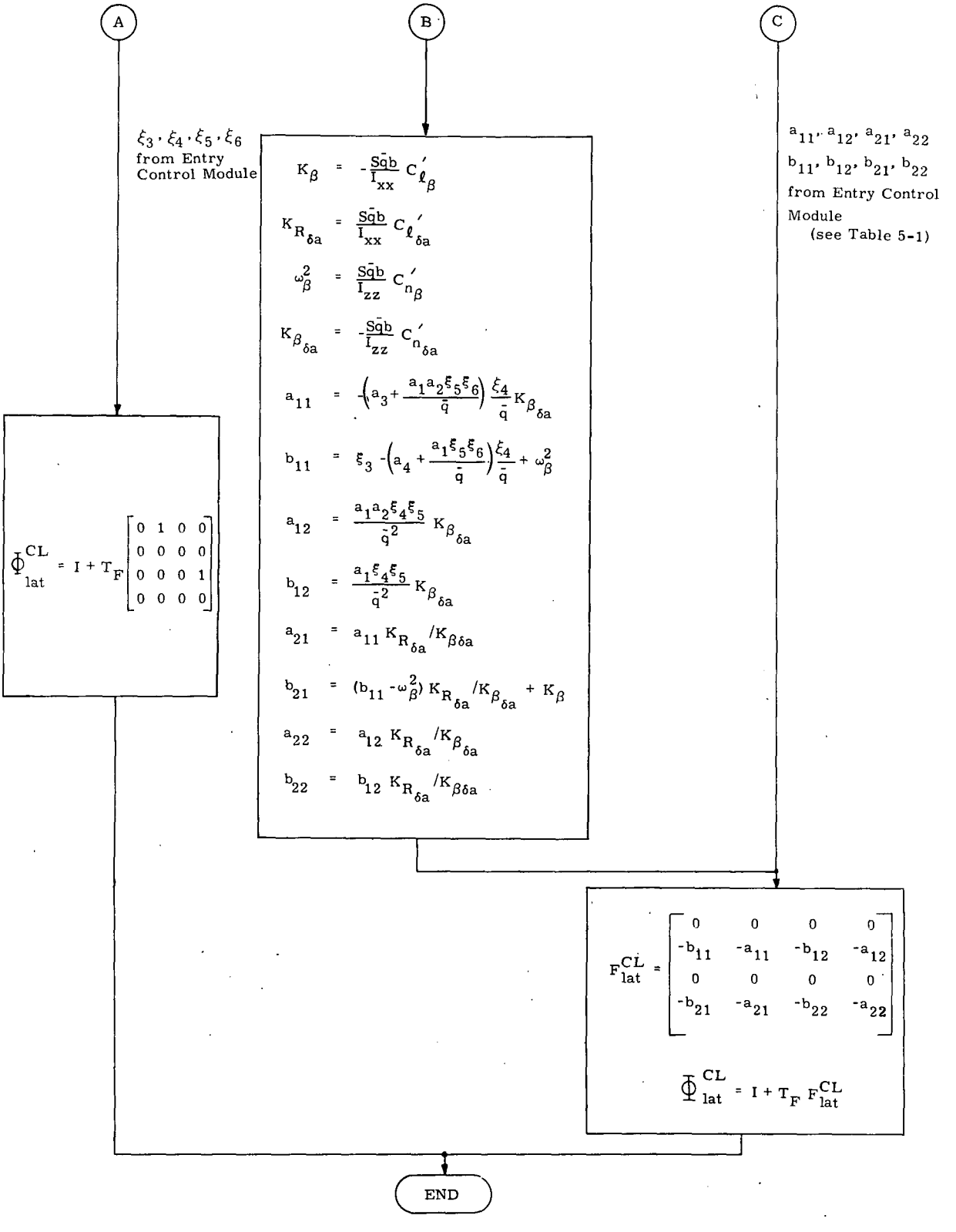



Fig. 3-4 Computation of Estimator State Transition Matrices (Cont)

Table 3-3 Parameter Values for State Extrapolation

Aerodynamic and Inertial Parameters

| | |
|---|---|
| $S$ | 2690. |
| $b$ | 78.056 |
| $I_{xx}$ | 805000 |
| $I_{zz}$ | 6067000 |

Fixed Control Gains

| | |
|---|---|
| $a_1$ | 0.2401 |
| $a_2$ | 6.0 |
| $a_3$ | 2.94 |
| $a_4$ | 2.9008 |
| $a_{11}$, $a_{12}$, $a_{21}$, $a_{22}$, $b_{11}$, $b_{12}$, $b_{21}$, $b_{22}$ | See Table 5-1 |

Scheduled Control Gains

| | | | |
|---|---|---|---|
| $\xi_1$, $\xi_2$, $\xi_3$, $\xi_4$, $\xi_5$, $\xi_6$ | See Figs. 5-16 through 5-21 | $C'_{\ell_{\delta a}}$, $C'_{n_{\delta a}}$, $C'_{\ell_\beta}$, $C'_{n_\beta}$ | See Fig. 3-5 |

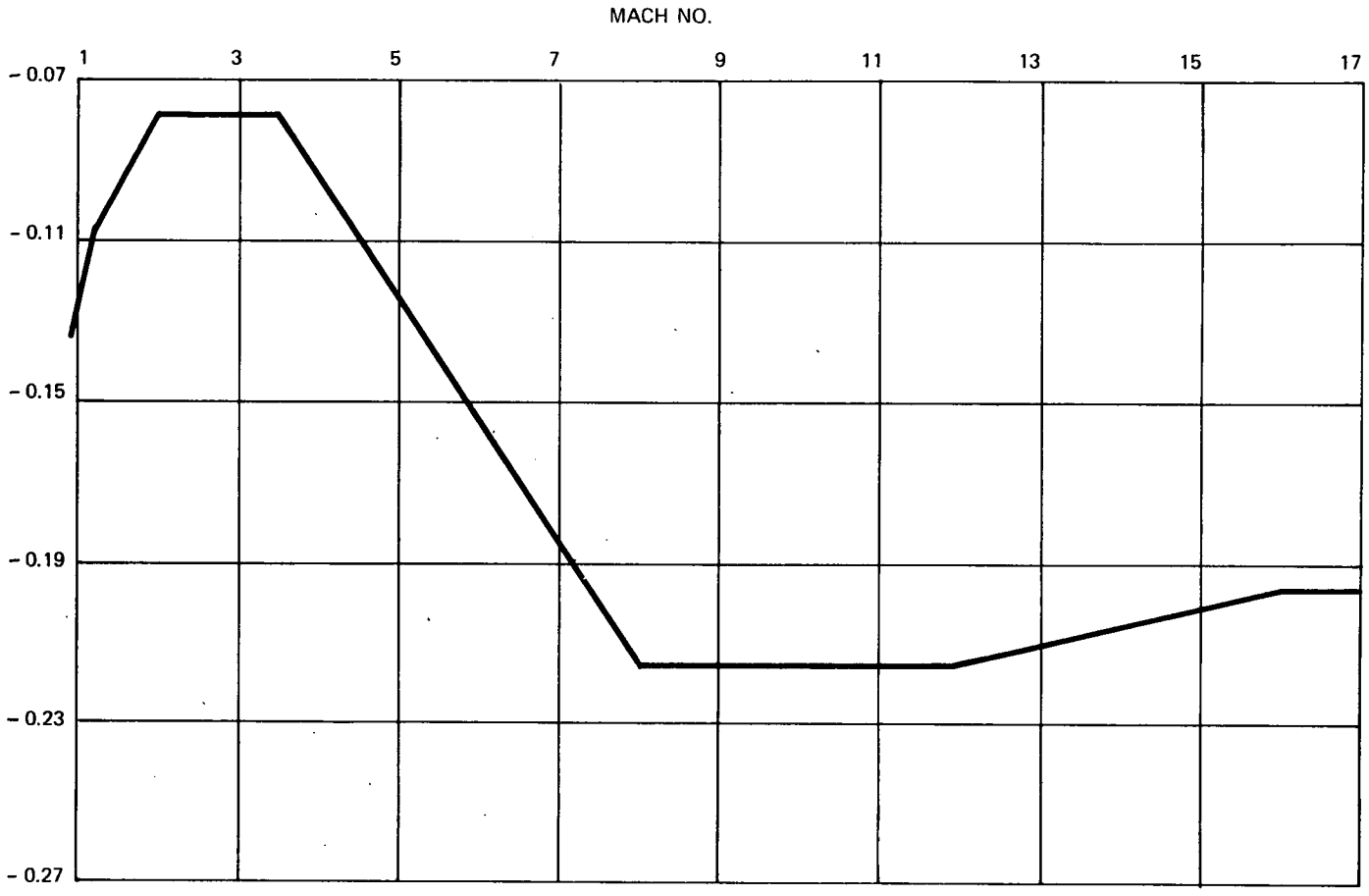



Fig. 3-5a $C_{\ell_{\delta a}}'$ vs Mach Number

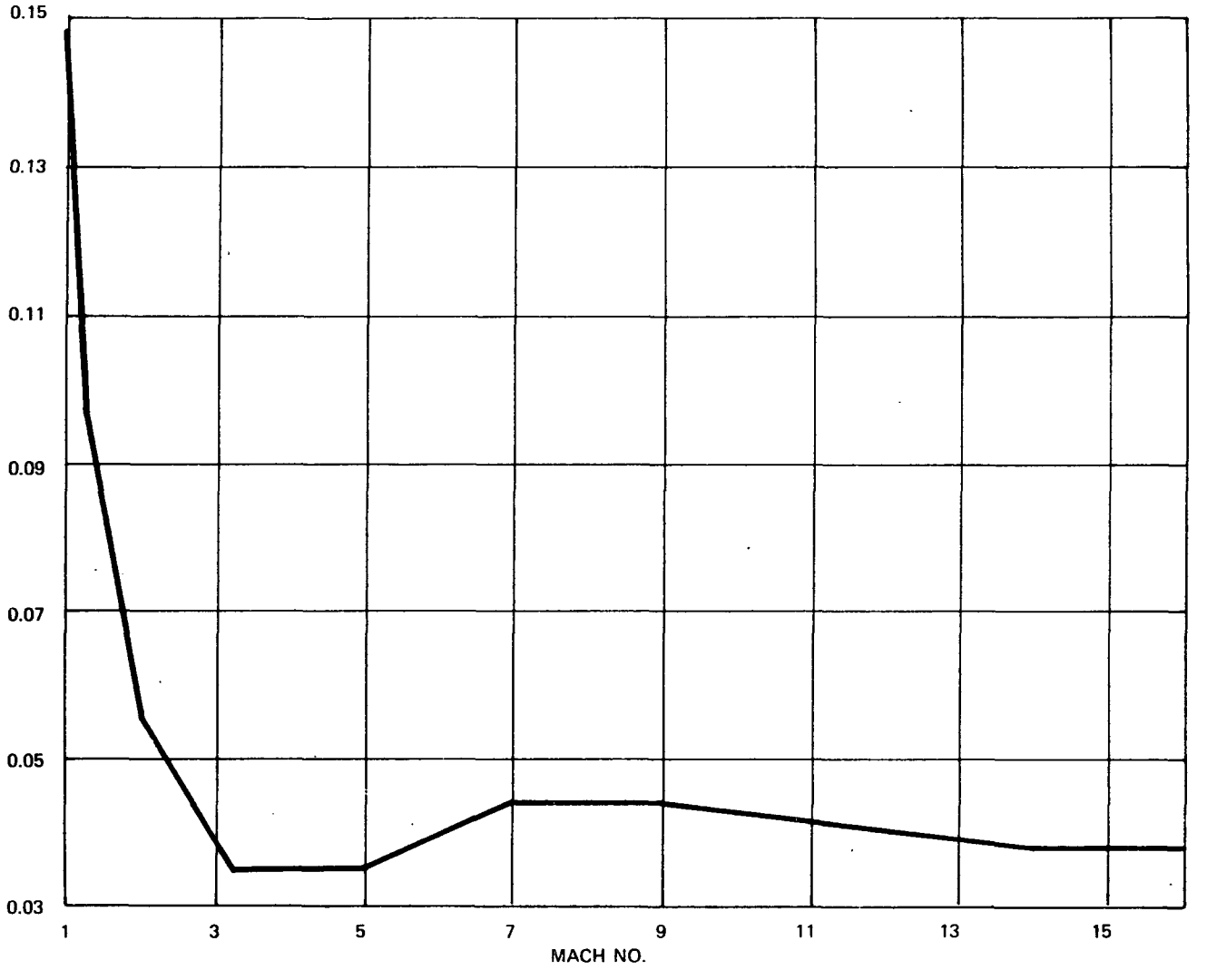



Fig. 3-5b $C_{n_{\delta a}}'$ vs Mach Number

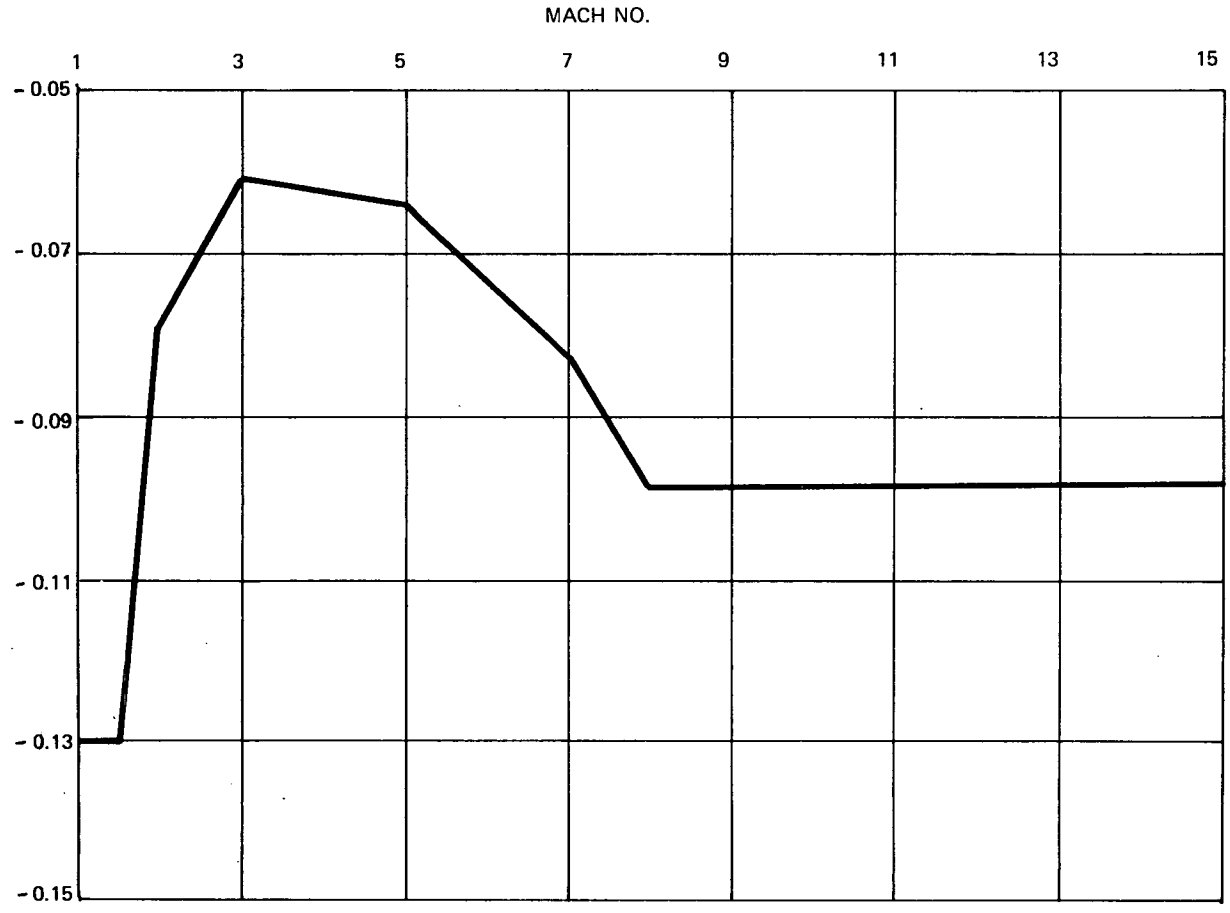



Fig. 3-5c $C_{\ell_\beta}'$ vs Mach Number

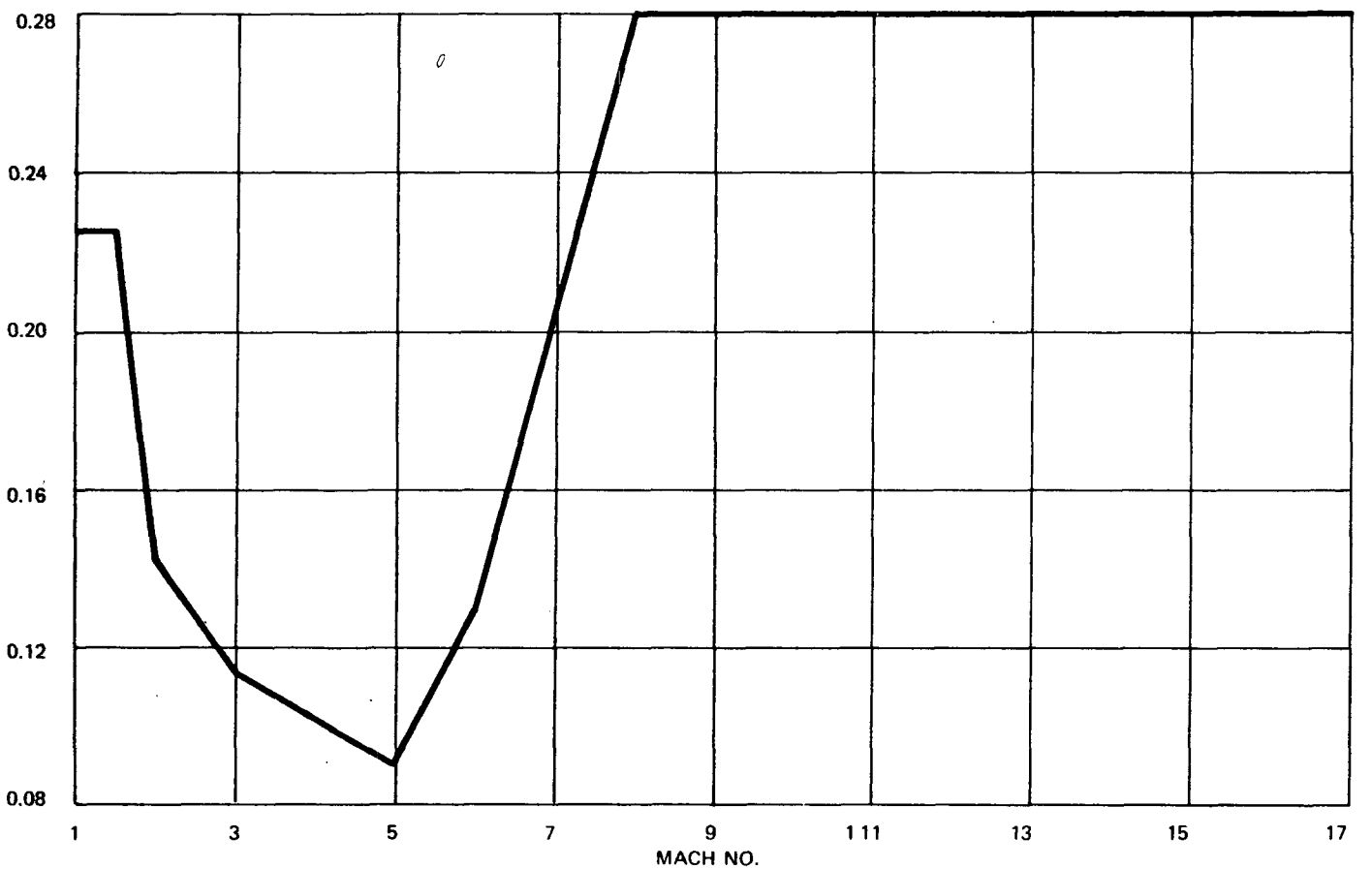



Fig. 3-5d $C_{n_\beta}'$ vs Mach Number

## 3.4 Preliminary Performance Summary

Simulated maneuvers have been performed on the Space Shuttle Functional Simulator using the state estimator in conjunction with the entry DFCS and the NR00089B (February version) vehicle. Both large and small step changes in angle of attack and bank were commanded under a wide range of flight conditions (Mach = 2, $\bar{q}$ = 150 lbs/ft$^2$ to Mach = 25, $\bar{q}$ = 0.25 lbs/ft$^2$).

Comparisons were made between simulations in which the DFCS controller operated on state estimator outputs (FILTER = 1) and corresponding simulations in which the DFCS controller operated directly on measured data (FILTER = 0). When random noise sources were not simulated (perfect measurements), comparisons between corresponding simulations were very good in every case. When random noise sources were simulated, corresponding simulations exhibited similar mean behavior, but aerodynamic surface commands were smoother for those cases which used the estimator. Figure 3-6, for example, shows actual angle of attack, roll angle, and aerodynamic surface position time histories for corresponding cases at Mach 14 and a dynamic pressure of 108 lbs/ft$^2$.

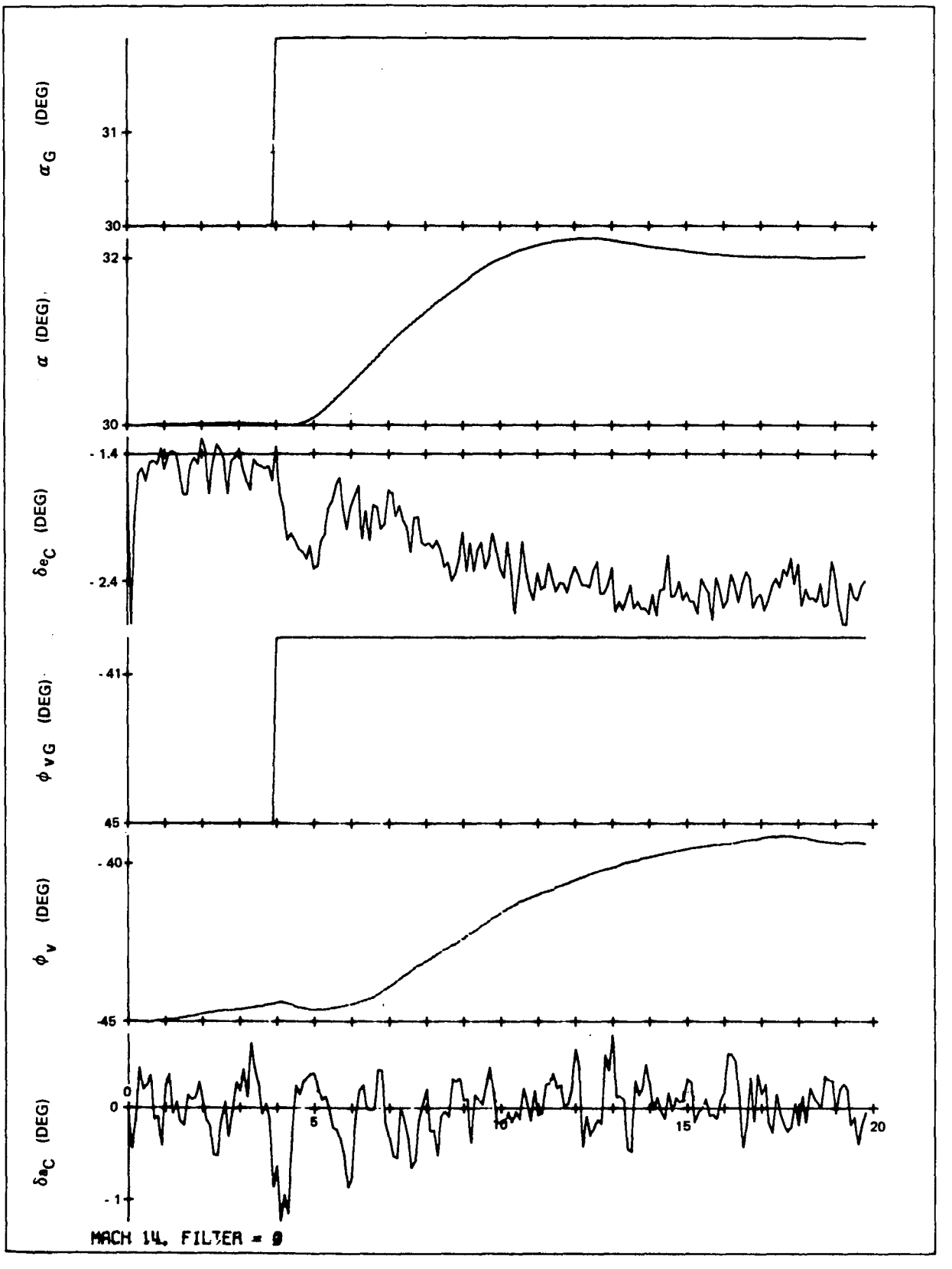



Fig. 3-6a Longitudinal and Lateral Responses with Noise Sources Simulated and FILTER = 0 (no estimator in loop)



C-2

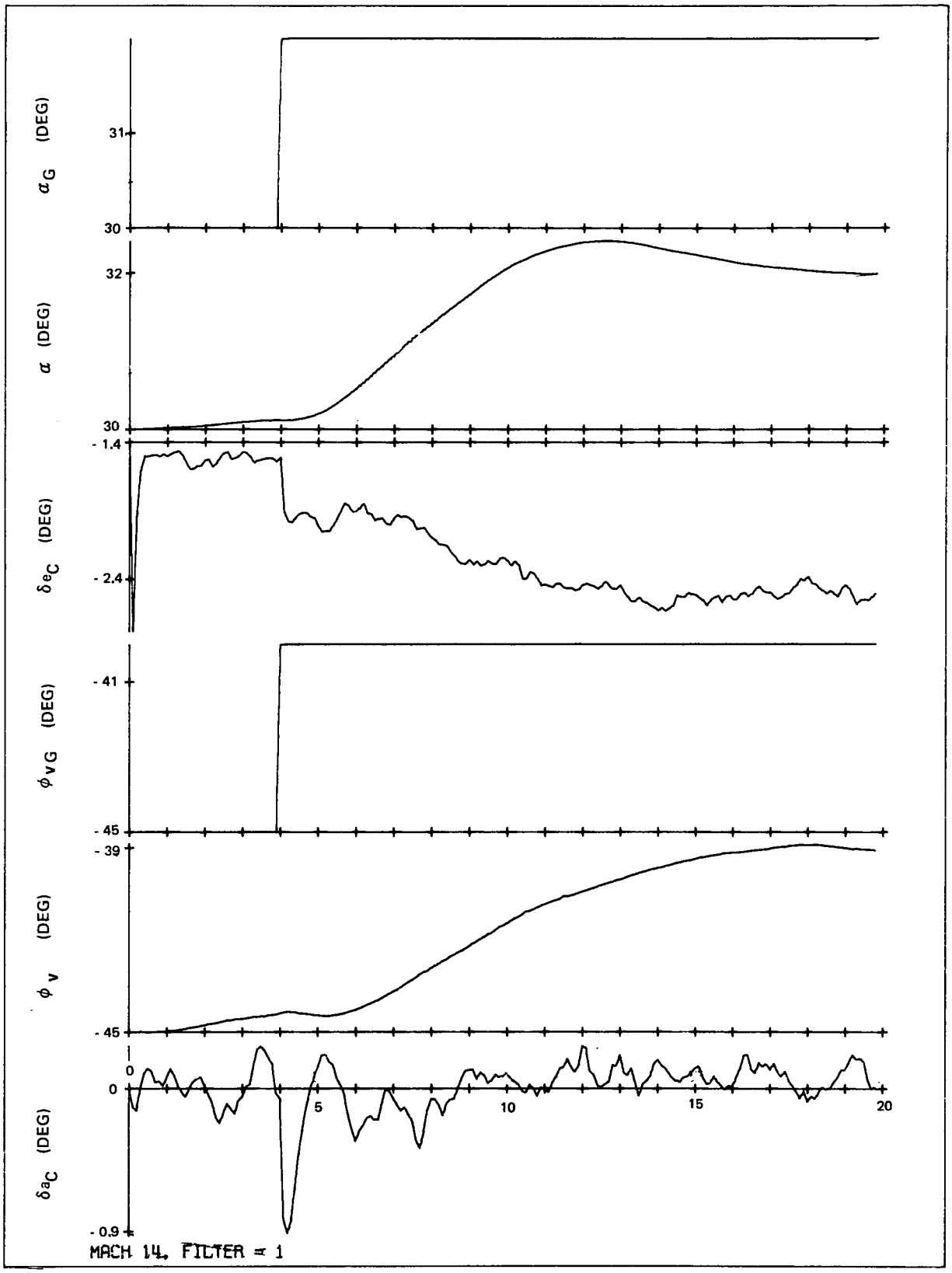



Fig. 3-6b Longitudinal and Lateral Responses with Noise Sources Simulated and FILTER = 1 (estimator in loop)

SECTION 4

# CRUISE PHASE

by

Albert G. Engel, Franklin H. Moss, Alexander Penchuk, Nicole Lamotte

## 4.1 Introduction

The cruising flight phase of an orbital Space Shuttle flight begins at about Mach 0.9, ends at touchdown, and follows the high-angle-of-attack/high-Mach-number entry phase. During the cruise phase the control is accomplished by moving the aerodynamic control surfaces: elevons, rudder, rudder flare (speed brake), and body flap. In addition, the cruise phase DFCS has been extended to deal with Shuttle powered horizontal flight, which encompasses developmental testing and ferry missions. During powered flights, thrust is an additional control variable.

For the cruise phase the DFCS contains three primary modes of operation: manual, semiautomatic and automatic. Manual mode consists of two types of control systems: Stability Augmentation System (SAS) and Command Augmentation System (CAS). By means of a side arm controller and rudder pedals, the pilot uses either of these two systems to manually control the orbiter. The pilot can reduce his workload while controlling the orbiter by selecting certain automatic functions provided by outer loops. This is the semiautomatic mode of operation. In the automatic mode of operation guidance algorithms provide the commands to the digital flight control system, and the pilot's control role is reduced to monitoring the overall performance. One further system available is the Backup Control System (BCS) which is nearly identical to the SAS. This system is completely manual, and following its initial flight test, is intended to remain unchanged throughout the Space Shuttle program.

Speed control is accomplished by using either speed brake or throttle. The settings can be controlled in an open loop fashion either by the pilot in the manual mode or by the guidance algorithms in the automatic mode. Alternatively the settings can be obtained from closed loop speed control loops. In this case, depending on the mode of operation, the speed commands come either from the pilot (select/hold mode) or from the guidance. For the purpose of pitch trimming, the body flap's setting is scheduled as a function of Mach number, which can be supplemented by a manual flap command.

Stabilization of the vehicle in the cruise phase is accomplished with compensation filters which appear in forward loop, feedback, and crossfeed paths. It was found that for control of the rigid body many of these filters are not needed and good

control performance can be obtained with simpler compensations than allowed for. However, these filters will become necessary and may, in fact, become insufficient in size as control of the flexible vehicle is considered. A table of gains suitable for control of the rigid body 89B vehicle at a particular flight condition (Mach 0.6, dynamic pressure 300 psf, altitude 6000 ft) is provided (Table 4-1). The gains which must be varied as a function of some flight parameter (such as dynamic pressure), are marked by an asterisk.

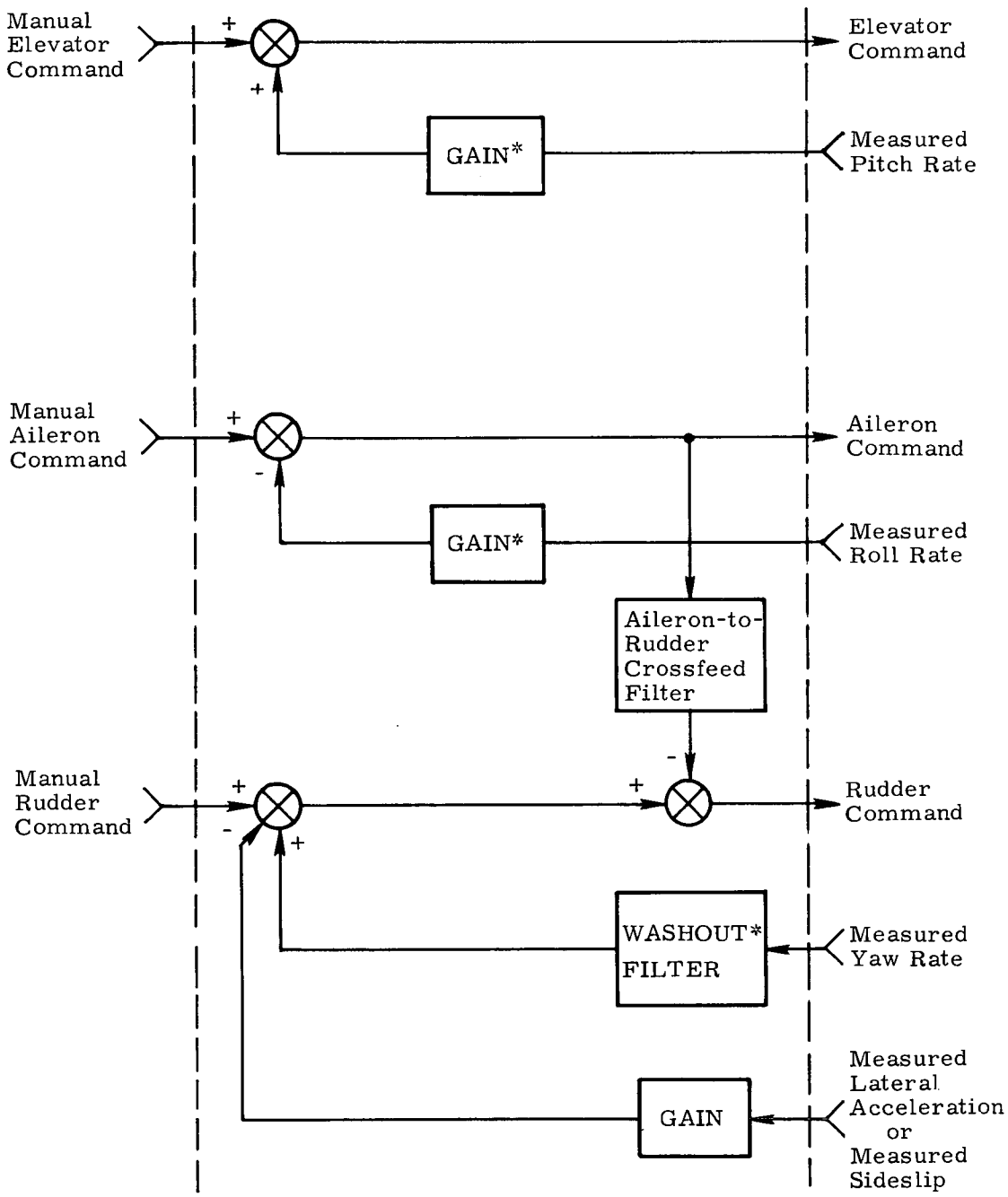
The description of the cruise phase autopilot is presented in the order of increasing detail, starting with a discussion of simplified block diagrams, proceeding to detailed functional block diagrams, and then finally to detailed flow charts. Subsection 4.2 describes manual, semiautomatic, and automatic control modes as well as speed and body flap controls. The description of the BCS is followed by that of mode selection and mode change trim initialization. Filter implementation is discussed in Subsection 4.3 where a generalized $n^{th}$-order digital filter implementation is detailed along with low-pass and washout filters, and integration functions. Subsection 4.4 describes preflight initialization of gains and filter coefficients, as well as limits, deadbands and other items. Subsection 4.5 presents summaries of input/output quantities for various channels such as elevator, aileron, etc. The detailed functional block diagrams in Subsection 4.6 are preceded by a list of conventions in order to facilitate their presentation. Subsection 4.7 contains functional flow charts for initialization subroutine of the cruise DFCS, and then detailed flow charts are presented in the Appendix, Subsection 4A.

## 4.2 Control Modes

### 4.2.1 Manual Modes of Operation

For manual operation the two types of control systems are a command augmentation system (CAS) for normal operation, and a stability augmentation system (SAS) for lower performance downmode operation. The pitch rate and roll rate feedback loops, with simple gain compensations in the feedback paths, are employed in SAS (Fig. 4-1). The manual commands to the elevator and aileron are summed with these feedback signals and sent out to the control surfaces. The manual rudder command is summed with two feedbacks and one crossfeed signal from the aileron before it is sent to the rudder. Washed-out yaw rate feedback augments the dutch-roll stability. The turn coordination in SAS is achieved by either a combination of aileron to rudder crossfeed and lateral acceleration, or sideslip feedback. For selecting the best control response, pitch rate, roll rate, and yaw rate gains can be scaled by the pilot from his control panel.

A simplified block diagram of CAS is shown in Fig. 4-2. The longitudinal and lateral control systems are pitch rate (or $C^*$) loop and roll rate loop closures having similar forward path compensation filters followed by a proportional path and an integral




* Gain-switch-sensitive gains

Fig. 4-1 Simplified Functional Block Diagram of The Stability Augmentation System (also Backup Control System) Mode

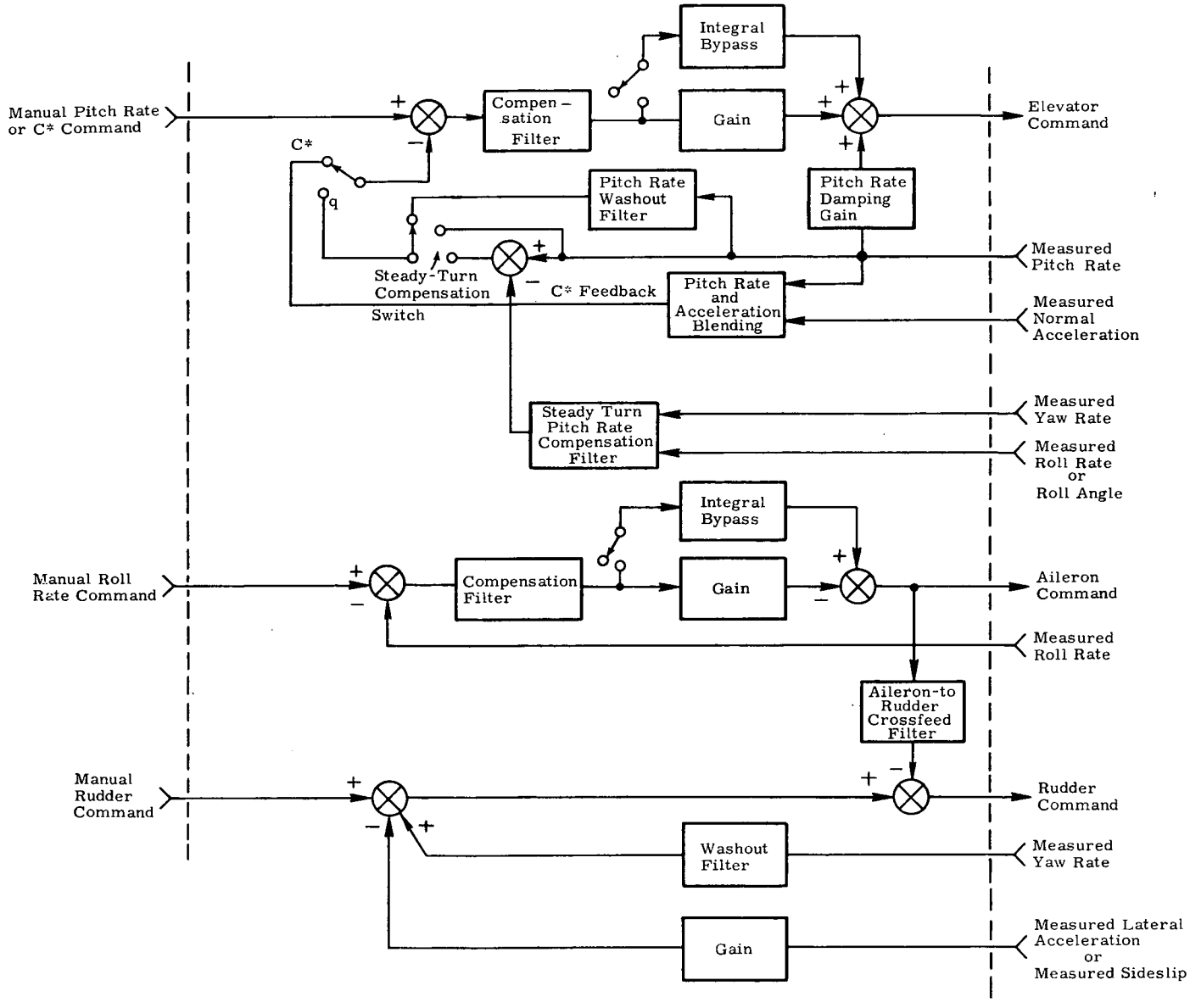



Fig. 4-2 Simplified Functional Block Diagram of the Command Augmentation System Mode

bypass. The integral bypass provides automatic trimming and assures unit-gain rate tracking. This function can be switched in or out from the control panel, and the pitch rate/$C^*$ loop has an inner damping loop and an outer feedback loop enabling a selection of either $C^*$ or pitch rate. $C^*$ is a blend of pitch rate and normal acceleration. The pitch rate feedback can be fed back directly, passed first through a washout filter, or fed back with an added steady turn compensating component. This component is proportional to the yaw rate and tangent of the roll angle which is measured or derived from the measured roll rate. Depending on the type of feedback ($C^*$ or pitch rate), the manual command is scaled according to either $\dot{C}^*$ or q for the control variable. The roll rate loop does not have an inner damping loop and thus uses roll rate feedback. The rudder channel is the same as in the SAS.

### 4.2.2 Semiautomatic Modes of Operation

Certain selectable outer loop closures provide semiautomatic modes of operation, the construction of which is shown in Fig. 4-3. For longitudinal control the inputs are either pitch attitude or altitude references with the controlled variables being pitch attitude and altitude. For lateral control the references and the controlled quantities are the bank angle and heading. In both cases, engaging the control stick opens these outer loops and enables the control stick steering function. Therefore, returning the stick to null reengages the selected semiautomatic mode once the body rates fall below a threshold value.

### 4.2.3 Automatic Modes of Operation

For one particular cruise phase guidance scheme the automatic mode is subdivided into two regions of operation according to the following table:

| Name of guidance phase | Approximate altitude range | Guidance commands |
|---|---|---|
| Approach | 40000 to 7000 feet | $\alpha_G$, $\phi_{v_G}$ |
| Terminal | 7000 feet to touchdown | $q_G$, $\phi_{b_G}$ |

During the approach phase the longitudinal control is of the angle-of-attack. In the terminal phase, longitudinal control is accomplished by the pitch rate loop which is identical to the CAS loop described in Subsection 4.2.1 (including pitch rate compensation during steady turns). In both cases the lateral control is of the roll attitude. However, the roll angle commands are different in the two guidance cases: roll angle is taken about the velocity vector in the approach phase, and about the body (vehicle x-axis) in the terminal phase. For both cases the form of the lateral control system and the gains are the same.

As illustrated in Fig. 4-4, guidance commands other than those shown above can also be accommodated. Thus the longitudinal control system can be configured

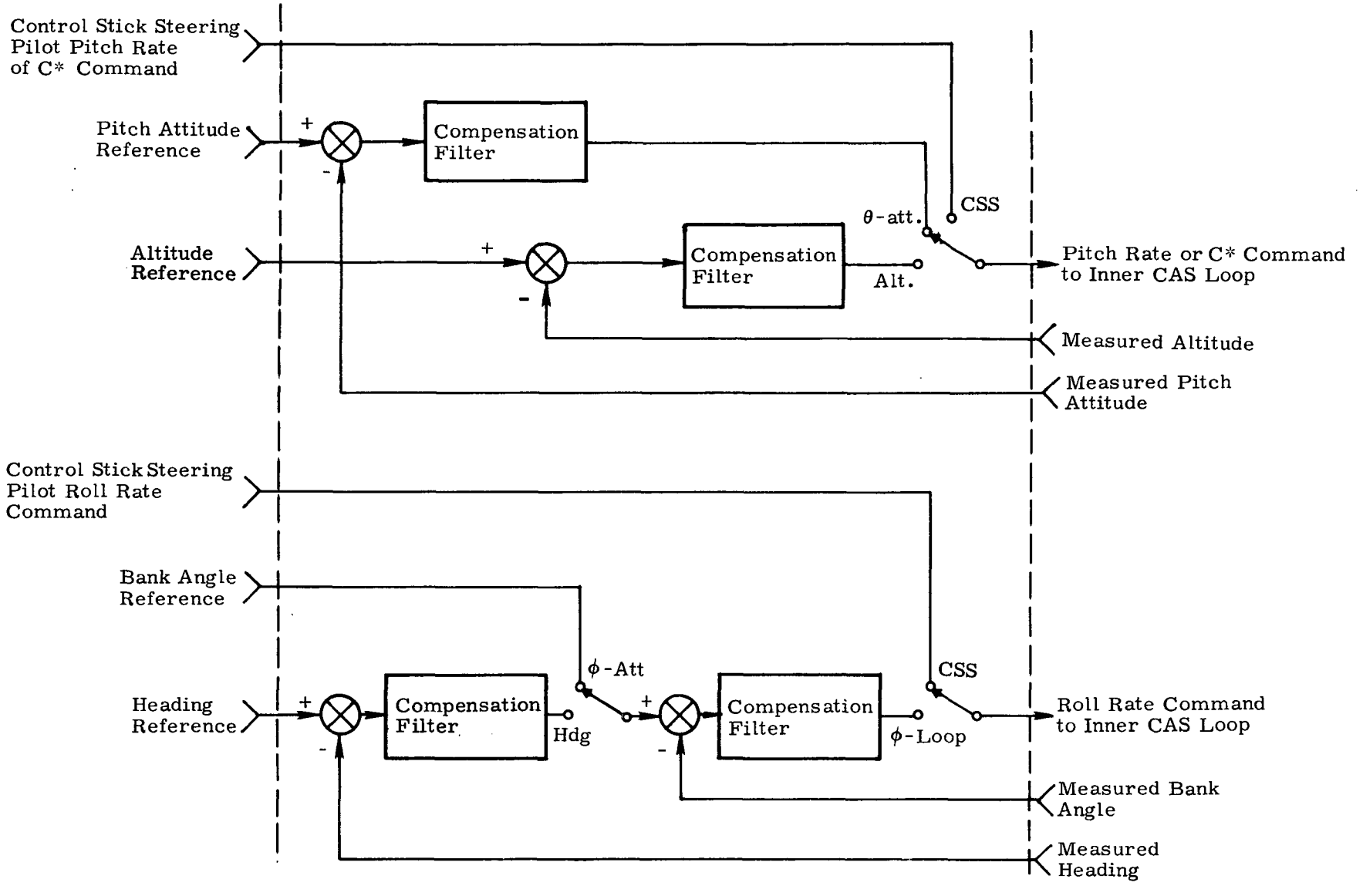



Fig. 4-3 Simplified Functional Block Diagram of Outer Loops for Semiautomatic Modes of Operation

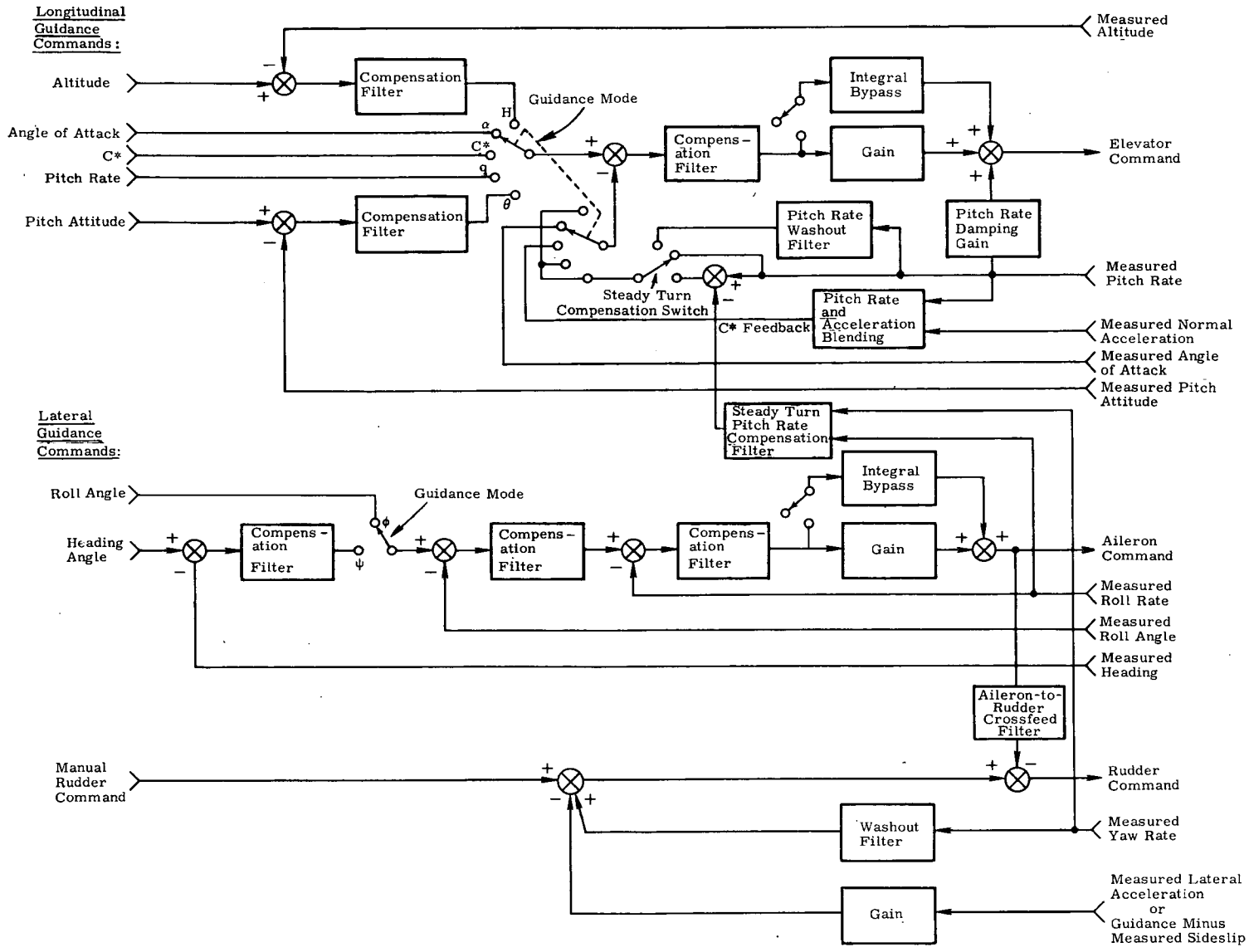



Fig. 4-4 Simplified Functional Block Diagram of the Automatic Mode

as either an altitude loop or a pitch attitude loop, both of which are outer loops closed around the inner CAS loops. Similar to the CAS, an additional longitudinal control mode has $C^*$ feedback. As well as the roll angle loop already discussed, the lateral control system is provided with a heading loop closed around the roll attitude loop.

The rudder loop is implemented similar to the SAS or CAS mode, except that a provision is made to accept sideslip guidance commands and direct or washed-out yaw rate feedback for automatic decrab maneuver. Rudder pedal commands enable this maneuver to be made manually.

It should be mentioned that, although not shown on the simplified block diagram, additional guidance commands can be applied in the automatic mode. These are elevator, aileron, and rudder commands applied directly to the appropriate summing junctions, and roll rate guidance command to the roll rate summing junction.

### 4.2.4 Speed Control

Control of indicated airspeed for the cruise phase is accomplished by two control effectors, the speed brake (also commonly known as "rudder flare"), and the air breathing engine system (ABES). Use of the ABES is anticipated only for purposes of powered horizontal flight operations, but the appropriate control algorithms resulting in throttle command are instituted in the current comprehensive cruise DFCS. Simplified functional block diagrams of the speed control modes are presented in Fig. 4-5.

In BCS and manual modes, which are functionally identical, the pilot's manual commands are passed directly to the control actuators. Although not indicated in Fig. 4-5, speed brake and throttle commands in BCS and manual modes, as in all speed modes, are crossfed to the elevator for the purpose of trimming out resulting pitching moment. Note that BCS and manual control may employ speed brake or throttle or both.

The IAS-hold mode is functionally similar to the automatic guidance mode which provides IAS commands, and is a velocity loop closure with forward loop compensation. Currently, the speed brake loop is implemented with proportional plus-integral compensation, and the throttle loop has lead-integral lag compensation. In this mode either speed brake or throttle is utilized and the nominal elevator mode selection is CAS.

The automatic (direct guidance) mode issues commands to the speed brake and throttle based directly on guidance issued speed brake and throttle commands. This may occur for either speed brake or throttle or both.

Finally, the automatic Terminal Area Energy Management (TAEM) guidance mode receives the TAEM guidance rate command and first passes through switching logic to decide whether the command is to be issued to speed brake or throttle. If

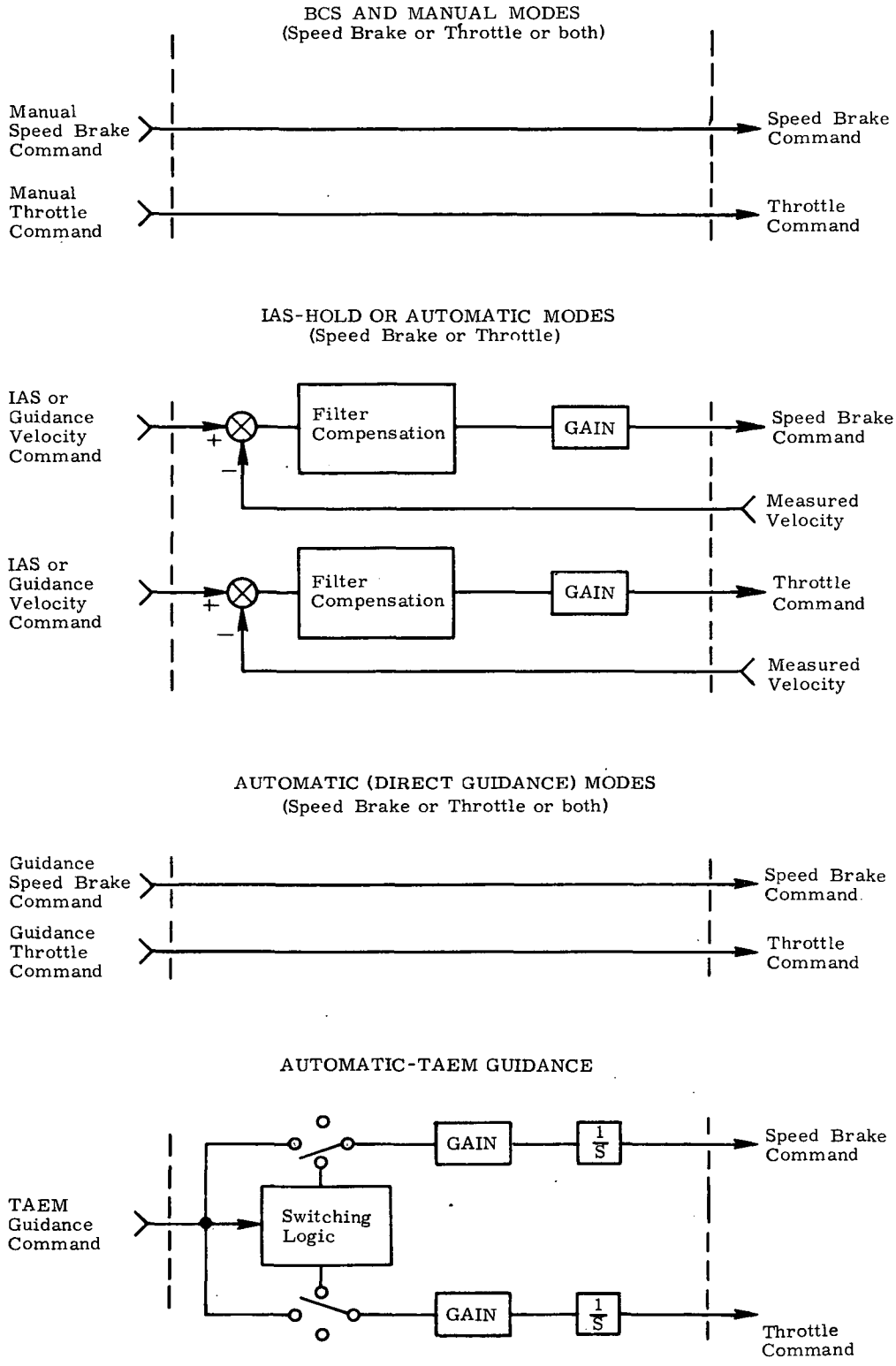



Fig. 4-5 Simplified Functional Block Diagrams - Speed Control

an indicated airspeed decrease is called for and the throttle is presently opened above its minimum, the command is issued to decrease throttle, or otherwise, the speed brake is opened. On the other hand, if an indicated airspeed increase is called for and the speed brake is presently opened above minimum, the command is issued to close the speed brake, or otherwise, the throttle is opened. These commands are scaled appropriately, integrated, and issued to the controller.

### 4.2.5 Body Flap

Control of the body flap, an aft pitching moment generator, is achieved via commands issued according to a piecewise linear body flap versus Mach number profile, biased with a direct pilot manual command (see Fig. 4-6). In addition, appropriate command biases are instituted to provide for full control surface deflection range.

### 4.2.6 Backup Control System (BCS)

The BCS mode is an autonomous, digitally implemented, multi-axis manual control capability which serves as pilot requested backup in the event of primary DFCS failure. Although the design presented here is only of a preliminary nature, it is intended to demonstrate the capabilities currently envisioned for BCS mode plus techniques for their implementation.

Whereas the primary DFCS is likely to be subject to modifications throughout the flight testing and orbital flight operations, the intent for BCS is to obtain an acceptable design prior to the first flight and to leave it unmodified thereafter. As the only BCS coding which is integrated into the primary DFCS structure is the BCS call flag from the executive routine, possibility of subsequent faulty modifications is virtually eliminated, thus providing the pilot with a continuously reliable alternative.

BCS mode is functionally similar to primary manual mode and shares the same simplified functional block diagrams (Figs. 4-1, 4-5, and 4-6). However, there are several fundamental differences which are not indicated: The axis-by-axis, multi-routine format of the primary DFCS is replaced by a single routine which processes all axes in succession; and the filter update, control and filter pushdown (wrap-up) functions are imbedded in the proper sequence. Whereas the primary system will have some gains scheduled as a function of dynamic pressure, a single set of fixed gains which provides suitable performance over as large a cruise flight spectrum as possible will be utilized for BCS. Finally, pilot signal preprocessing is not included for elevator, aileron, and rudder, and there is no Mach number profile for body flap.

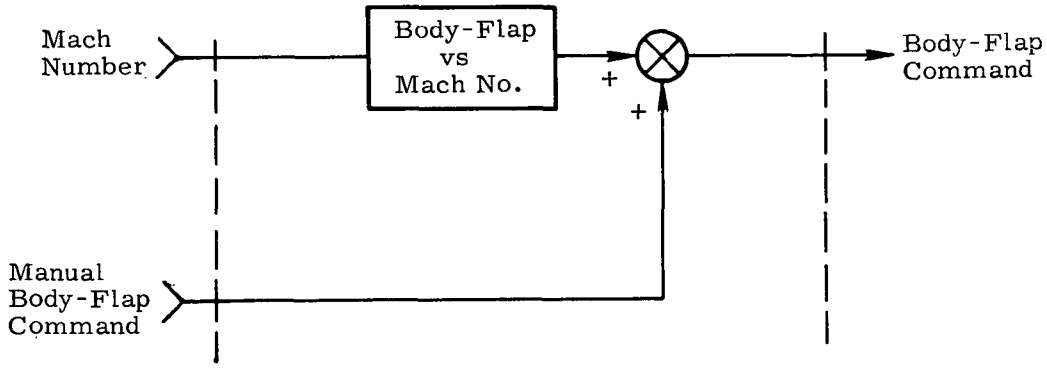



Fig. 4-6 Functional Block Diagram of Body Flap

### 4.2.7 Mode Selection

Mode select/indicate interfaces with the pilot are helpful in understanding the capabilities of the cruise phase DFCS, especially where manual control and a multiplicity of modes complicate the picture. Figure 4-7a is compatible with simultaneous multi-axis mode selection in contrast to axis-by-axis mode selection. Thus it is compatible with earlier versions of the cruise phase DFCS wherein axis-by-axis mode changes were not considered. Multi-axis mode selection is not unlikely for the ultimate cruise phase DFCS version, considering such things as control loop design failure modes, pilot training, and computer input/output simplifications. The gain-select switches and integrator moding select/indicators are on a separate panel. This panel might be associated with the ABES console since these capabilities are intended primarily for the flight test phases of development.

Figure 4-7b, on the other hand, suggests an axis-by-axis mode selection capability compatible with the present cruise phase DFCS design, including all peripheral moding capability. All indicated functions are implemented, except for the pilot selectable pitch turn-coordination-override capability. Gain-select switches are active only in the SAS mode. Pilot-selectable integrator-moding-override capability is provided only in the CAS mode (pitch and/or roll).

## 4.3 Digital Filters and Their Implementation

### 4.3.1 Generalized $n^{th}$-Order Filters

Control filters for the cruising mode are classical, recursive, $n^{th}$-order digital filters, whose Z transform is D(Z):

$$D(Z) = \frac{a_1 + a_2 Z^{-1} + \ldots + a_{2n} Z^{-n}}{1 + a_3 Z^{-1} + \ldots + a_{2n+1} Z^{-n}}$$

The general form of their implementation is presented in block diagram format in Fig. 4-8. The symbol x represents the controlled variable and the a's are either fixed or scheduled gains (currently fixed) which correspond to the gains of form "Gx" which appear in the flow charts and block diagrams presented later in this section.

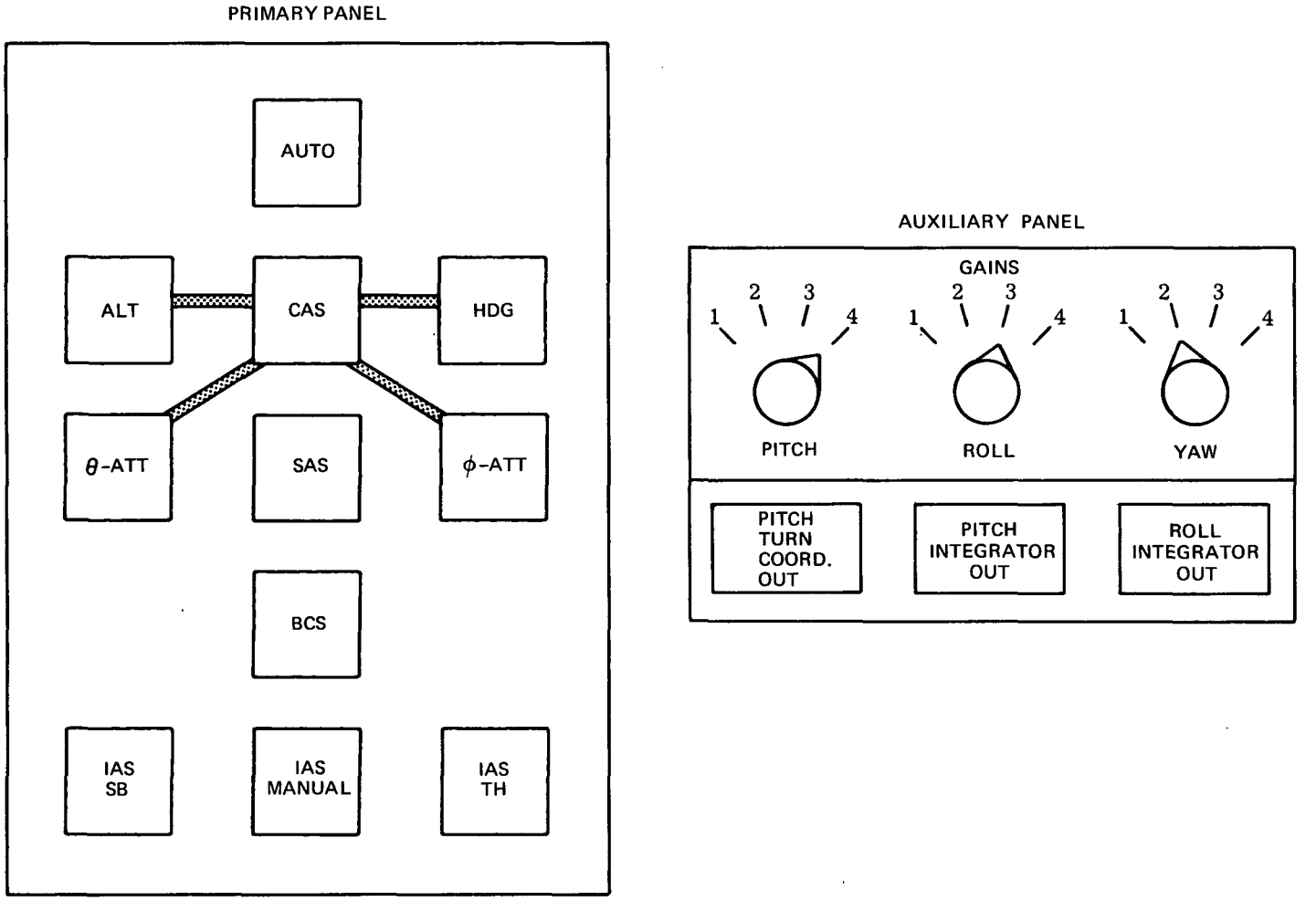



Fig. 4-7a Mode Select/Indicate Panels, Type 1, Cruise Phase DFCS

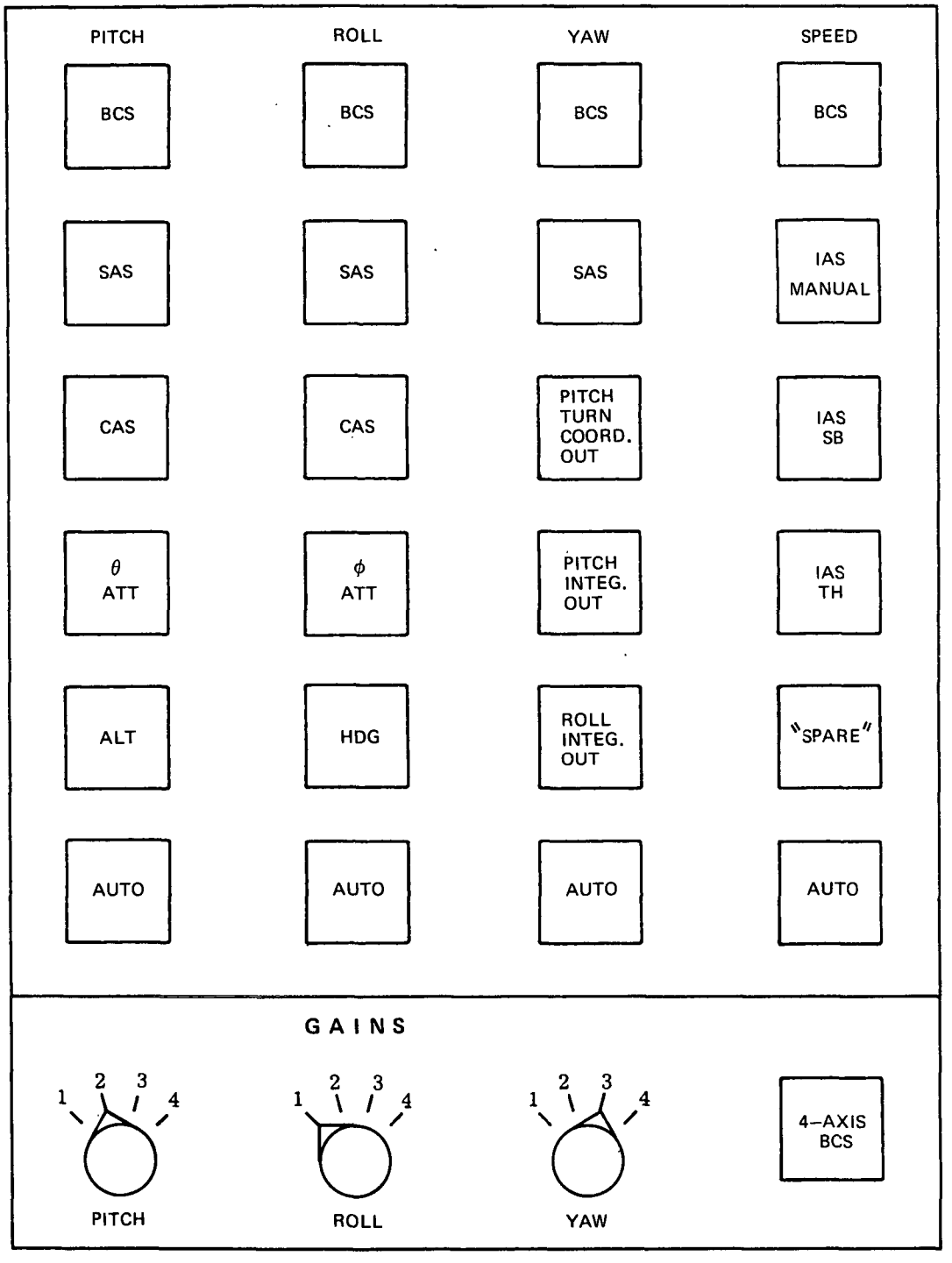



Fig. 4-7b Mode Select/Indicate Panels, Type 2 (Axis-by-Axis) Cruise Phase DFCS

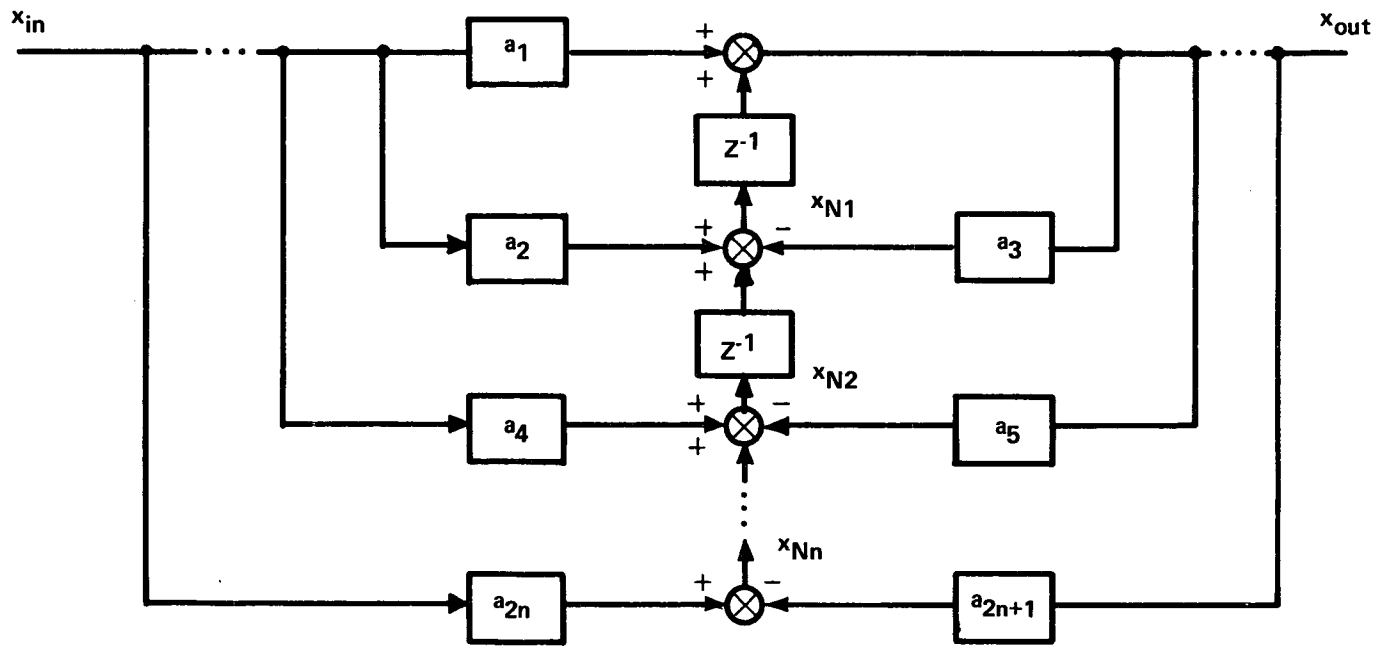



Fig. 4-8 $N^{th}$ -Order Digital Filter

The current filter implementation provides for no greater than $2^{nd}$-order filter capability, but extension to higher order filters (if needed) is straightforward. The $Z^{-1}$ calculation blocks are achieved in the DFCS programming by the relative sequence of instructions, which permits the DFCS cycle time lag to occur at the point so designated. In the coding, each $2^{nd}$-order state filter is divided into two parts in order to minimize the computational transport lag between data input and control command. The first part, which occurs in the Filter Update Routine is limited to updating the filter estimate according to the most current measurement:

$$x_{out} = a_1 x_{in} + x_{N1}$$

Where $x_{Nn}$ represents the $n^{th}$-node, or summing junction, of the filter. Note that exactly n nodes are required to implement an $n^{th}$-order filter.

The Filter Update Routine is sequentially followed by the Control Routine which issues control commands based upon the most current filter output value $x_{out}$. Filter data "pushdown", which is actually the tabulation of the current value associated with each node, is then accomplished in the Filter Pushdown Routine by the following two equations for a 2nd-order filter:

$$x_{N1} = a_2 x_{in} - a_3 x_{out} + x_{N2}$$

$$x_{N2} = a_4 x_{in} - a_5 x_{out}$$

This sequence of instructions results in the following equivalent difference equation for the filter output:

$$x_{out} = a_1 x_{in} + a_2 x_{in_{-1}} + a_4 x_{in_{-2}} - a_3 x_{out_{-1}} - a_5 x_{out_{-2}}$$

Where subscripts -1 and -2 denote past values, one and two samples previous.

Control loops which utilize $2^{nd}$-order filter capability are available for control of the following quantities:

| BCS | Manual Modes | Automatic and Semi-Automatic Modes | Feedback Quantities | Others |
|---|---|---|---|---|
| roll rate | roll rate | roll angle | measured pitch rate | turn coordination |
| pitch rate | pitch rate | pitch angle | measured yaw rate | rudder crossfeed |
| yaw rate | yaw rate | yaw angle | measured lateral acceleration; | |
| | | angle-of-attack | normal acceleration (with gravity) | |
| | | sideslip angle | | |
| | | indicated air-speed | | |
| | | altitude | | |

### 4.3.2 Specialized $1^{st}$-Order Filters

Recursive, $1^{st}$-order washout filters are utilized in the pitch rate and yaw rate feedback paths of the primary system and in the yaw rate feedback path of BCS. The Z transform of the filter is:

$$D(Z) = \frac{1 - Z^{-1}}{1 + G_{xW} Z^{-1}}$$

A block diagram and the update and pushdown equations for the washout filter are presented below

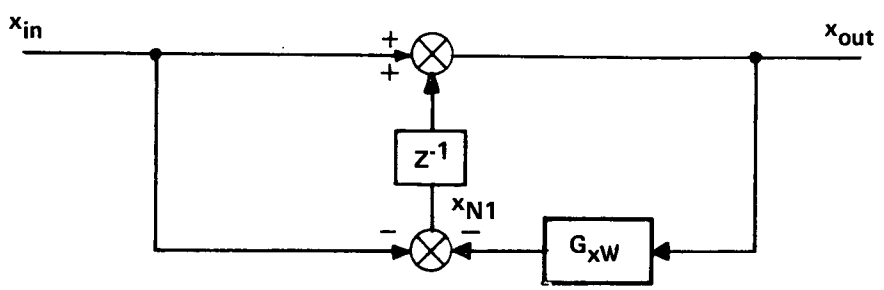



Fig. 4-9 Digital Washout Filter

Filter Update: $x_{out} = x_{in} + x_{N1}$

Filter Pushdown: $x_{N1} = -x_{in} - G_{xW} x_{out}$

Recursive, $1^{st}$-order low pass filters are utilized for the filtering of the following quantities:

1) pilot stick inputs
2) crossfeeds to elevator (from speed brake, throttle, and body flap)
3) elevator trim tracker
4) yaw rate for pitch turn coordination

The Z transform of the low pass filter is

$$D(Z) = \frac{1 + G_x}{1 + G_x Z^{-1}}$$

and the block diagram and update and pushdown equations are presented below

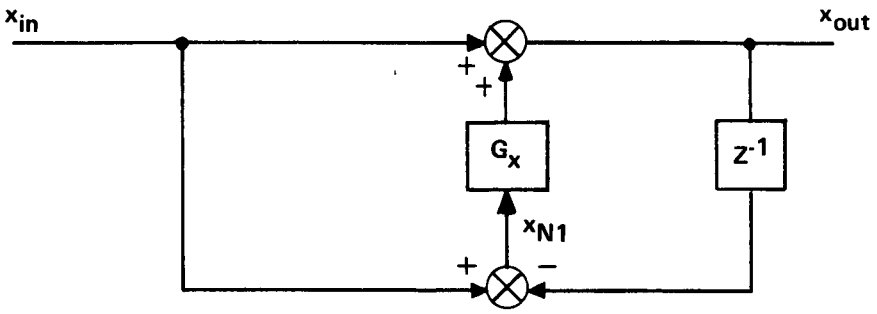



Fig. 4-10 Digital Low Pass Filter

Filter Update: $x_{out} = x_{in} + G_x x_{N1}$

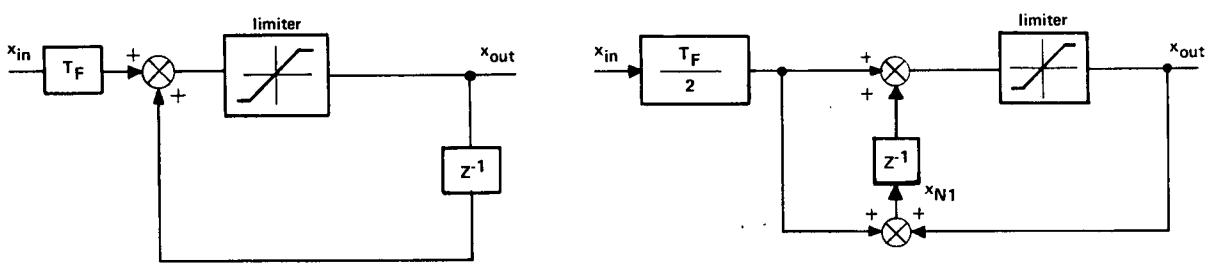
Filter Pushdown: $x_{N1} = -x_{out} + x_{in}$

### 4.3.3 Integrators

Integration is achieved by both trapezoidal and rectangular integrators. Trapezoidal integrators are employed for the integral bypass functions in the elevator and aileron loops, while rectangular integrators are utilized for functions in which trapezoidal integration proves to be of no additional advantage. Thus, rectangular integrators are utilized for the integration of pilot trim discretes (elevator, aileron, and rudder in primary system and BCS) and for the integration of TAEM guidance rate commands (speed brake and throttle). The Z transforms of these integrators are

Rectangular: $D(Z) = \dfrac{T_F}{1-Z^{-1}}$ Trapezoidal: $D(Z) = \dfrac{T_F}{2} \dfrac{1+Z^{-1}}{1-Z^{-1}}$

and the block diagrams and update and pushdown equations are as follows:




Rectangular Integrator Trapezoidal Integrator

Fig. 4-11 Digital Integrator Filters
Rectangular Integrator Trapezoidal Integrator

| | Rectangular | Trapezoidal |
|---|---|---|
| Filter Update: | $x_{out} = x_{out} + T_F x_{in}$ | $x_{out} = \frac{T_F}{2} x_{in} + x_{N1}$ |
| Filter Pushdown: | none | $x_{N1} = \frac{T_F}{2} x_{in} + x_{out}$ |

Note from the block diagrams that the output of the integrators is limited; in practice this occurs in the filter update routine. Also notice that rectangular integration requires no pushdown equation, thus substantiating its use whenever possible.

## 4.4 Preflight Parameter Specifications

Basic parameters for the cruise phase DFCS include filter coefficients, gains, limits, deadbands, logic-control flags, etc. Each of these basic parameters must be numerically specified prior to the cruise phase initialization pass. A representative preflight parameter specification is presented alphabetically: Table 4-1a for non-subscripted parameters and Table 4-1b for subscripted parameters. All numerical data is based upon a single-point design at Mach 0.6 and 300 psf (about 6000 feet) utilizing aerodynamic data for the February 1973 AERO16 version of the 89B orbiter. The RCAH version of the CAS elevator mode is indicated (ICSTAR = 0); the rudder sideslip feedback option is chosen (IBFDBK = 1); pitch turn coordination is bypassed (IQTRN = 1).

Most of the basic parameters are utilized directly, but a number of "working" quantities are employed, such as sample-interval-multiplied trim rates or $\alpha$-guidance values of CAS filter coefficients, limits and loop gains. These are computed on the initialization first pass (Fig. 4-24) or upon mode initialization as required (Fig. 4-25 through 4-28).

Earlier orbiter cruise phase DFCS designs have utilized scheduled gains and coefficients to provide desirable response characteristics over the cruise phase flight envelope (including Powered Horizontal Flight). The scheduling parameter has been dynamic pressure. The scheduling operations (when enabled) are carried out every slow pass by the Parameter Estimation Routine (Fig. 4-12), which operates on preflight-specified tables of parameter value vs dynamic pressure. Typically seven data points are used: 100, 150, 225, 300, 425, 525, and 650 psf. The thirteen parameters corresponding to the scheduled quantities of earlier designs are identified by asterisks in Table 4-1. Tabular specifications for these gains and coefficients will be added to the preflight parameter specification tables, as they become available.

## 4.5 Input/Output Summaries

Figures 4-13a through 4-13e detail those quantities which are input to and output by the five basic control channels (speed, elevator and body flap, aileron, rudder and BCS). Input includes information transferred across the various interface boundaries – executive, guidance, and pilot, as well as miscellaneous moding information, crossfeeds from other loops, and sensor measurements. Output contains control effector commands, pertinent mode status for display, and crossfeeds to other loops.

## Table 4-1a Preflight Parameter Specifications for the Cruise Phase

(Non-subscripted Parameters)

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | AINTMX | .2400 | | ETMN | -.1745 | | GFSB | 0. |
| | CCAMN | -.0873 | | ETMX | .0654 | | GET | 0. |
| | CCAMX | .3491 | * | GAINT | .707 | | GETB | 0. |
| | CCCMN | -.0873 | * | GAP | .288 | | GETF | -.9900 |
| | CCCMX | .3491 | | GAPMB | .288 | | GEX | .2327 |
| | DACMX | .4800 | | GAX | .2400 | | GEXB | .2327 |
| | DECMN | -.6981 | | GAXB | .2400 | | GFX | .7679 |
| | DECMX | .2618 | | GCFQM | 12.422 | | GFXB | .7679 |
| | DFCMN | -.3840 | | GCH | .3 | * | GPHPS | 4. |
| | DFCMNB | -.3840 | * | GCTH | 1.4 | * | GPPH | 1.19 |
| | DRCMX | .2618 | | GCX | -.1163 | | GPX | .1745 |
| | DSCMN | -.3072 | | GDFC | -.9048 | | GQW | -.9802 |
| | DSCMNB | -.3072 | | GDSC | -.3679 | | GRAY | 0. |
| | DSCMX | 1.2217 | | GDTC | -.9753 | * | GRB | .314 |
| | DSDMX | .1257 | * | GECA | -.5 | | GRCPH | 0. |
| | DTCMN | 500 | * | GECC | -1. | | GRDAC | 0. |
| | DTCMNB | 500 | | GEF | 0. | | GRDACB | 0. |
| | DTCMX | 18000. | | GEFB | 0. | | GRM | -.9802 |
| | DTDMX | 8750. | * | GEINTA | .4 | | GRRE | 0. |
| | DTDMXB | 8750. | * | GEINTC | .707 | * | GRRWA | 1.19 |
| | FINTMN | -.3491 | * | GFQMA | .2 | | GRRWB | 1.19 |
| | FINTMX | .1309 | | GFQMB | 1. | | GRW | -.9802 |
| | ETDB | .0087 | * | GFQMC | 0. | | GRWB | -.9802 |
| | ETDMX | .0174 | | GFS | 0. | | GRXAUT | .2094 |

*Scheduled Gains

Table 4-1a Preflight Parameter specifications for the Cruise Phase (Cont)

(Non-subscripted Parameters)

| | | | | | |
|---|---|---|---|---|---|
| GRXB | .2094 | ICSTAR | 0 | XFMX | .5406 |
| GRXSAS | .2094 | IPHRC | 1 | XETMN | -0.75 |
| GSDDG | 1. | IQTRN | 1 | XETMNB | -0.75 |
| GSU | 1. | IRWP | 1 | XETMX | 0.75 |
| GSX | 2.0385 | PCMX | .3491 | XETMXB | 0.75 |
| GSXB | 2.0385 | PHCMX | .5236 | XETR | 0.075 |
| GTDDG | 152.25 | PMDIS | .0873 | XETRB | 0.3 |
| GTU | 8.3X10000 | QCTMX | .0367 | XFMMN | 0. |
| GTX | 35000 | TFB | .1 | XFMMNB | 0. |
| GTXB | 35000 | XAB | 0. | XRB | 0. |
| GXA | -.6065 | XADB | .02 | XRDB | .02 |
| GXAX | 1. | XADMX | .5 | XRDMX | .25 |
| GXE | -.6065 | XAMDIS | .05 | XRMX | 1. |
| GXEX | 2. | XAMX | 1. | XRTMX | .3125 |
| GXR | -.6065 | XATMX | .5 | XRTMXB | .3125 |
| GXRX | .25 | XATMXB | .5 | XRTR | .0156 |
| GXSX | .25 | XATR | .05 | XRTRB | .0625 |
| GXSXB | .25 | XATRB | .2 | XSMDIS | .1 |
| GXTX | .5 | XEB | 0. | XSMMN | 0. |
| GXTXB | .5 | XEDB | .02 | XTMDIS | .1 |
| IABES | 1 | XEDMX | .3 | XTMMN | 0. |
| IANGRV | 3 | XEMDIS | .05 | | |
| IBFDBK | 1 | XEMN | -1 | | |

*Scheduled Gains

Table 4-1b Preflight Parameter Specifications for the Cruise Phase

(Subscripted Parameters)

```
DFCMCH(I,J) (-.2094, .1), (-.2094, .2), (-.2094, .5),
            (-.0873, .52), (-.0873, 6), (.1745, 8), (.1745, 9)
GAN(I)    1., 0., 0., 0., 0.
GAPMS(I) .144,.2304,.288,.3456
GAYM(I)   1., 0., 0., 0., 0.
GBE(I)    1., 0., 0., 0., 0.
GCEA(I)   1., 0., 0., 0., 0.
GCEC(I)   1., 0., 0., 0., 0.
GDAC(I)   1., 0., 0., 0., 0.
GEOMS(I) .5, .8, 1., 1.2
GHE(I)    1., -.9905, .01, 0., 0.
GPE(I)    1., 0., 0., 0., 0.
GPHE(I)   1., 0., 0., 0., 0.
GPHT(I)   .1, 0., -.999, 0., 0.
GPM(I)    1., 0., 0., 0., 0.
GPMB(I)   1., 0., 0., 0., 0.
GPSE(I)   1., 0., 0., 0., 0.
GQM(I)    1., 0., 0., 0., 0.
GQMB(I)   1., 0., 0., 0., 0.
GRE(I)    1., 0., 0., 0., 0.
GREB(I)   1., 0., 0., 0., 0.
GRRWS(I) .595, .952, 1.19, 1.428
GSUE(I)   1., -.93, -1., 0., 0.
GTHE(I)   1., 0., 0., 0., 0.
GTUE(I)   1., -1.9713, -1.905, .9716, .9048
```

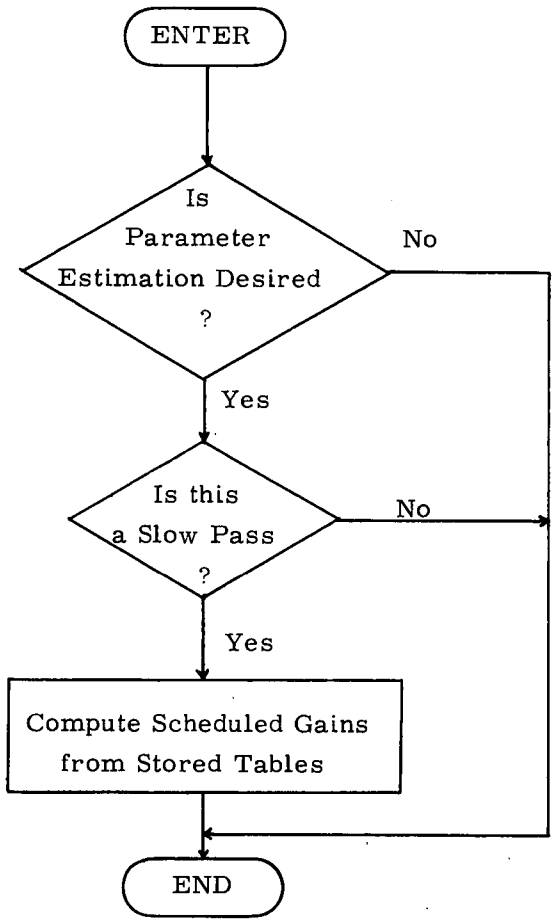



Fig. 4-12 Parameter Estimation Routine for Cruise Phase

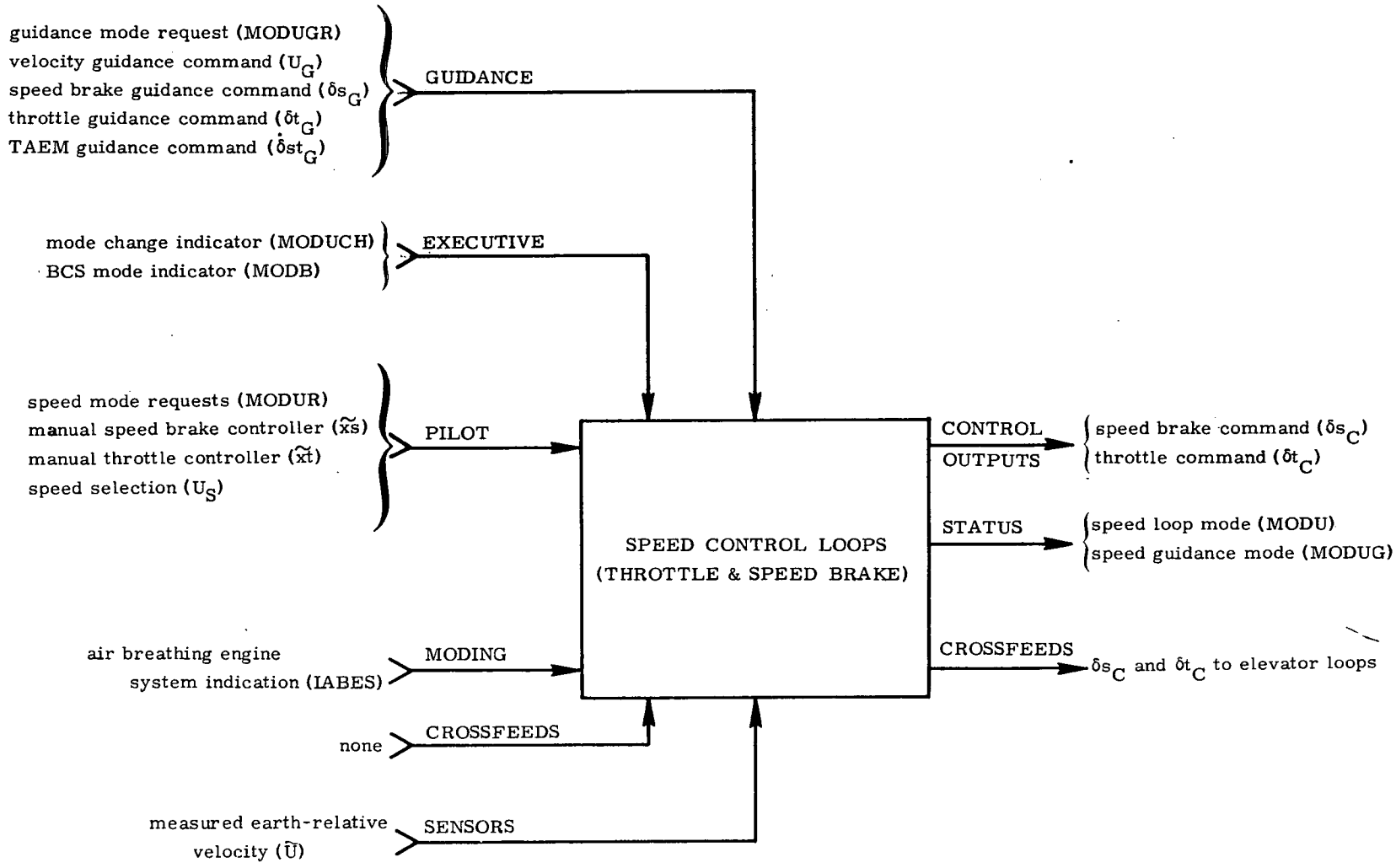



Fig. 4-13a Speed Control Input/Output Interface Variables

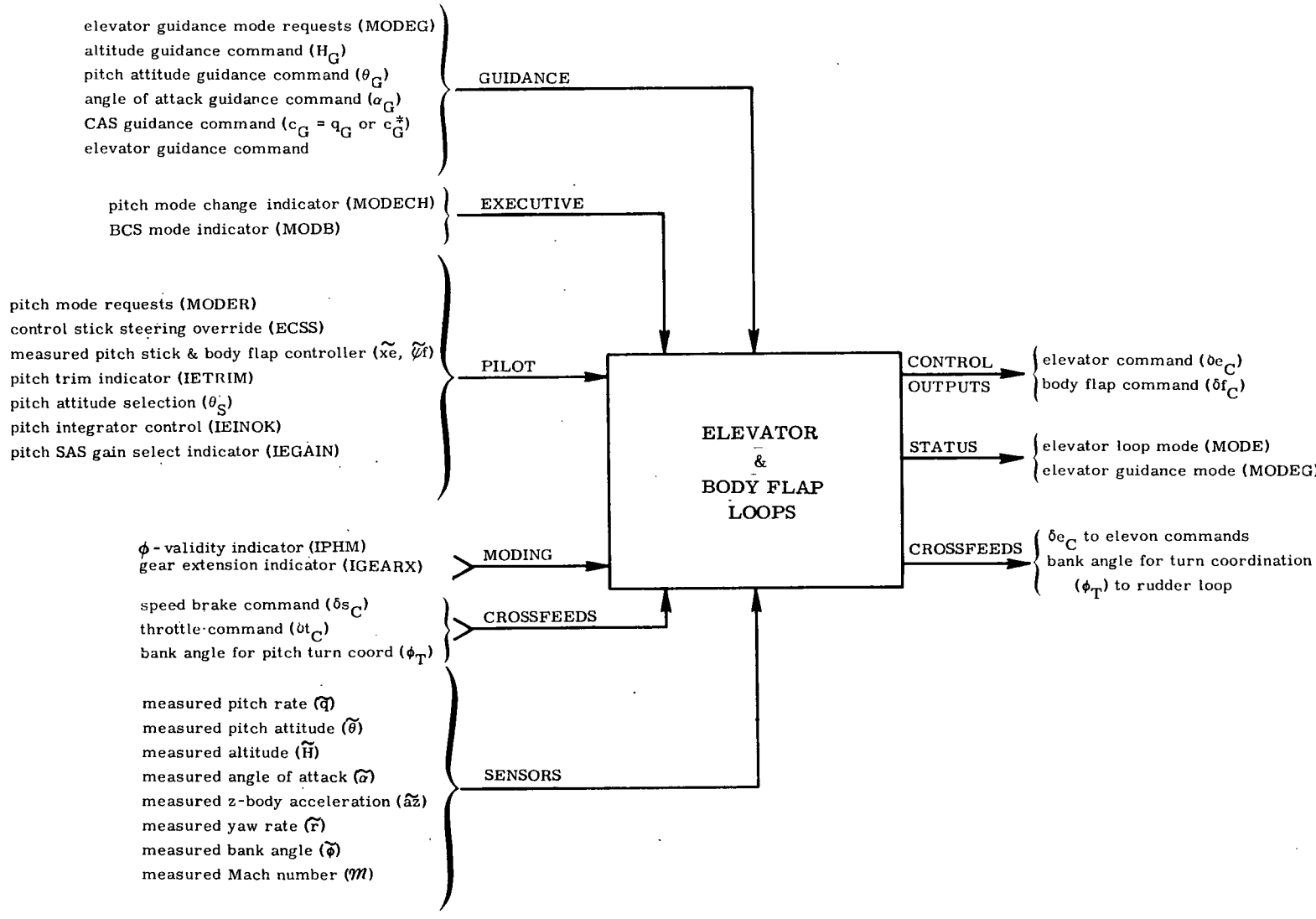



Fig. 4-13b Elevator and Flap Input/Output Interface Variables

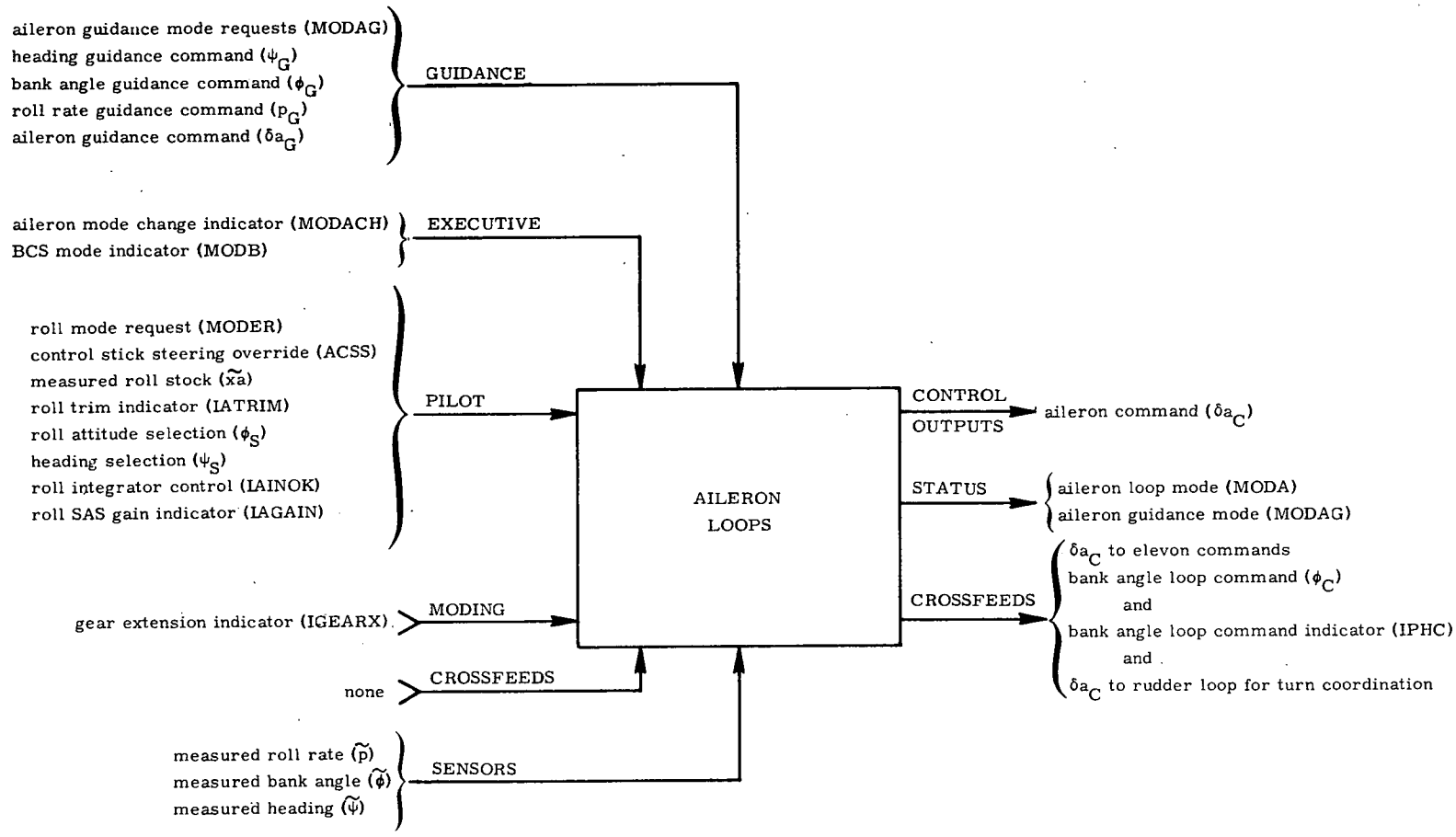



Fig. 4-13c Aileron Input/Output Interface Variables

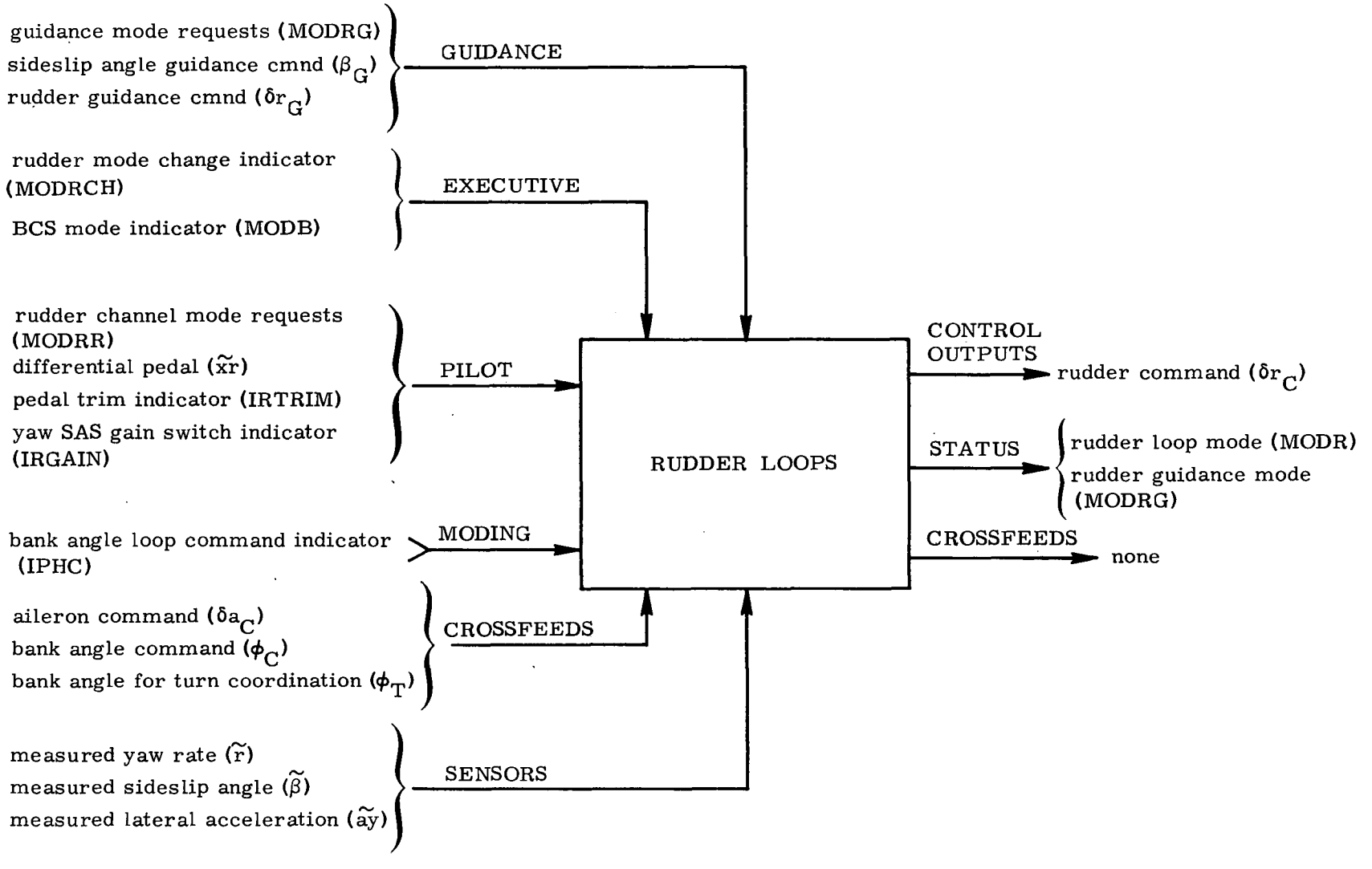



Fig. 4-13d Rudder Input/Output Interface Variables

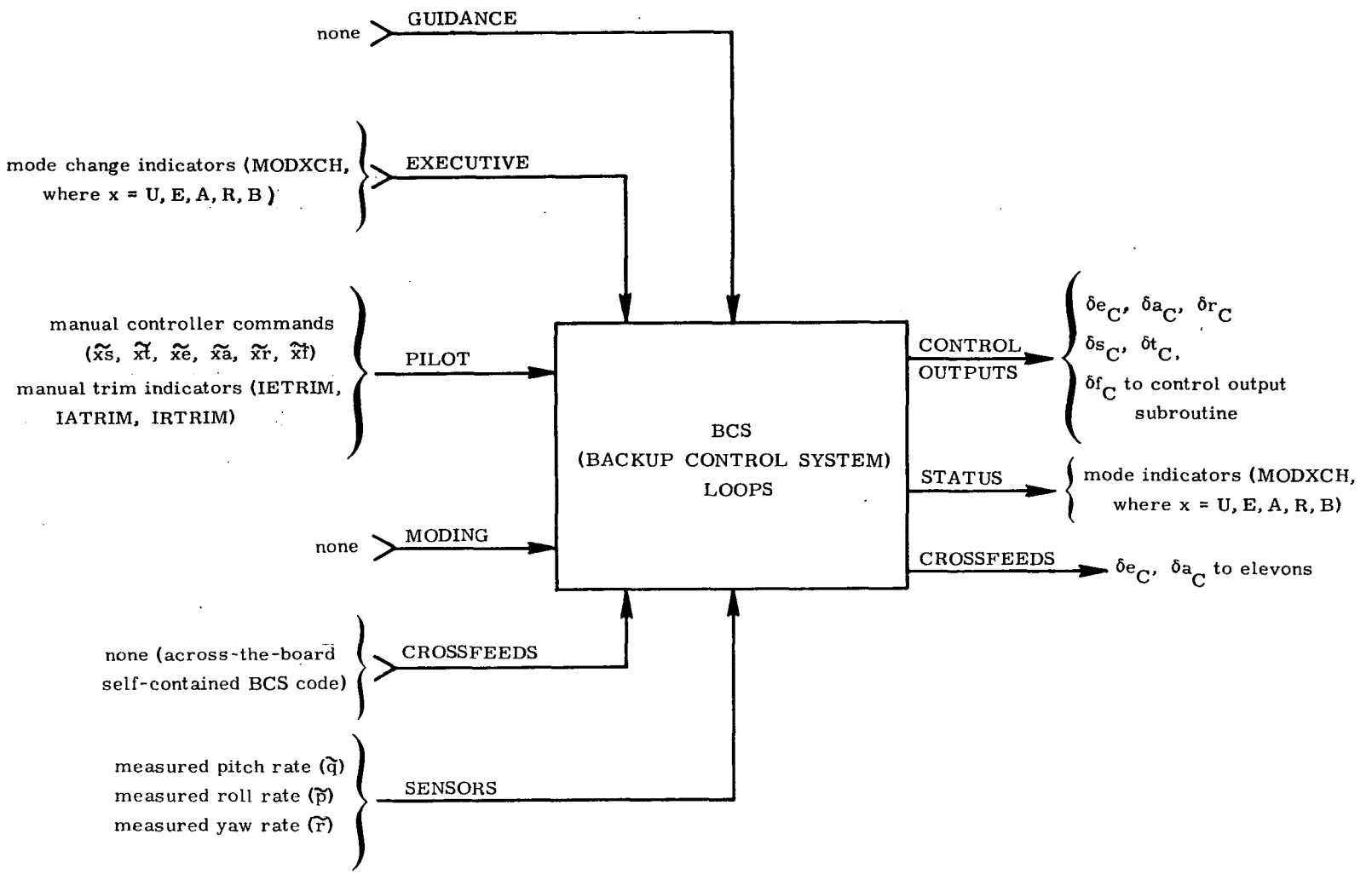



Fig. 4-13e Backup Control System Input/Output Interface Variables

## 4.6 Functional Block Diagrams for Cruise Phase DFCS

This section presents a detailed control-engineer's description of the cruise phase DFCS, using functional block diagrams. These include all the feedbacks, crossfeeds, limiters, switches, and compensation filters currently available with preflight parameter specification options.

A number of simplifying conventions have been employed, particularly in the areas of filter and limiter descriptions. These conventions are summarized in Fig. 4-14.

Tables 4-2 through 4-4 indicate the realizable cruise phase DFCS modes of operation, and the corresponding settings of all flags. Note for example, that while MODS = 3 (speed brake using u-filter for IAS-hold) and MODT = 3 (throttle using u-filter for IAS-hold) are both valid modes of operation, they must not occur simultaneously. The mode change Initialization Subroutine (Fig. 4-25 through 4-28) handles all the setting of all derived flags, according to the logic indicated in these tables.

Pilot input signal processing for speed brake throttle and flap controllers, for pitch and roll sidearm stick, and for pedals, is shown in Fig. 4-15. Pitch, roll and yaw manual trim circuits are shown in Fig. 4-16. Outer loop (semi-automatic or automatic) error signal generation is shown in Fig. 4-17. Note that the pilot "Select" option is implied (except for altitude) simply by "writing" into the appropriate storage location, e.g., $H_{REF}$.

Speed control, including throttle and speed brake options, is shown in Fig. 4-18a, which indicates an input from TAEM-guidance interface signal processing as shown in Fig. 4-18b. Elevator control loops are shown in Fig. 4-19. Also shown in Fig. 4-19 are the generation of $\phi_{TF}$ for turn coordination (pitch and yaw rate command circuits), the generation of body flap commands (manual override of a Mach profile), and an elevator trim-tracker network akin to conventional stabilizer trim-tracking. (This trim indication is being studied for trim-initialization on emergency downmodes.)

Figure 4-20 presents the aileron control loops, and Fig. 4-21 presents the rudder control loops, including provision for both $\beta$ feedback and "$\dot{\beta}$" feedback (a combination of yaw rate command, yaw rate feedback without washout filtering, lateral acceleration, and aileron). Note that pedal and trim are available even in auto mode (through the gain GRXAUT) for manual decrab maneuvers.

Figure 4-22 indicates the override capability of the BCS mode, indicated by MODB = 1. Details of BCS loops (essentially the corresponding MANUAL or SAS loops with independently specified gains, coefficients, and nonlinearities) are given in Fig. 4-23.

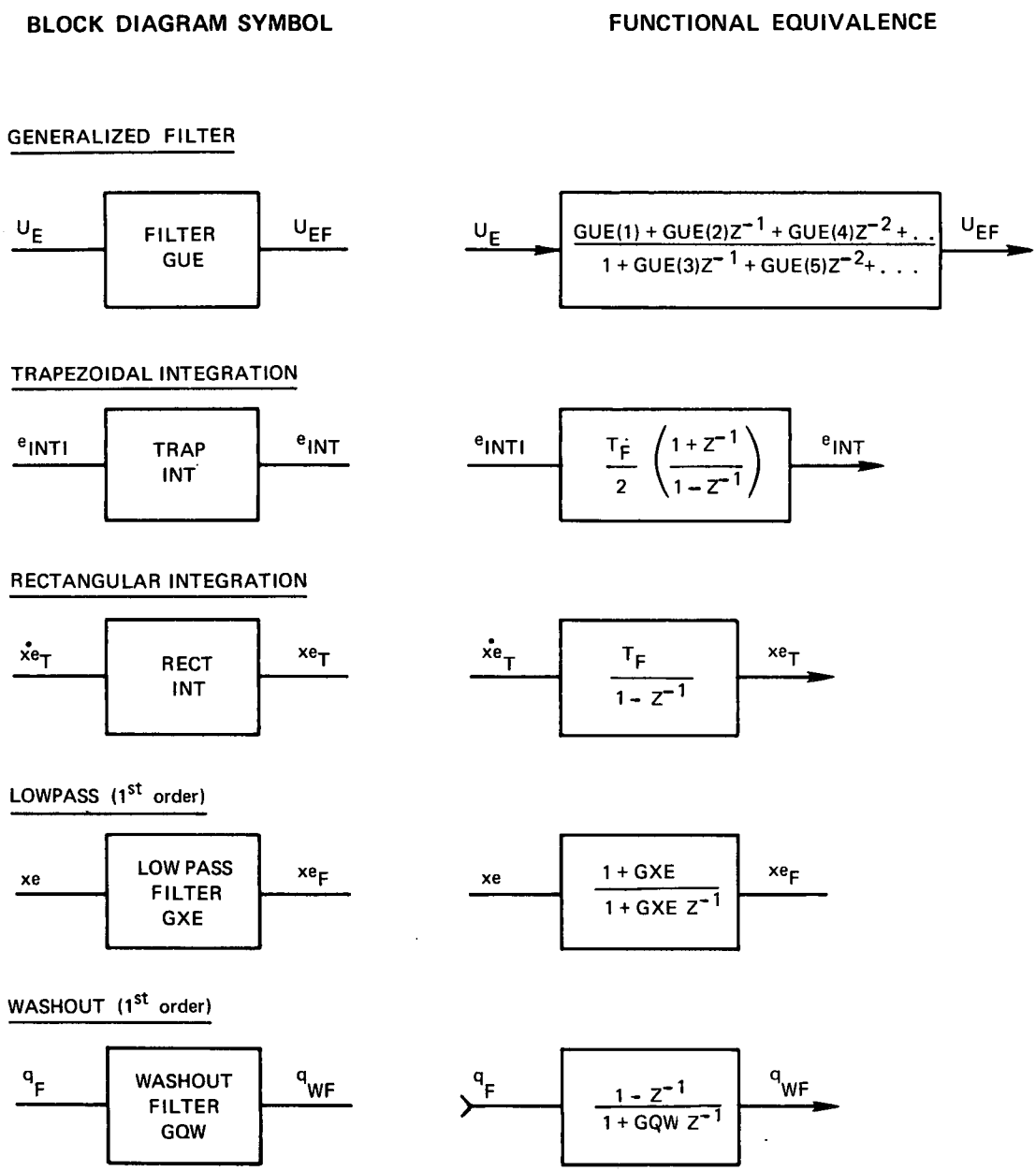



Fig. 4-14 Simplifying Conventions for Block Diagrams

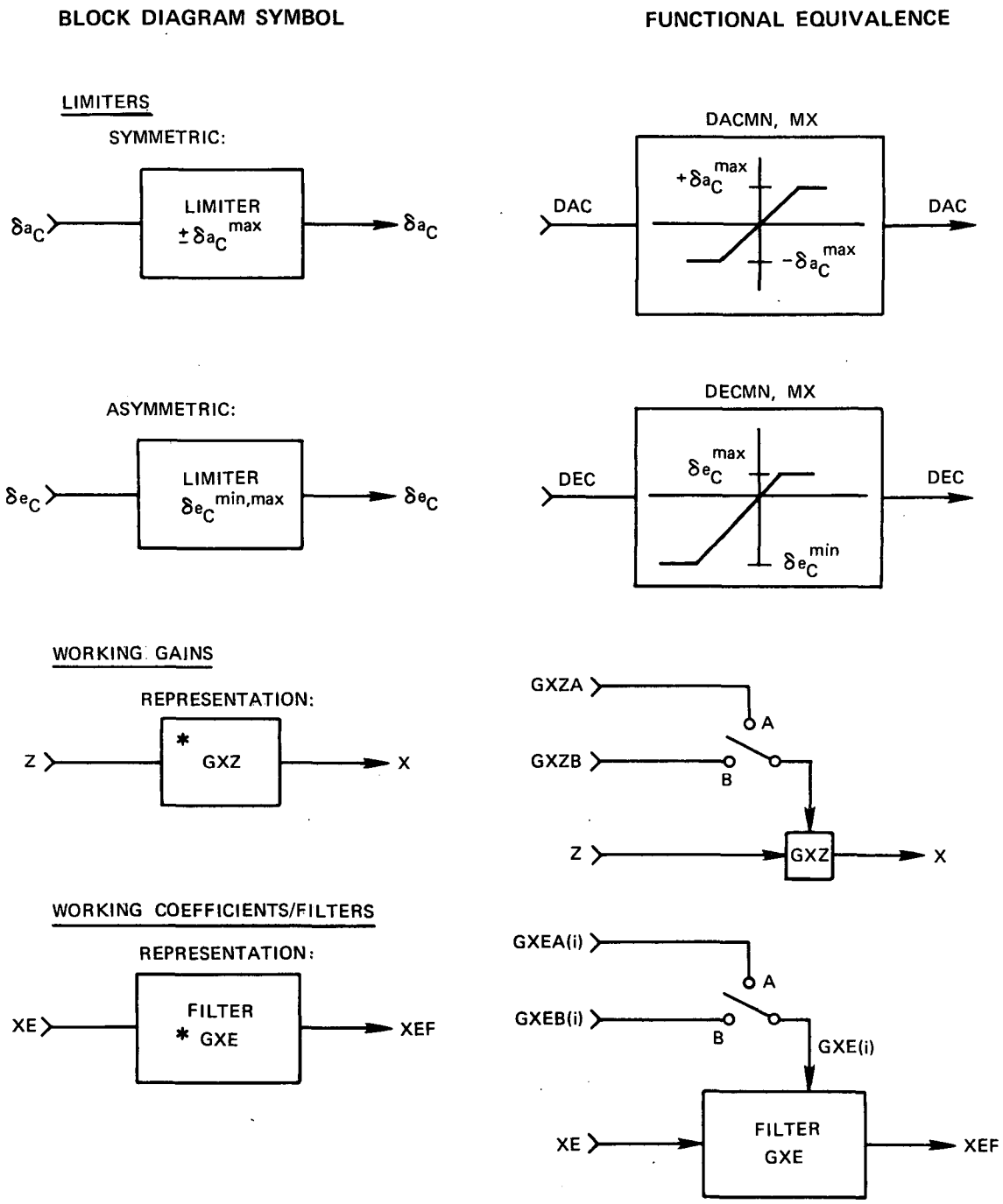



Fig. 4-14 Simplifying Conventions for Block Diagrams (Cont)

Table 4-2 Logic for Speed Control Flag Settings

| | Input Flags | | Derived Flags | | | |
|---|---|---|---|---|---|---|
| | MODU | MODUG | MODS | MODT | MODUFT | MODUFP |
| BCS, UBCS | 1 | X | 1 | 0 | X | X |
| | 1 | X | 1 | 1 | X | X |
| UMAN | 2 | X | 2 | 0 | 0 | 1 |
| (IAS - MAN) | 2 | X | 2 | 2 | 0 | 1 |
| UASB | 3 | X | 3 | 0 | 1 | 2 |
| (IAS - SB) | 3 | X | 3 | 2 | 1 | 2 |
| AUTO, UAUT | 4 | 1 | 5 | 0 | 0 | 1 |
| | | 1 | 5 | 5 | 0 | 1 |
| | | 2 | 4 | 0 | 1 | 3 |
| | | 2 | 4 | 2 | 1 | 3 |
| | | 3 | 4 | 5 | 1 | 3 |
| | | 4 | 2 | 4 | 2 | 3 |
| | | 5 | 5 | 4 | 2 | 3 |
| | | 6 | 6 | 0 | 0 | 4 |
| | | 6 | 6 | 2 | 0 | 4 |
| | | 7 | 6 | 5 | 0 | 4 |
| | | 8 | 2 | 6 | 0 | 4 |
| | | 9 | 5 | 6 | 0 | 4 |
| | | 10 | 6 | 6 | 0 | 4 |
| UATH | 5 | X | 2 | 3 | 2 | 2 |
| (IAS - TH) | | | | | | |

| Table 4-3 Logic for Elevator Control Flag Settings | | | | | | | |
|---|---|---|---|---|---|---|---|
| | Input Flags | | Derived Flags | | | | |
| | MODE | MODEG | ICC1 | ICC2 | ITHC | IHC | ICF |
| BCS, EBCS | 1 | X | X | X | X | X | X |
| SAS, ESAS | 2 | X | X | X | X | X | X |
| CAS, ECAS | 3 | X | 1 | X | X | X | 1, 2 |
| AUTO, EAUT | 4 | 1 | 4 | X | X | X | 1, 2 |
| | 4 | 2 | 3 | 1 | 2 | X | 1, 2 |
| | 4 | 3 | 5 | X | X | X | 3 |
| | 4 | 4 | 3 | 2 | X | 2 | 1, 2 |
| EATT | 5 | X | 2 | 1 | 1 | X | 1, 2 |
| EALT | 6 | X | 2 | 2 | X | 1 | 1, 2 |

| Table 4-4 Logic for Aileron Control Flag Settings | | | | | |
|---|---|---|---|---|---|
| | Input Flags | | Derived Flags | | |
| | MODA | MODAG | IPC | IPHC | IPSC |
| ABCS | 1 | X | X | X | X |
| ASAS | 2 | X | X | X | X |
| ACAS | 3 | X | 1 | X | X |
| AUTO | 4 | 1 | 4 | 0 | X |
| | 4 | 2 | 3 | 4 | X |
| | 4 | 3 | 3 | 3 | 2 |
| AATT | 5 | X | 2 | 1 | X |
| AHDG | 6 | X | 2 | 2 | 1 |

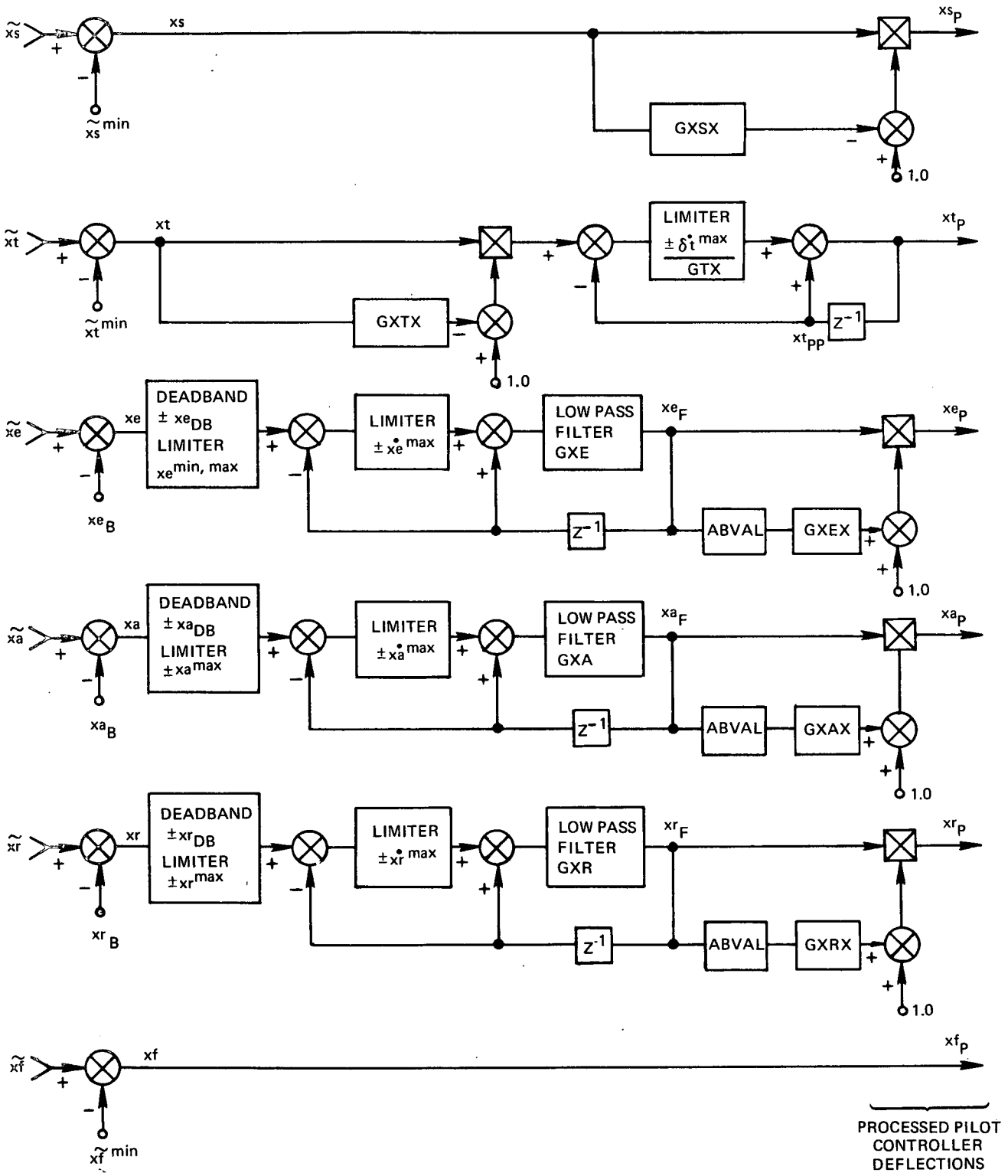



Fig. 4-15 Block Diagram, Pilot Input Signal Processing

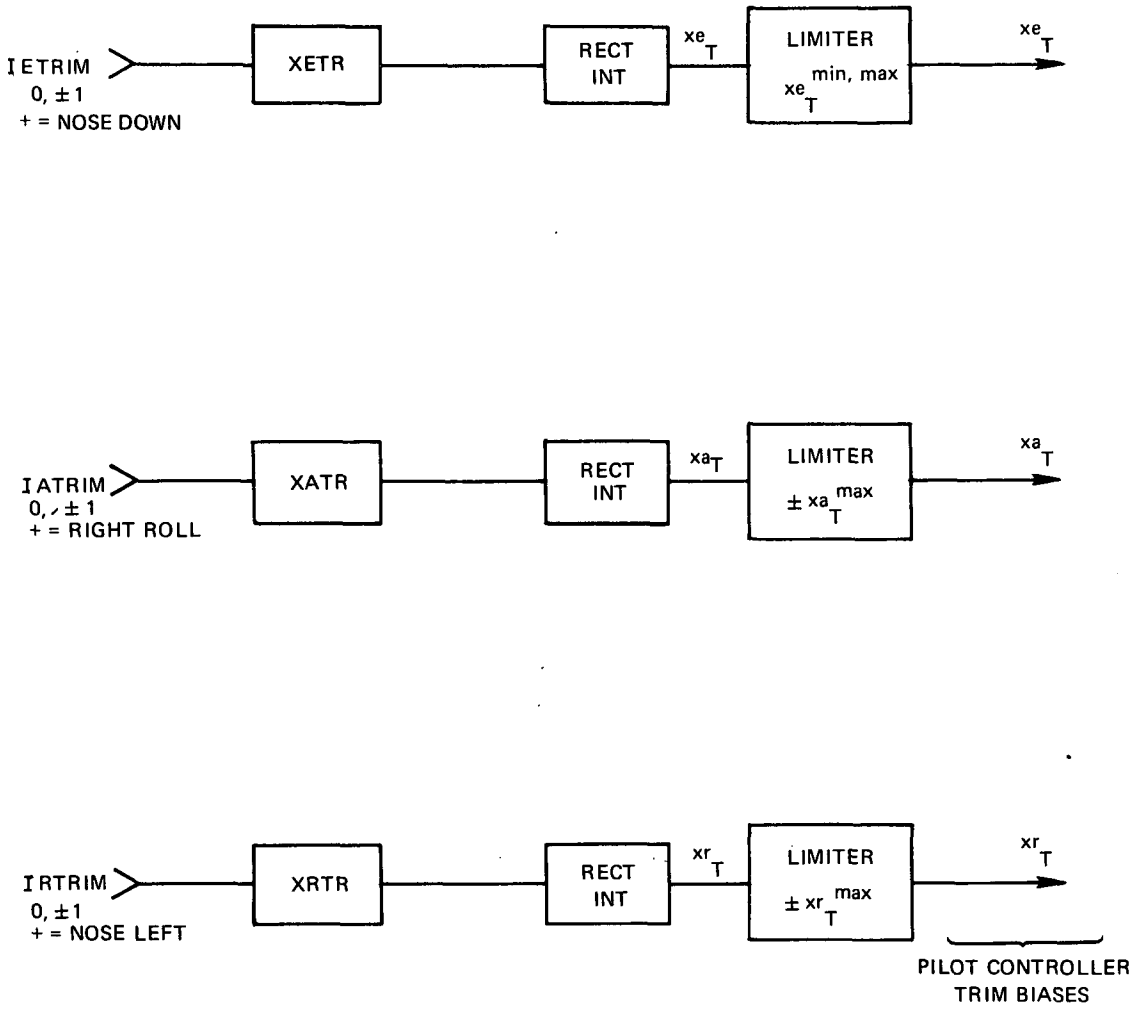



Fig. 4-16 Block Diagram, Pilot Manual Trim Circuits

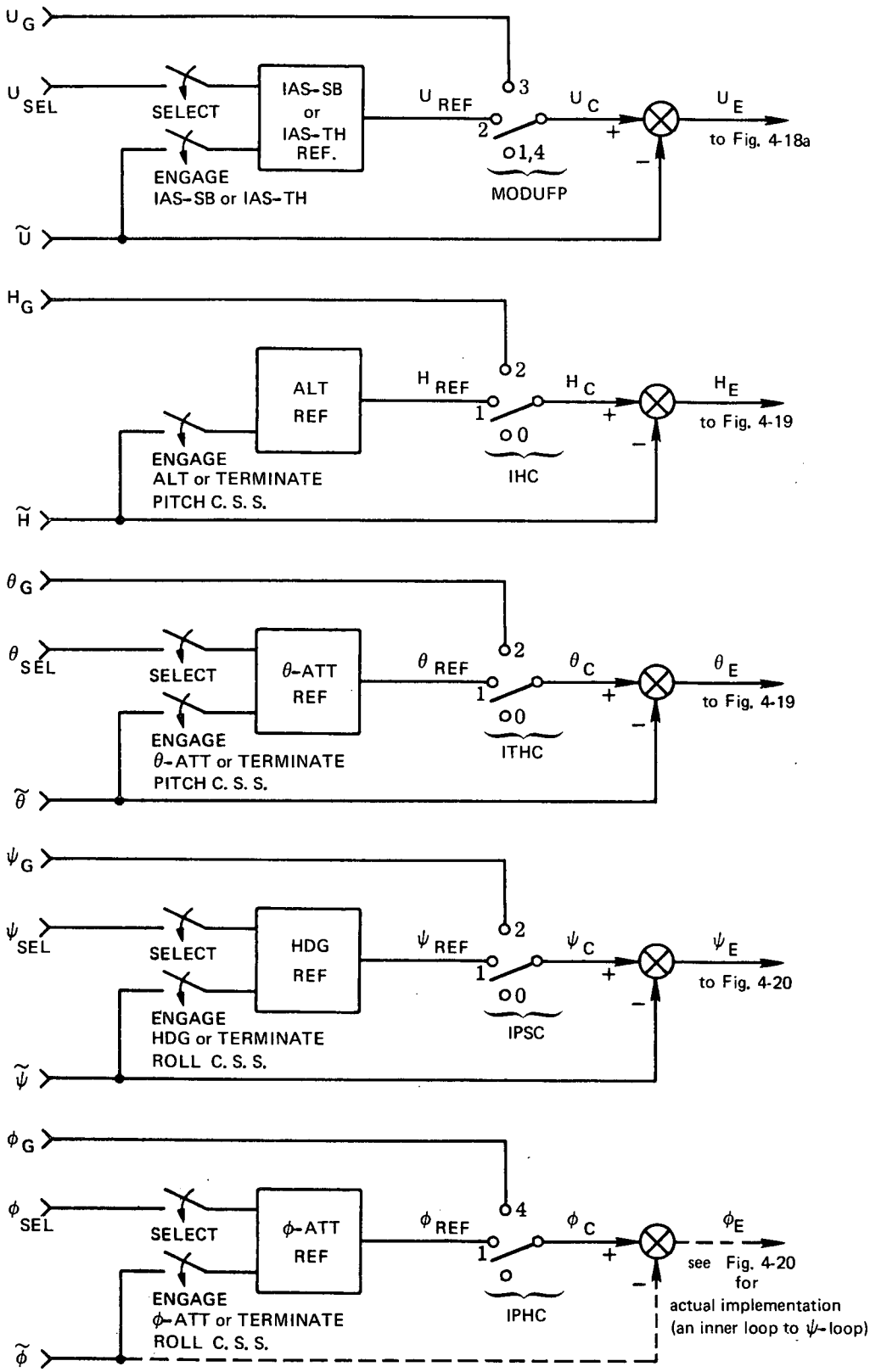



Fig. 4-17 Block Diagram, Outer Loop Error Signal Generation

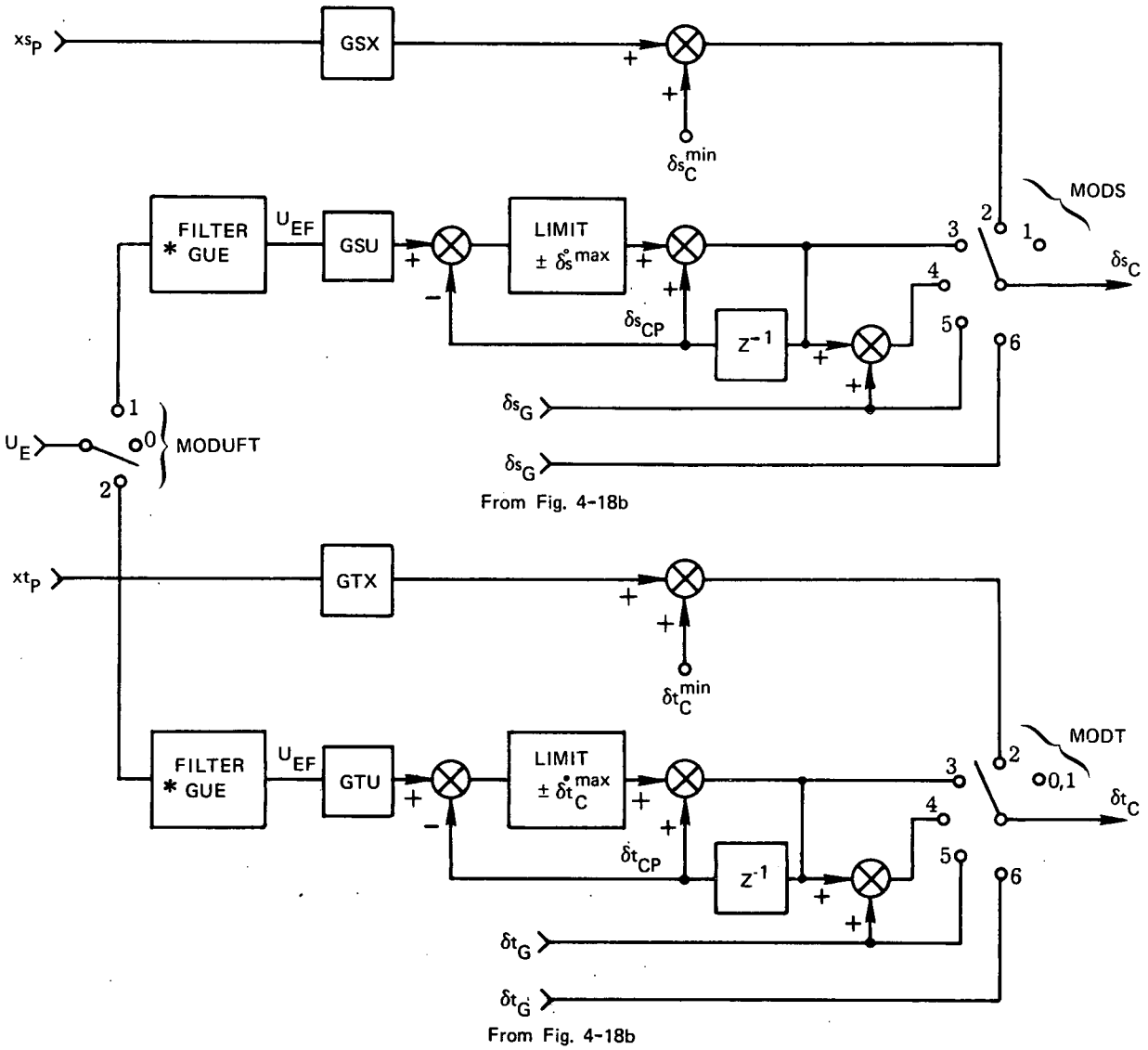



Fig. 4-18a Speed Control Loops

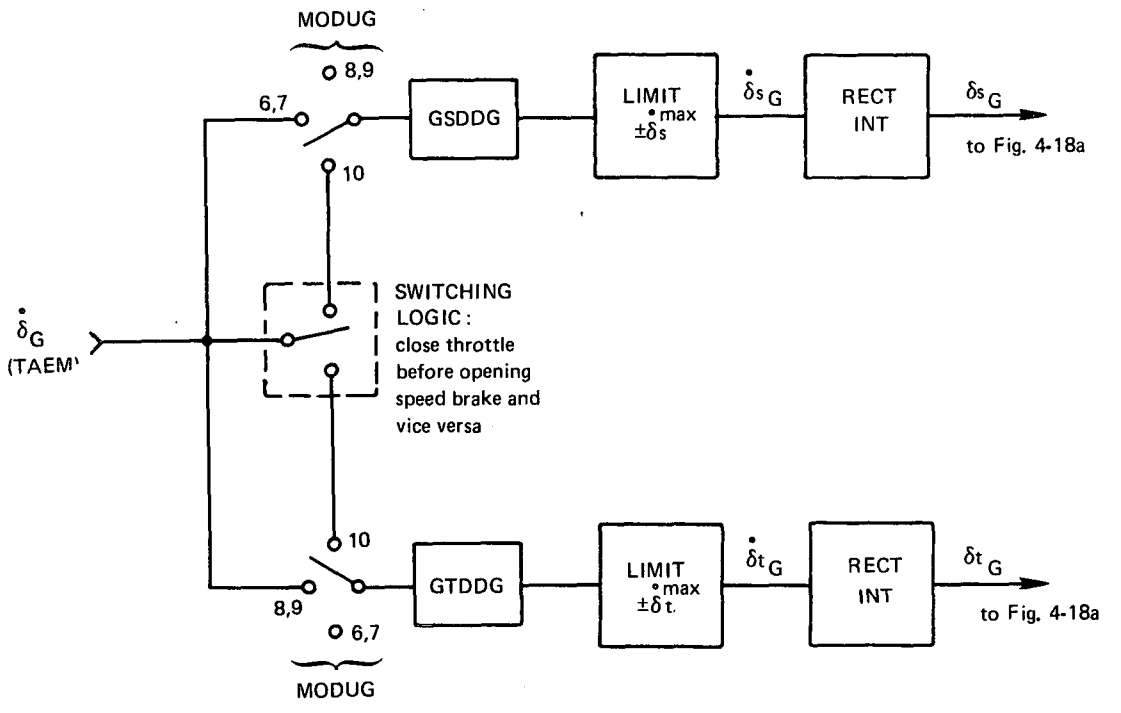



Fig. 4-18b TAEM Guidance Processing for Speed Control

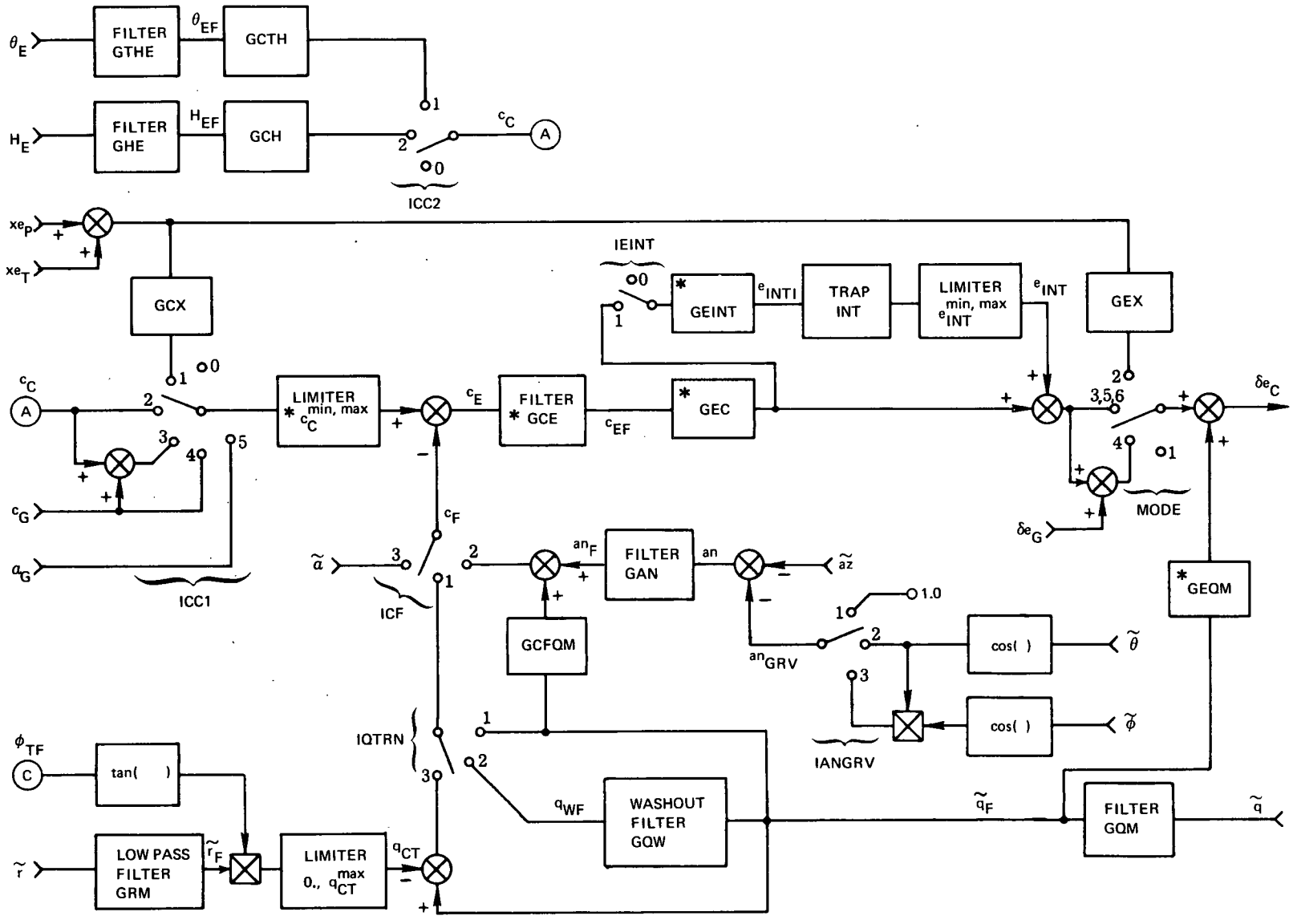



Fig. 4-19 Elevator Control Loops

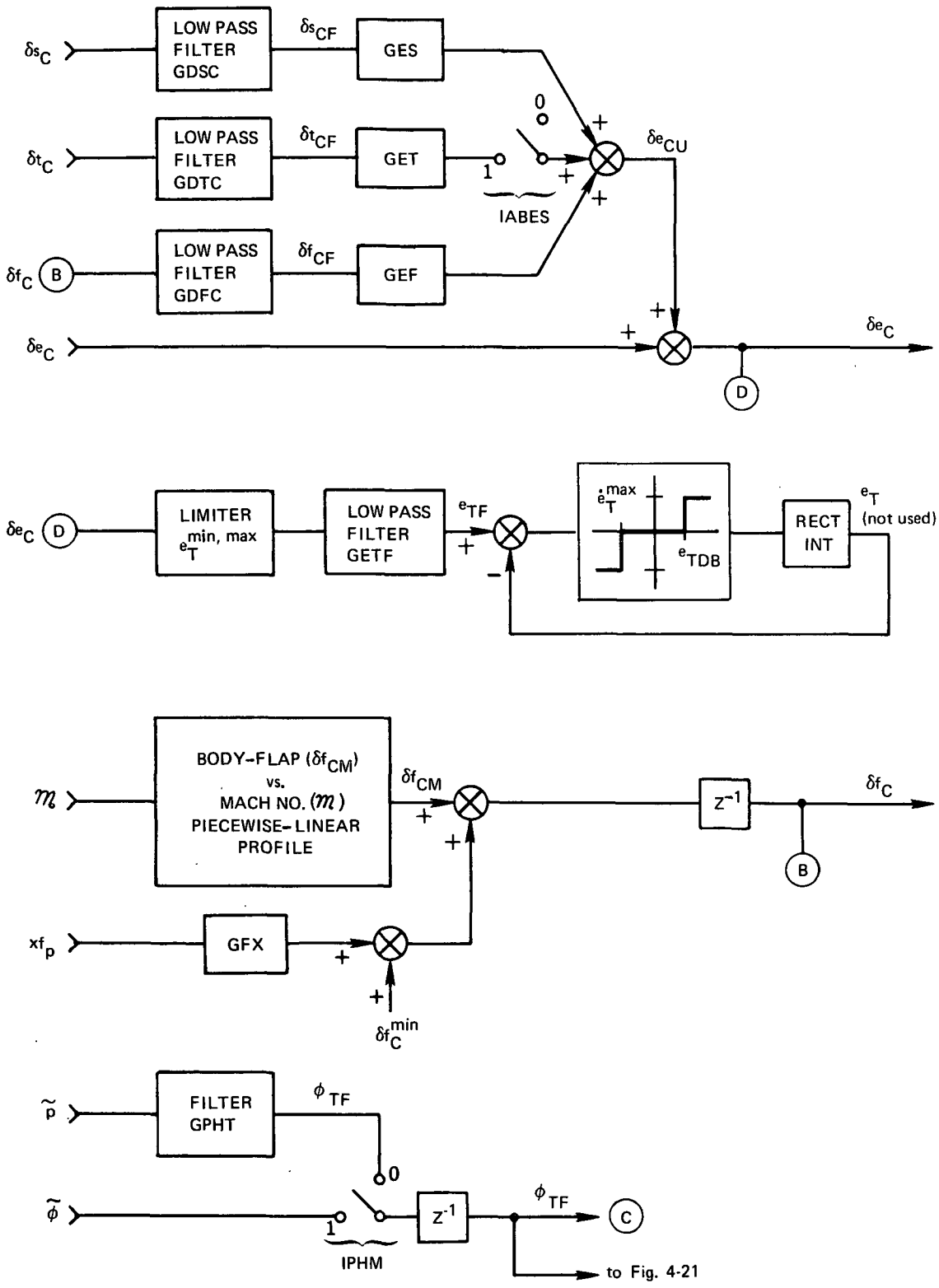



Fig. 4-19 Elevator Control Loops (Cont)

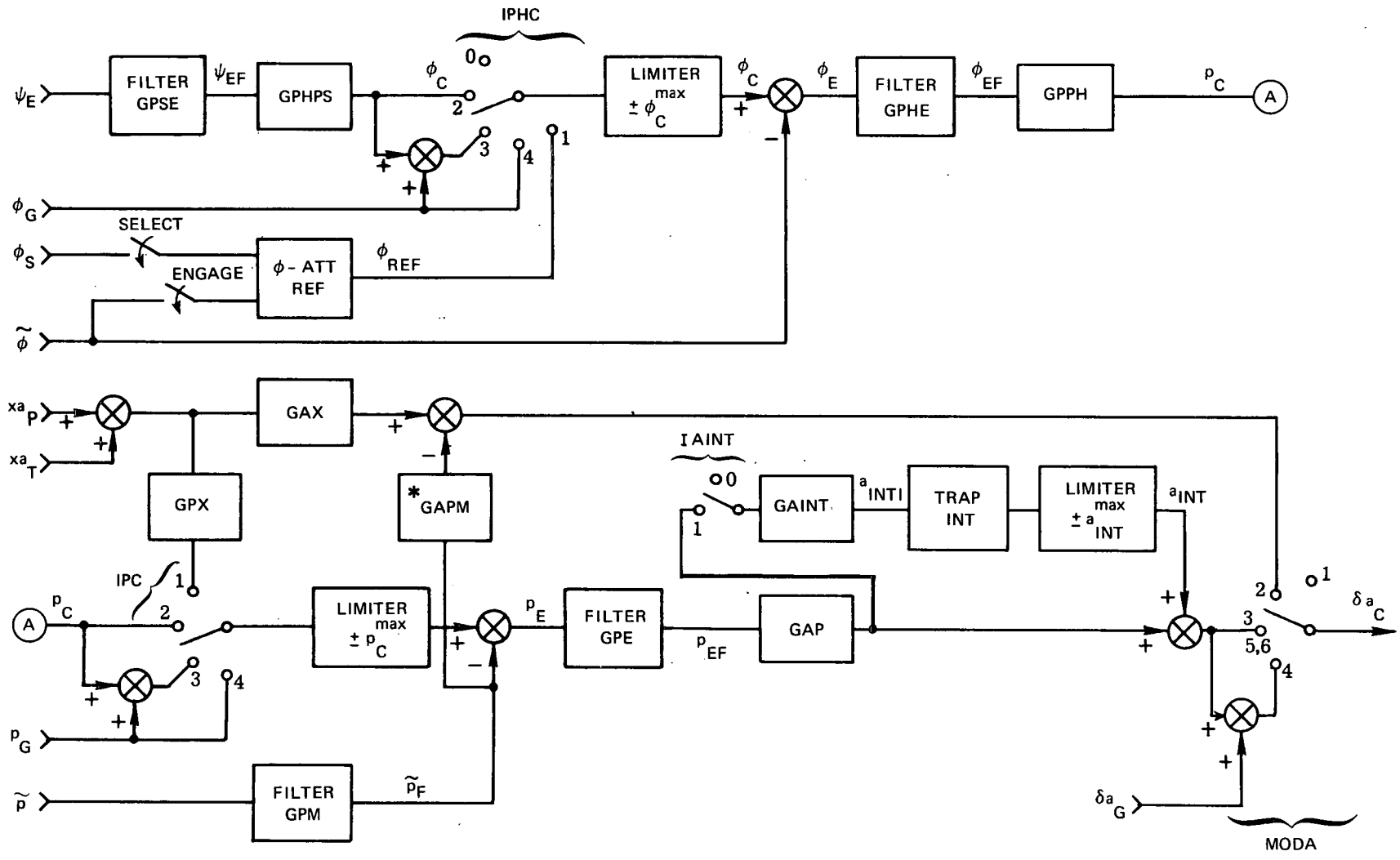



Fig. 4-20 Aileron Control Loops

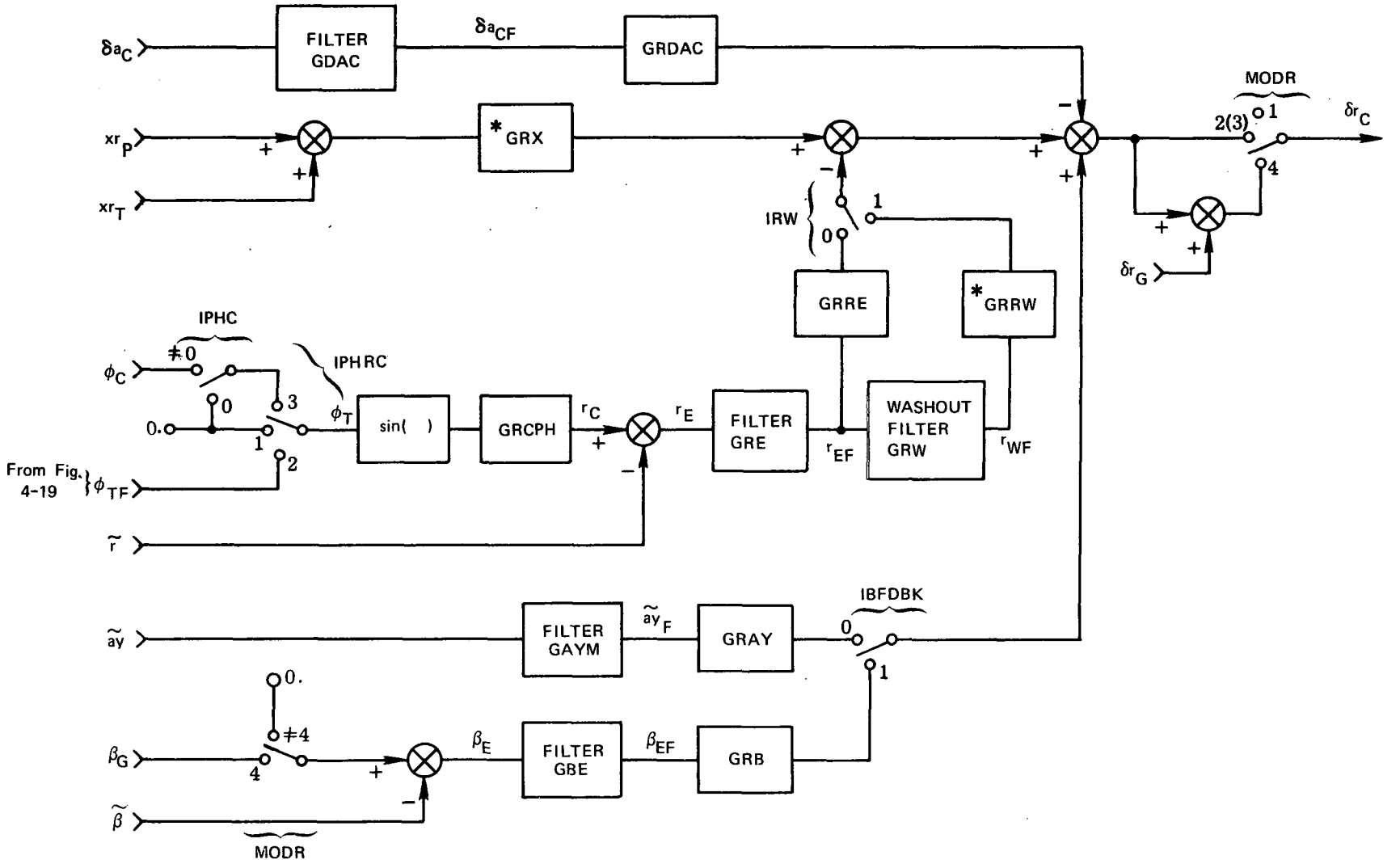



Fig. 4-21 Rudder Control Loops

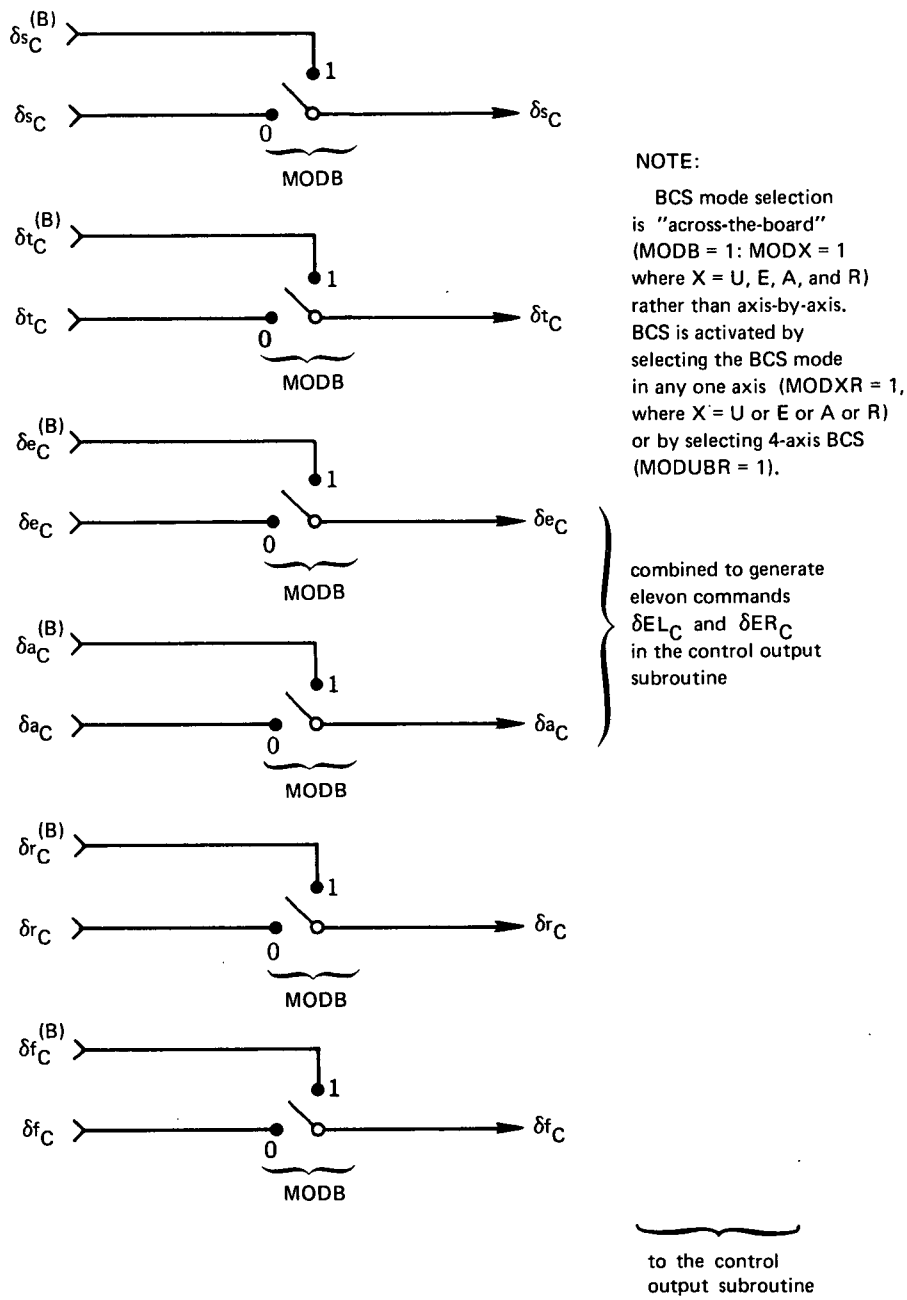



Fig. 4-22 Control Output Signals

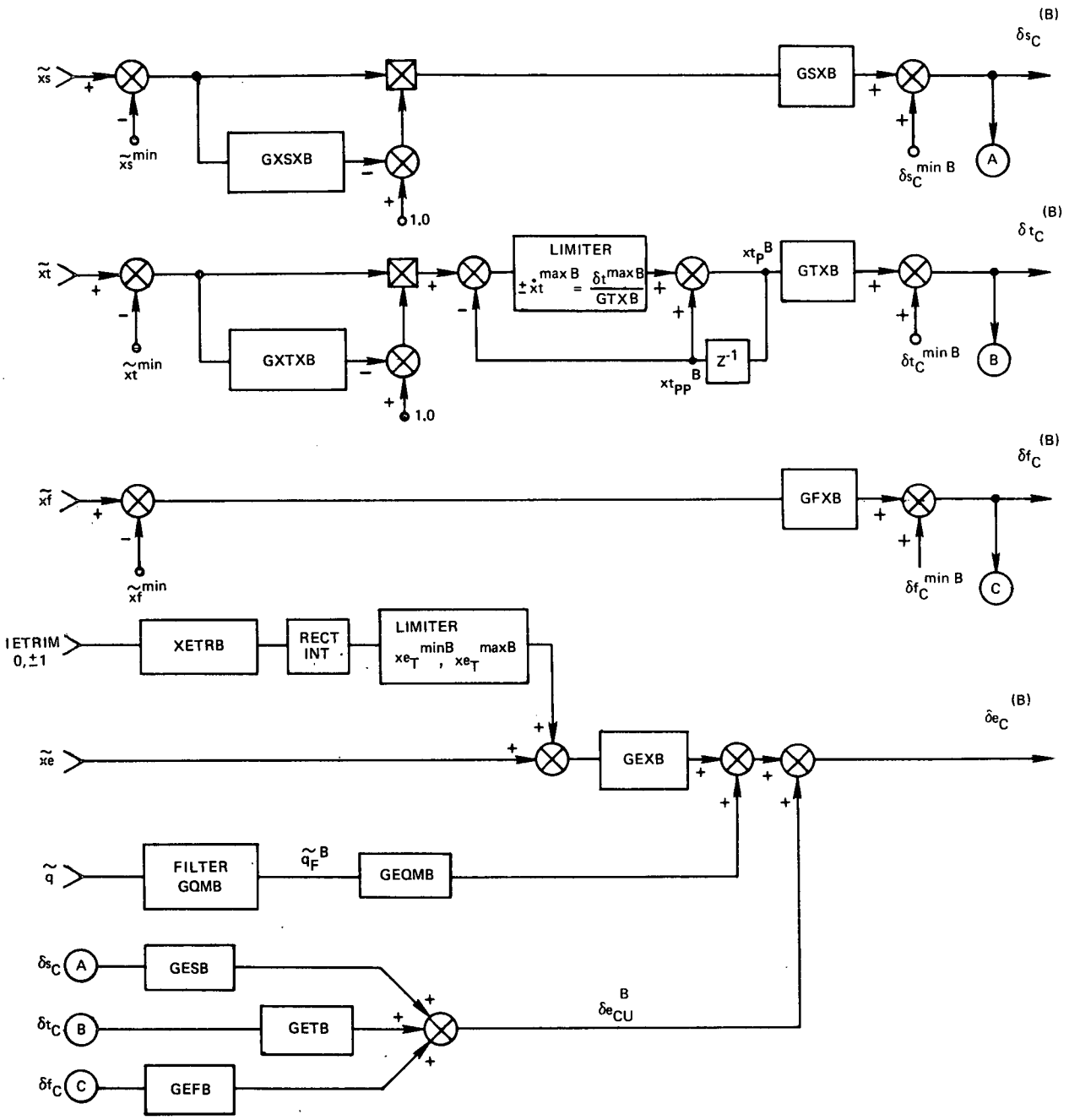



Fig. 4-23 Backup Control System

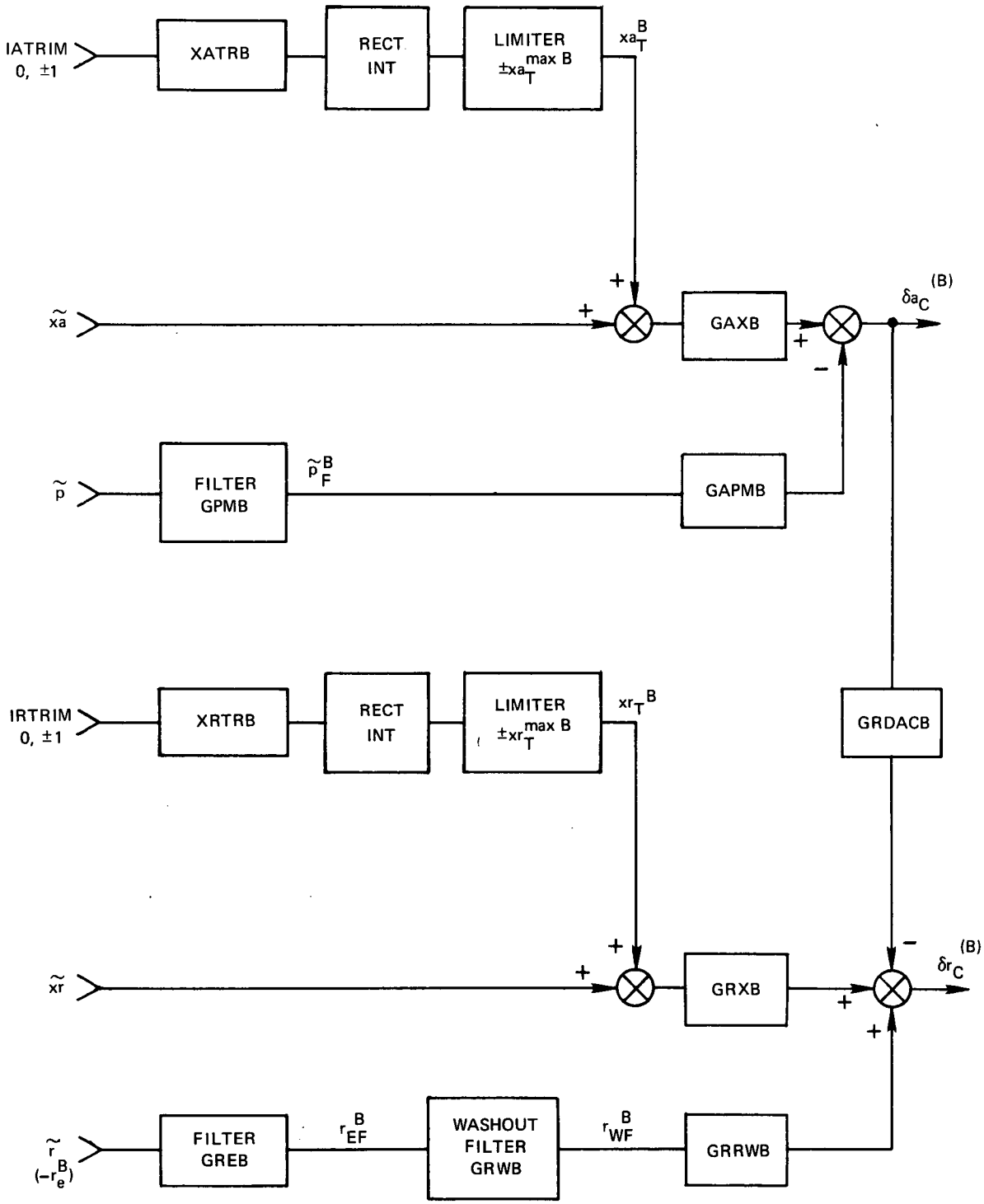



Fig. 4-23 Backup Control System (Cont)

## 4.7 Functional Flow Charts - Cruise Phase Initialization

The figures in this subsection represent the detailed logic flow of part of the cruise phase DFCS at the engineering/functional level rather than at the coded software level presented in Appendix 4A. Implementation (especially with respect to programming language syntax) should not be restricted by overspecification. Therefore, in future revisions of this document functional flow charts similar to those in this subsection will replace the detailed flow charts in the appendix.

Figure 4-24 presents the cruise phase segment of the routine which is called by the DFCS executive when the logic for a mission phase is to be completely initialized. Figures 4-25 through 4-28 present the cruise phase segment of the routine which is called for any phase or mode initialization; they interface with the structure shown in Fig. 4A-2.

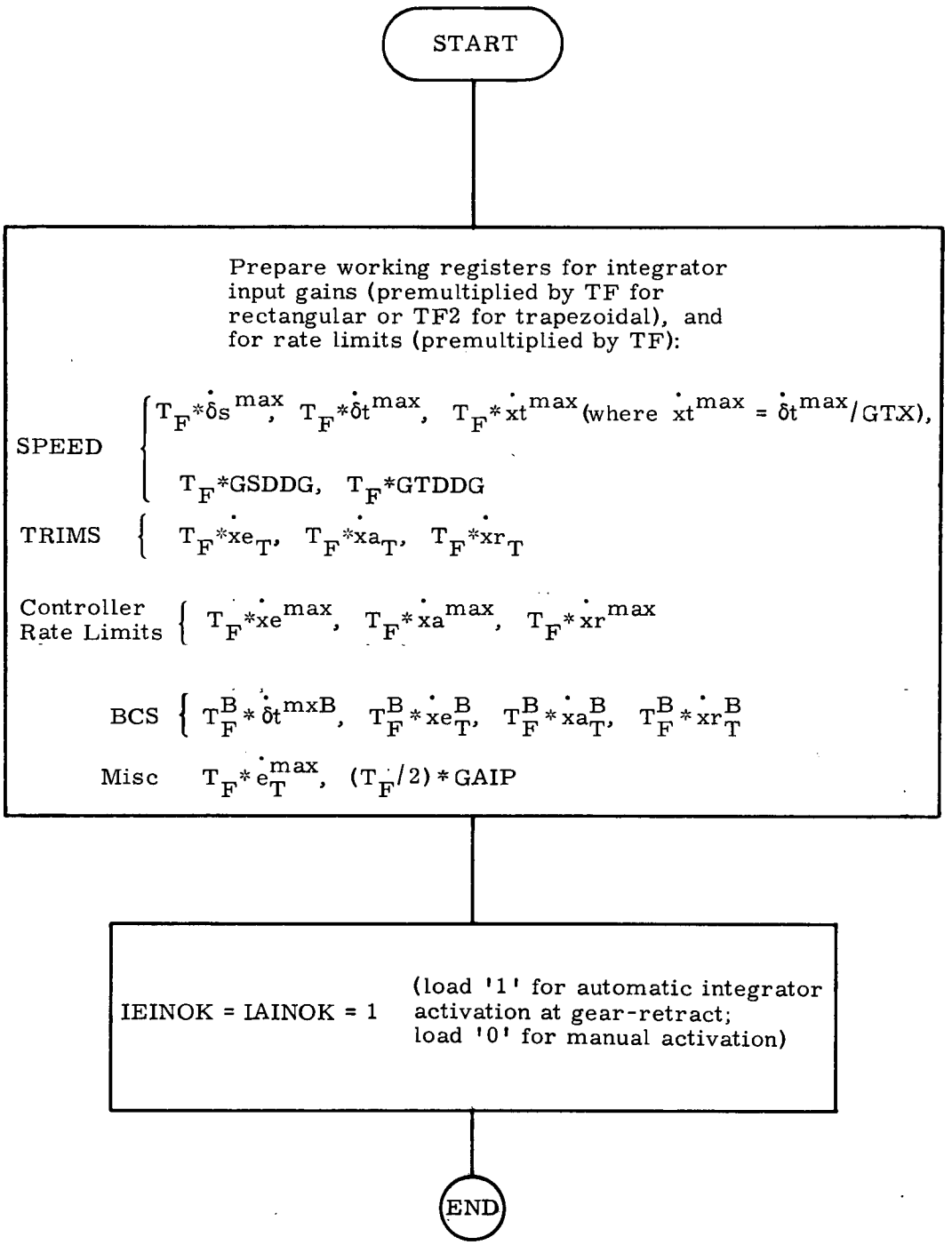
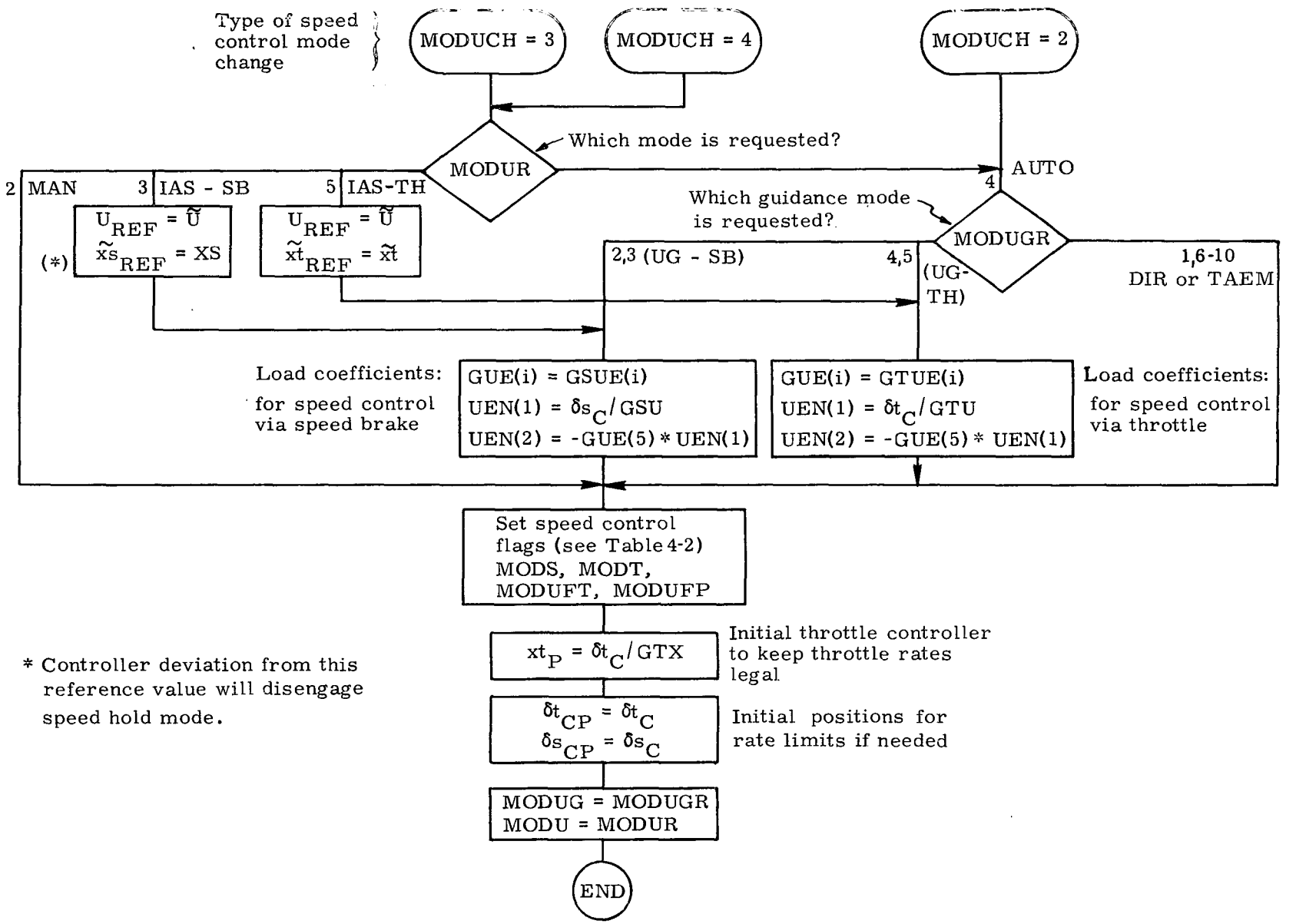



Fig. 4-24 Functional Flow Diagram, Load Manipulation Routine




* Controller deviation from this reference value will disengage speed hold mode.

Fig. 4-25 Functional Flow Diagram, Speed Initialization Routine

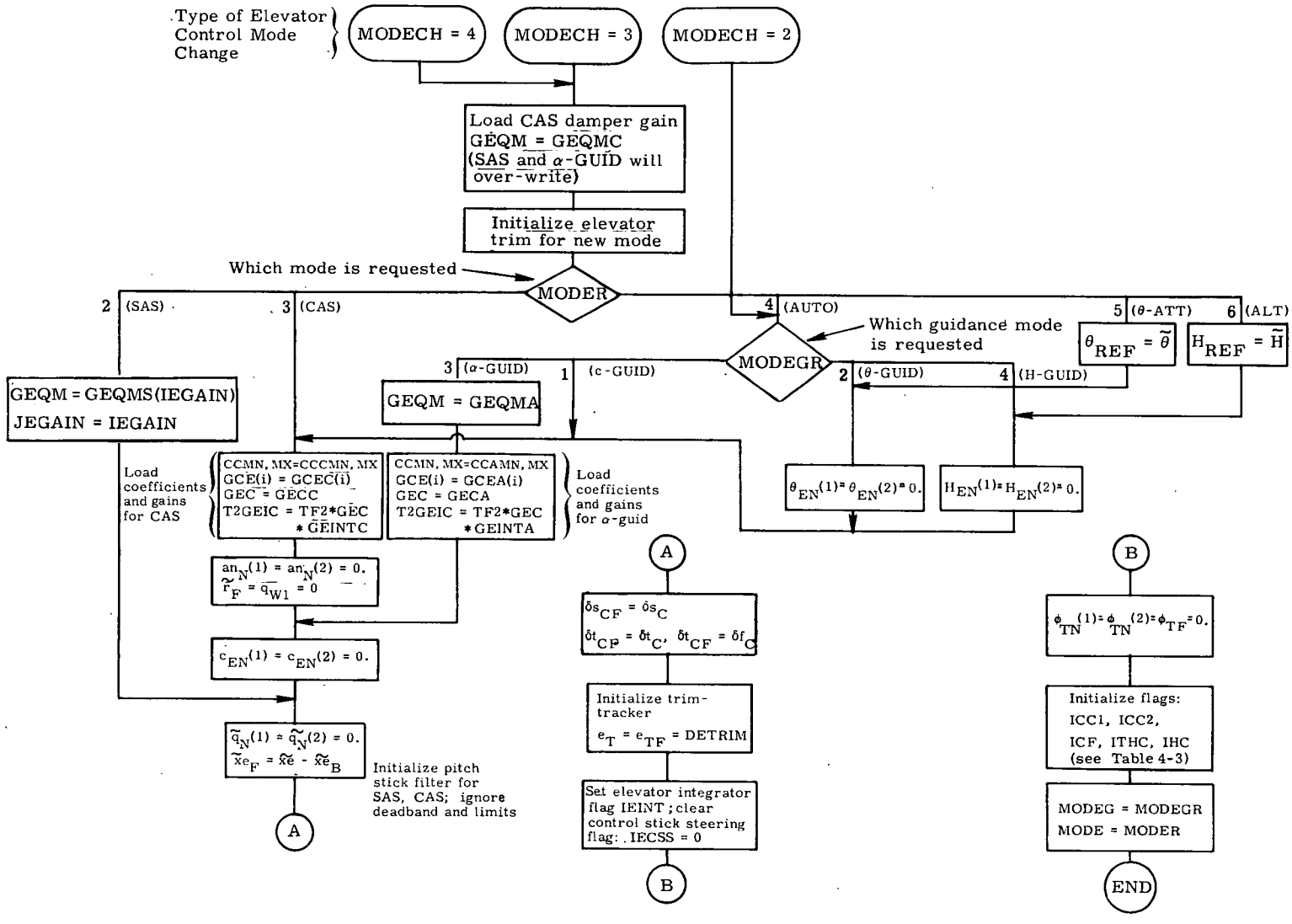



Fig. 4-26 Functional Flow Diagram, Elevator Initialization Routine

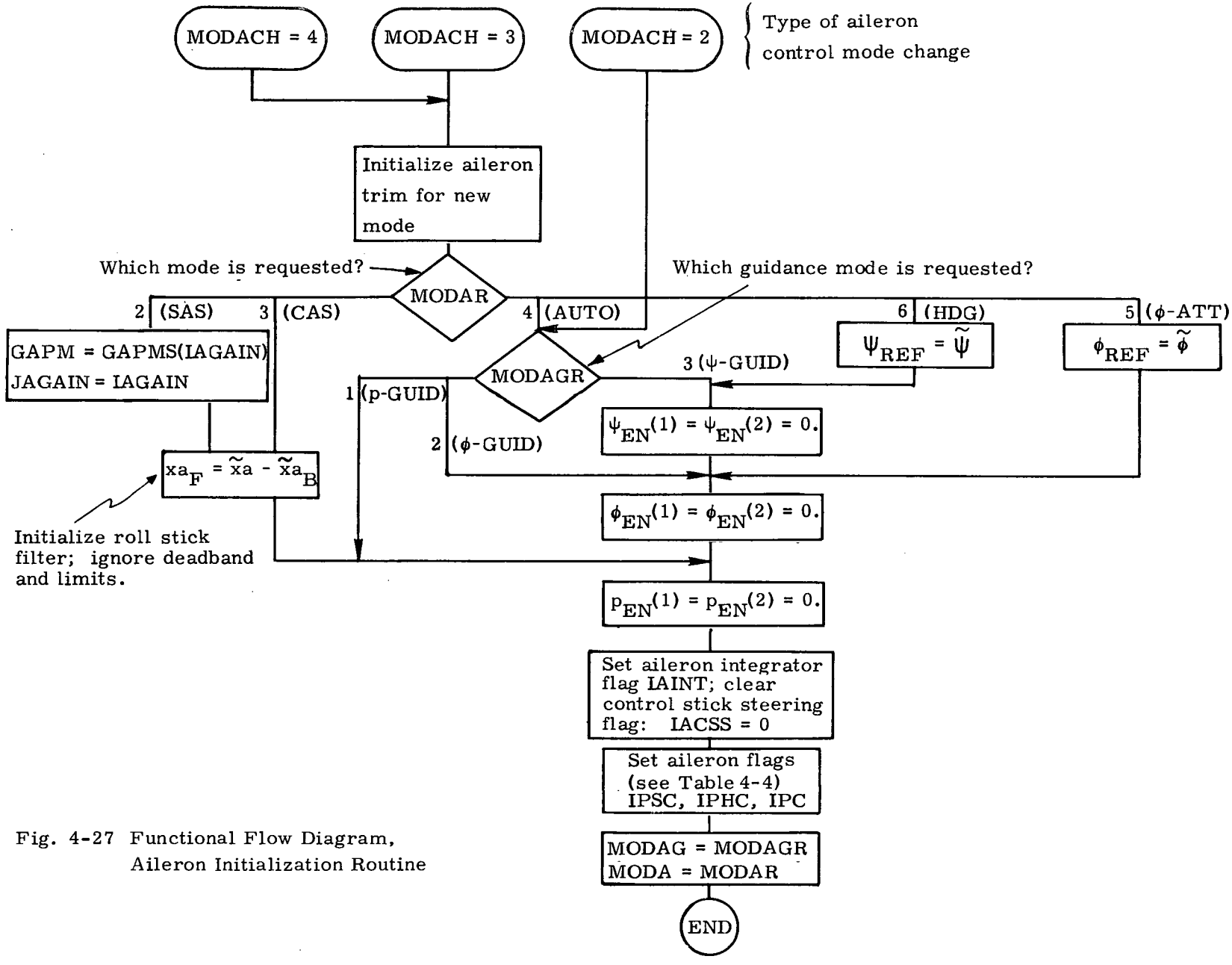



Fig. 4-27 Functional Flow Diagram, Aileron Initialization Routine

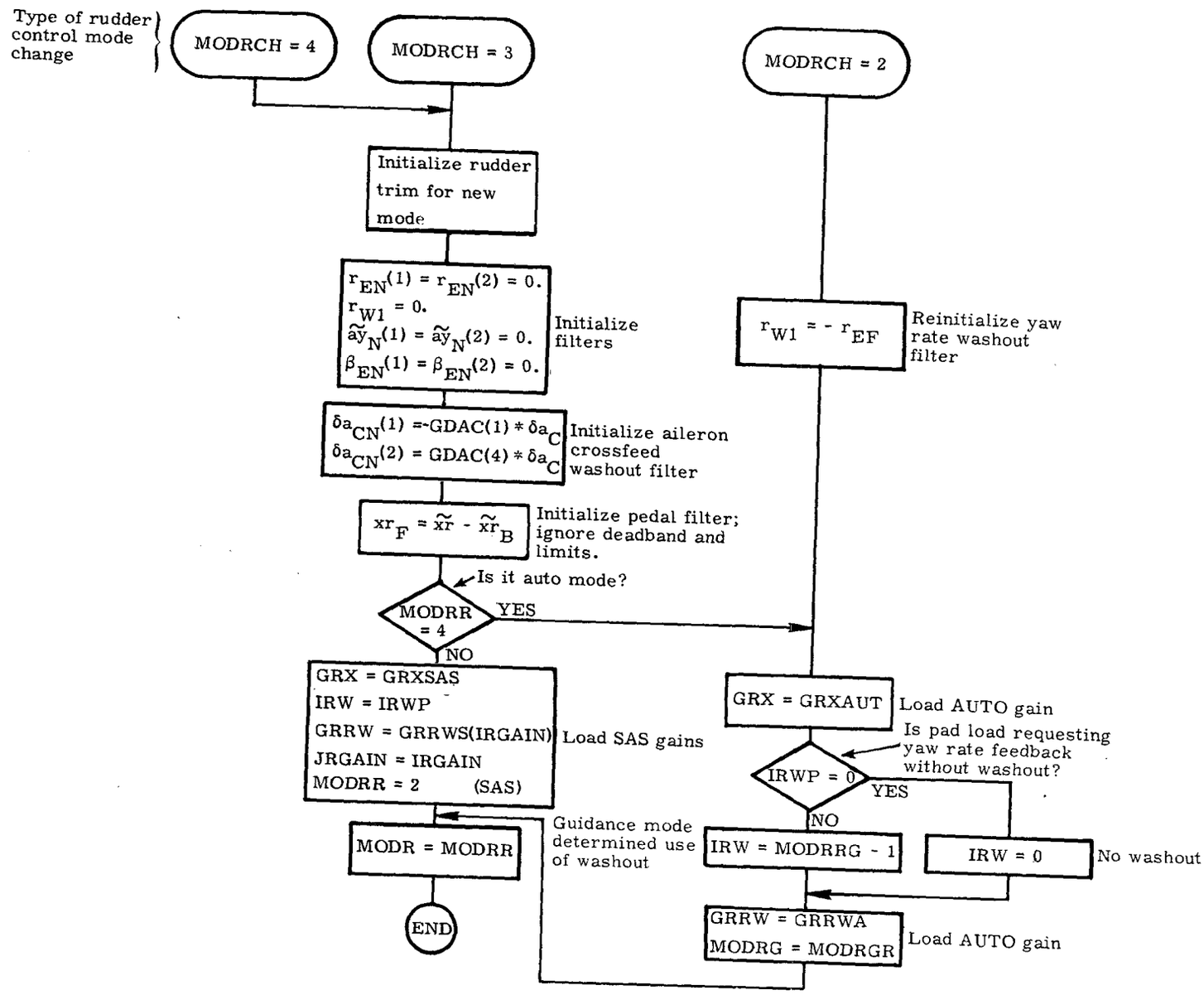



Fig. 4-28 Functional Flow Diagram, Rudder Initialization Routine

## APPENDIX 4A

## DETAILED FLOWCHARTS FOR CRUISE PHASE DFCS

Detailed specifications for the cruise phase DFCS, in terms of FORTRAN flow charts of an operating version of the DFCS, are presented in Figs. 4A-1 through 4A-20. Minute functional details are covered, but unfortunately they are often overwhelmed by implementational aspects (e.g., higher-order-language syntax), and are thus less-than-ideal for specification purposes.

Figure 4A-1 depicts the cruise phase preflight (or first-pass) load manipulation operations. Figure 4A-2 presents the skeletal structure into which the remaining figures fit. In particular, there are five basic subroutines. These subroutines interface directly with the DFCS executive as shown in Section 1.2. Their functions for cruise phase are as follows:

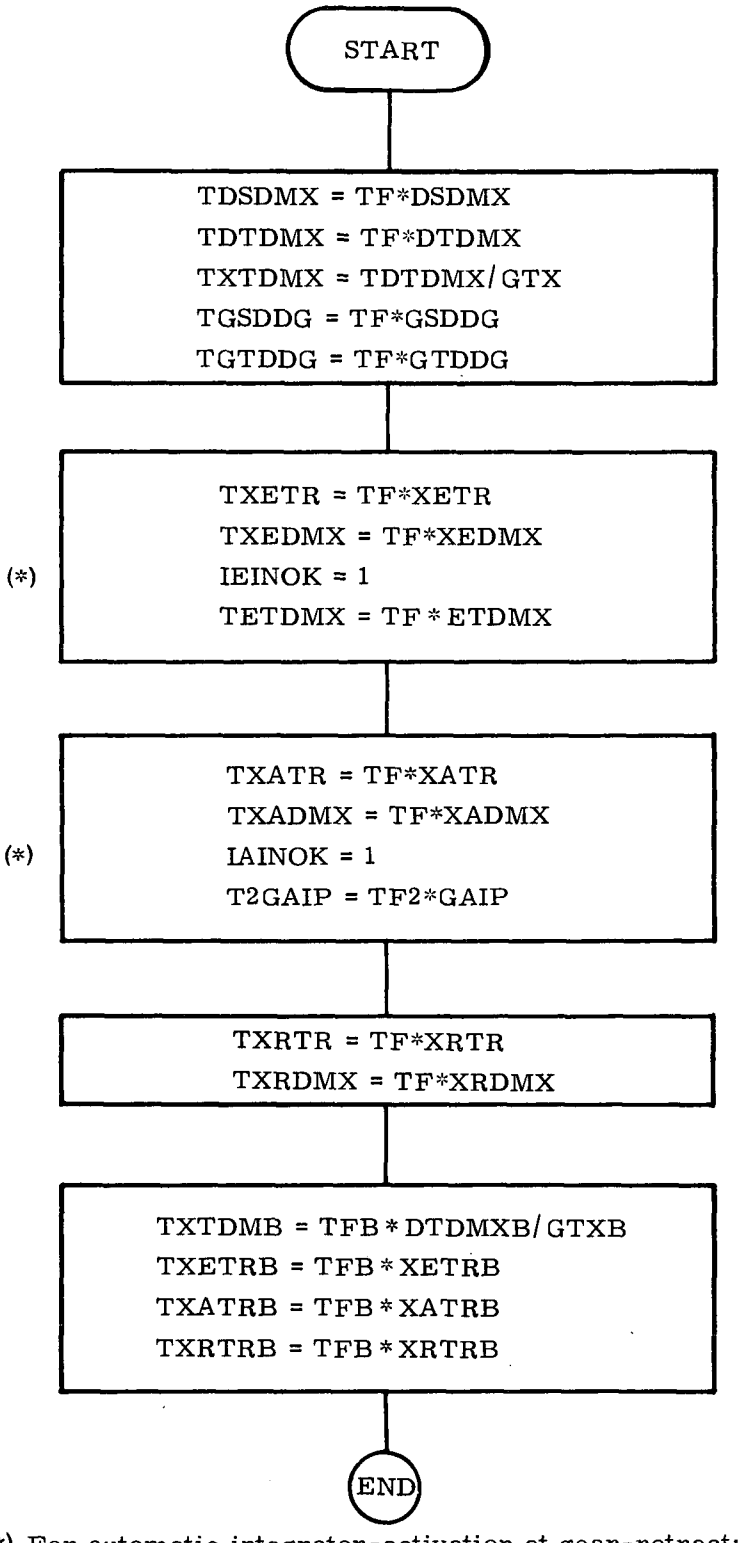
Initialization Subroutine (Fig. 4A-3 through Fig. 4A-6)
handles all first-pass (preflight turn-on or inflight phase-change) and mode change initialization activity.

Filter Part 1 Subroutine (Fig. 4A-7 through Fig. 4A-10)
handles all preparatory calculations for the Control Output Subroutine, including pilot input signal processing, filter output node computations, control stick steering override detection, etc.

Control Law Subroutine (Fig. 4A-11 through Fig. 4A-13)
handles all control output calculations in preparation for the Control Output Subroutine (Subsection 2.2).

Filter Part 2 Subroutine (Fig. 4A-14 through Fig. 4A-18)
handles all filter wrapup processing and miscellaneous activity, such as body flap command calculations, bank-angle filter calculations, and reengaging semiautomatic operation following a control stick steering override.

BCS Subroutine (Fig. 4A-19 through Fig. 4A-20)
handles sequentially, the Filter Part 1, Control Law, Control Output, and Filter Part 2 calculations for a 4-axis BCS mode of operation.




(*) For automatic integrator-activation at gear-retract; for manual reactivation load IEINOK = IAINOK = 0.

Fig. 4A-1 Load Manipulator Routine Cruise Section

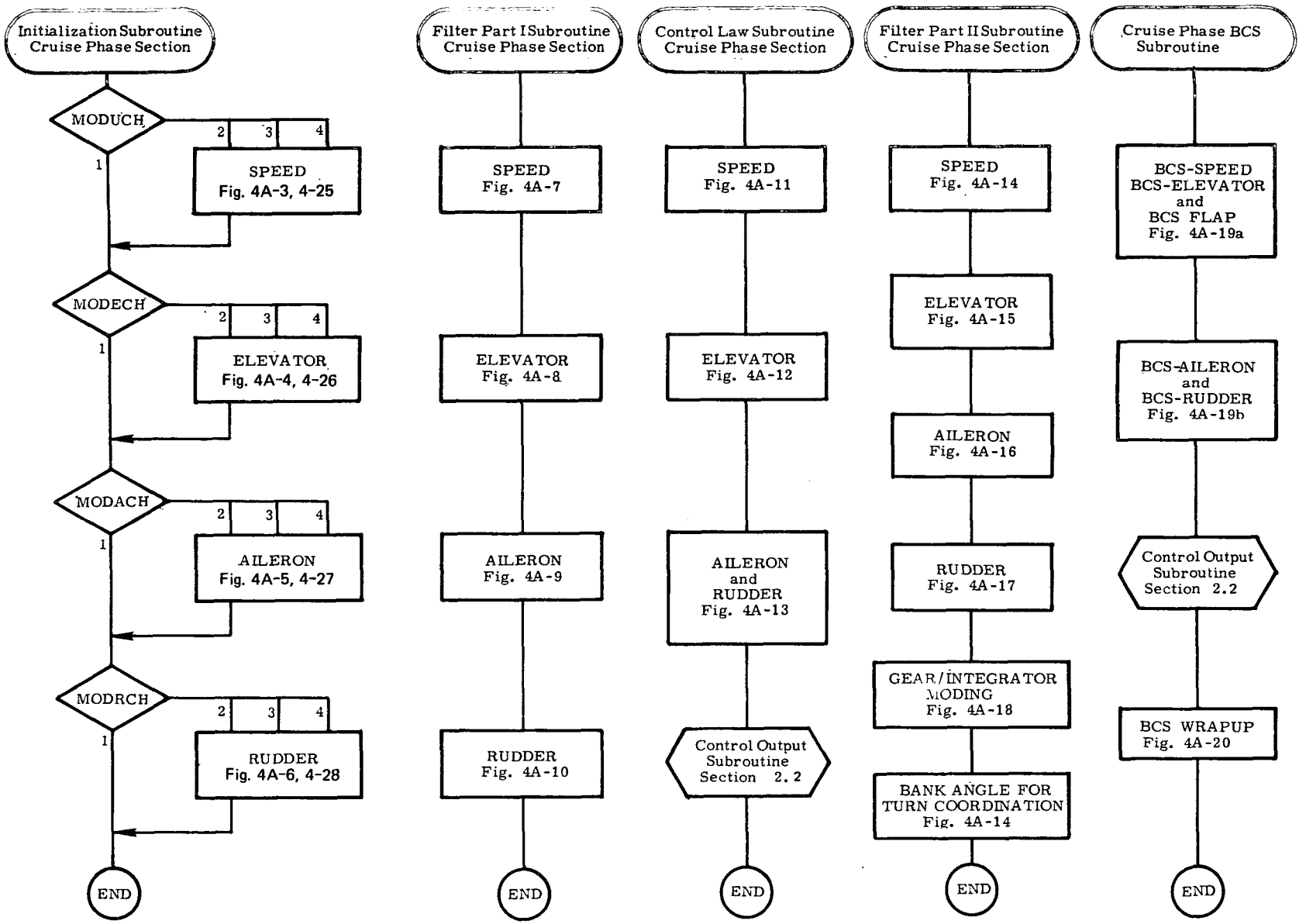



Fig. 4A-2 Overview of Cruise Phase Flow Charts

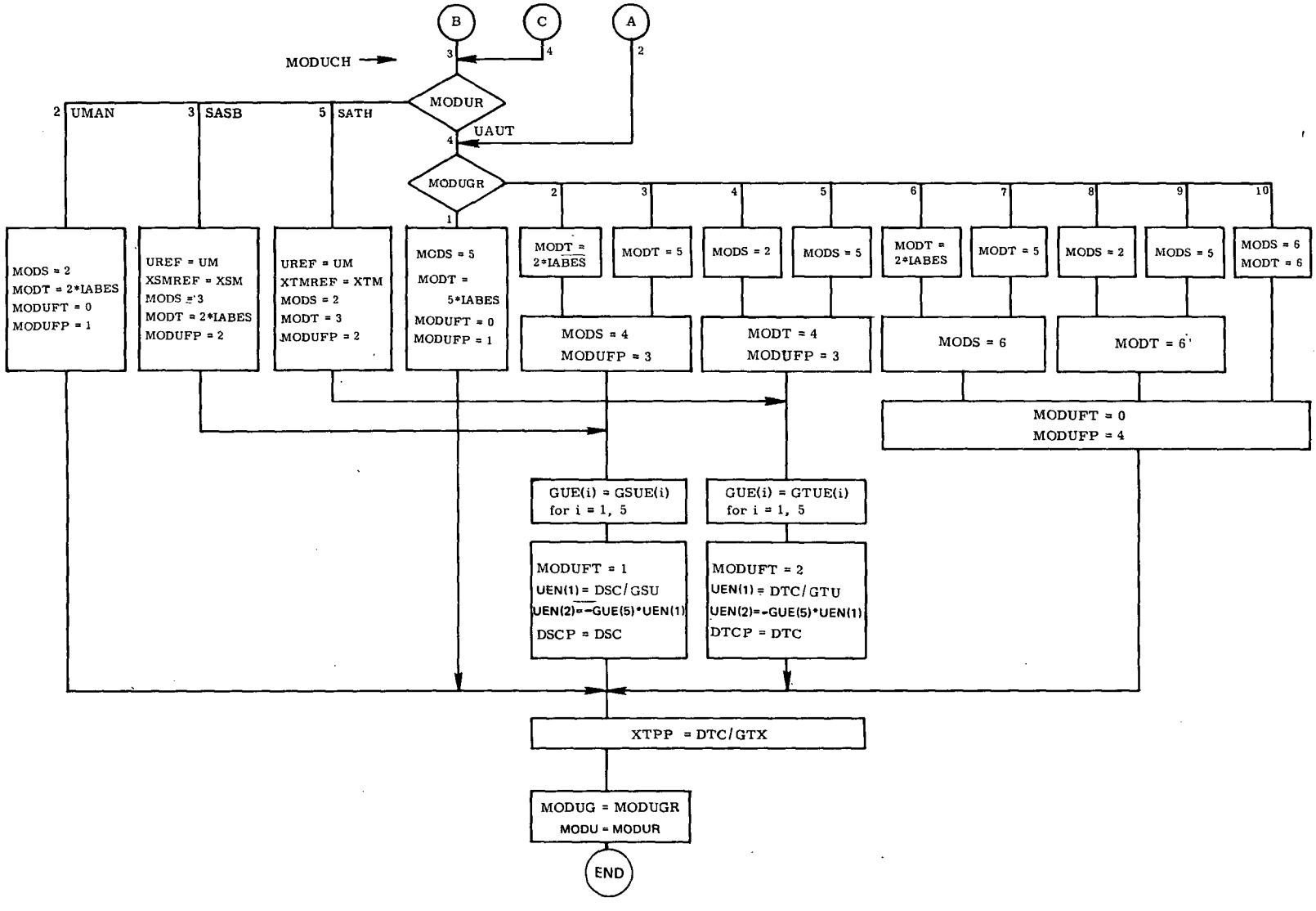



Fig. 4A-3 Initialization Subroutine, Cruise Phase Section, Speed Subsection

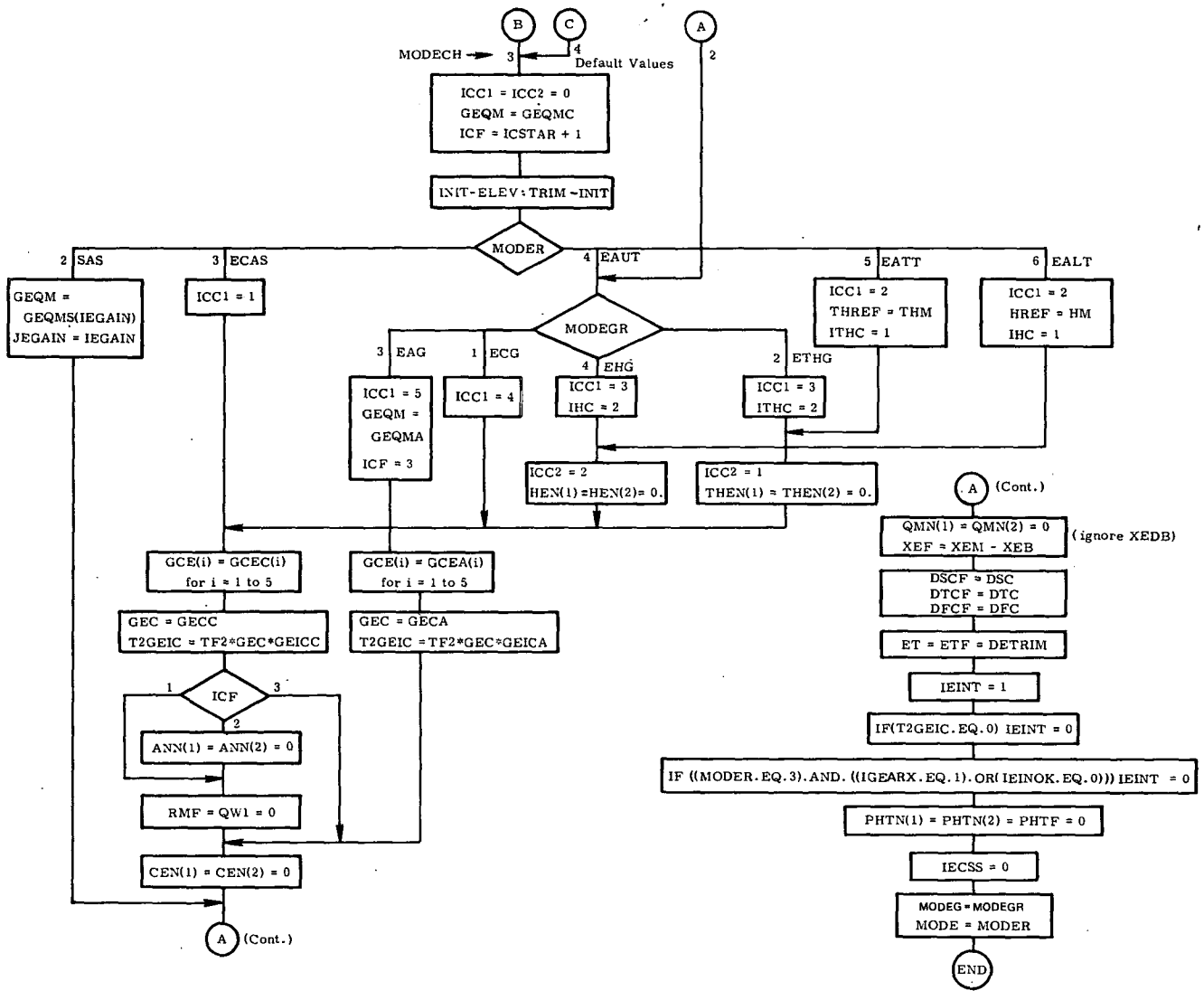



Fig. 4A-4a Initialization Subroutine, Cruise Phase Section, Elevator Subsection

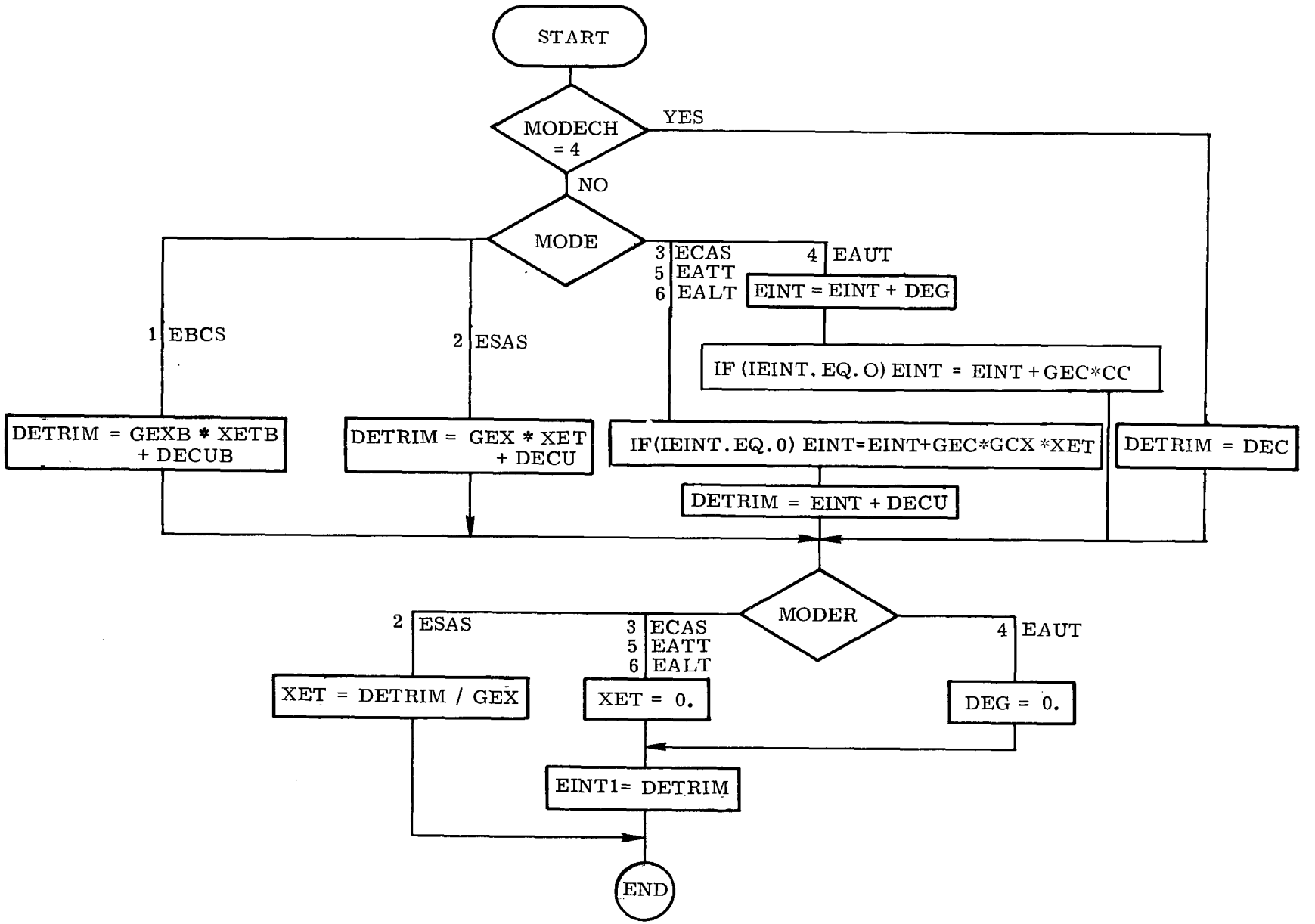



Fig. 4A-4b Initialization Subroutine, Cruise Phase Section, Elevator Trim Subsection

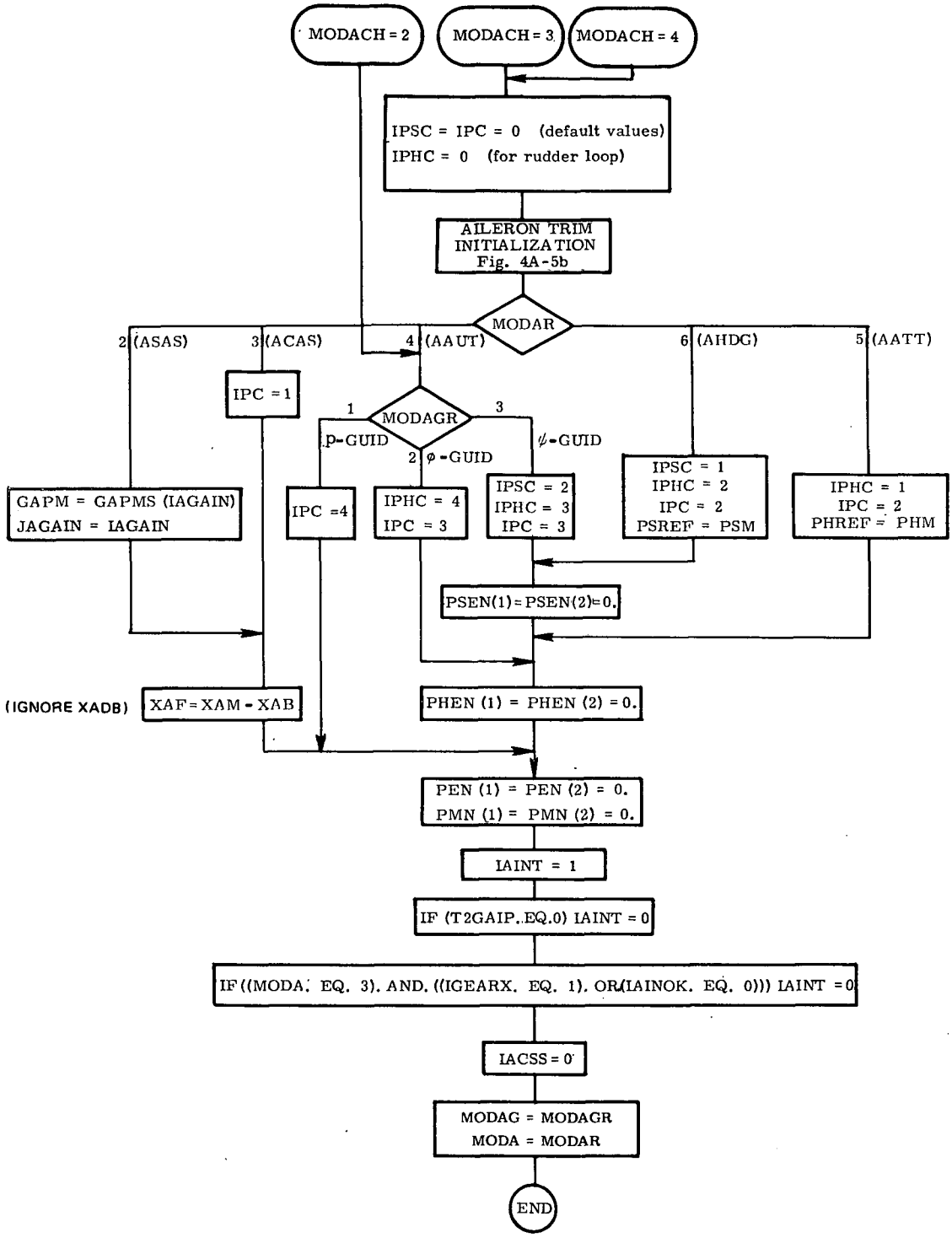



Fig. 4A-5a Initialization Subroutine, Cruise Phase Section, Aileron Subsection

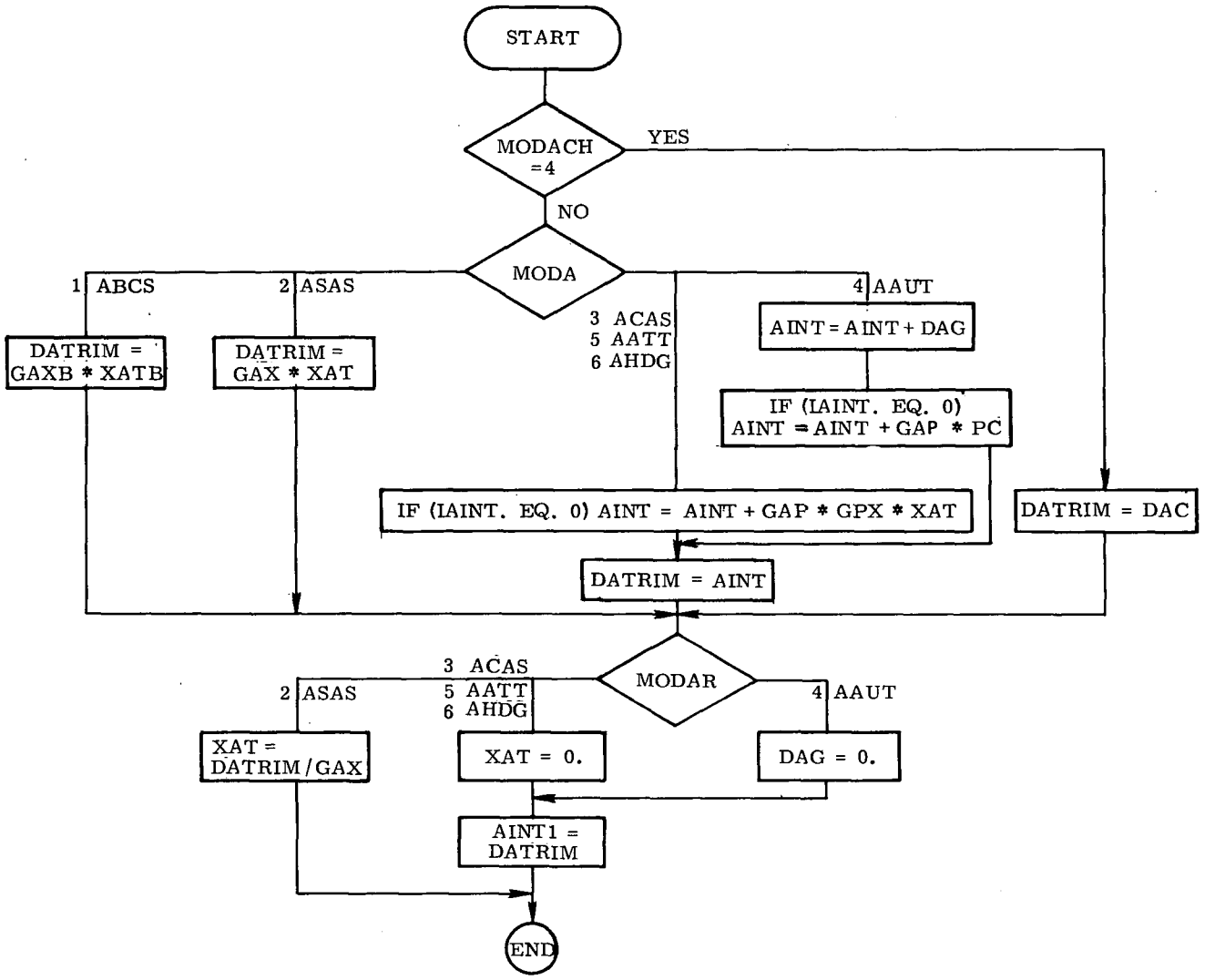



Fig. 4A-5b Initialization Subroutine, Cruise Phase Section, Aileron Trim Subsection

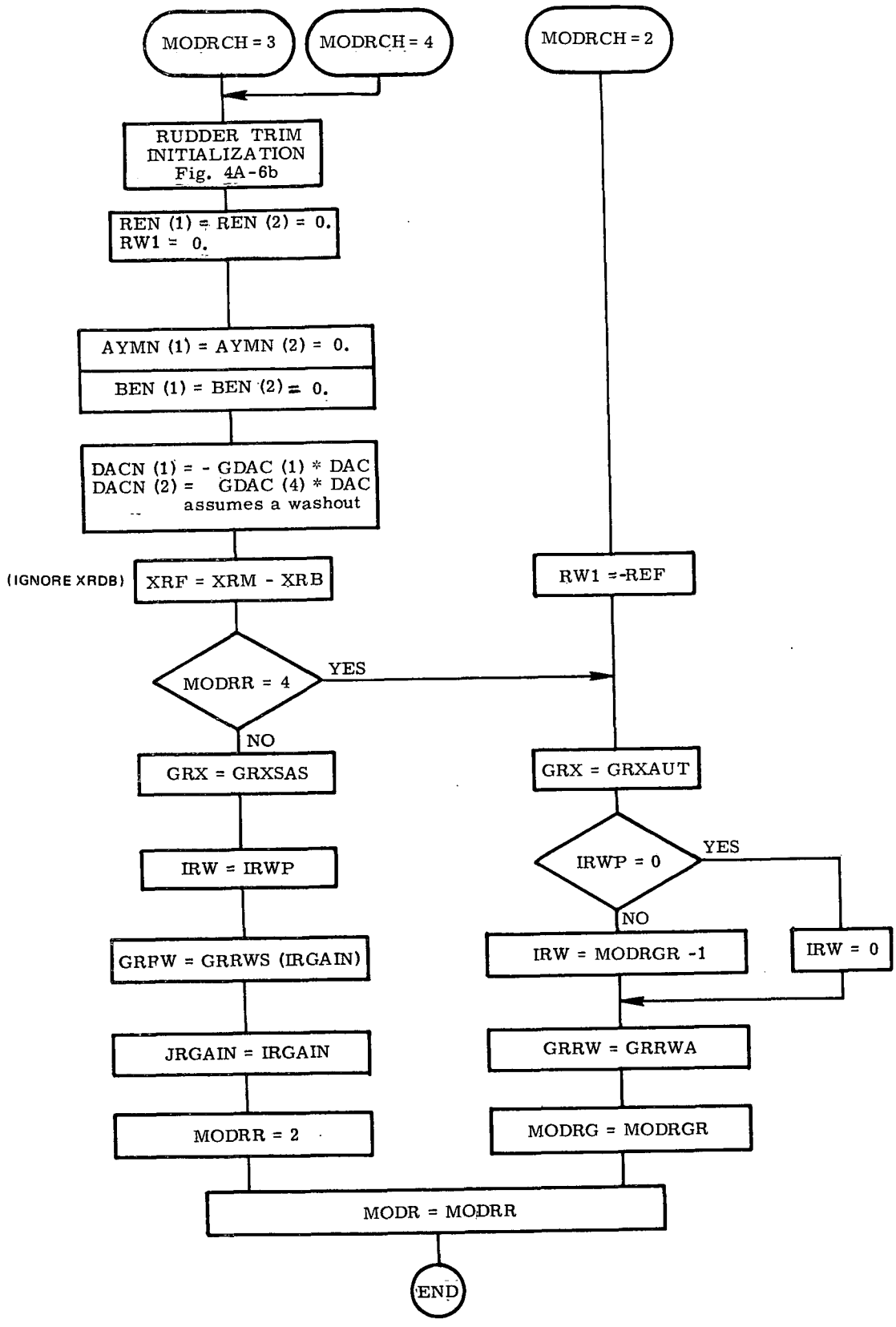



Fig. 4A-6a Initialization Subroutine, Cruise Phase Section, Rudder Subsection

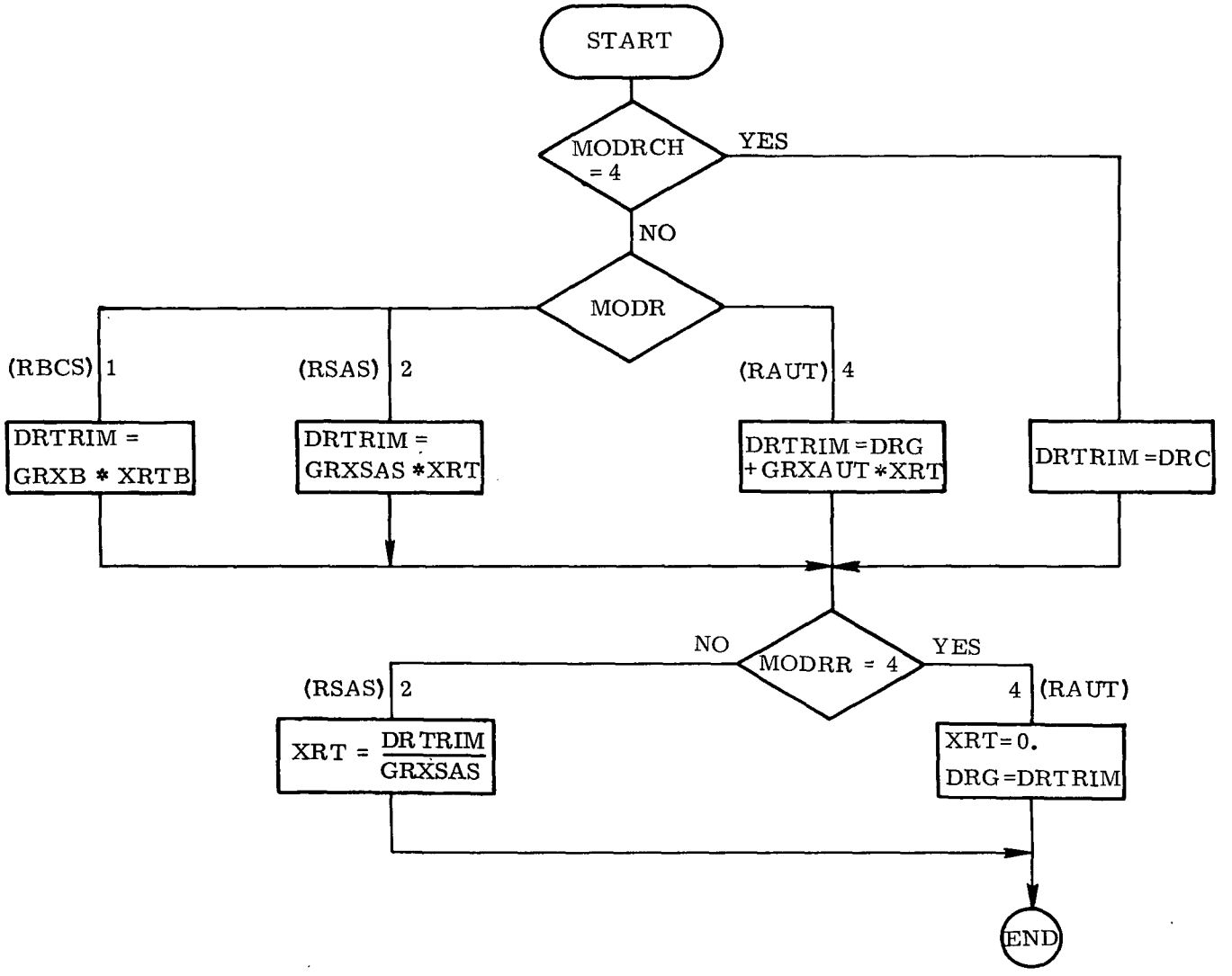



Fig. 4A-6b Initialization Subroutine, Cruise Phase Section, Rudder Trim Subsection

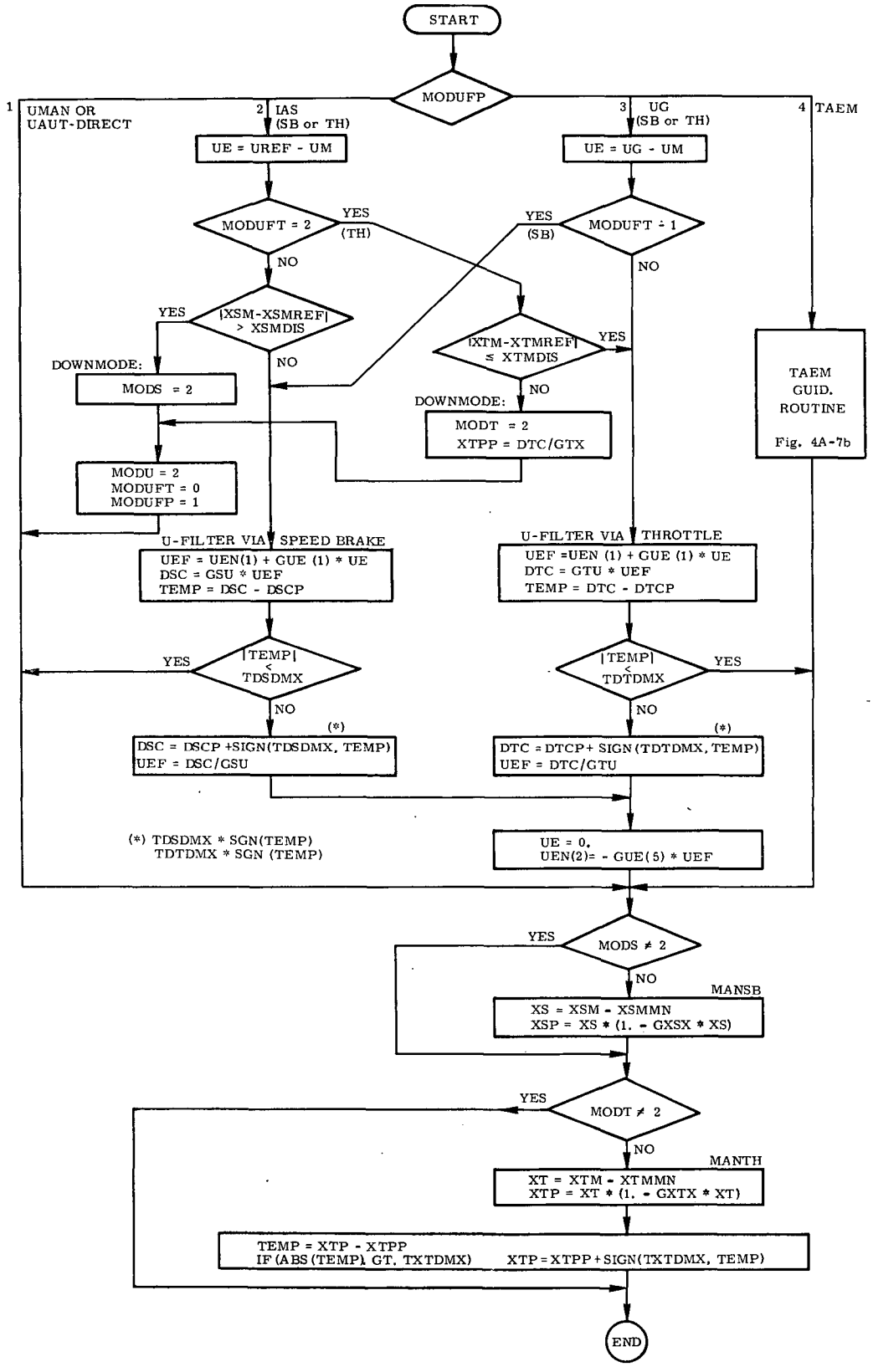



Fig. 4A-7a Filter Part I Subroutine, Cruise Phase Section Speed Subsection

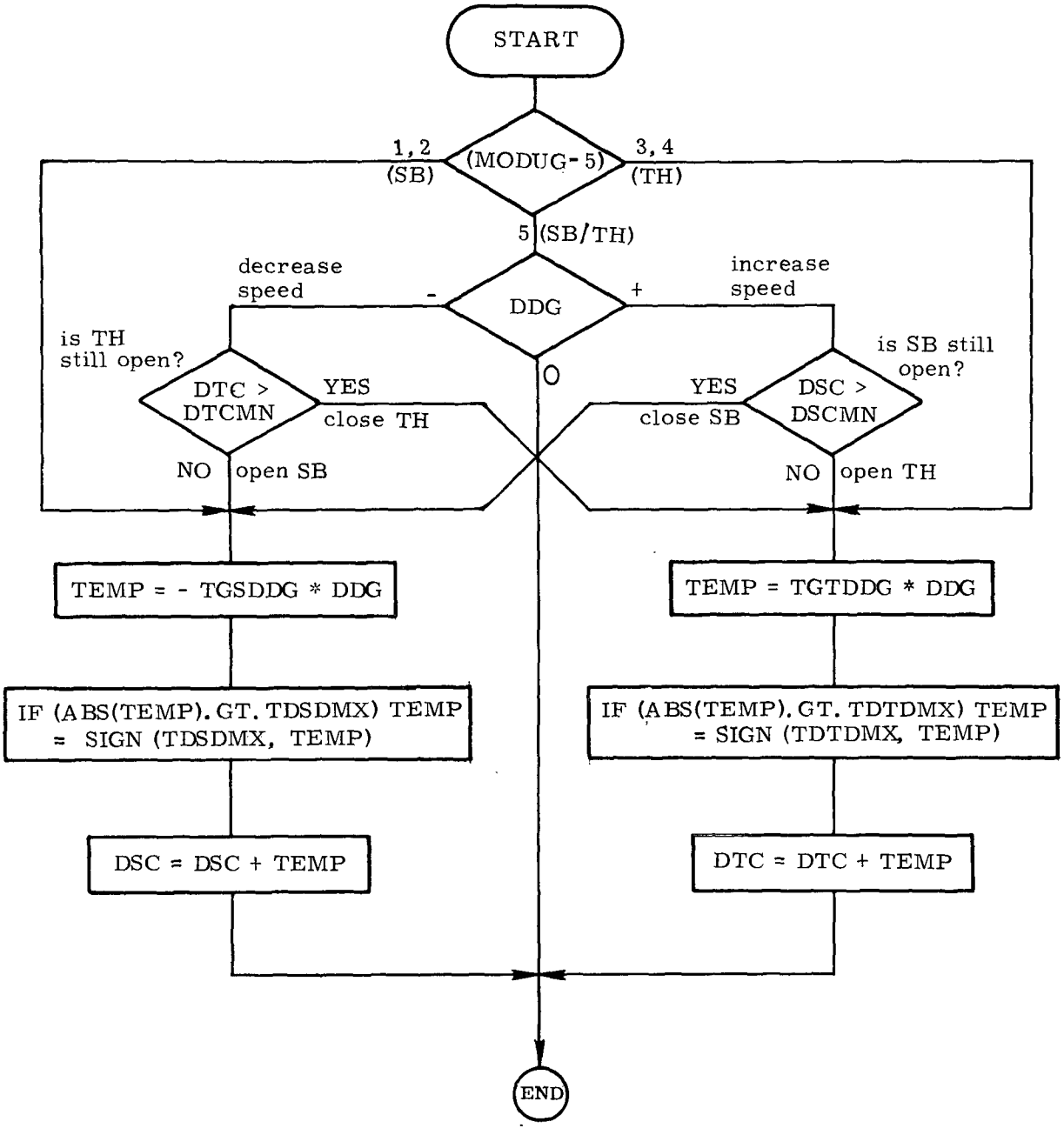



Fig. 4A-7b Filter Part I Subroutine, Cruise Phase Section Speed Subsection, (TAEM Guidance Routine)

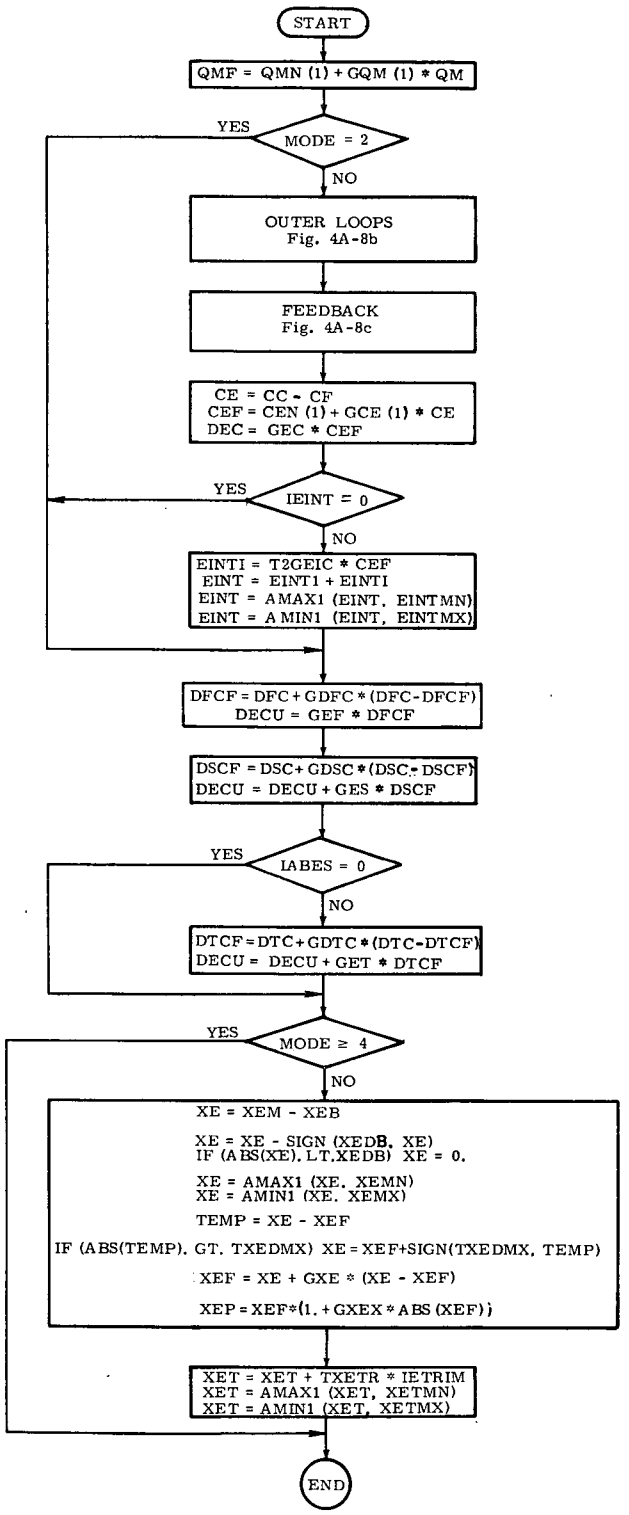



Fig. 4A-8a Filter Part I Subroutine, Cruise Phase Section, Elevator Subsection,

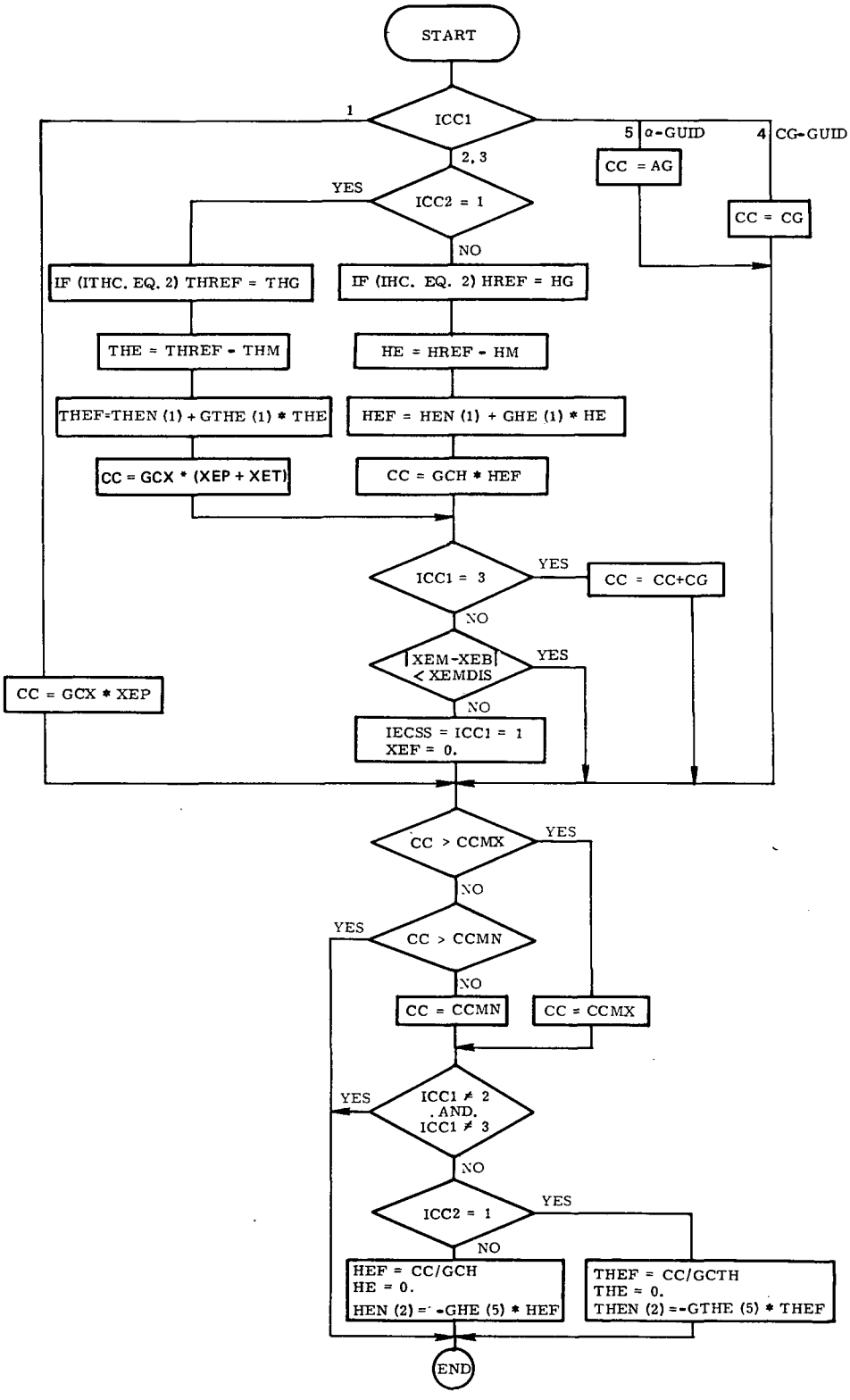



Fig. 4A-8b Filter Part I Subroutine, Cruise Phase, Elevator Subsection, Outer Loops

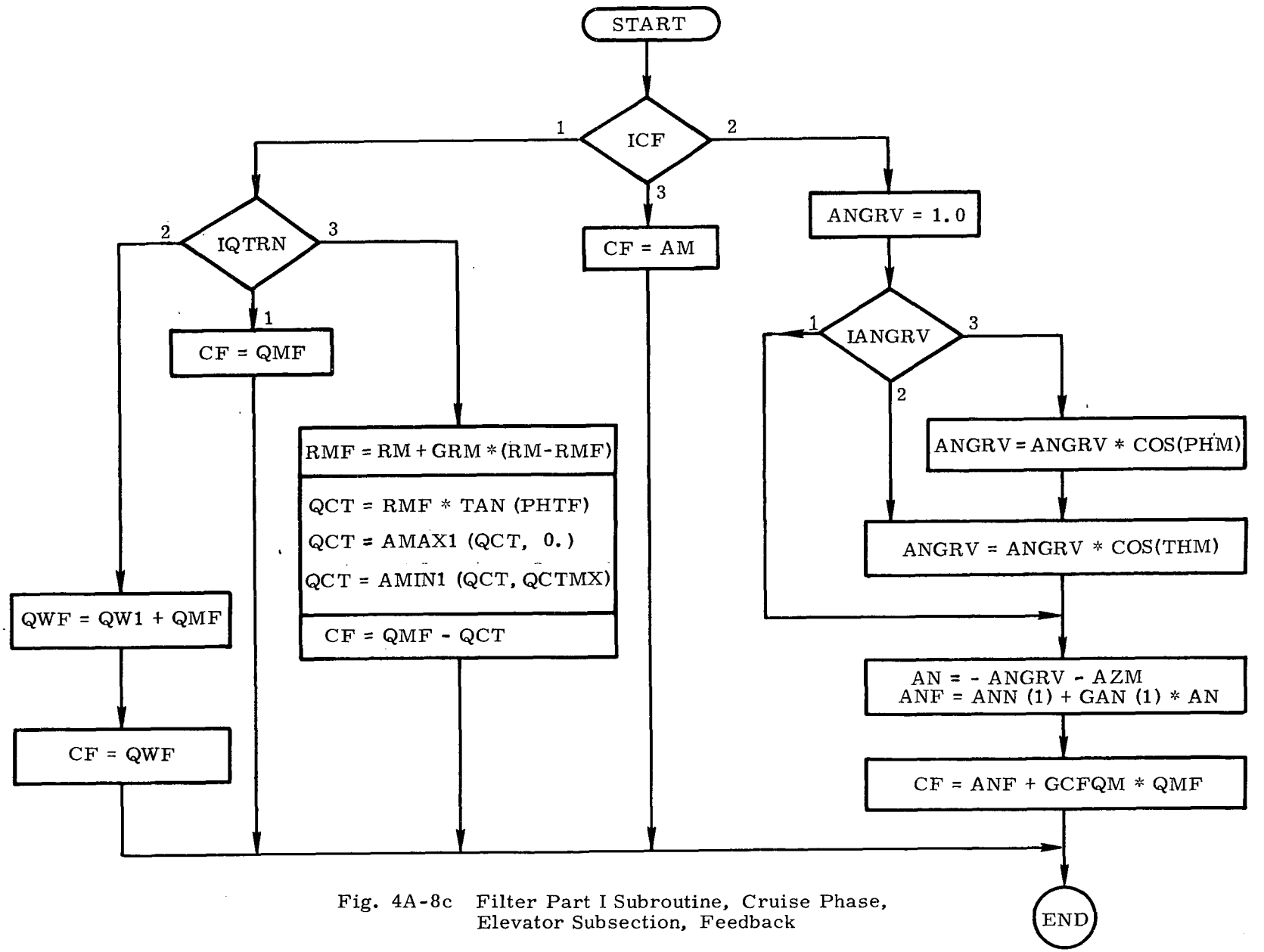



Fig. 4A-8c Filter Part I Subroutine, Cruise Phase, Elevator Subsection, Feedback

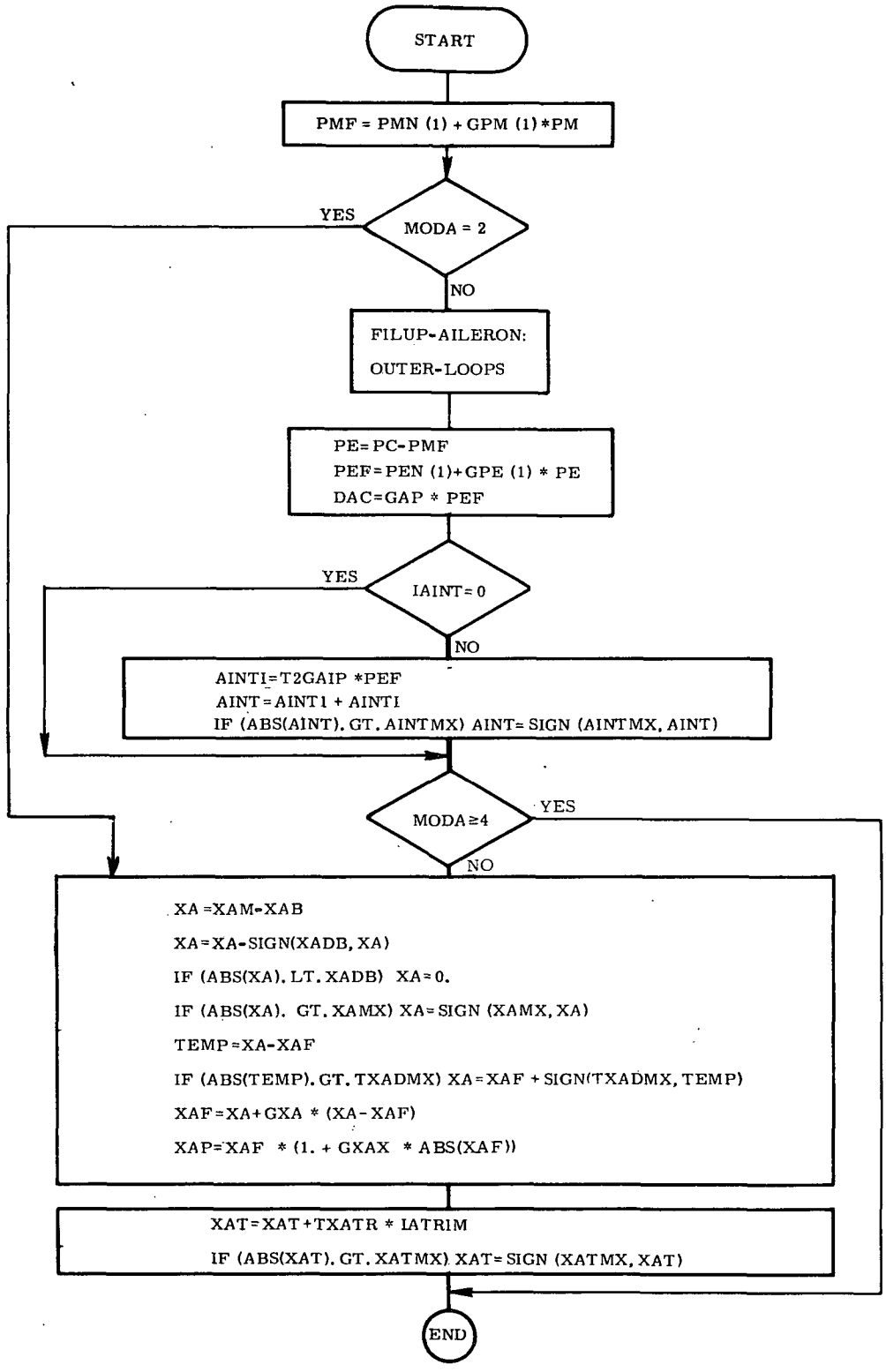



Fig. 4A-9a Filter Part I Subroutine, Cruise Phase Section, Aileron Subsection

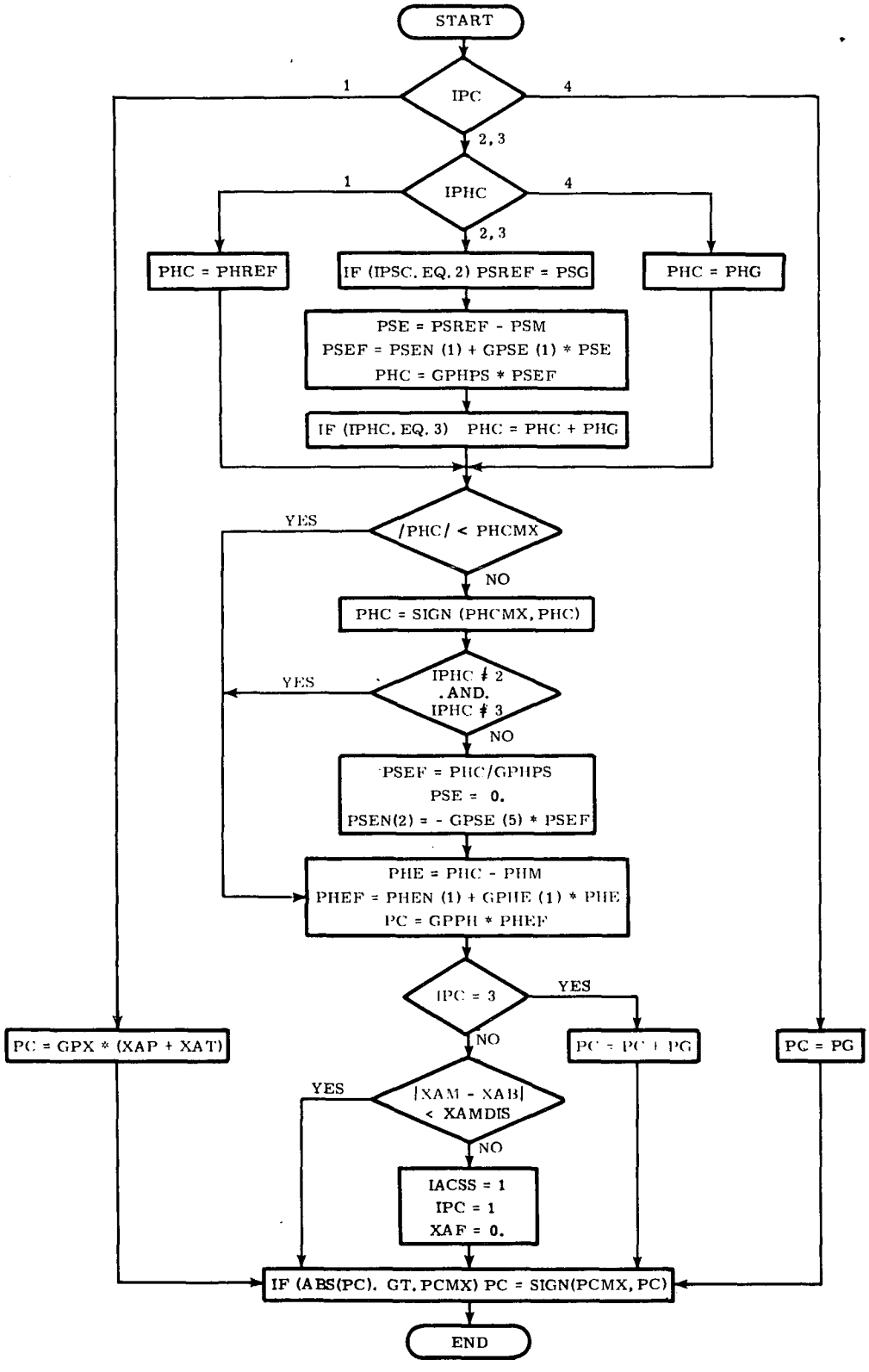



Fig. 4A-9b Filter Part I Subroutine, Cruise Phase Section, Aileron Subsection, Outer Loops

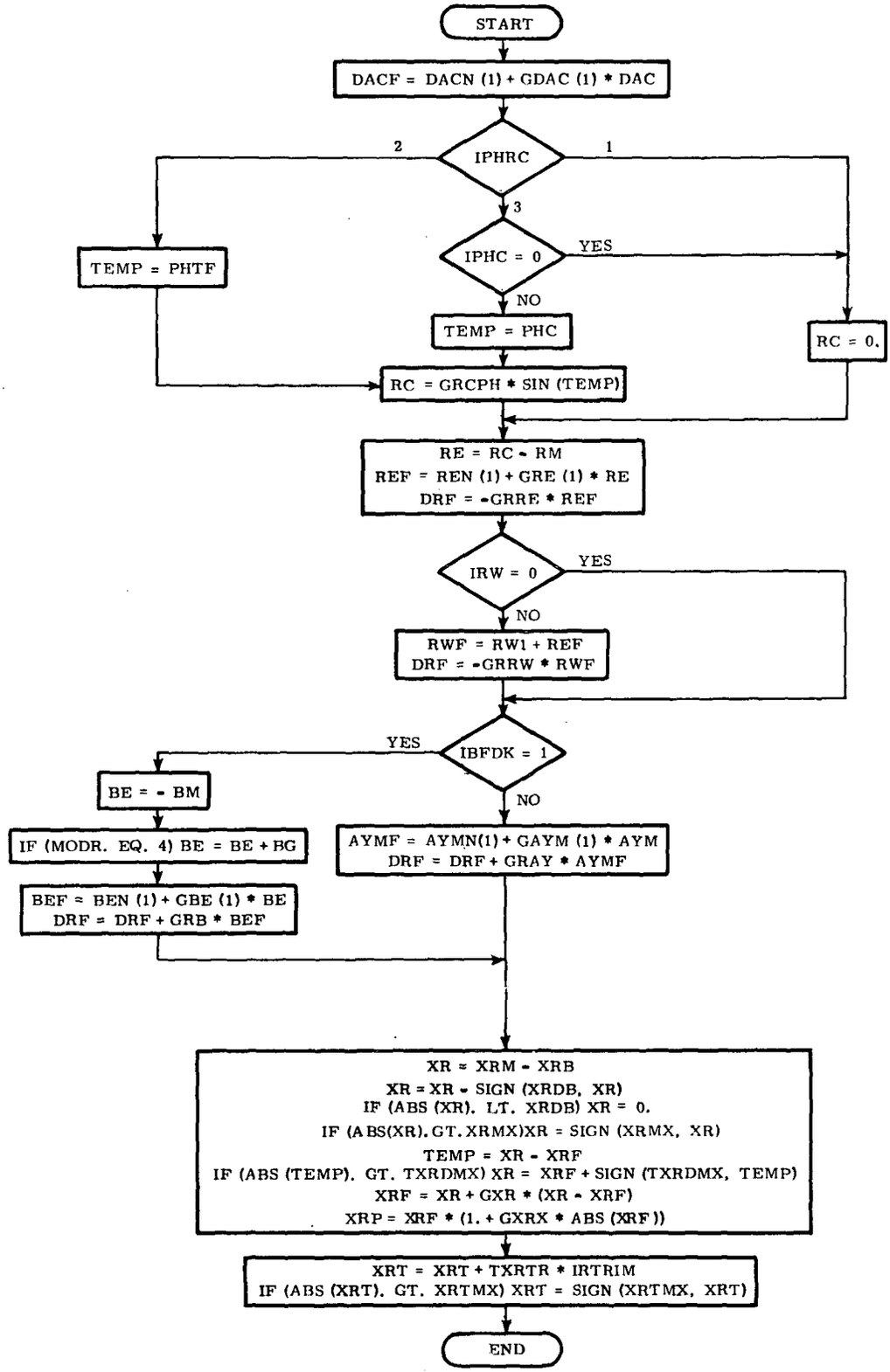



Fig. 4A-10 Filter Part I Subroutine, Cruise Phase Section, Rudder Subsection

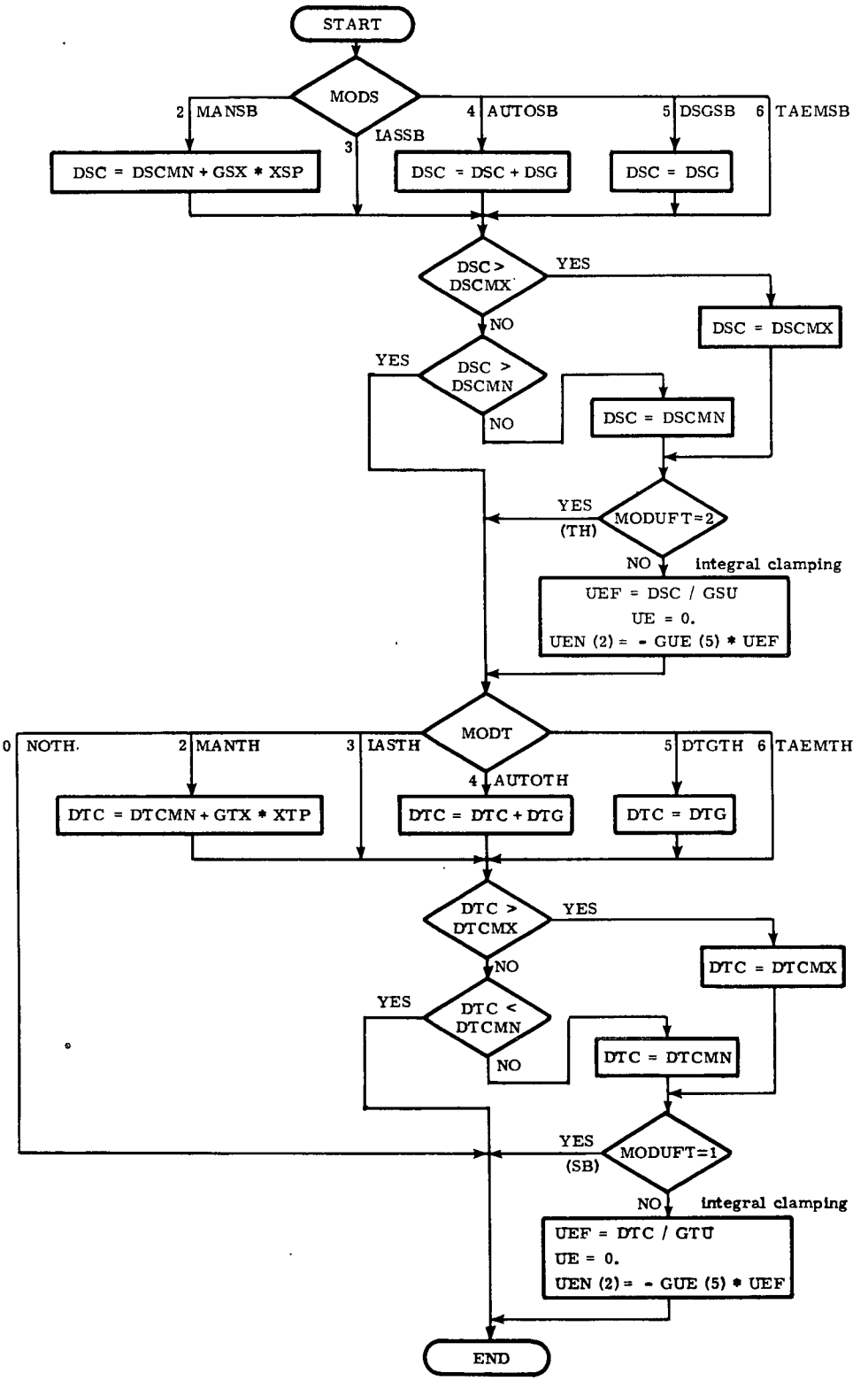



Fig. 4A-11 Control Law Subroutine, Cruise Phase Section, Speed Subsection

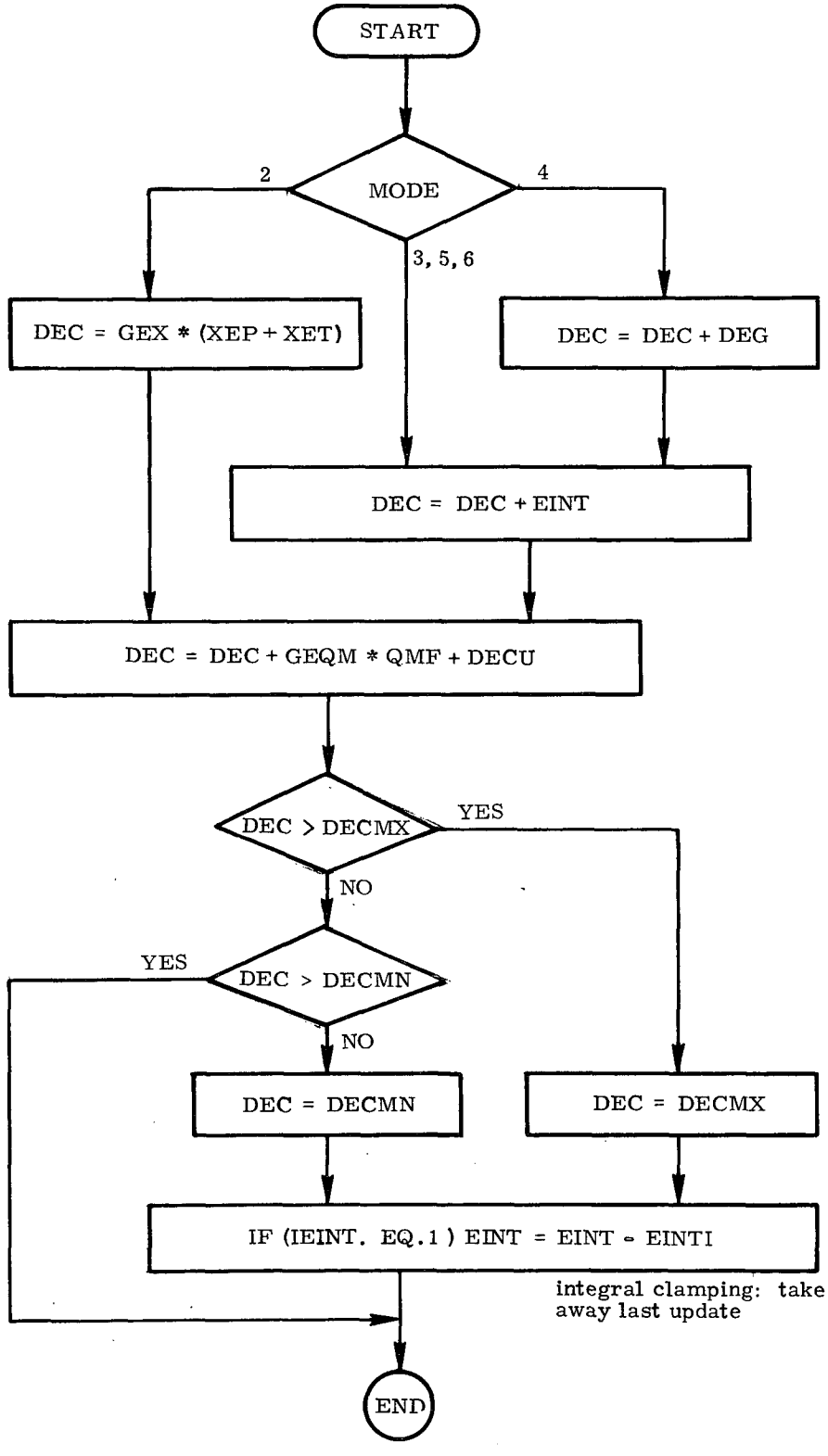
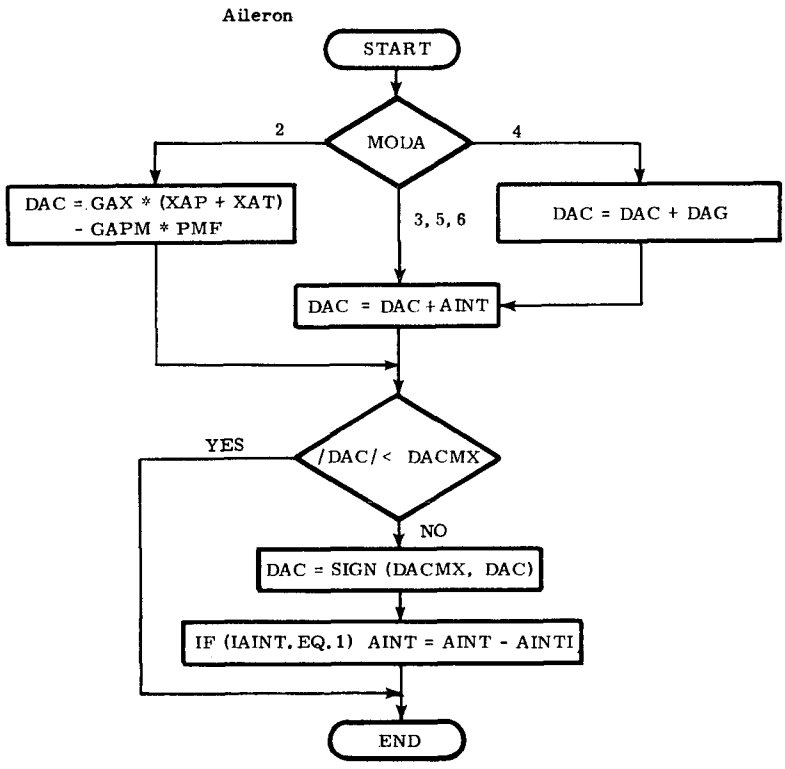



Fig. 4A-12 Control Law Subroutine, Cruise Phase Section, Elevator Subsection



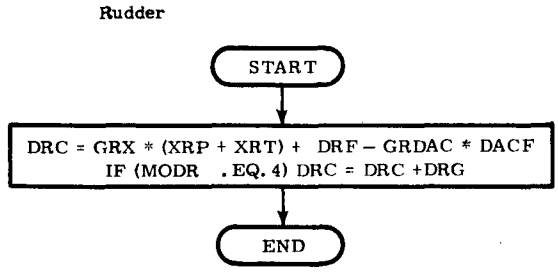
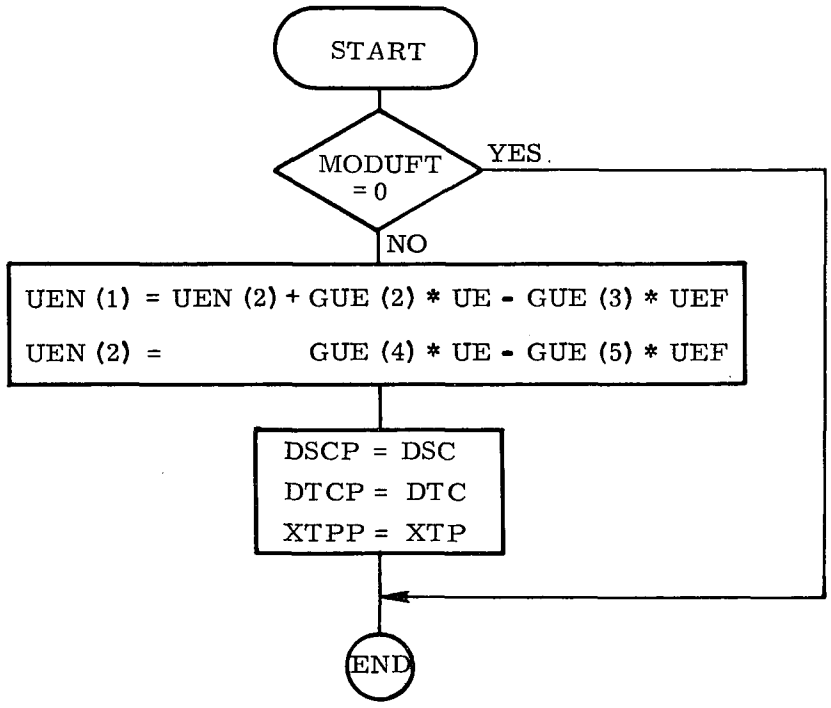




Fig. 4A-13 Control Law Subroutine, Cruise Phase Section, Aileron and Rudder Subsections



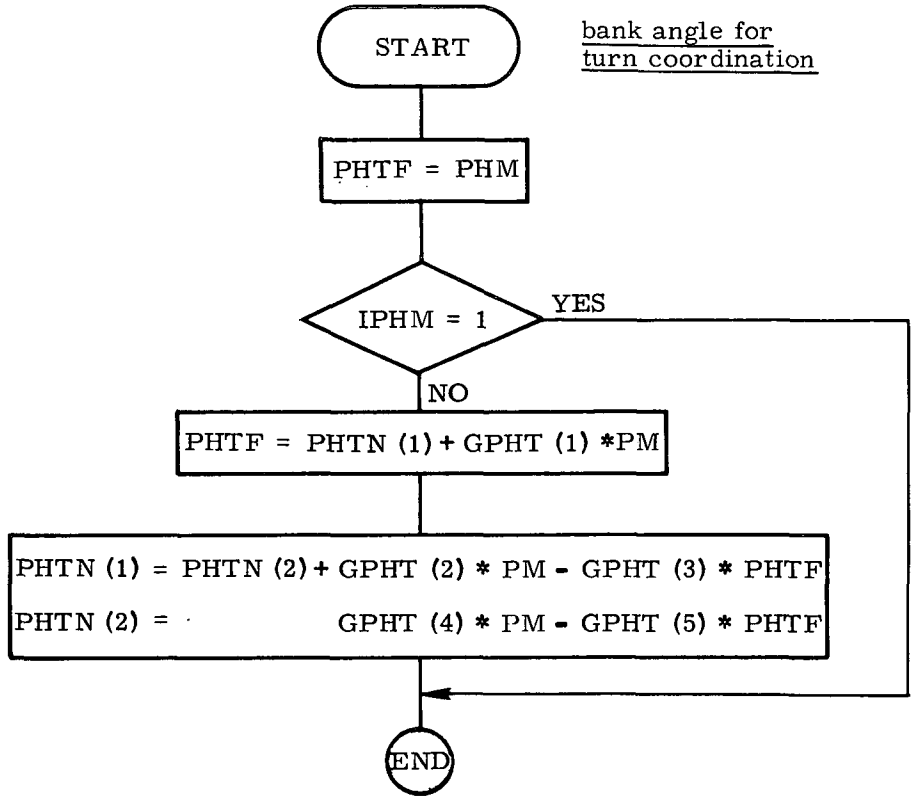
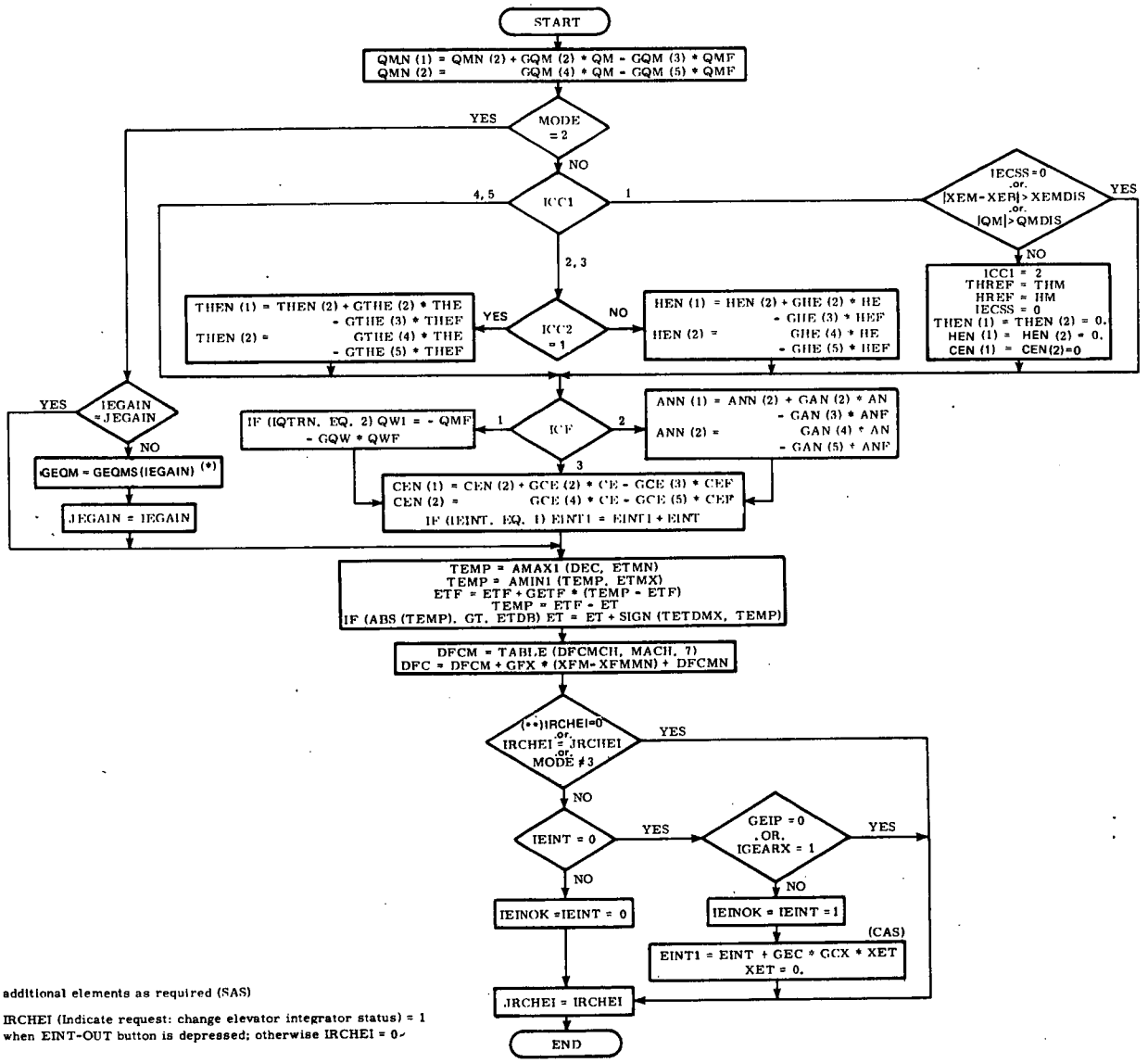



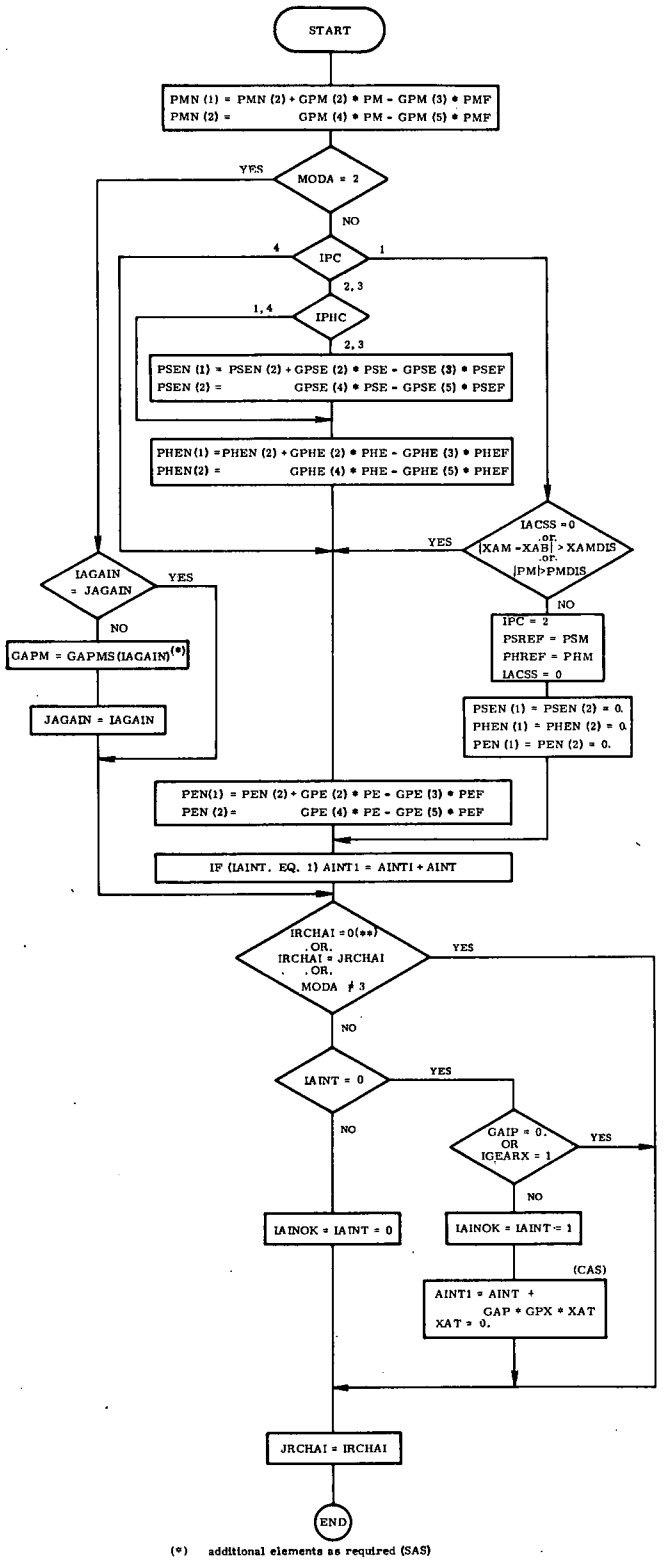

Fig. 4A-14 Filter Part II Subroutine, Cruise Phase Section, Speed Subsection




(*) additional elements as required (SAS)

(**) IRCHEI (Indicate request: change elevator integrator status) = 1 when EINT-OUT button is depressed; otherwise IRCHEI = 0

Fig. 4A-15 Filter Part II Subroutine, Cruise Phase Section, Elevator Subsection




(*) additional elements as required (SAS)

(**) IRCHAI (indicate request: change aileron integrator status)
= 1 when AINT-OUT button is depressed; otherwise IRCHAI = 0

Fig. 4A-16 Filter Part II Subroutine, Cruise Phase Section, Aileron Subsection

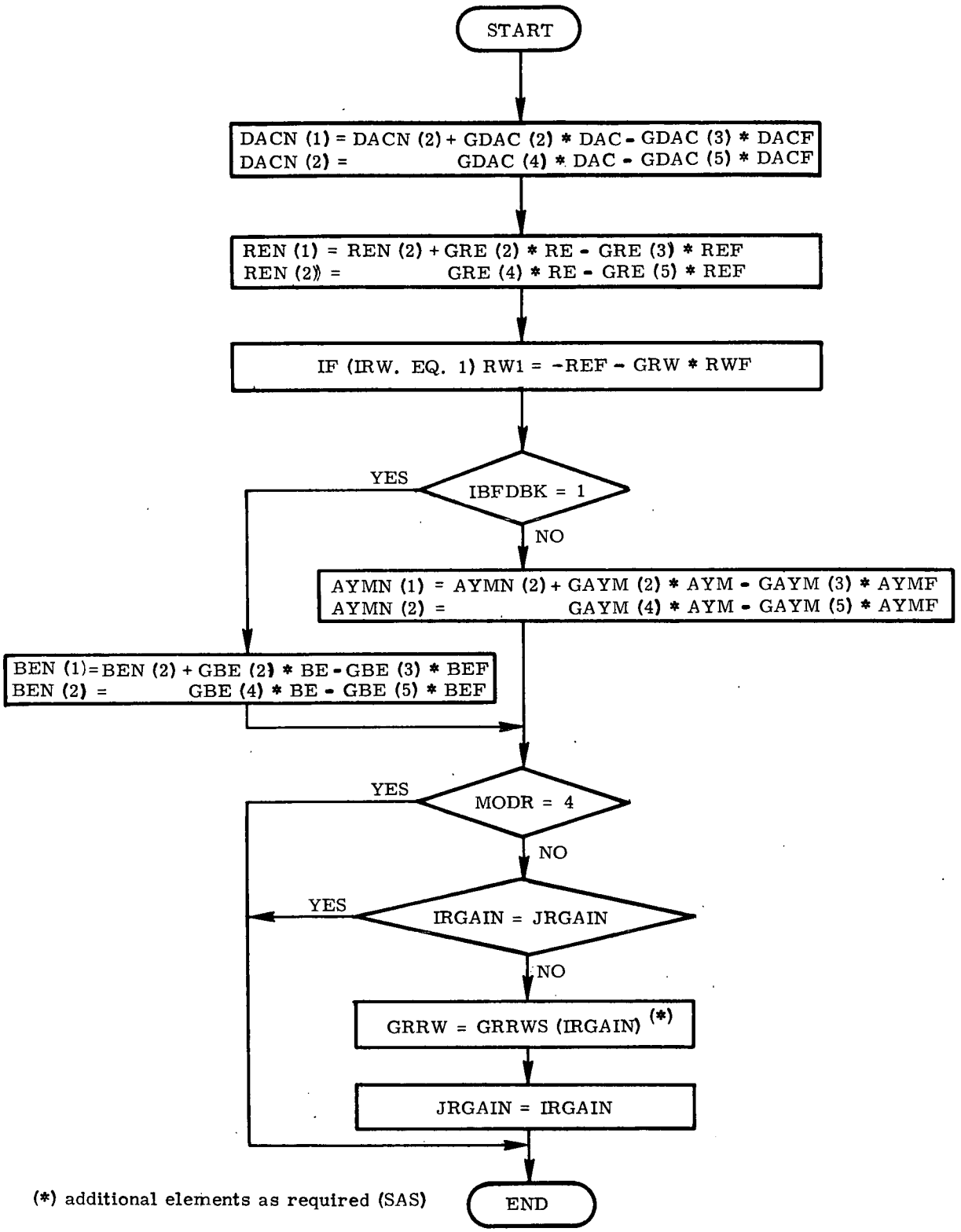



Fig. 4A-17 Filter Part II Subroutine, Cruise Phase Section, Rudder Subsection

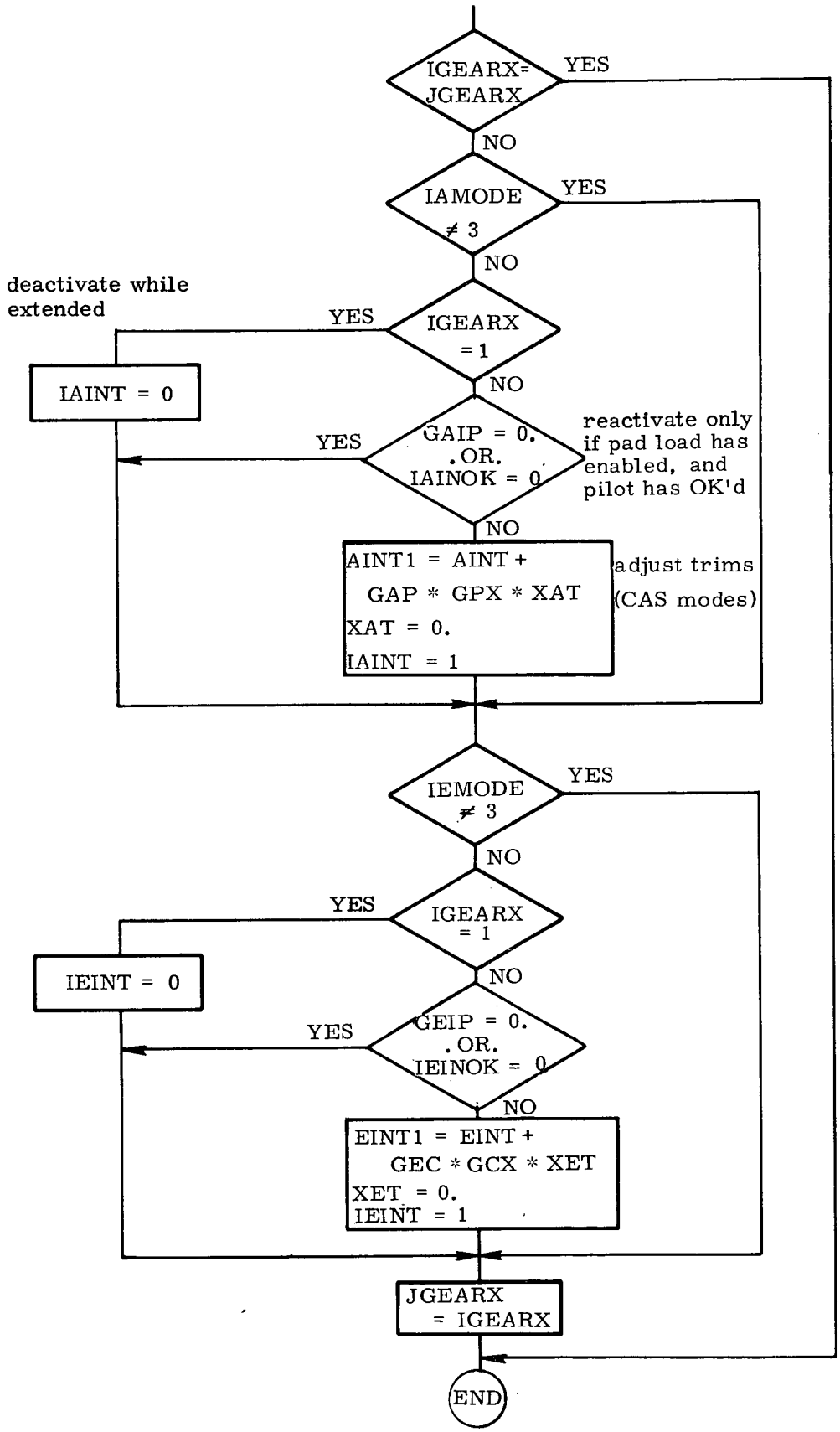



Fig. 4A-18 Filter Part II Subroutine, Cruise Phase Section, Gear/Integrating Moding

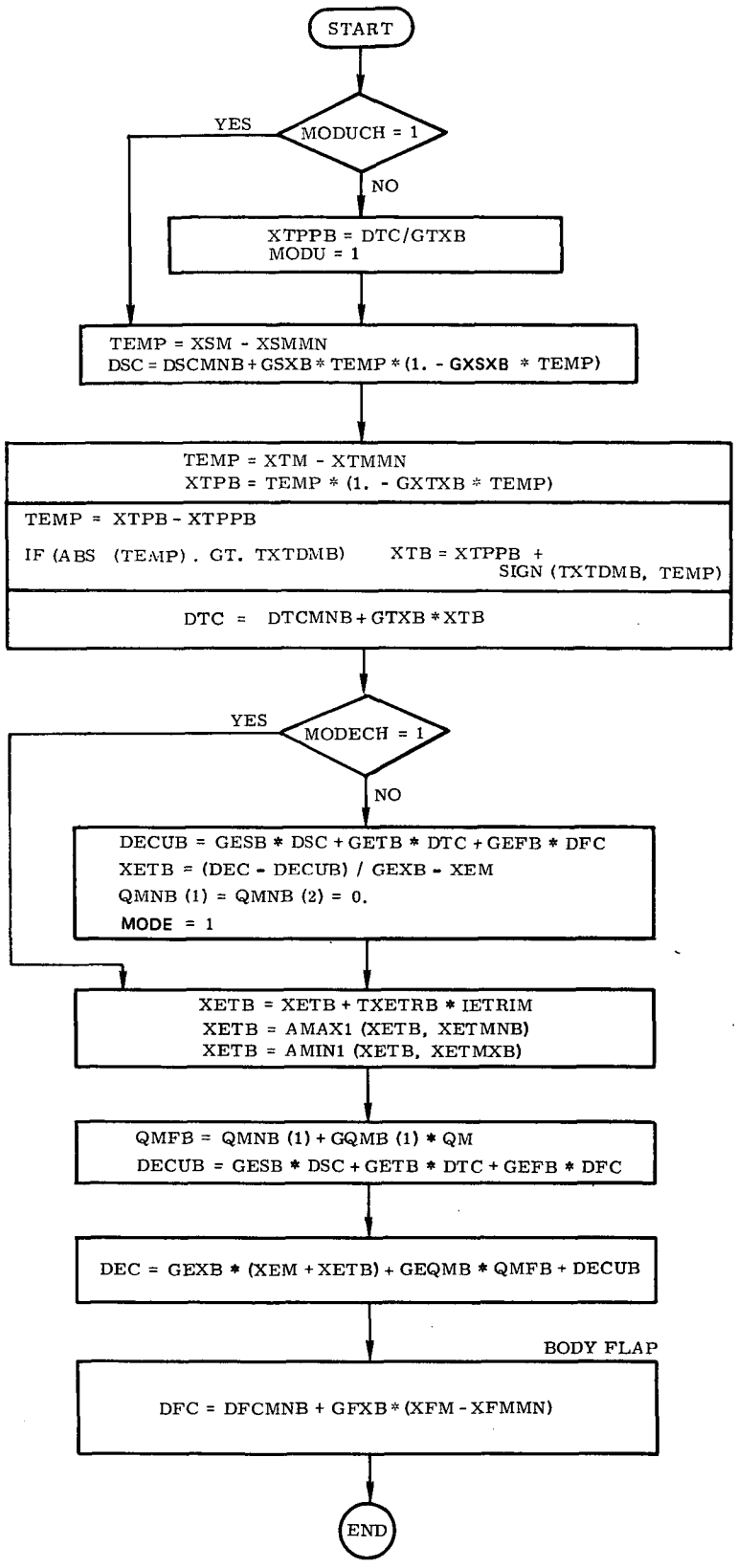



Fig. 4A-19a Cruise Phase BCS Subroutine, Speed, Elevator, and Flap Subsection

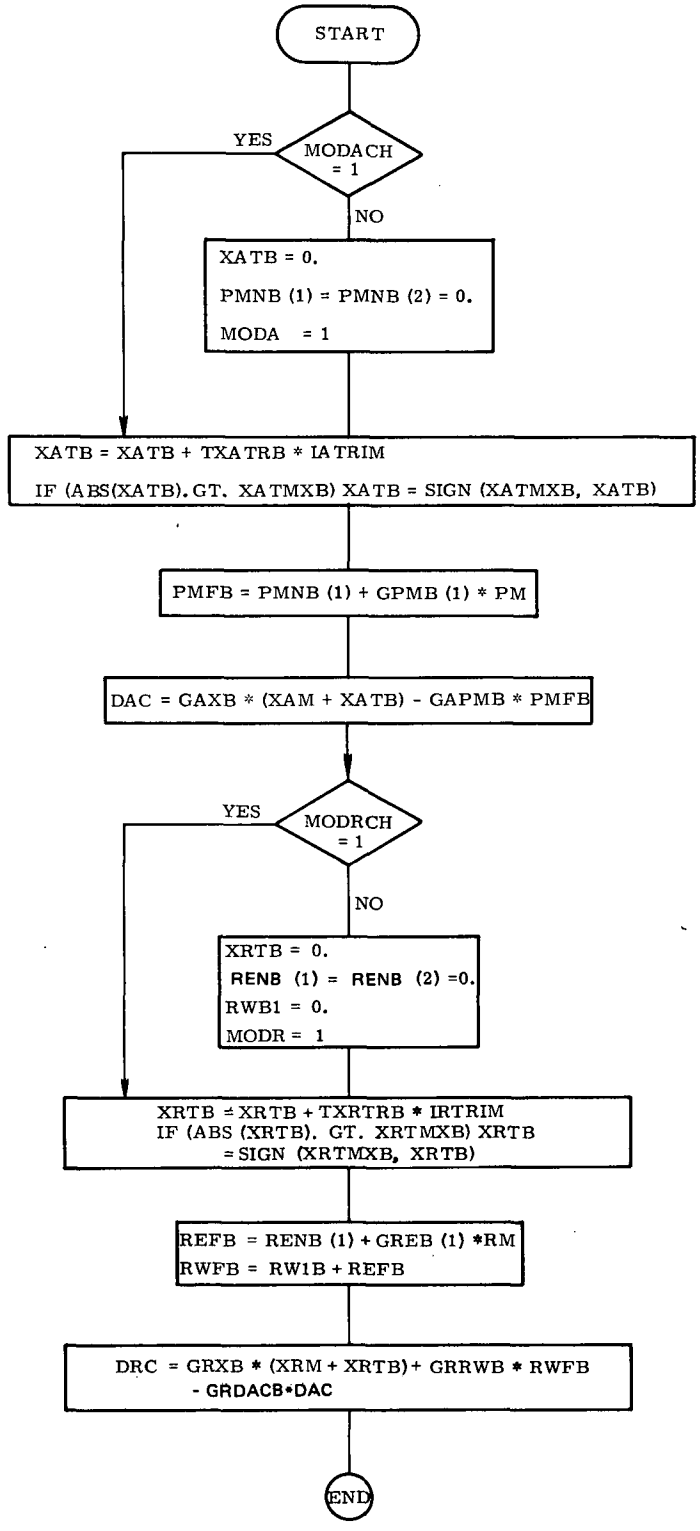



Fig. 4A-19b Cruise Phase BCS Subroutine, Aileron and Rudder Subsection

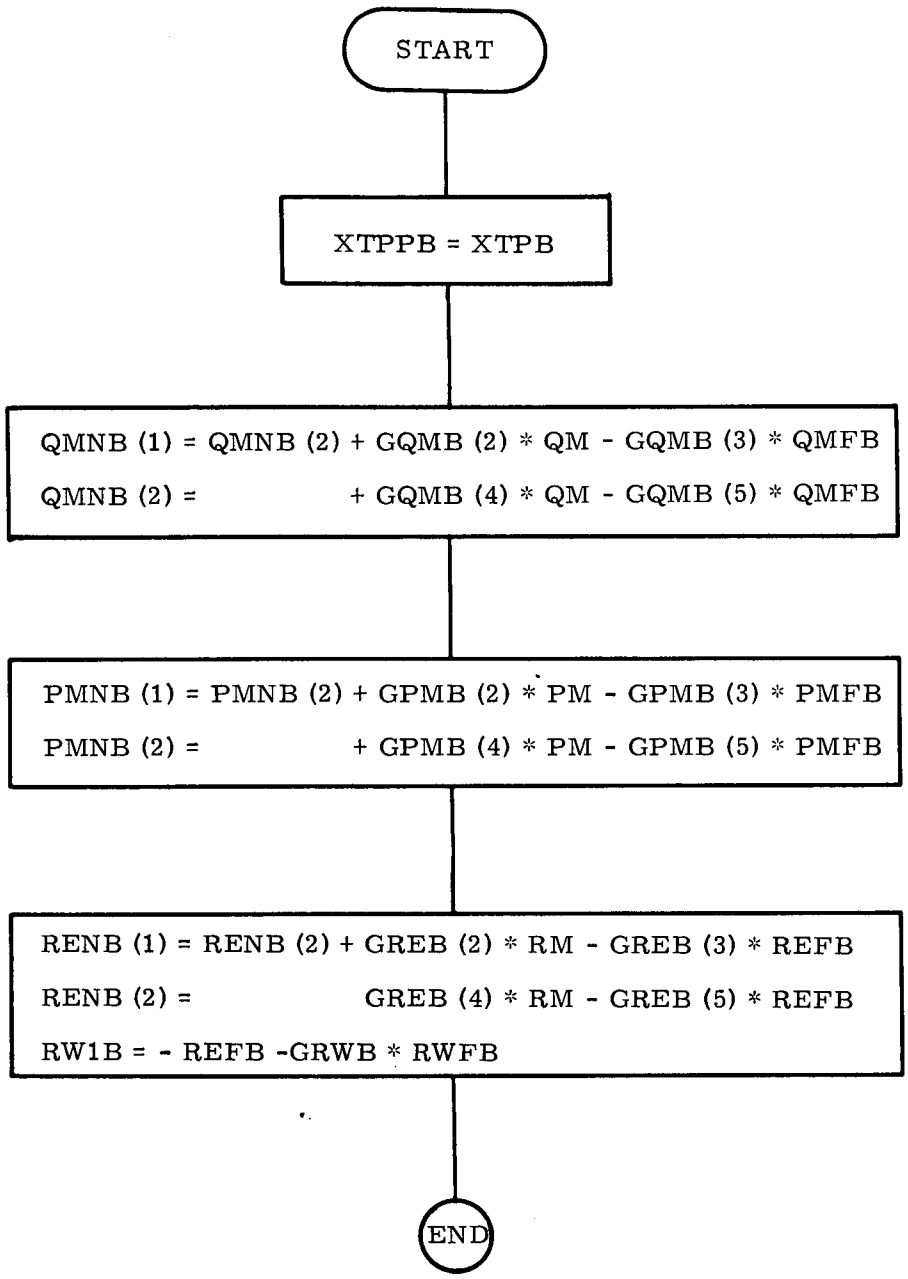



Fig. 4A-20 Cruise Phase BCS Subroutine, Wrapup Subsection

SECTION 5

# ENTRY PHASE

by

Richard D. Goss and Greg L. Zacharias

## 5.1 Introduction

The DFCS design presented in this section is intended to provide stability-axis attitude control for the vehicle during entry, where this mission phase begins outside the sensible atmosphere (e.g., at approximately 400,000 ft altitude) and ends in the high subsonic flight regime. The control logic makes use of both the ACPS and the aerodynamic control surface torques, blended together in such a manner as to minimize ACPS jet activity, while providing capability for simultaneous parallel use when necessary.

The basic design features include gain scheduling and control submode switching (both as functions of selected trajectory parameters) to account for variations in the vehicle dynamics and to allow for reasonably constant closed-loop transient response characteristics. In addition, except for very low dynamic pressure operation, maximum use is made of the aerodynamic control surfaces to minimize ACPS fuel. Finally, the controller synthesis, and thus the design presented here makes advantageous use of the vehicle's aerodynamic torques resulting from trim attitude deviations, allowing the vehicle's rigid body aerodynamics to aid in attitude maneuvers.

This section is organized into three subsections. Subsection 5.2 provides an overview of the entry DFCS organization and operational characteristics, while Subsection 5.3 is a detailed description of the entry DFCS, both functionally and as implemented in the present FORTRAN simulator. Finally, Subsection 5.4 is a brief summary of simulation experience to date.

## 5.2 Overview

Figure 5-1 provides an overview of the entry phase control logic structure. The interface, state estimation, and air data routines are described in Section 2 (Subsection 2.1), Section 3, and Section 2 (Subsection 2.2), respectively. The purpose of the first two is to provide estimates of the stability axis angles and rates, along with the guidance commands, while the purpose of the latter is to provide

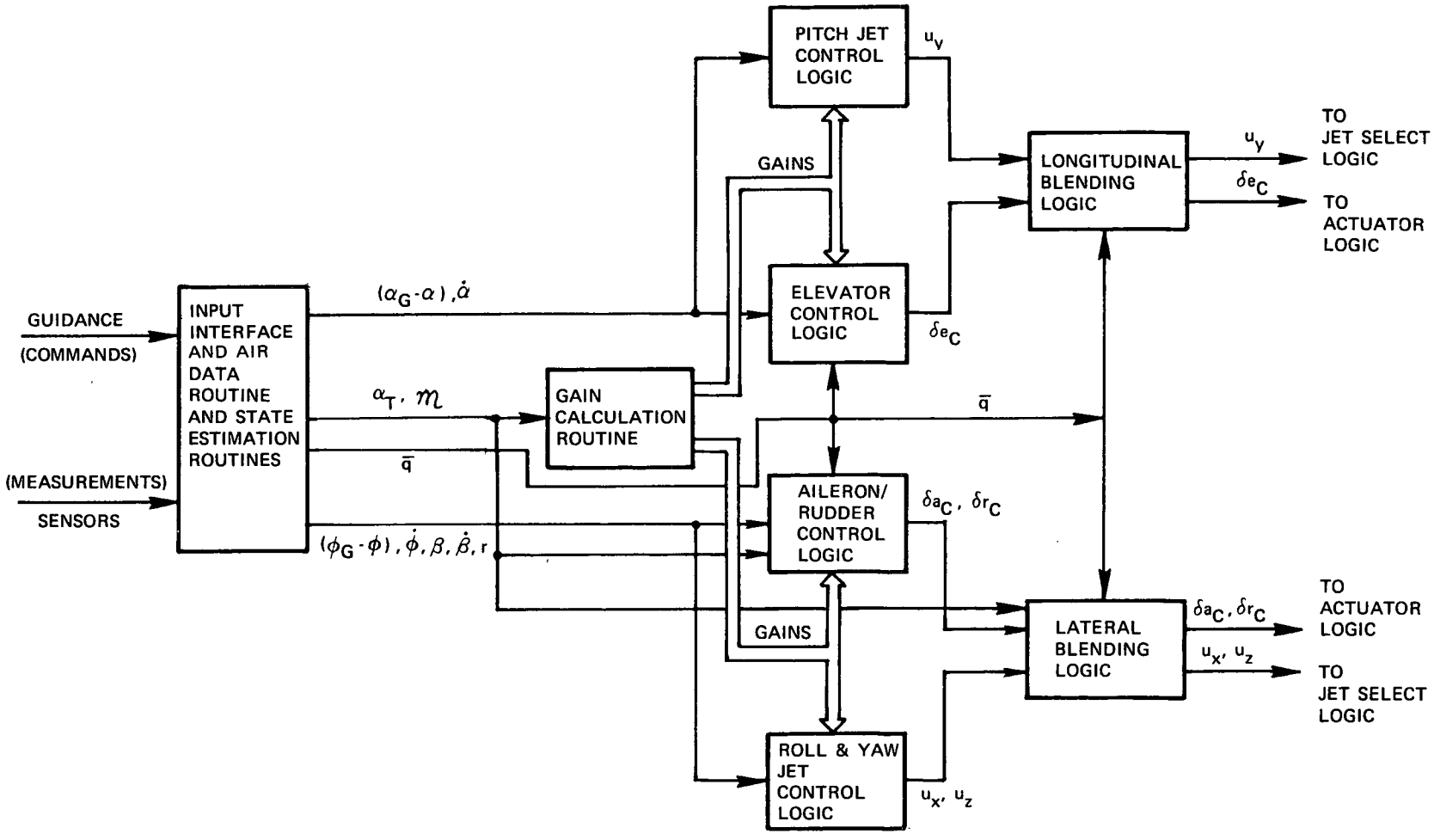



Fig. 5-1 Overview of Entry Control

estimates of the trajectory parameters (trim angle-of-attack, Mach number, and dynamic pressure).

Longitudinal control of angle-of-attack and pitch rate is provided by elevator (tandem elevon) deflections and pitch ACPS jet firings. The elevator logic uses conventional rate and position feedback (with feedforward trim integration) through scheduled gains to maintain relatively constant closed-loop pitch dynamics, while the ACPS logic uses phase-plane control to maintain a desired limit-cycle operation.

Lateral control of sideslip and bank angle is provided by aileron (differential elevon) and rudder deflections, and roll and yaw ACPS jet firings. The aileron/rudder logic uses rate and position feedback of the lateral variables (with feedforward trim integration for center-of-gravity offset compensation) to provide attitude and rate control of both sideslip and bank. The ACPS logic uses phase-plane control to maintain a desired limit-cycle operation when there is low dynamic pressure, and sideslip rate damping to maintain lateral stability when there is high dynamic pressure.

In both the lateral and longitudinal channels, blending of control by the surfaces and the ACPS is provided through a logic which determines ACPS inhibition on the basis of the magnitude of the current surface command, and on the basis of the vehicle's Mach number and angle-of-attack.

## 5.3 Description of Control Laws

This section presents the control system block diagrams, equations, system parameters, and detailed flow charts defining the entry DFCS design. For convenient reference, this section is divided into four subsections. Subsection 5.3.1 and 5.3.2 describe the longitudinal and lateral control laws, while Subsection 5.3.3 specifies the fixed and scheduled parameters used in the present design.

### 5.3.1 Longitudinal Control

Longitudinal control uses both elevator (tandem elevon) and pitch ACPS jets to maintain the commanded angle-of-attack. As shown in Fig. 5-2, which illustrates a typical sequence of control activity during entry, there are two control submodes for the elevator and one for the pitch jets.

Early in the entry, when the dynamic pressure is low, the elevator is used strictly for trim, so as to minimize pitch disturbance torques and thus fuel usage. The trim control law operates in a closed-loop fashion by slowly driving the elevator in the direction which nulls the pitch disturbance. The estimate of the disturbance torque is obtained by observing the pitch ACPS firing history. During this flight

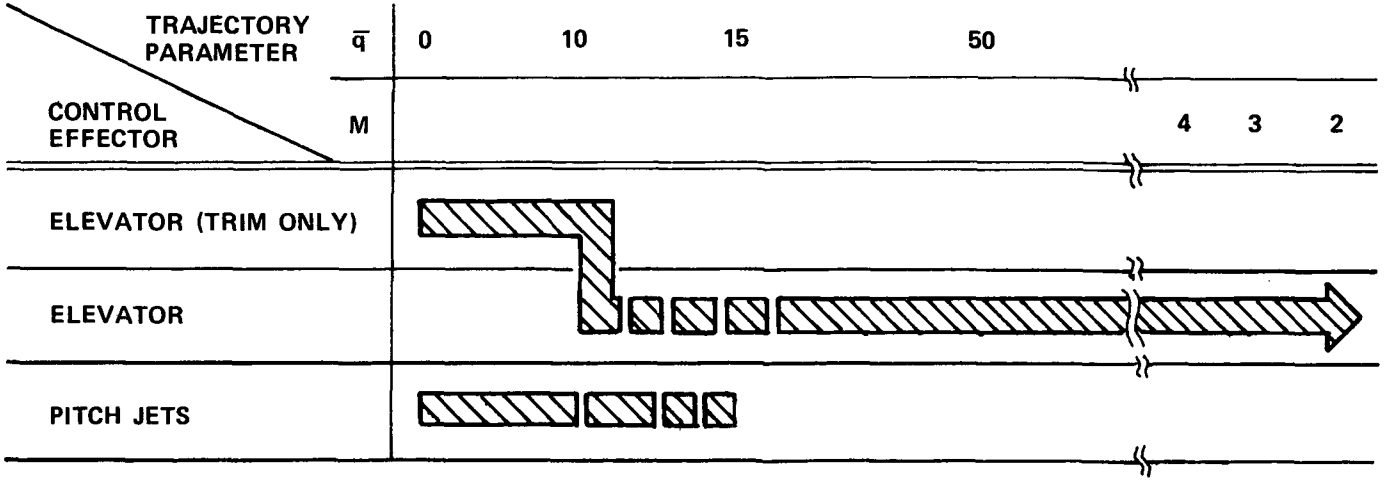



Fig. 5-2 Controller Activity during Entry (Longitudinal)

regime, the pitch ACPS jets, controlled by an angle-of-attack/pitch-rate phase-plane logic, provide attitude and rate control in response to guidance commands. There is no jet inhibition in this regime.

When the dynamic pressure gets high enough, ($\bar{q}_1$), the elevator is used for both trim and transient control. Trim is initially maintained through the use of the trim logic of the previous submode. At a slightly higher dynamic pressure, ($\bar{q}_1^*$), trim is maintained through the use of an integrator acting upon the angle-of-attack error; the initialization value of the integrator is obtained from the value generated by the previous trim logic. Elevator control of transient errors is maintained through feedback of angle-of-attack and pitch rate, with appropriate gains. During this flight regime, pitch jet control identical to that of the previous regime may be utilized together with the elevator, or control with the jets may be inhibited. This is done by the blending logic which is essentially a two-sided deadbanded relay with hysteresis. The commanded elevator deflection is compared with preset fractions of its maximum and minimum values; should the command be "small", it is assumed that the elevator has sufficient control authority, and thus the pitch jets are not required. Conversely, should the elevator command be "large", the pitch jet control is not inhibited, but is allowed to assist the elevator in controlling the vehicle. This blended control continues throughout the rest of the entry, so that the pitch jets are always available for control assistance.

The logic provides for a gradual turn-on of the elevator control so as to avoid a switching transient when the second control submode is entered.

The remainder of this section describes the longitudinal control laws, and their digital implementation, in detail. The block diagrams and equations defining the control logic for the elevator and pitch jets are given in Subsections 5.3.1.1 and 5.3.1.2. Subsection 5.3.1.3 consists of the controller coding summarized in flow chart form.

### 5.3.1.1 Elevator Control

Figure 5-3a is a block diagram of the elevator control loop, showing the two control submodes, with switching determined by $\bar{q}_1$†.

When the dynamic pressure is less than $\bar{q}_1$, only an elevator trim setting is sent to the elevator actuator. This trim setting is determined by integrating the pitch ACPS commands ($u_y$), or, equivalently, by slowly driving the elevator so as to null the longitudinal disturbance torques.

When the dynamic pressure reaches $\bar{q}_1$, closed-loop control of the angle-of-

† See Subsection 5.3.3 for numerical values of the parameters

Note: $\bar{q}_1 < \bar{q}_1^*$

Nonlinearities: (1) Limiter; unity slope; $\delta e_C^{max} = (1 - K_e)\,\delta e_t + K_e\,\delta e^{max}$

$\delta e_C^{min} = (1 - K_e)\,\delta e_t + K_e\,\delta e^{min}$

(2) Clamped integrator; unity slope

(3) Limiter; unity slope

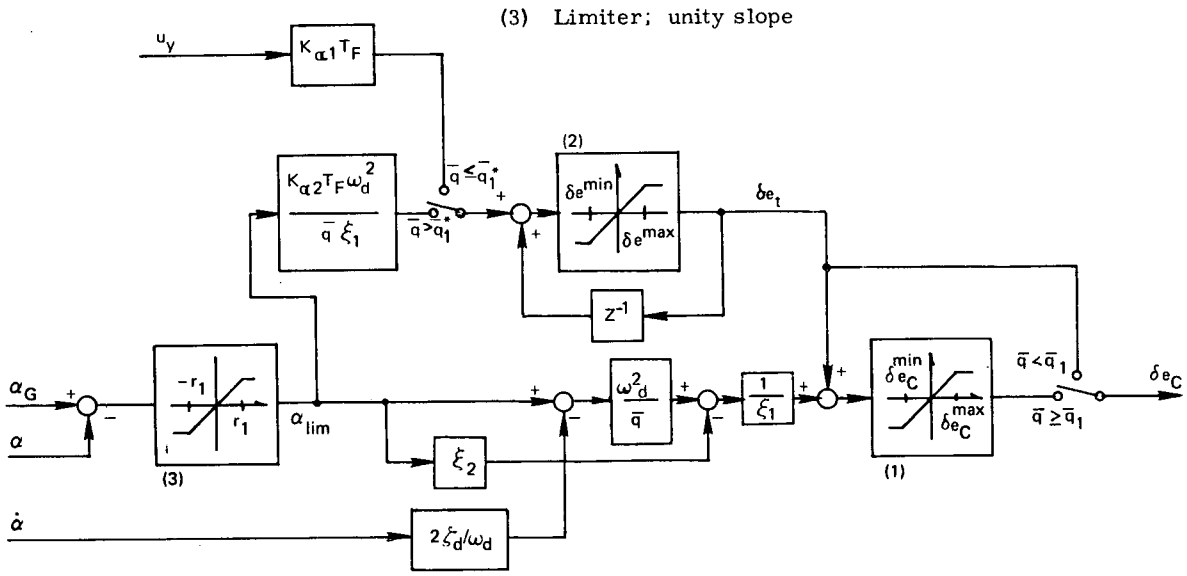



Fig. 5-3a Elevator Control Block Diagram

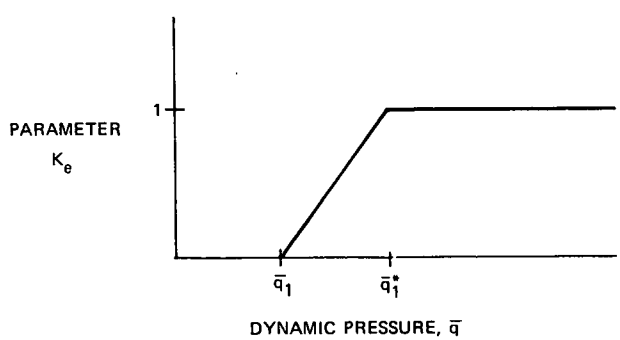



Fig. 5-3b Elevator Deflection Limit Parameter

attack is provided by conventional position ($\alpha$) and rate ($\dot{\alpha}$) feedback through a network which attempts to maintain a constant transient response throughout the flight envelope. Specifically, the desired closed-loop transfer function is chosen to be a well-damped second-order system, or

$$\left[\frac{\alpha}{\alpha_G}\right]_{desired} = \frac{k_\alpha}{s^2 + 2\zeta_d\omega_d s + \omega_d^2} \tag{5-1}$$

where ($\zeta_d$, $\omega_d$) are specified for acceptable performance ($k_\alpha$ is not explicitly specified). The gains shown in the diagram may be separated into one of three types: (1) prespecified by the desired closed-loop characteristics; (2) dynamic pressure dependent ($1/\bar{q}$); or (3) scheduled gains which are functions of the aerodynamic flight regime (e.g., $\xi_1$)†. Trim control is accomplished in this submode either by the trim logic of the previous submode (if $\bar{q} \leq \bar{q}_1^*$) or by a conventional clamped trim integrator acting on the angle-of-attack error, thus providing a steady-state trim elevator command in the absence of transient angle-of-attack errors.

As can be seen from Fig. 5-3a and 5-3b, in order to allow a gradual blending in of active elevator control, the elevator command limits are functions of dynamic pressure which gradually open up from the trim setting to the physical deflection limits.

### 5.3.1.2 Pitch ACPS Control

Shown in Fig. 5-4 is the ACPS phase-plane switch logic representation used for control of the pitch jets. For longitudinal control, the phase-plane coordinates are angle-of-attack error and angle-of-attack rate, or

$$(\epsilon_1, \epsilon_2) = (\alpha - \alpha_G, \dot{\alpha}) \tag{5-2a}$$

The phase-plane is separated into three regions (by the switch curves), in which the commanded jet firing is either positive ($u = +1$), negative ($u = -1$), or zero ($u = 0$). Thus, for the particular application of pitch jet control

$$(u, U) = (u_y, U_y) \tag{5-2b}$$

where $u_y$ and $U_y$ symbolize the jet command and the available acceleration level, respectively.

The equations for the switch curves themselves are based on the fuel-time optimal solution to the double integrator control problem and are modified by

† See Subsection 5.3.3 for numerical values of the parameters

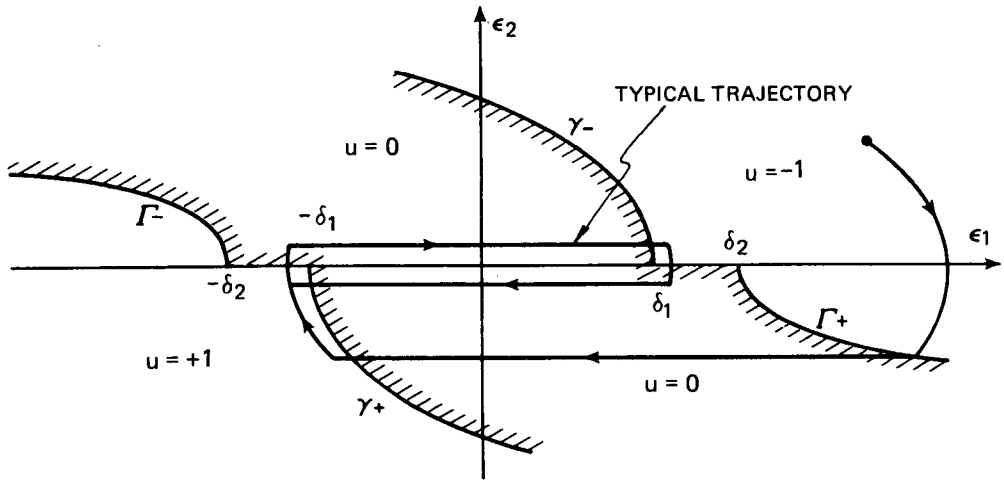
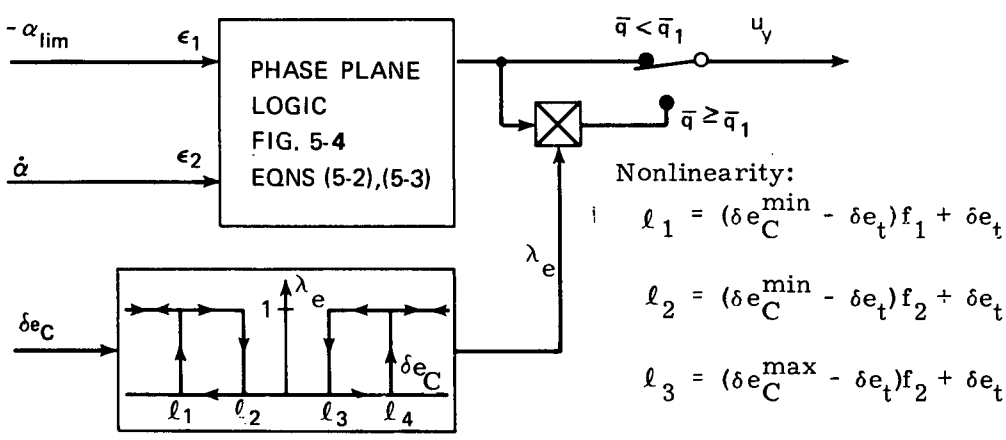



Fig. 5-4 ACPS Switch Logic




Nonlinearity:

$$\ell_1 = (\delta e_C^{min} - \delta e_t) f_1 + \delta e_t$$

$$\ell_2 = (\delta e_C^{min} - \delta e_t) f_2 + \delta e_t$$

$$\ell_3 = (\delta e_C^{max} - \delta e_t) f_2 + \delta e_t$$

$$\ell_4 = (\delta e_C^{max} - \delta e_t) f_1 + \delta e_t$$

Output: $\lambda_e$ = 1 or 0

[Note: $\alpha_{lim}$, $\delta e_C^{max}$, $\delta e_C^{min}$ from Fig. 5-3a]

Fig. 5-5 Pitch ACPS Control Block Diagram

deadband incorporation to ensure practical limit-cycle convergence (see typical trajectory of Fig. 5-4). The switch curves are defined by:

$$\gamma\pm: \ \epsilon_1 = \mp(\delta_1 - \frac{1}{2U}\epsilon_2^2)$$
$$\Gamma\pm: \ \epsilon_1 = \pm(\delta_2 + \frac{\sigma}{2U}\epsilon_2^2) \tag{5-3a}$$

where, in this case, the deadbands are pitch attitude deadbands, or:

$$(\delta_1, \delta_2) = (\delta_{\alpha 1}, \delta_{\alpha 2}) \tag{5-3b}$$

and $\sigma$ is a fuel-time weighting constant greater than unity given for pitch control by:

$$\sigma = \sigma_\alpha \tag{5-3c}$$

The interface (or blending) logic between commanded pitch jet firings and the commanded elevator deflections is shown in Fig. 5-5. The approach taken is to inhibit firing when there is "sufficient" pitch control acceleration from the elevator. This measure of sufficiency is obtained by comparing commanded elevator ($\delta e_C$ shown in Fig. 5-3a) with threshold values to determine whether or not "excessive" elevator deflection is being called for. The hysteresis path included in the blending interface avoids chatter in elevator and ACPS activity due to cross-coupling effects. Note that this design includes an additional switch to preclude jet inhibition when the dynamic pressure is low and the elevator is used solely for trim.

#### 5.3.1.3 Digital Implementation

Coding implementation for longitudinal control during the entry phase is summarized in flow-chart form in Fig. 5-6.

### 5.3.2 Lateral Control

Lateral control uses both aileron (differential elevons) and rudder, and roll and yaw ACPS jets to maintain small sideslip angles and follow commanded bank maneuvers. As shown in Fig. 5-7, which illustrates a typical sequence of control activity during entry, there are three control submodes for the aerodynamic control surfaces, one for the yaw jets, and three for the roll jets.

Early in the entry, when the dynamic pressure is low, the aileron and rudder commands are zero. During this regime, the yaw jets, controlled by a bank angle/yaw-rate phase-plane logic, provide attitude and rate control of the bank angle in response to guidance commands. The roll jets are controlled by one of two logics, depending on the yaw jet activity. When there are no yaw commands, the roll jets, controlled by a sideslip/sideslip-rate phase-plane logic, maintain small sideslip

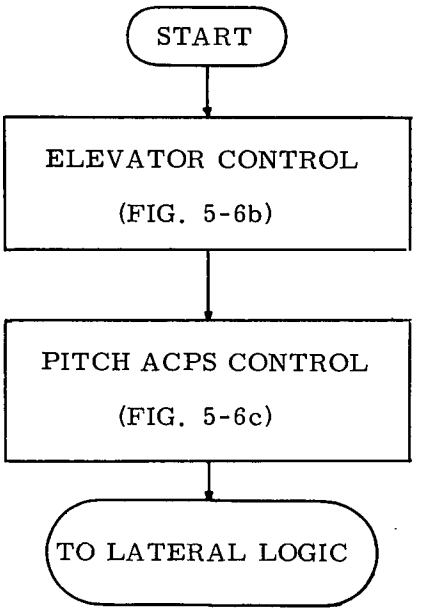



Fig. 5-6a Logic Flow for the Longitudinal Control Channel

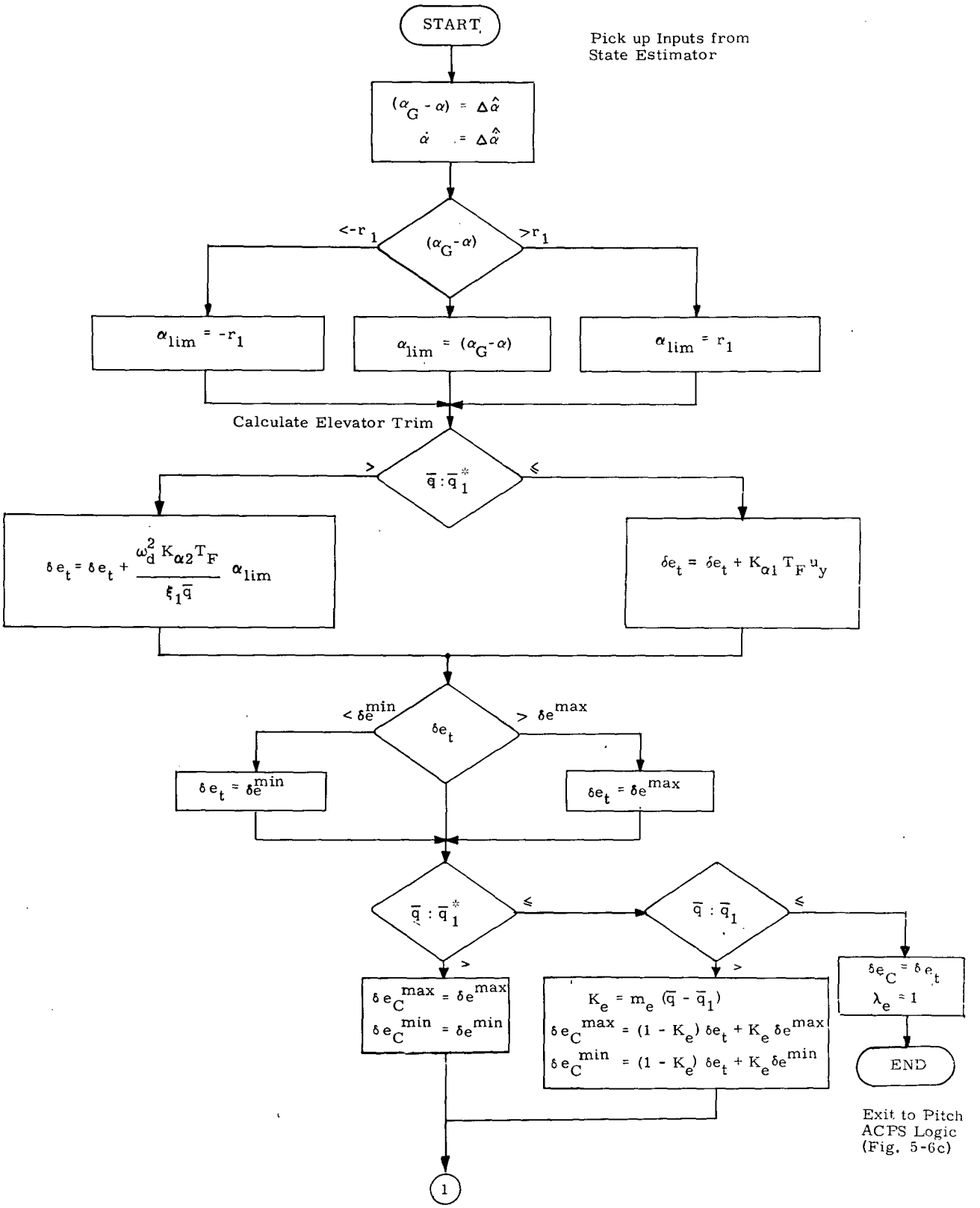



Fig. 5-6b Logic Flow for Elevator Control

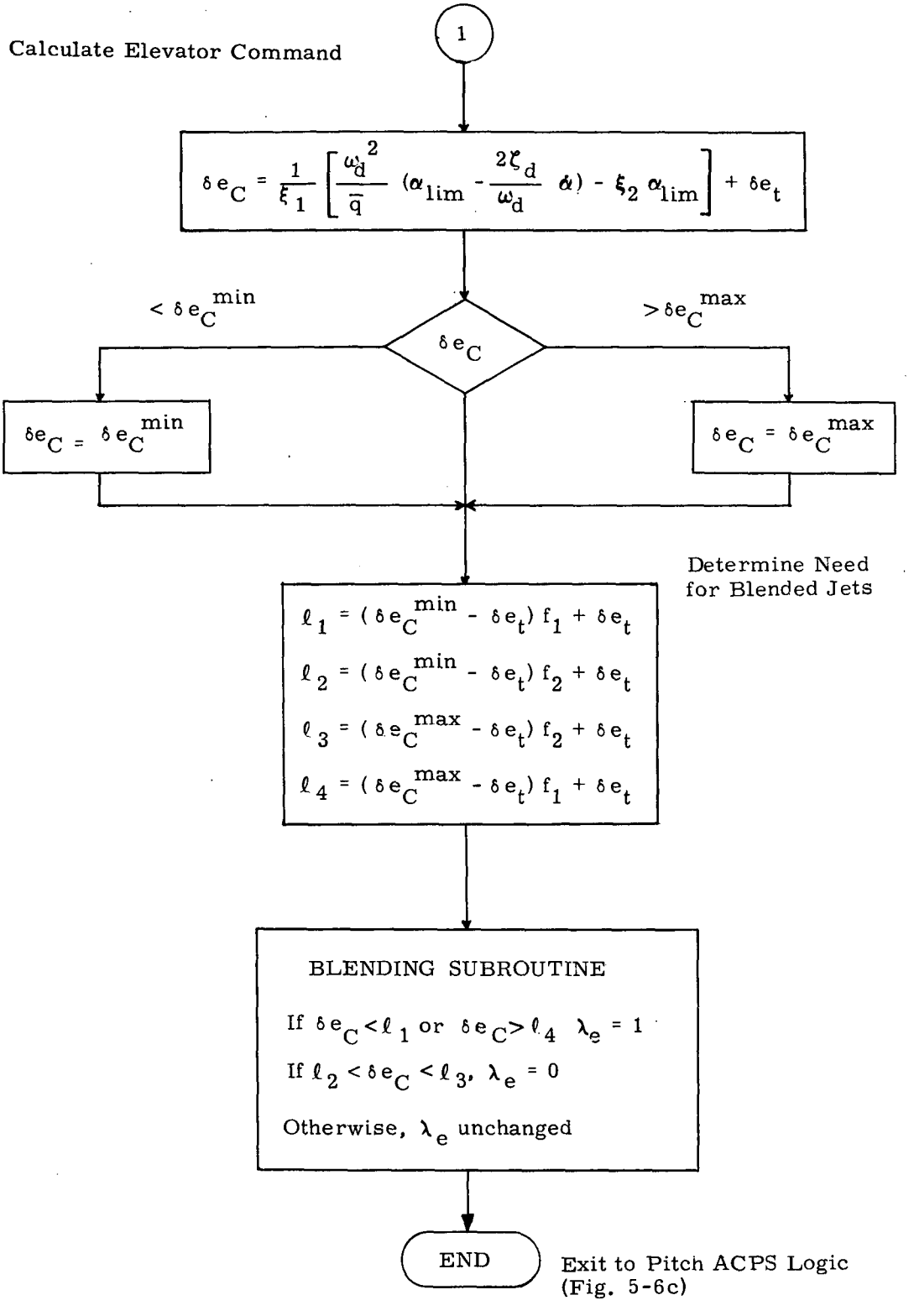



Fig. 5-6b Logic Flow for Elevator Control (Cont)

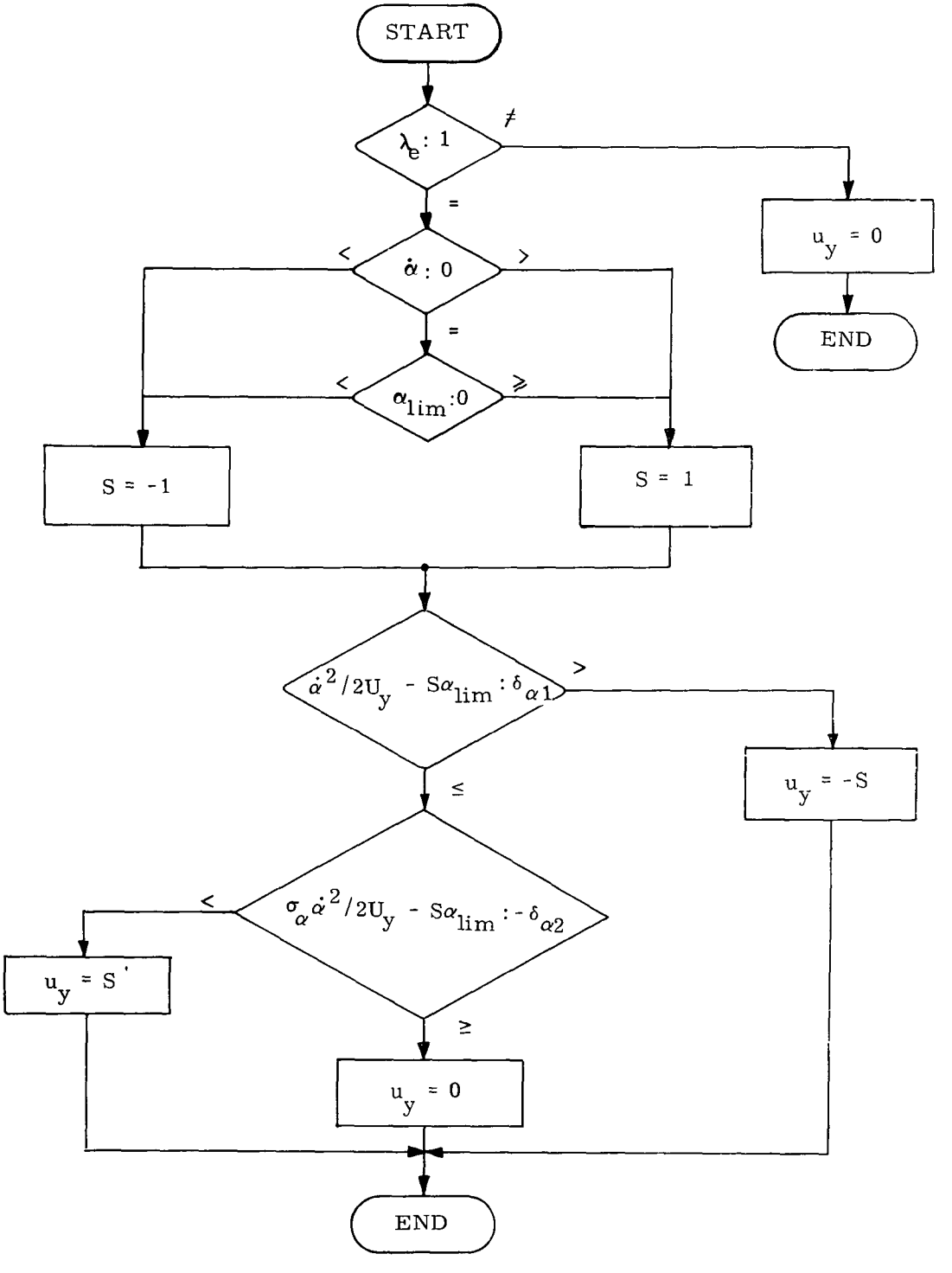



Fig. 5-6c Logic Flow for Pitch ACPS Control

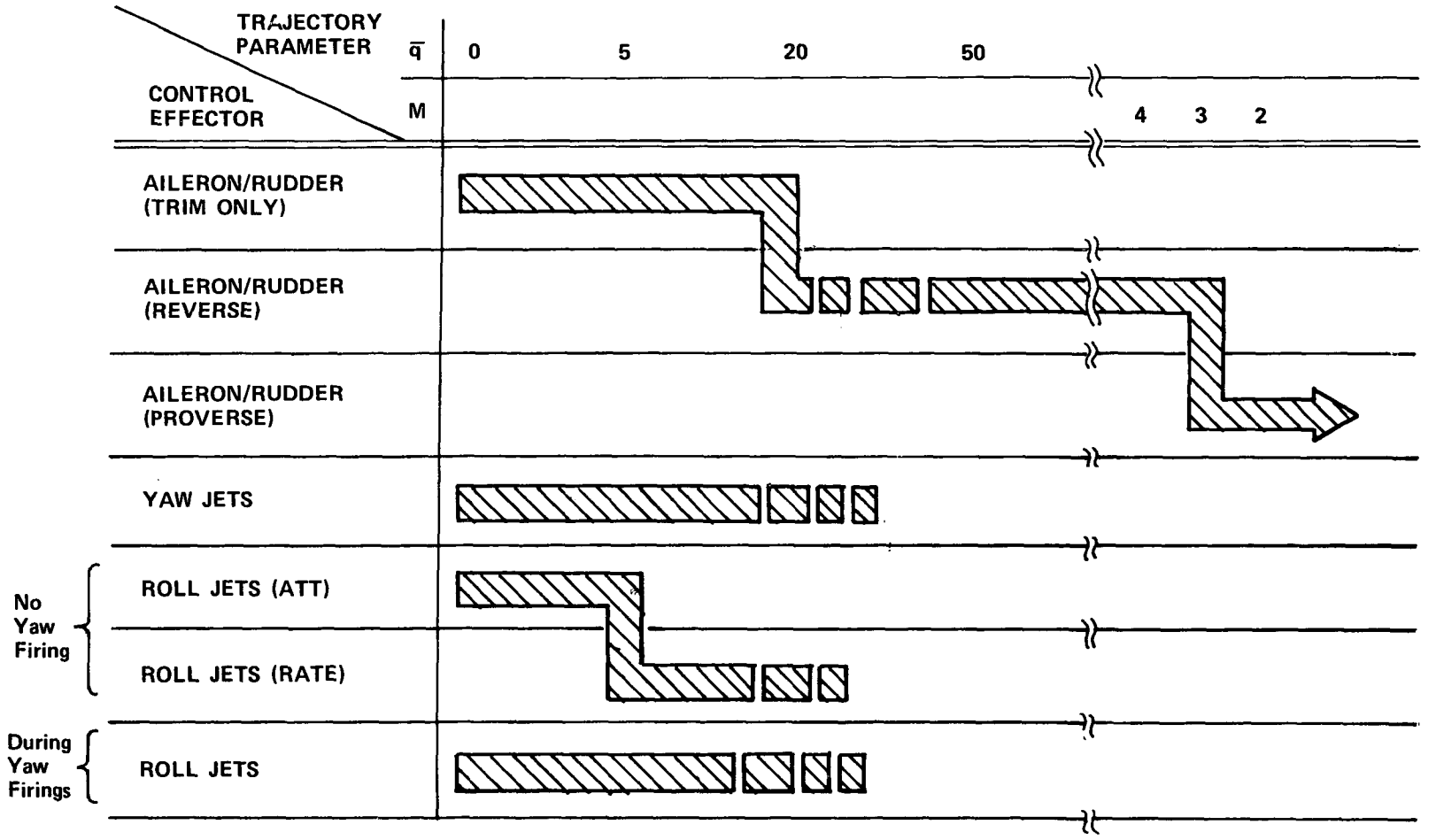



Fig. 5-7 Controller Activity during Entry (Lateral)

attitudes and rates. When the yaw jets are commanded to maneuver the vehicle, the roll jets are controlled by a sideslip/sideslip-rate phase-plane logic, which is similar to the first submode except that compensation for yaw jet torques is included. Again, the purpose of this logic is to maintain small sideslip attitudes and rates. There is no jet inhibition in this flight regime.

Later in the entry, when the dynamic pressure reaches a certain prespecified value ($\bar{q}_3$), the aileron and rudder commands are still zero. The yaw jets are controlled by the same logic as in the previous regime. The roll jets are controlled by one of two logics, again depending on yaw jet activity. When there are no yaw jet commands, the roll jets are commanded by a deadbanded relay logic which simply provides rate damping in sideslip. When the yaw jets are commanded to maneuver the vehicle, the roll jets are commanded by the same corresponding roll jet logic of the previous regime. There is no jet inhibition in this flight regime.

Still later in the entry, when the dynamic pressure is higher ($>\bar{q}_2$), the aileron is used for both trim and transient control. Trim is initially maintained at zero. At a slightly higher dynamic pressure ($\bar{q}_2^*$), trim is maintained through the use of an integrator acting upon the bank angle attitude error; the initialization value of the integrator is zero. Aileron control of transient errors (in a "reverse" sense, as it has come to be labeled) is maintained through feedback of bank and sideslip attitudes and rates with appropriate gains. During this regime, the rudder command is null until the Mach number reaches a certain level ($\mathcal{M}_1$), after which it is proportional to the commanded aileron deflection from trim. The proportionality variable is a constant. Also during this regime, roll and yaw jet control identical to that of the previous regime may be utilized together with the aileron and rudder, or control with the jets may be inhibited. This is done with the same type of blending logic used longitudinally; here, the commanded aileron deflection determines jet inhibition. Also as in the longitudinal case, there is a gradual turn-on of the allowed aerodynamic control authority.

Still later in the entry, when the vehicle has crossed a prespecified curve in the Mach-alpha plane, only the aileron and rudder are used for transient control, the jets being inhibited for the remainder of the entry. Aileron and rudder control (in a "proverse" sense to distinguish it from that of the previous submode) of transient errors is maintained through feedback of bank and sideslip attitudes and rates (to both surfaces), with appropriate gains. Lateral trim for this regime has not yet been implemented.

The remainder of this section describes the lateral control laws, and their digital implementation, in detail. The block diagrams and equations defining the control logic for the aileron and rudder are given in Subsection 5.3.2.1, while those for the yaw and roll jets are given in Subsection 5.3.2.2. Subsection 5.3.2.3 consists of the controller coding summarized in flow-chart form.

5.3.2.1 Aileron/Rudder Control

Figure 5-8 is an overview of the aileron/rudder control logic, illustrating the two control submodes, with switching determined by the flag $I_{LAT}$. This flag is specified by vehicle controllability considerations (see Subsection 5.4) and is defined by a curve in the Mach-alpha plane as shown in Fig. 5-9.

5.3.2.1.1 Early Aileron/Rudder Control

Figure 5-10a illustrates the "reverse" control submode. When the dynamic pressure is less than $\bar{q}_2$, aileron trim is set to zero, as are the aileron and rudder commands. When the dynamic pressure reaches $\bar{q}_2$, aileron trim and rudder command are still set to zero, but closed-loop control of bank and sideslip (by the aileron alone) is provided by position ($\beta$, $\phi$) and rate ($\dot{\beta}$, $\dot{\phi}$) feedback of the lateral state variables, through a network which attempts to maintain a constant response throughout the trajectory. Specifically, the desired closed-loop transfer function for bank angle response is chosen to have four specified complex poles, or:

$$\left[\frac{\phi}{\phi_G}\right]_{desired} = \frac{k_\phi (s^2 - \omega_\phi^2)}{(s^2 + 2\zeta_1\omega_1 s + \omega_1^2)(s^2 + 2\zeta_2\omega_2 s + \omega_2^2)} \tag{5-4}$$

where ($\zeta_i$, $\omega_i$) are chosen for acceptable transient response ($k_\phi$ is not explicitly specified and $\omega_\phi$ is a function of the airframe dynamics.). This response is obtained, in the absence of rudder effectiveness, by making use of sideslip mistrims to bank the vehicle.

As with the longitudinal controller, the gains of Fig. 5-10a may be separated into one of three types: (1) prespecified by the desired closed-loop characteristics (e.g., $a_1$); (2) dynamic pressure dependent; or (3) scheduled gains which are functions of the aerodynamic flight regime (e.g., $\xi_3$). For convenient reference, the gains of the first category are defined in terms of desired response characteristics with the following equation set:

$$\begin{aligned} a_1 &= \omega_1^2 \omega_2^2 \\ a_2 &= 2(\zeta_1\omega_2 + \zeta_2\omega_1)/(\omega_1\omega_2) \\ a_3 &= 2(\zeta_1\omega_1 + \zeta_2\omega_2) \\ a_4 &= \omega_1^2 + \omega_2^2 + 4\zeta_1\zeta_2\omega_1\omega_2 \end{aligned} \tag{5-5}$$

where, as noted above, the ($\zeta_i$, $\omega_i$) are chosen for acceptable roll response.



C-3

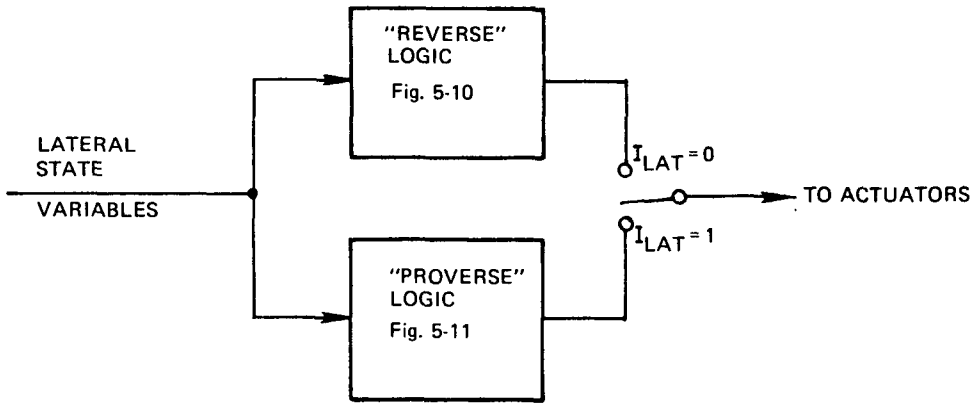



Fig. 5-8 Overview of Aileron/Rudder Control Logic

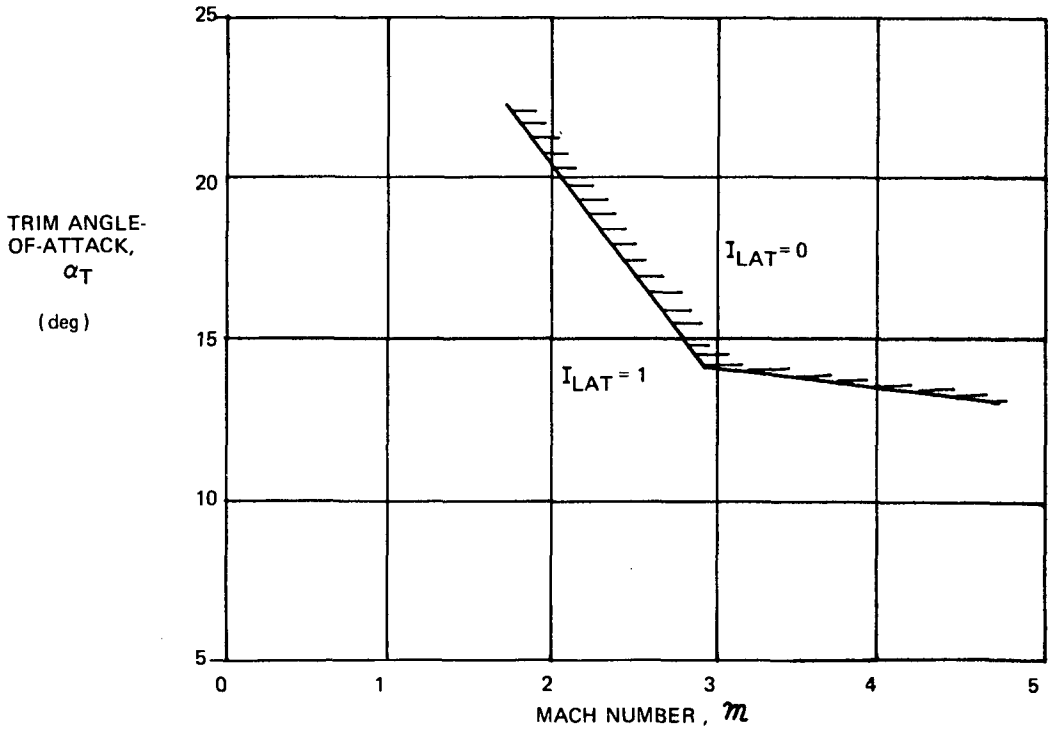
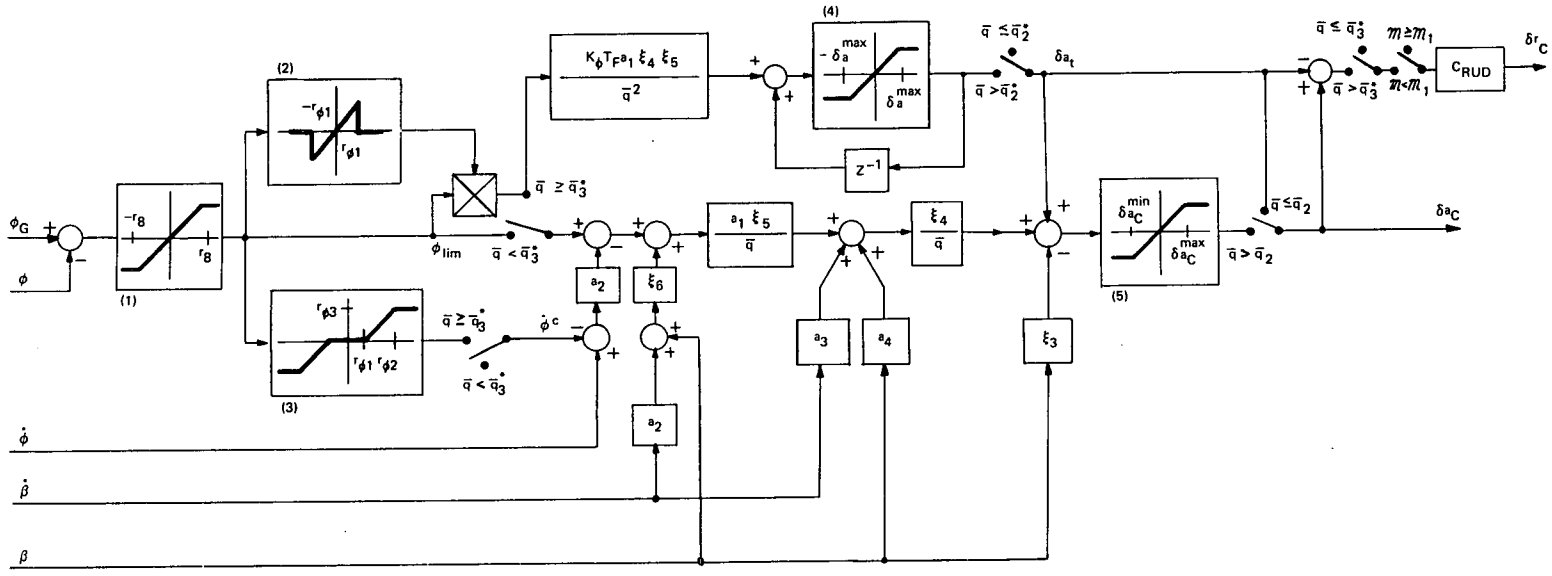



Fig. 5-9 Control Submode Switch Parameter Definition




NON-LINEARITIES:
(1) Limiter; unity slope
(2) Inverse deadband; unity slope
(3) Limiter with deadband (symmetric)
(4) Limiter; unity slope
(5) Limiter; unity slope; $\delta a_C^{max} = K_a \delta a^{max}$, $\delta a_C^{min} = -K_a \delta a^{max}$

NOTE: $\bar{q}_2 < \bar{q}_2^* < \bar{q}_3^*$

Fig. 5-10a Aileron/Rudder Control Block Diagram (Reverse Mode, $I_{Lat} = 0$)

Aileron trim is commanded zero until the dynamic pressure reaches $\bar{q}_2^*$, and it is then determined through the use of a conventional clamped integrator in the bank angle error channel. This allows the vehicle to trim to a non-zero sideslip angle (by commanding a non-zero trim aileron deflection) in the presence of body-axis roll disturbance torques due to lateral displacements in the vehicle's center-of-gravity. To avoid interpreting large transients (such as occur during a bank reversal) as lateral cg-offset induced mistrims, integration of large magnitude errors is inhibited by the $\lambda_\phi$ switch. The gradual expansion of the aileron command limits during this submode as dynamic pressure increases is shown in Figs. 5-10a and 5-10b.

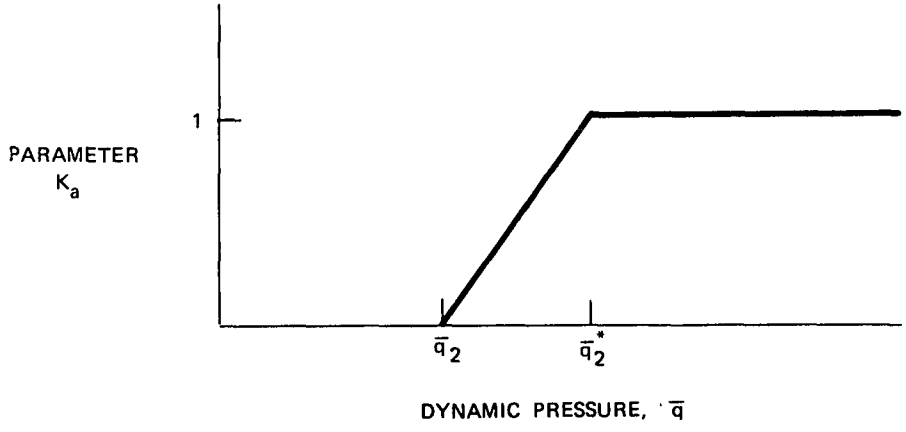



Fig. 5-10b Aileron/Deflection Limit Parameter

When the dynamic pressure reaches $\bar{q}_3^*$, later in this "reverse" submode, a rate command loop becomes effective when large attitude errors are detected (as in a bank reversal). As shown in Fig. 5-10a, when the attitude error is greater than $r_{\phi 1}$, the logic is no longer concerned with maintaining a bank attitude, but instead is configured to drive the vehicle to a desired bank rate. This allows the vehicle to complete the maneuver at higher rates than normally obtained with the attitude control loop providing the feedback.

Finally, when the Mach number drops below $\mathcal{M}_1$, the rudder is switched in to augment the aileron effectiveness. The aileron to rudder crossfeed gain is a constant.

5.3.2.1.2 Late Aileron/Rudder Control

Figure 5-11 illustrates the "proverse" control submode. Closed-loop control of bank and sideslip is provided by position ($\beta$, $\phi$) and rate ($\dot{\beta}$, $\dot{\phi}$) feedback of the lateral state variables, through a network which attempts to maintain a constant response throughout the remainder of the trajectory. Specifically, the desired closed-loop transfer matrix describing bank and sideslip attitude response to a guidance commanded step is chosen to be:

$$\begin{bmatrix} (s^2 + a_{11}s + b_{11}) & (a_{12}s + b_{12}) \\ (a_{21}s + b_{21}) & (s^2 + a_{22}s + b_{22}) \end{bmatrix} \begin{bmatrix} \beta \\ \phi - \phi_G \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \qquad (5\text{-}6)$$

where the $a_{ij}$ and $b_{ij}$ are chosen for acceptable transient response. It may be noted that in this "proverse" submode, the availability of the rudder allows for considerably more freedom in shaping the transient response, so that both sideslip and bank characteristics may be specified, in contradistinction to the previous submode (see Eq (5-4)).

As shown in the block diagram, there are eight gains ($r_{ij}$ and $p_{ij}$) which are derived functions of both the desired closed-loop parameters (in Eq (5-6)) and of four selected scheduled gains. They are given by:

$$\begin{aligned} r_{11} &= a_{11}\gamma_{11} + a_{21}G_1\gamma_{12} & p_{11} &= b_{11}\gamma_{11} + b_{21}G_1\gamma_{12} \\ r_{12} &= a_{12}\gamma_{11} + a_{22}G_1\gamma_{12} & p_{12} &= b_{12}\gamma_{11} + b_{22}G_1\gamma_{12} \\ r_{21} &= a_{11}\gamma_{21} + a_{21}G_1\gamma_{22} & p_{21} &= b_{11}\gamma_{21} + b_{21}G_1\gamma_{22} \\ r_{22} &= a_{12}\gamma_{21} + a_{22}G_1\gamma_{22} & p_{22} &= b_{12}\gamma_{21} + b_{22}G_1\gamma_{22} \end{aligned} \qquad (5\text{-}7a)$$

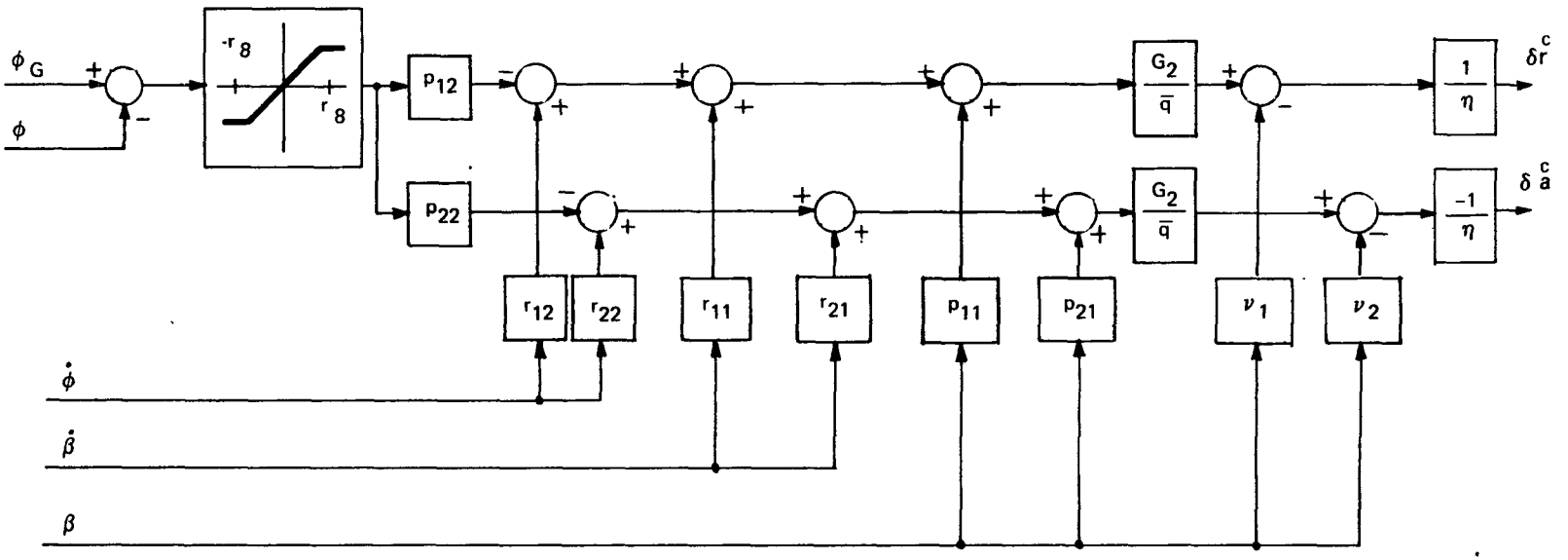
where $G_1$ is a constant and the $\gamma_{ij}$ are scheduled gains. Also shown in the block diagram are a constant gain, $G_1$, two additional scheduled gains $\nu_1$ and $\nu_2$, and one additional derived gain, a function of the $\gamma_{ij}$:

$$\eta = \gamma_{11}\gamma_{22} - \gamma_{12}\gamma_{21} \qquad (5\text{-}7b)$$

5.3.2.2 Yaw and Roll ACPS Control

Yaw and roll jet activity is completely inhibited while in the "proverse" aileron/rudder control submode ($I_{LAT} = 1$). This section describes the lateral ACPS control logic while in the "reverse" surface control submode.

Yaw jet ACPS control synthesis is based on double integrator modeling of the bank angle dynamics, so that the switch logic presented for pitch ACPS control may




NON-LINEARITY:
Limiter ; Unity Slope.

Fig. 5-11 Aileron/Rudder Control Block Diagram (Proverse Mode, $I_{Lat} = 1$)

be used. Roll jet logic consists of three models, one based on double integrator modeling of the sideslip dynamics, a second based on oscillator modeling, and a third which accounts for control axis cross-coupling.

5.3.2.2.1 Yaw Jet Control

A double integrator model of the bank angle dynamics, combined with the assumption of turn coordination provided by the roll jets, allows for the use of the ACPS pitch jet control logic, with the appropriate redefinition of control parameters. Specifically, yaw jet control is defined by the phase-plane switch logic of Fig. 5-4 and Eq (5-3a), where the coordinates are given by:

$$(\epsilon_1, \epsilon_2) = \left[ -\phi_{lim} \sin \alpha_T, r \right] \tag{5-8a}$$

and the switch curve parameters are given by:

$$(u, U) = (u_z, U_z) \tag{5-8b}$$

$$(\delta_1, \delta_2) = (\delta_{\phi 1}, \delta_{\phi 2}) \tag{5-8c}$$

$$\sigma = \sigma_\phi \tag{5-8d}$$

The blending interface logic is discussed below.

5.3.2.2.2 Roll Jet Control

Roll jet control of vehicle sideslip is accomplished by the use of three control modes. Two of these use a double integrator model for sideslip; the remaining one uses an oscillator model.

The purpose of the first mode is to provide turn coordination when the yaw jets are firing ($u_z \neq 0$). This is accomplished by the phase-plane logic of Fig. 5-12, which is a generalized ($\beta$, $\dot{\beta}$) plane. The figure illustrates a typical state trajectory and the resultant limit cycle (with switch curve overshoots). The equations for the switch curve $\mu$ are:

$$\begin{aligned} \mu: \; S\beta &= \frac{1}{2U_1} \left[ (\dot{\beta}^m)^2 - \dot{\beta}^2 \right] \text{ if } S\dot{\beta} > \dot{\beta}^m \\ &= -\frac{1}{2U_2} \left[ (\dot{\beta}^m)^2 - \dot{\beta}^2 \right] \text{ if } S\dot{\beta} \leq \dot{\beta}^m \end{aligned} \tag{5-9a}$$

where

$$
\begin{aligned}
U_1 &= U_z \cos \alpha_T \\
U_2 &= U_x \sin \alpha_T - U_z \cos \alpha_T \\
S &= \operatorname{sgn}(u_z) \\
\dot{\beta}^m &= \sqrt{2U_2\delta_1}
\end{aligned}
\tag{5-9b}
$$

and where $\delta_1$ is defined below in Eq (5-10).

The purpose of the second mode is to provide direct control over sideslip when the yaw jets are inactive ($u_z = 0$) and when the dynamic pressure is low. With double integrator modeling, the phase-plane logic of Fig. 5-4 and Eq (5-3a) may be used, where the coordinates are given by:

$$(\epsilon_1, \epsilon_2) = (\beta, \dot{\beta}) \tag{5-10a}$$

and the switch curve parameters are given by:

$$(u, U) = (u_x, U_x \sin \alpha_T) \tag{5-10b}$$

$$(\delta_1, \delta_2) = (\delta_{\beta 1}, \delta_{\beta 2}) \tag{5-10c}$$

$$\sigma = \sigma_\beta \tag{5-10d}$$

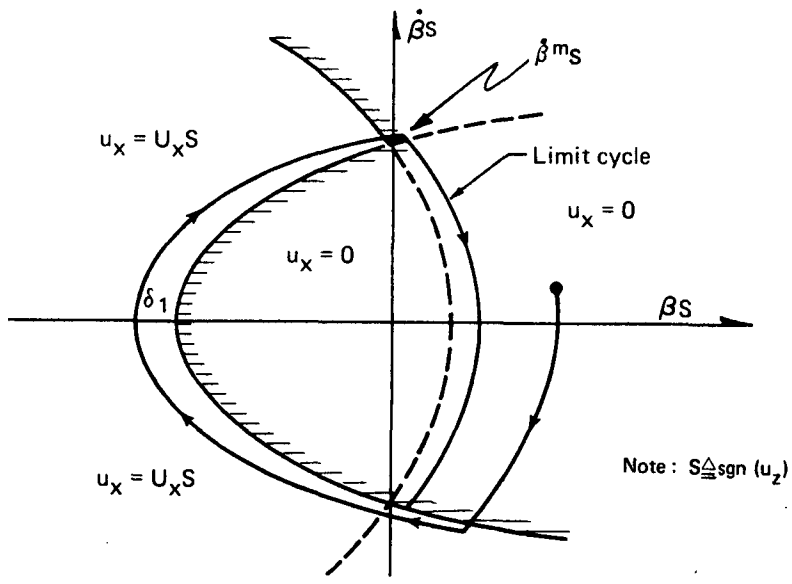



Fig. 5-12 Roll Jet Control when $u_z \neq 0$

It is convenient to label this control mode (used at low $\bar{q}$ and when $u_z = 0$) as attitude hold, since both rate and attitude error are driven toward zero.

The third control mode takes advantage of the vehicle's dynamic stability in sideslip, which becomes significant once the dynamic pressure becomes sufficiently high. What is required is simply rate damping of the sideslip angular velocity as shown below, where $\dot{\beta}^m$ is given by Eq (5-9b).

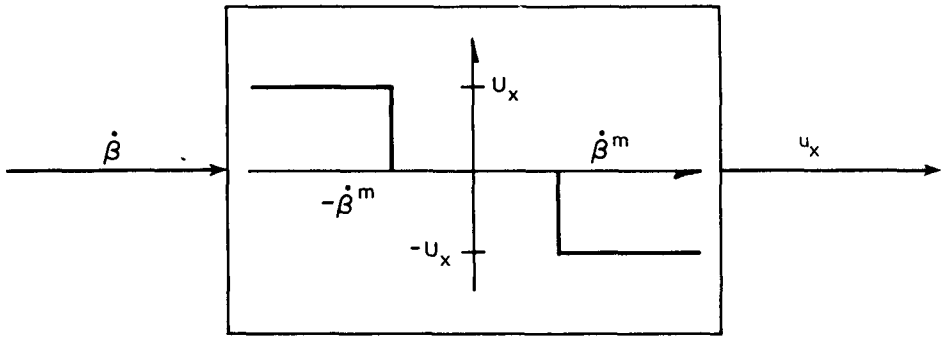



Fig. 5-13 Roll Jet Control when $u_z = 0$ ($\bar{q} \geq \bar{q}_3$)

The control modes described above, defining lateral control with yaw and roll jets, are summarized in block diagram form in Fig. 5-14. Several points should be noted. First, the mode select block which defines the roll jet control mode to be used is defined by:

$$\begin{array}{llll} \text{If} & u_z \neq 0 & \text{then Mode} = & \text{I} \\ & u_z = 0,\ \bar{q} < \bar{q}_3 & & \text{II} \\ & u_z = 0\ \ \bar{q} \geq \bar{q}_3 & & \text{III} \end{array} \qquad (5\text{-}11)$$

where the modes are coordination, attitude hold, and rate damping, respectively. The three roll control blocks correspond with the above modes. A second point to note is the gating by the blending parameter $\lambda_a$, which is defined by logic that is identical to that of the pitch channel. Finally, it may be noted that the logic disallows ACPS inhibition during aileron trim-only (null aileron command) operation.

5.3.2.3 Digital Implementation

Coding implementation for lateral control during the entry phase is summarized in flow-chart form in Fig. 5-15.

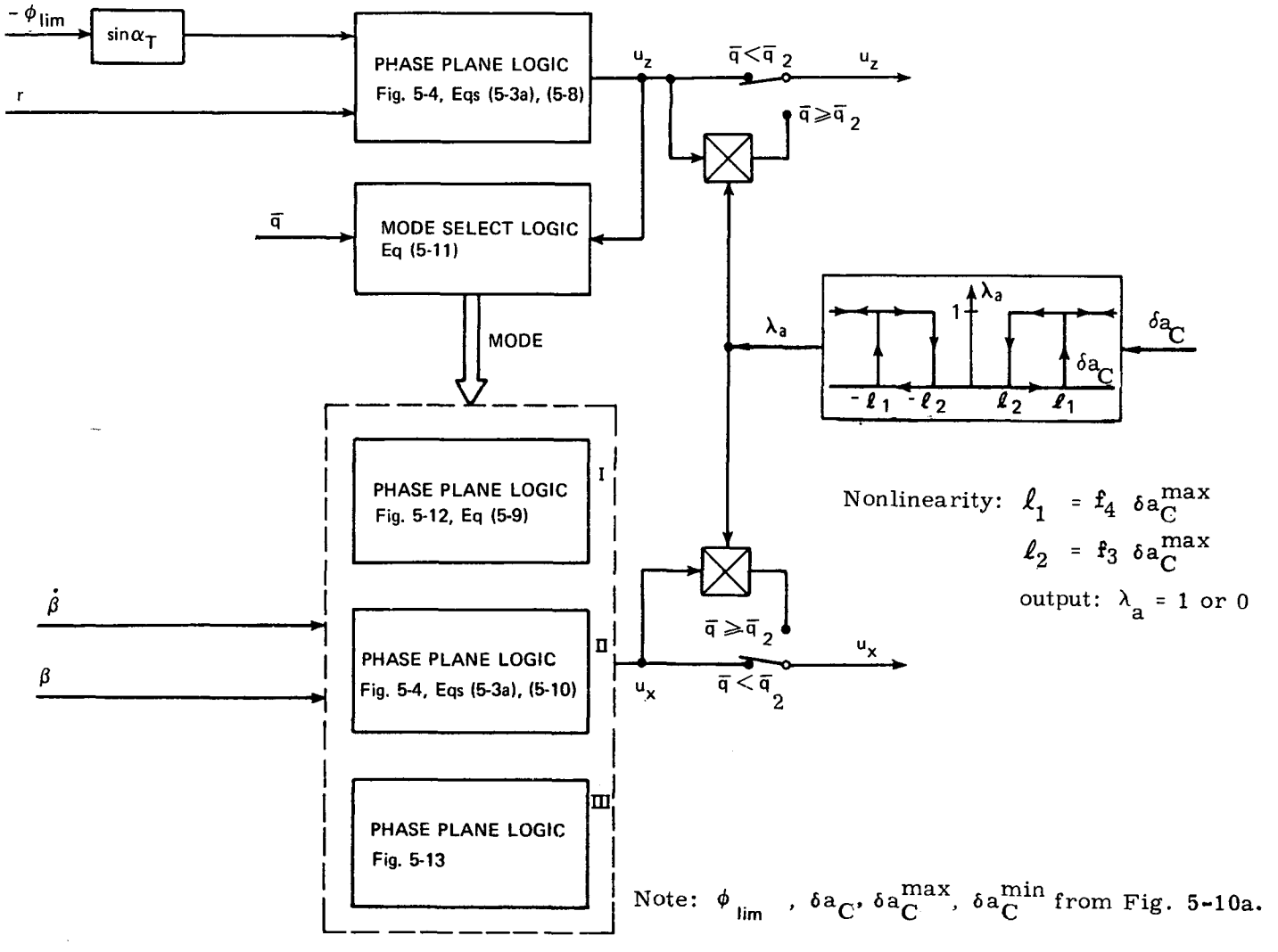



Fig. 5-14 Yaw and Roll ACPS Control Block Diagram

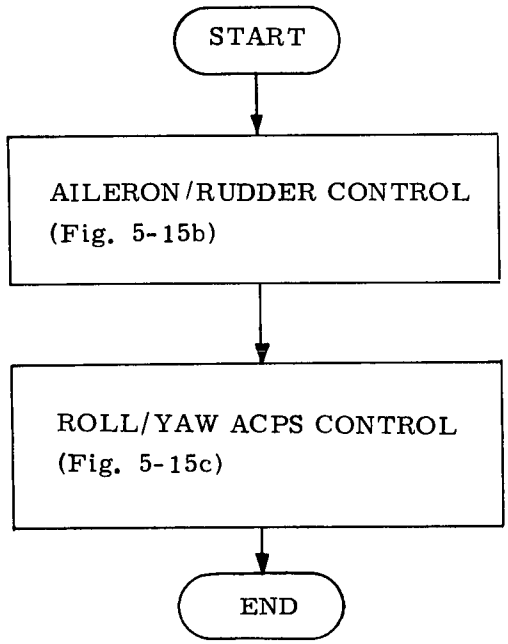



Fig. 5-15a Logic Flow for the Lateral Channel

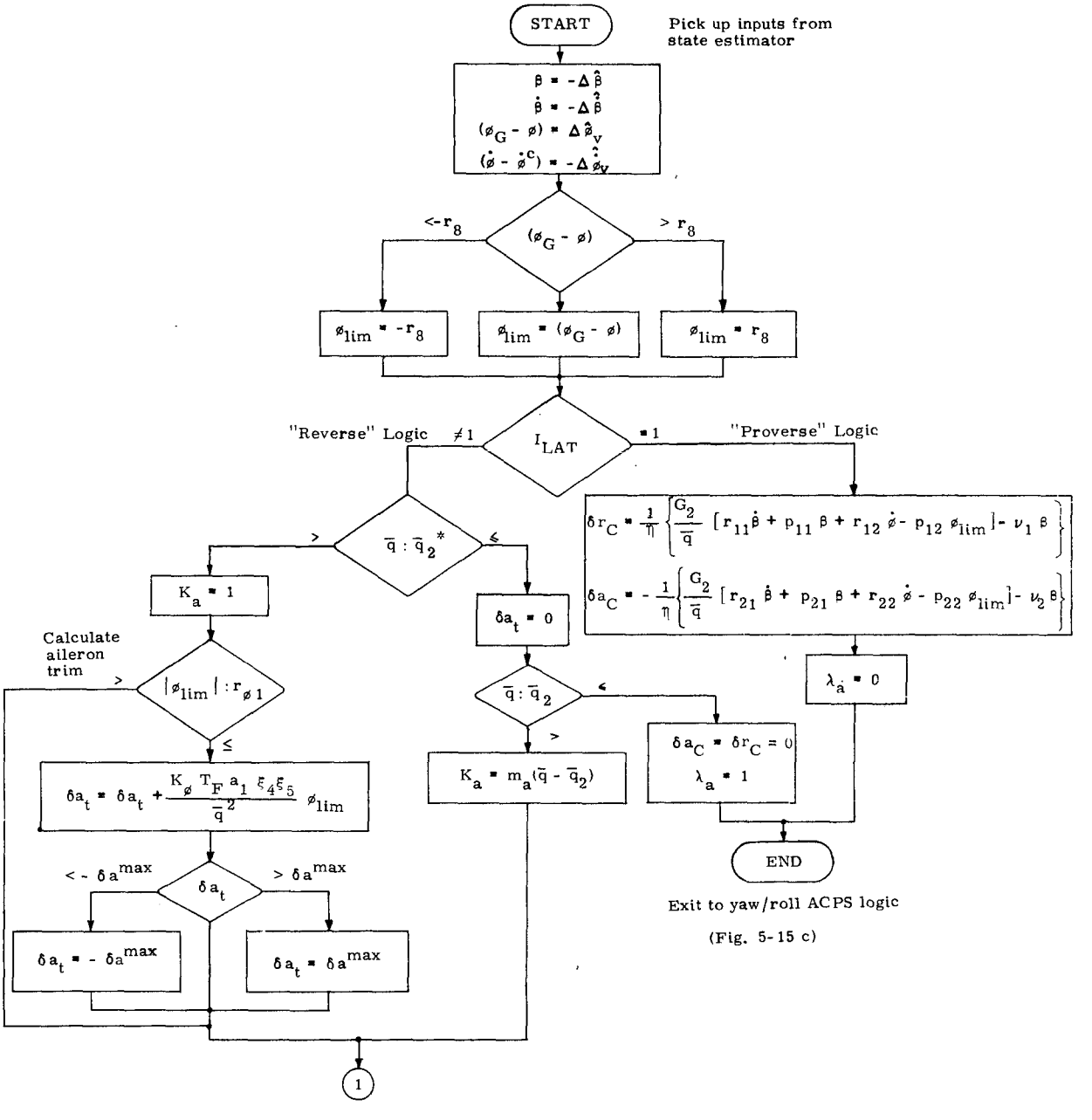



Fig. 5-15b Logic Flow for Aileron/Rudder Control

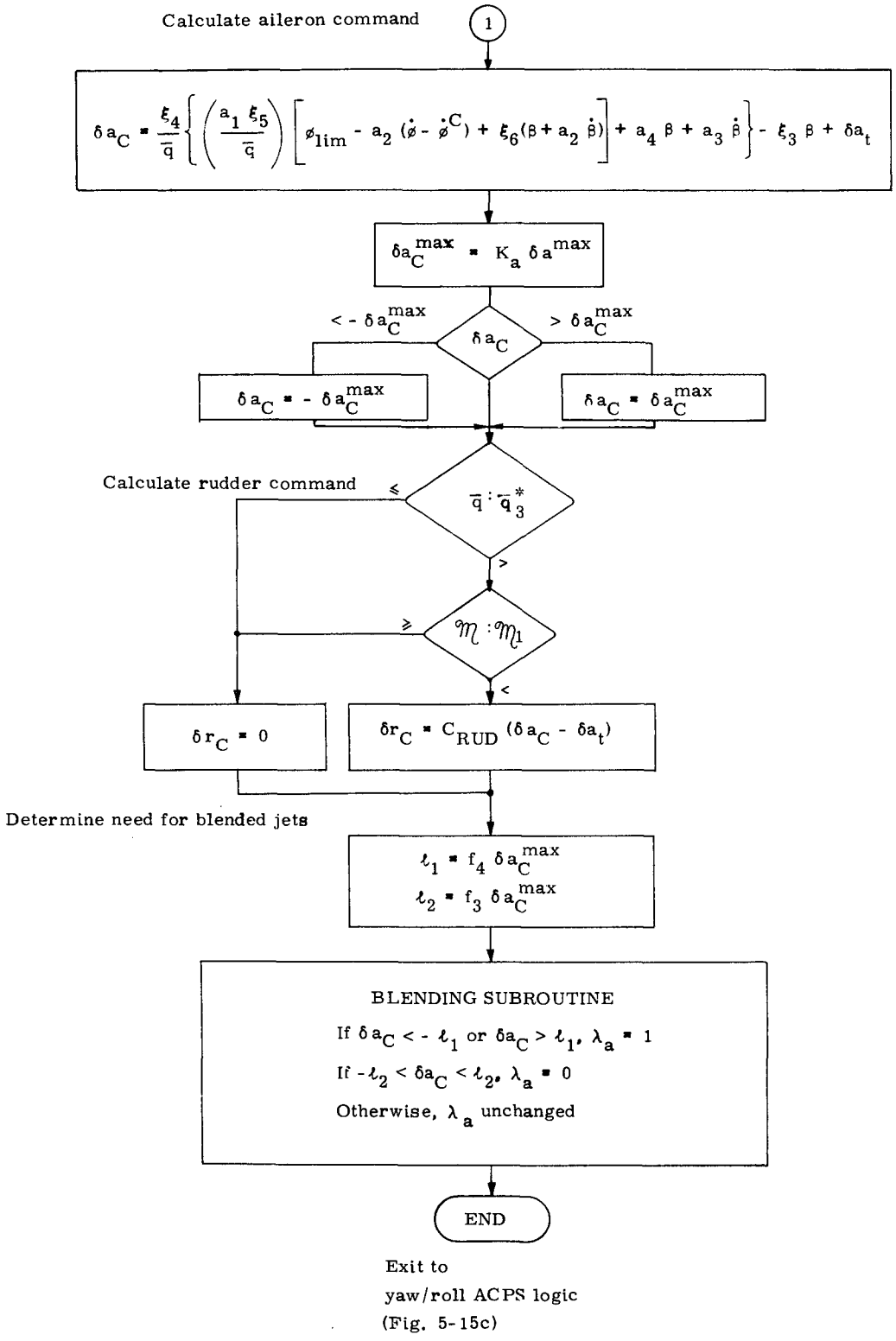



Fig. 5-15b Logic Flow for Aileron/Rudder Control (Cont)

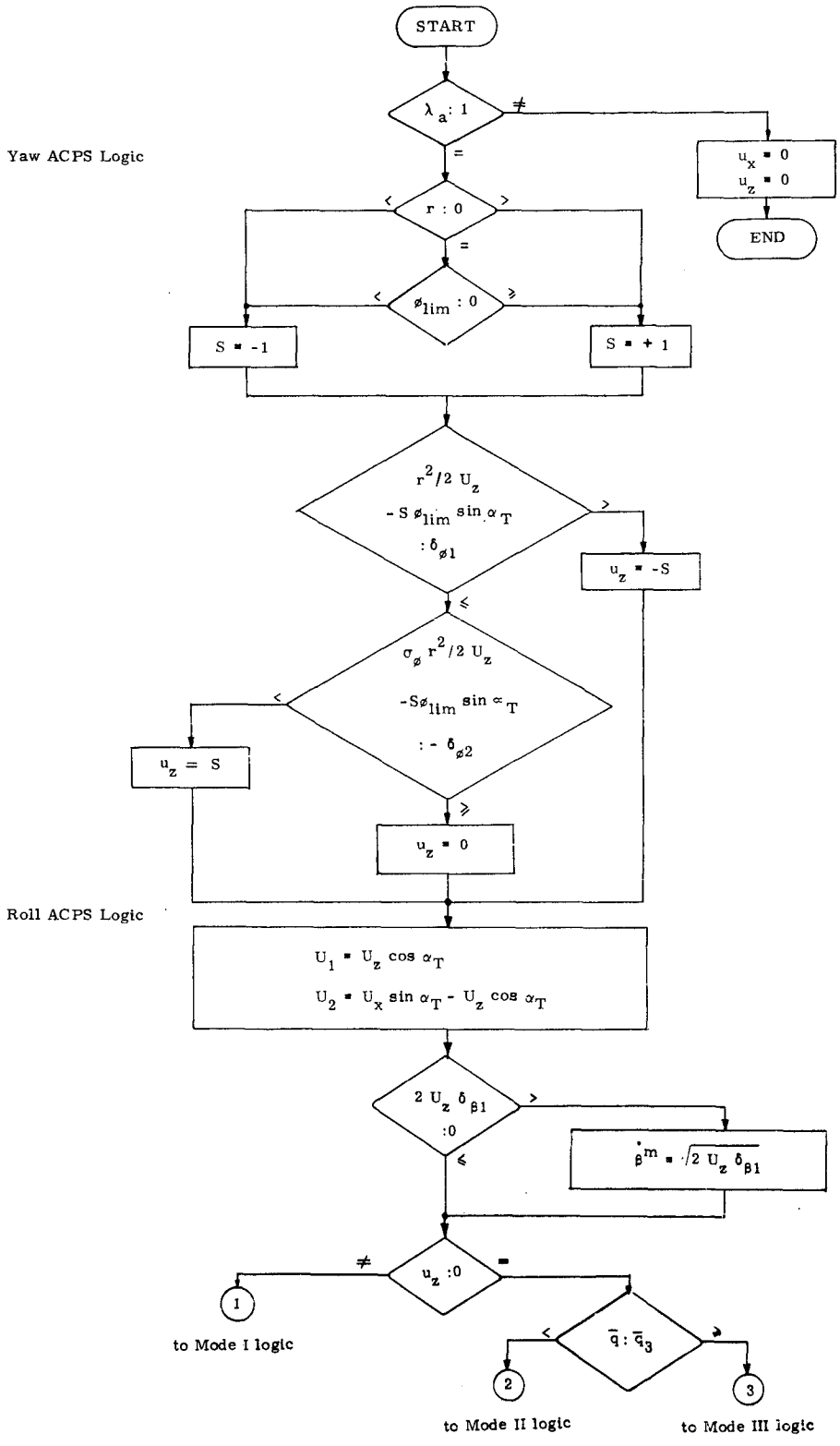



Fig. 5-15c Logic Flow for Yaw/Roll ACPS Control

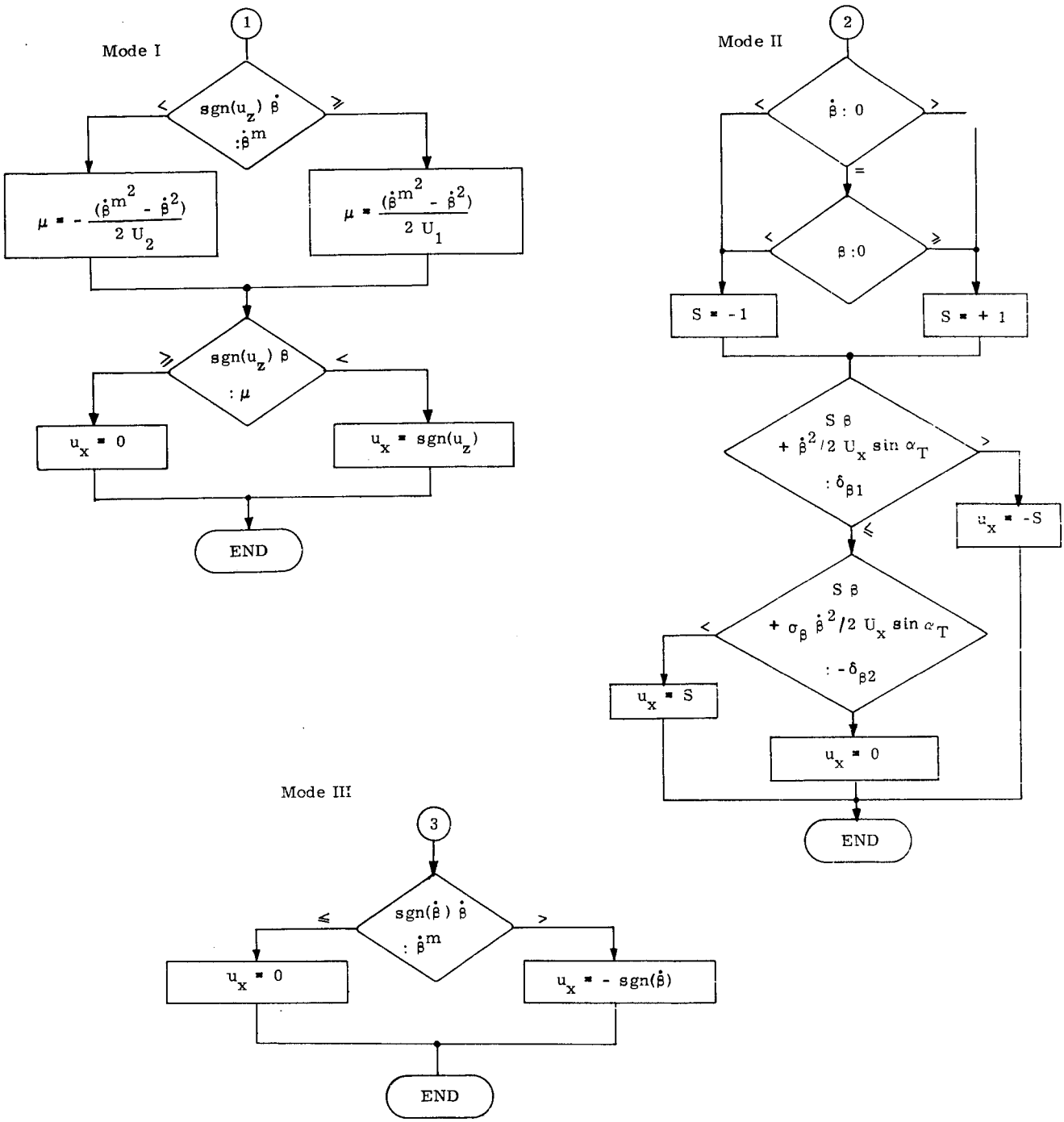



Fig. 5-15c Logic Flow for Yaw/Roll ACPS Control (Cont)

### 5.3.3 DFCS Parameter Specification

All parameters used in control computations for the entry phase are either fixed or are scheduled on the basis of Mach number, $\mathcal{M}$. Table 5-1 tabulates all entry-phase fixed control parameters. The scheduled control gains $\xi_1$ through $\xi_6$ and $\nu_1$ and $\nu_2$ are plotted in Figs. 5-16 through 5-23. The scheduled control parameters $\gamma_{11}$ through $\gamma_{22}$ are plotted in Figs. 5-24 through 5-27. As noted in Subsection 5.3.2.1.1, the lateral gains $r_{ij}$, $p_{ij}$, and $\eta$ are calculated as functions of the $\gamma_{ij}$, according to Eq (5-7).

In an initialization pass through the entry control logic, several parameters which are functions of pad-loaded values are calculated and stored for future use. The Load Manipulation Routine for the entry phase is shown in Fig. 5-28.

## 5.4 Performance Summary

Shown in Fig. 5-29 are three curves in the Mach-alpha plane which define regions of lateral controllability using differing control modes. Also shown is a typical late entry Mach-alpha profile. The curves were derived by conducting simulated maneuvers with the entry DFCS and the NR 00089B (Feb. version) vehicle at various Mach numbers and angles-of-attack. The curves separate desirable from undesirable simulated vehicle response.

Curve A indicates the lower boundary (in the Mach-alpha plane) to which the vehicle may be flown using ailerons alone for lateral control. By driving the rudder proportionally to the aileron command (with a crossfeed gain of 2), the region of "reverse" control may be considerably extended farther down the entry profile (curve B). Finally, the region of controllability using the "proverse" control mode is bounded (above) by curve C. The resultant overlapping region, defined by curves B and C, provides the motivation for the switch curve choice shown in Fig. 5-9.

It should be noted that an on-going effort in determining design sensitivity to aerodynamic coefficient uncertainties will reduce the area of the overlap region. However, the control design used in deriving the curves of Fig. 5-29 was not "tuned" so as to maximize the resultant overlap area. It is anticipated that future efforts in this direction will enlarge this region for the nominal aerodynamic coefficient values, while counteracting the shrinking effect of off-nominal aerodynamics.

Table 5-1 Fixed Parameters

| Symbol | Value | Dimension |
|---|---|---|
| $U_x$ | 1.65 | $deg/sec^2$ |
| $U_y$ | 1.32 | $deg/sec^2$ |
| $U_z$ | 0.66 | $deg/sec^2$ |
| $\delta_{\alpha 1}$ | 0.75 | deg |
| $\delta_{\alpha 2}$ | 1.0 | deg |
| $\delta_{\beta 1}$ | 0.2 | deg |
| $\delta_{\beta 2}$ | 0.4 | deg |
| $\delta_{\phi 1}$ | 0.75 | deg |
| $\delta_{\phi 2}$ | 1.5 | deg |
| $\sigma_\alpha$ | 5 | - |
| $\sigma_\beta$ | 2 | - |
| $\sigma_\phi$ | 5 | - |
| $f_1$ | 0.8 | - |
| $f_2$ | 0.6 | - |
| $f_3$ | 0.6 | - |
| $f_4$ | 0.8 | - |
| $\bar{q}_1$ | 10 | $lbf/ft^2$ |
| $\bar{q}_2$ | 20 | $lbf/ft^2$ |
| $\bar{q}_3$ | 5 | $lbf/ft^2$ |
| $\bar{q}_1^*$ | 20 | $lbf/ft^2$ |
| $\bar{q}_2^*$ | 30 | $lbf/ft^2$ |
| $\bar{q}_3^*$ | 80 | $lbf/ft^2$ |

Table 5-1 Fixed Parameters (Cont)

| Symbol | Value | Dimension |
|---|---|---|
| $\mathcal{M}_1$ | 6 | - |
| $K_{\alpha 1}$ | -1.0 | $sec^{-1}$ |
| $K_{\alpha 2}$ | 0.2 | $sec^{-1}$ |
| $K_{\phi}$ | 0.01 | $sec^{-1}$ |
| $\omega_d$ | 0.7 | $sec^{-1}$ |
| $\omega_1$ | 0.7 | $sec^{-1}$ |
| $\omega_2$ | 0.7 | $sec^{-1}$ |
| $\zeta_d$ | 0.7 | - |
| $\zeta_1$ | 1.4 | - |
| $\zeta_2$ | 0.7 | - |
| $r_1$ | 5 | deg |
| $r_8$ | 25 | deg |
| $r_{\phi 1}$ | 5 | deg |
| $r_{\phi 2}$ | 25 | deg |
| $r_{\phi 3}$ | 8 | deg/sec |
| $C_{RUD}$ | 2 | - |
| $a_{11}$ | 1.0 | $sec^{-1}$ |
| $a_{12}$ | 0.0 | $sec^{-1}$ |
| $a_{21}$ | 0.0 | $sec^{-1}$ |
| $a_{22}$ | 1.3 | $sec^{-1}$ |
| $b_{11}$ | 1.0 | $sec^{-2}$ |
| $b_{12}$ | 0.0 | $sec^{-2}$ |

Table 5-1 Fixed Parameters (Cont)

| Symbol | Value | Dimension |
|---|---|---|
| $b_{21}$ | 1.0 | $sec^{-2}$ |
| $b_{22}$ | 0.8 | $sec^{-2}$ |
| $G_1$ | 0.133 | - |
| $G_2$ | 29.0 | $lbf\text{-}sec^2/ft^2$ |
| $\delta e^{min}$, $\delta e^{max}$, $\delta a^{max}$, $\delta r^{max}$ | see Subsection 2-2 | |

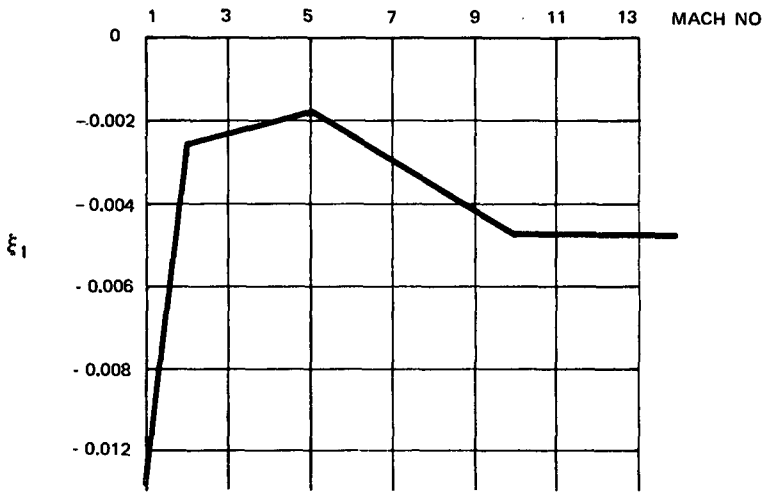



Fig. 5-16 Gain Schedule for $\xi_1$

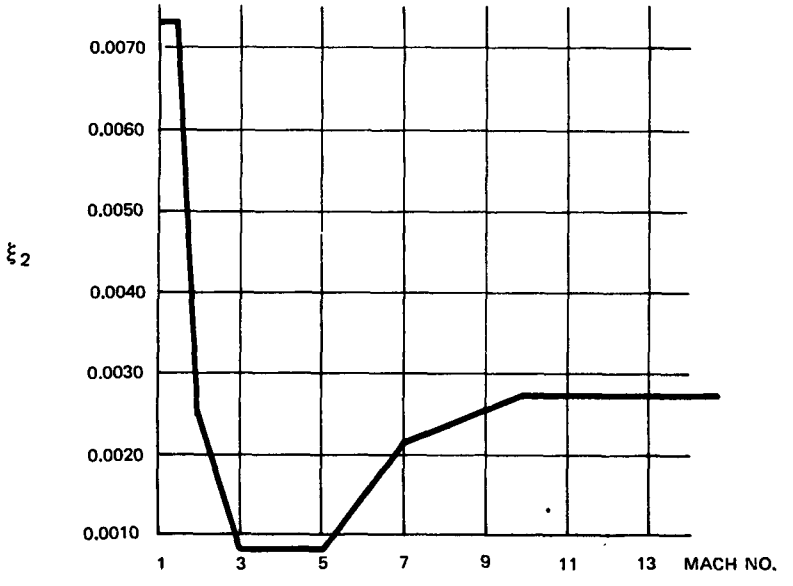



Fig. 5-17 Gain Schedule for $\xi_2$

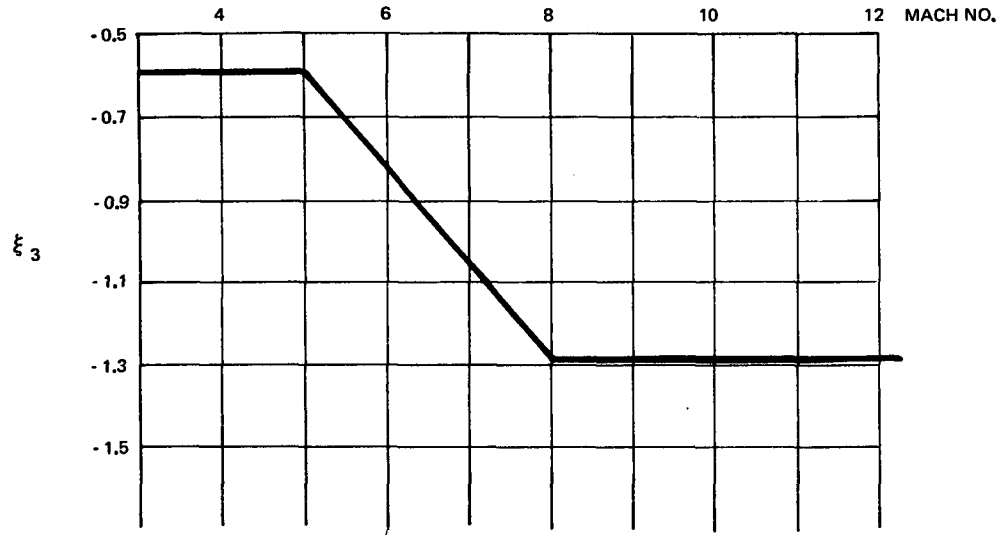



Fig. 5-18 Gain Schedule for $\xi_3$

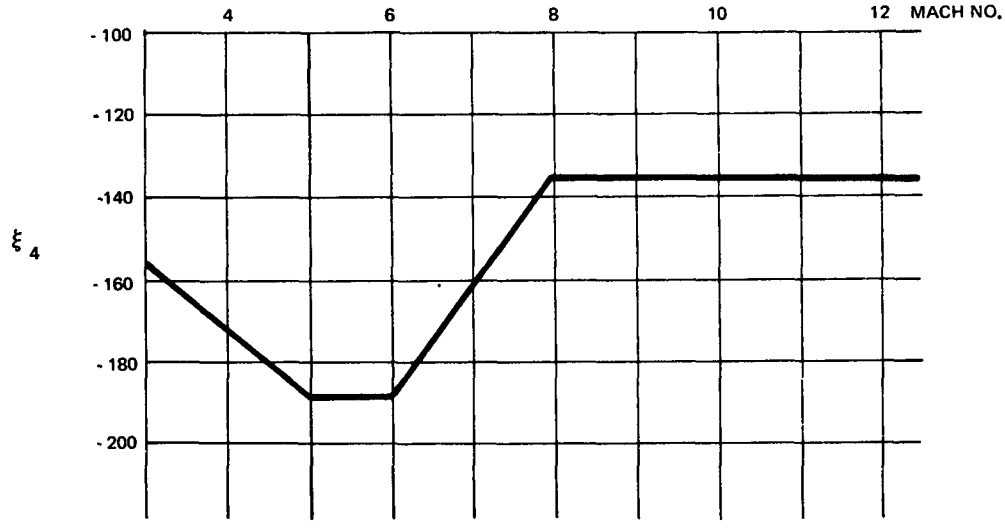



Fig. 5-19 Gain Schedule for $\xi_4$

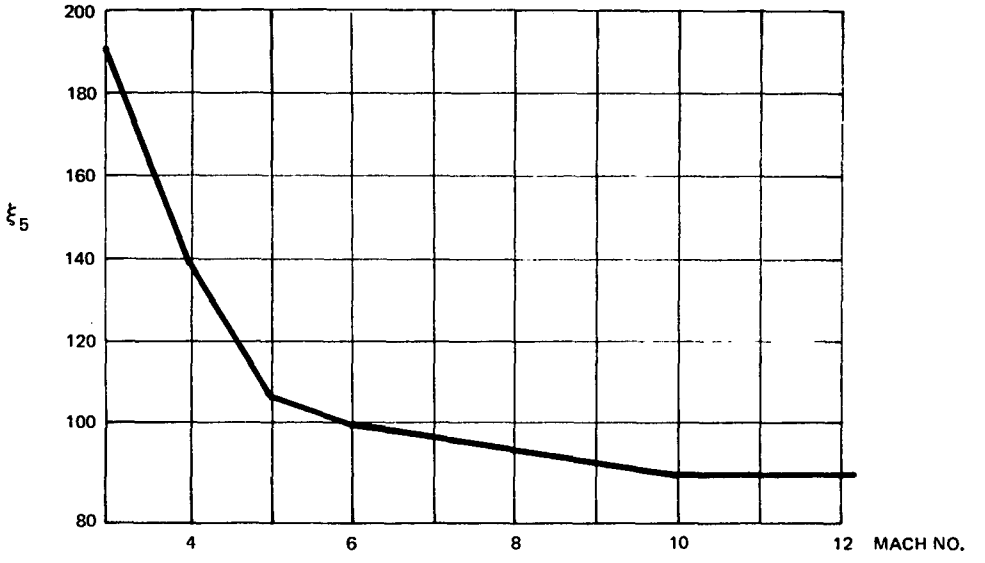



Fig. 5-20 Gain Schedule for $\xi_5$

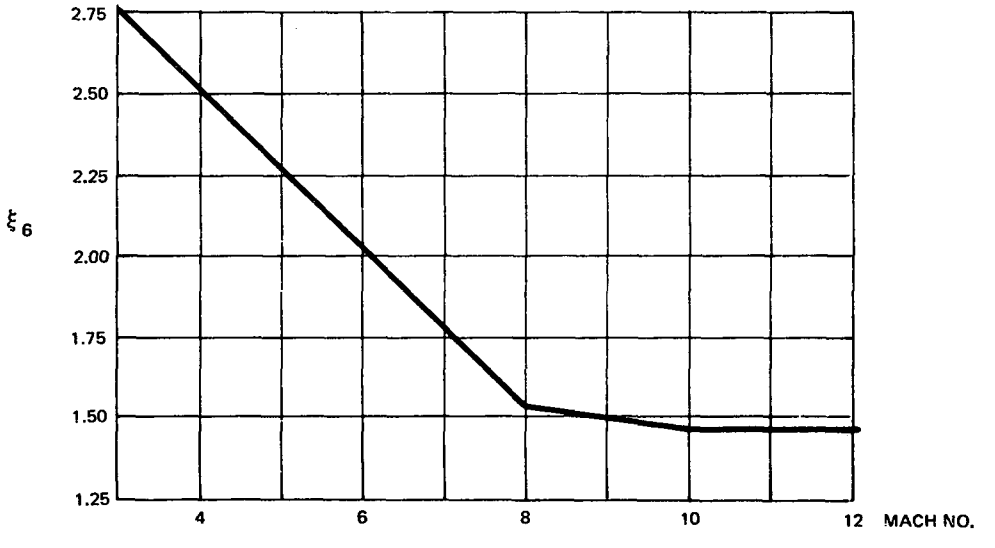



Fig. 5-21 Gain Schedule for $\xi_6$

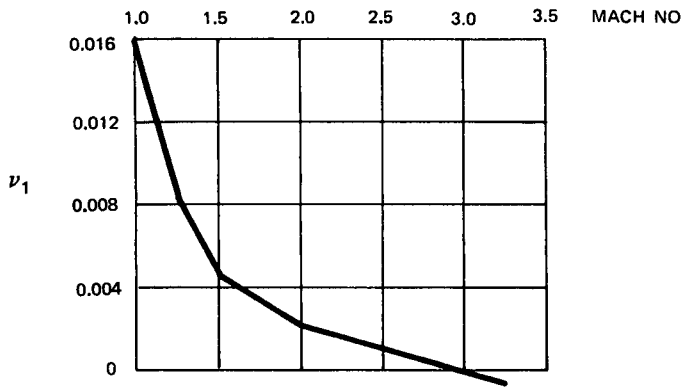



Fig. 5-22 Gain Schedule for $\nu_1$

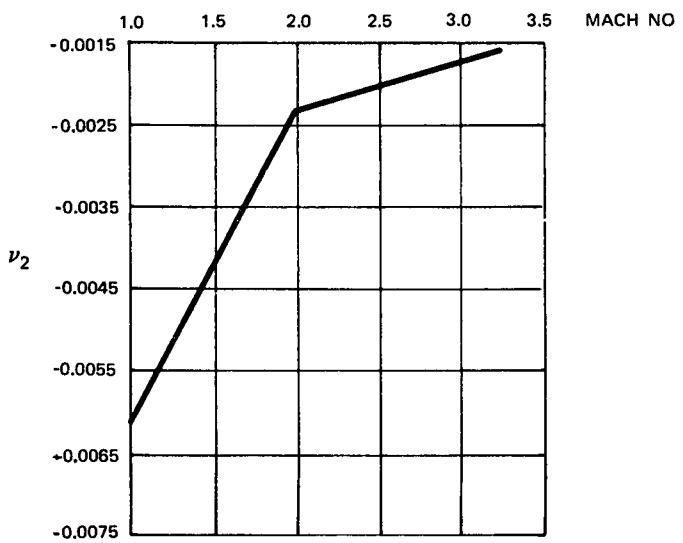



Fig. 5-23 Gain Schedule for $\nu_2$

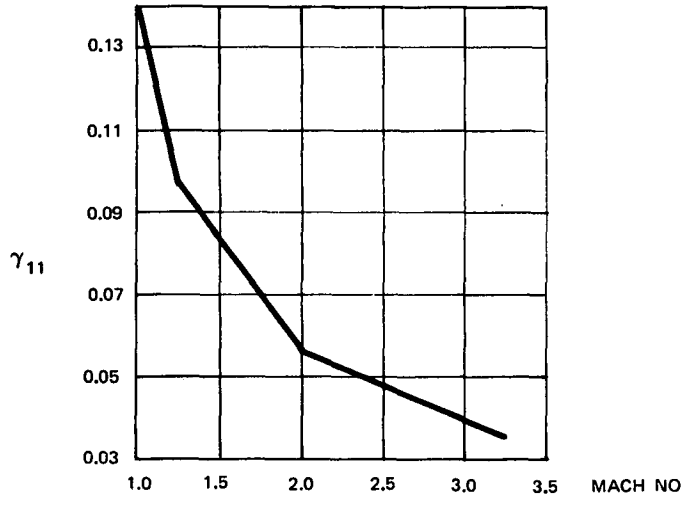



Fig. 5-24 Gain Schedule for $\gamma_{11}$

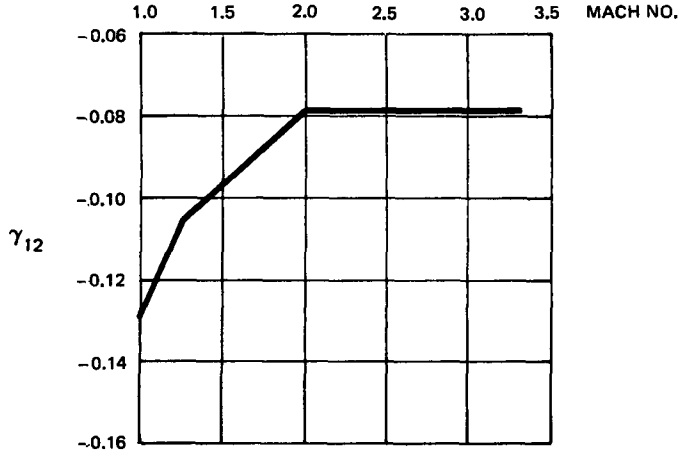



Fig. 5-25 Gain Schedule for $\gamma_{12}$

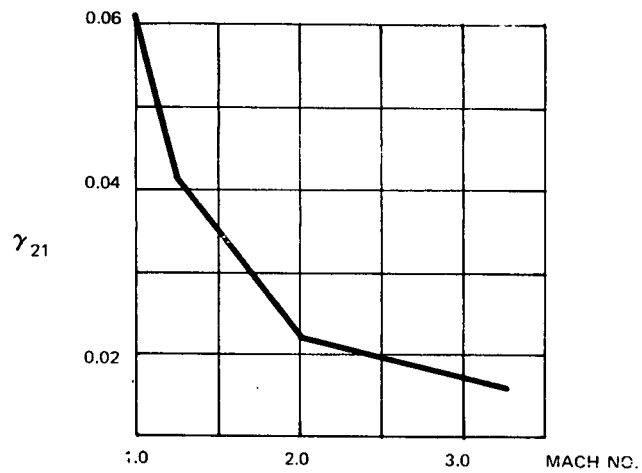



Fig. 5-26 Gain Schedule for $\gamma_{21}$

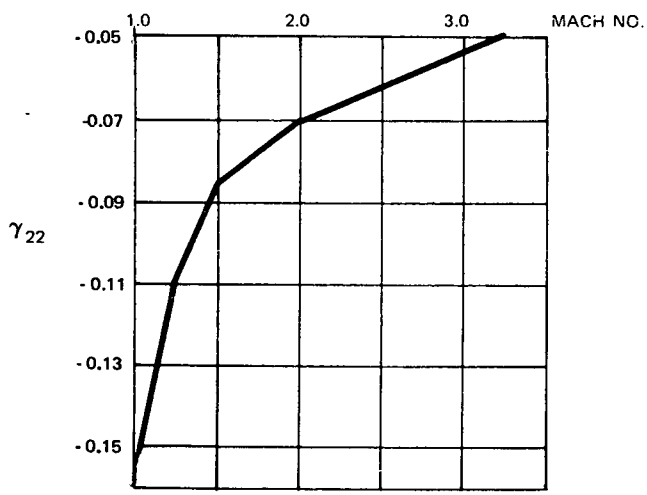



Fig. 5-27 Gain Schedule for $\gamma_{22}$

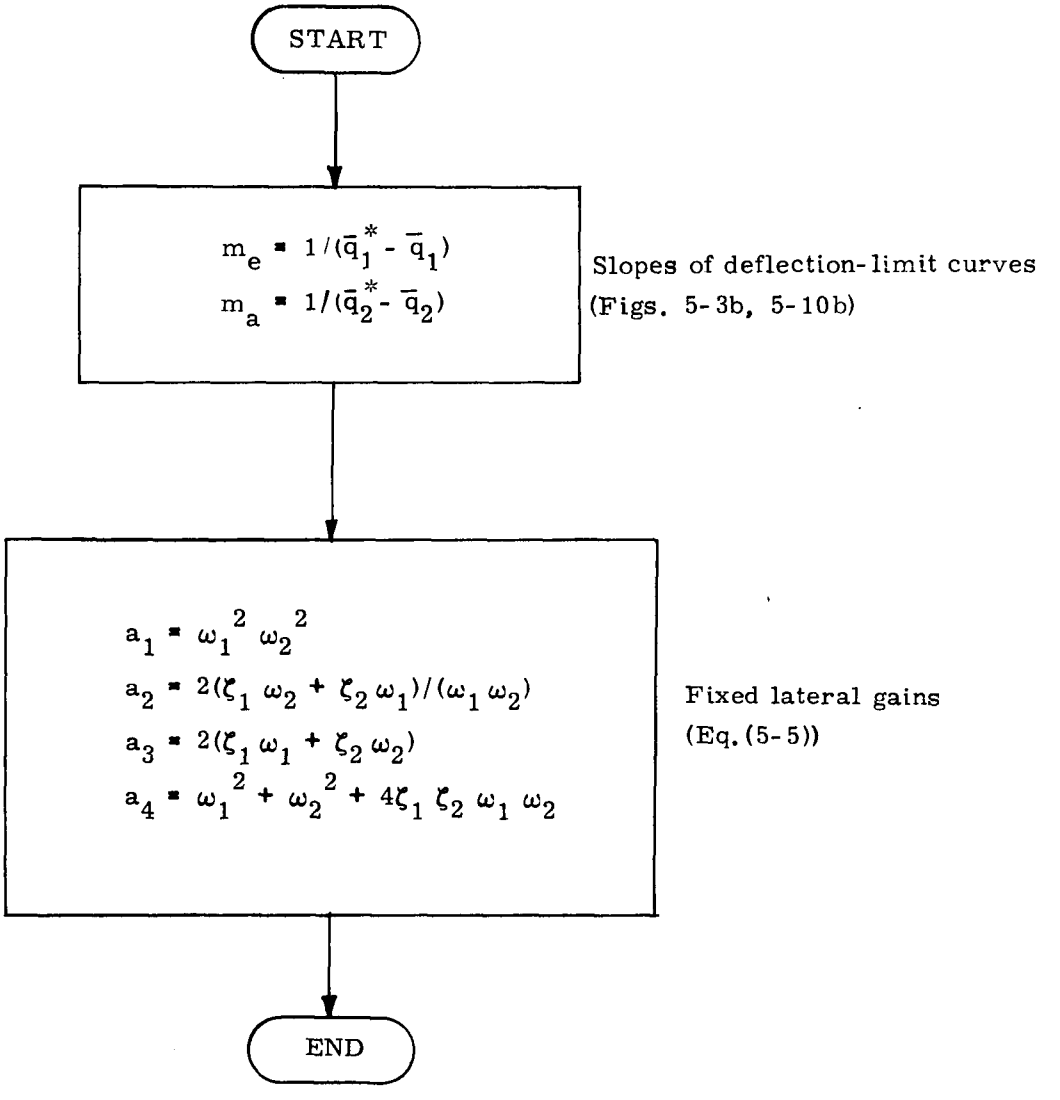



Fig. 5-28 Entry Load Manipulation

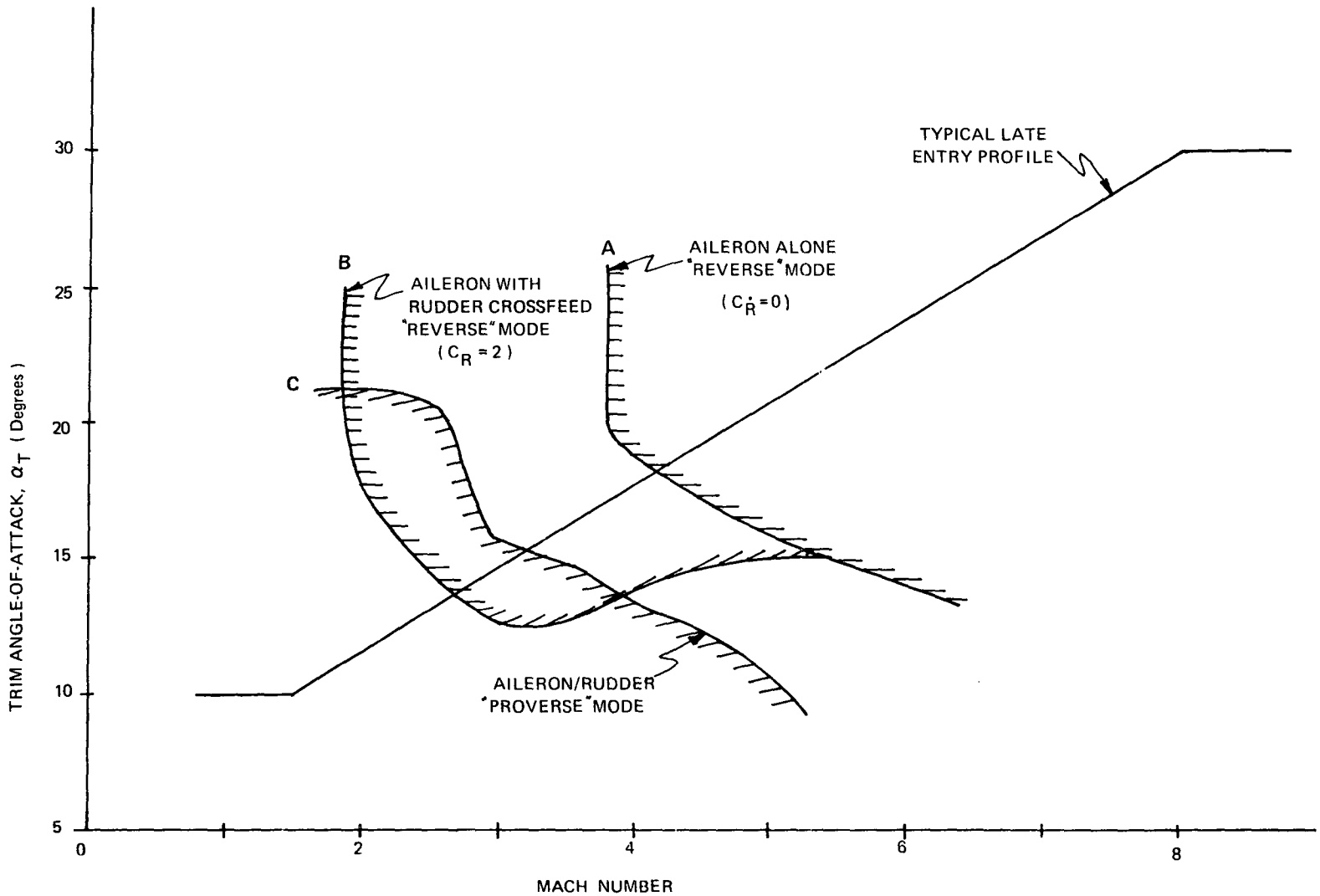



Fig. 5-29 Controllability Regimes

SECTION 6

# ON-ORBIT PHASE

by

Steven R. Croopnick, Donald W. Keene and John J. Turkovich

## 6.1 Introduction

The DFCS (Digital Flight Control System) will control the attitude and the translation of the Space Shuttle during the various orbital activities by using either the ACPS (Attitude Control Propulsion System) or the OMS (Orbital Maneuvering System). For the purpose of control, there are two orbital phases: the ACPS phase when the jets are the control effectors, and the TVC (Thrust Vector Control) phase when the gimbaled engines are used for control. The ACPS phase is presented first, in Subsection 6.2, followed by a preliminary description of the on-orbit TVC phase in Subsection 6.3. Digital implementation of the ACPS phase, presented in Subsection 6.4, and preliminary testing was accomplished using structured programming techniques in the HAL language. All symbols relevant to the on-orbit phase may be found in a separate glossary at the end of this section.

## 6.2 On-orbit ACPS Phase: Engineering Description

A block diagram of the DFCS during the on-orbit ACPS phase is illustrated in Fig. 6-1. The on-orbit attitude maneuver routine or another interface routine (see Part II) supplies to the DFCS:

1) a set of reference gimbal angles, $[\theta_d]$
2) a set of incremental reference gimbal angles, $[\Delta\theta_d]$
3) the desired angular velocity vector, $\underline{\omega}_d$
4) a set of bias gimbal angles, $[\beta_{BIAS}]$
5) a bias acceleration vector, $\underline{\alpha}_m$

Every $t_{fast}$ seconds the DFCS adds $[\Delta\theta_d]$ to $[\theta_d]$ to give a new set of reference gimbal angles. The DFCS then compares these to the measured set of gimbal angles, $[\theta_{GA}]$, and computes the angular errors in control axes. The bias angles are added to these errors in control axes and fed to the phase plane switching logic. Also fed to the phase plane switching logic are the angular velocity error, $\underline{\omega}_e$, the net estimated disturbance acceleration, $\underline{\alpha}_{dm}$, the magnitudes of control accelerations, $[\alpha_C]$, and

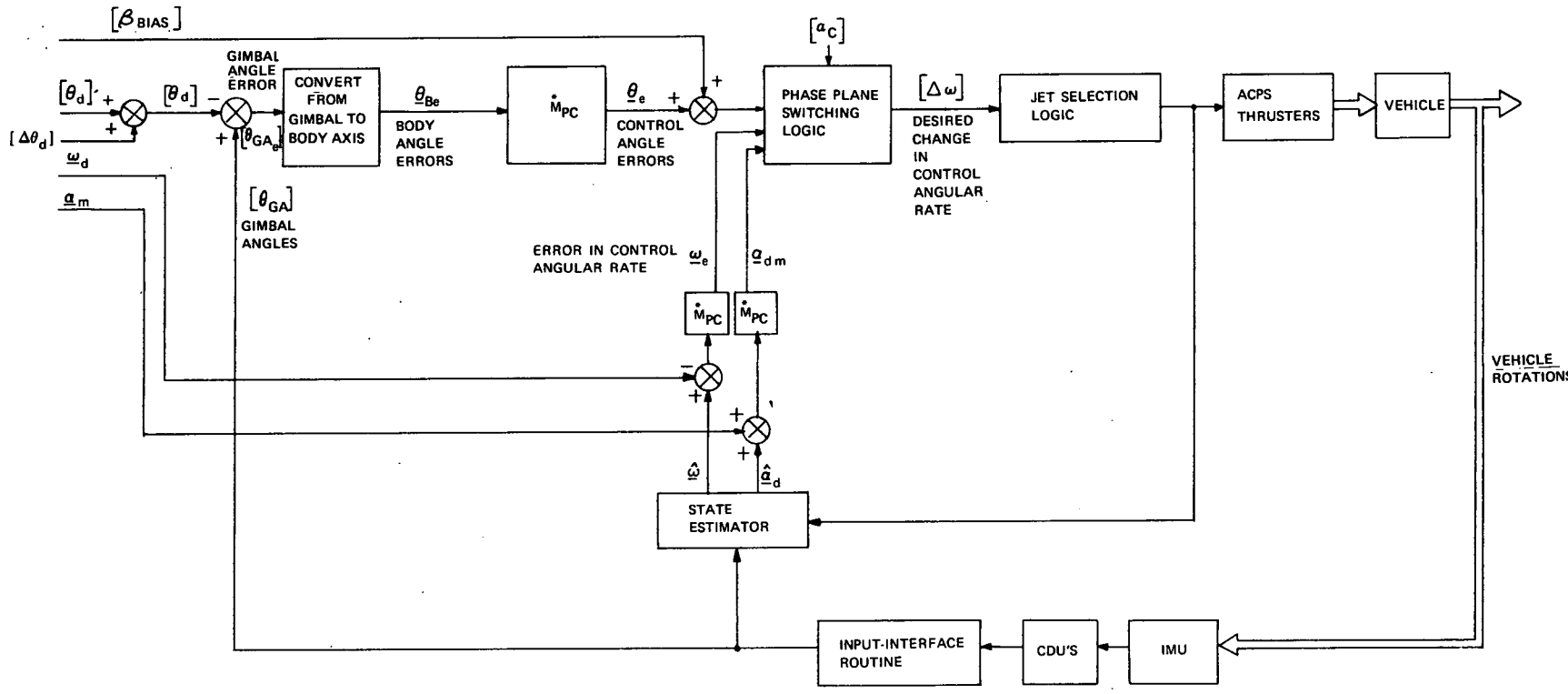



Fig. 6-1 On-Orbit ACPS

the last desired change in angular velocity. The angular velocity error, $\underline{\omega}_e$, is computed by subtracting the desired angular velocity, $\underline{\omega}_d$, from the estimated angular velocity, $\hat{\underline{\omega}}$, and transforming the result into control coordinates. The net estimated disturbance acceleration, $\underline{\alpha}_{dm}$, is computed by adding the bias acceleration, $\underline{\alpha}_m$, to the estimated disturbance acceleration, $\hat{\underline{\alpha}}_d$, and transforming the result into control coordinates. From these inputs the phase plane switching logic determines the desired angular rate changes, $[\Delta\omega]$, in control coordinates. From these desired rate changes the jet selection logic determines the appropriate jets to be fired, if any, and the corresponding firing times, and then turns the jets on and schedules their turn-off.

### 6.2.1 ACPS Control Axes

This subsection presents a technique for decoupling the ACPS control of the Shuttle Orbiter into three essentially independent axes. This technique permits one to treat the rigid body rotational dynamics of the vehicle, normally represented by a $6^{th}$ order system, as three independent $2^{nd}$ order systems, thus justifying the use of phase plane control logic. A similar technique was used in the design of the Lunar Module autopilot. The jet selection logic which has been implemented to utilize the decoupled axes is limited to specified jet pairs and provides only attitude control, not translation, in its present form. A more general scheme will be used in the future.

The ACPS thruster group locations for the NR ATP baseline configuration are given in Subsection 2.3. Define a +u thruster combination as one or more thrusters from group 2 and/or group 12: e.g., jet #14 and jet #31. Also define a -u thruster combination as one or more thrusters from groups 1 and 11: e.g., jets #10 and 25. The only restriction in selecting the jets is that the torque produced by the +u combination is opposite (or nearly opposite) in direction to that of the -u jets. Similarly, select a +v combination from groups 4 and 10, e.g., #9 and #26; and a -v combination from groups 3 and 9, e.g., #13 and #32. Finally, select a +w combination from groups 6 and 8, e.g., #5 and #18; and a -w combination from groups 5 and 7, e.g., #6 and #17. The suggested thruster combinations give moderate rotational control authority while minimizing the net translation.

The control axes are defined by the accelerations produced by the u, v and w jets. More specifically, let

$\underline{U}_{BASIS}$ = unit vector in the direction of the +u axis

$\underline{V}_{BASIS}$ = unit vector in the direction of the +v axis

$\underline{W}_{BASIS}$ = unit vector in the direction of the +w axis

$\overset{*}{J}$ = orbiter inertia matrix

$\underline{M}_{P2U}, \underline{M}_{M2U}, \underline{M}_{P2V}, \underline{M}_{M2V}, \underline{M}_{P2W}, \underline{M}_{M2W}$ = control torques produced by the $\pm u$, $\pm v$, $\pm w$ thruster combinations

then

$$\underline{U}_{BASIS} = \text{unit } (\overset{*}{J}^{-1} (\underline{M}_{P2U} - \underline{M}_{M2U})/2)$$

$$\underline{V}_{BASIS} = \text{unit } (\overset{*}{J}^{-1} (\underline{M}_{P2V} - \underline{M}_{M2V})/2)$$

$$\underline{W}_{BASIS} = \text{unit } (\overset{*}{J}^{-1} (\underline{M}_{P2W} - \underline{M}_{M2W})/2)$$

(The components are given in vehicle axes.)

The control axes orientations are shown in Fig. 6-2 for the thruster combinations given above. Note that the $\pm u$ accelerations are not exactly opposite. Neither are the $\pm v$ accelerations.

The transformation from control axes to vehicle (or pilot) axes is defined as:

$$\overset{*}{\underline{M}}_{CP} = (\underline{U}_{BASIS}, \underline{V}_{BASIS}, \underline{W}_{BASIS})$$

The transformation from vehicle (or pilot) axes to control axes is simply:

$$\overset{*}{M}_{PC} = (\overset{*}{M}_{CP})^{-1}$$

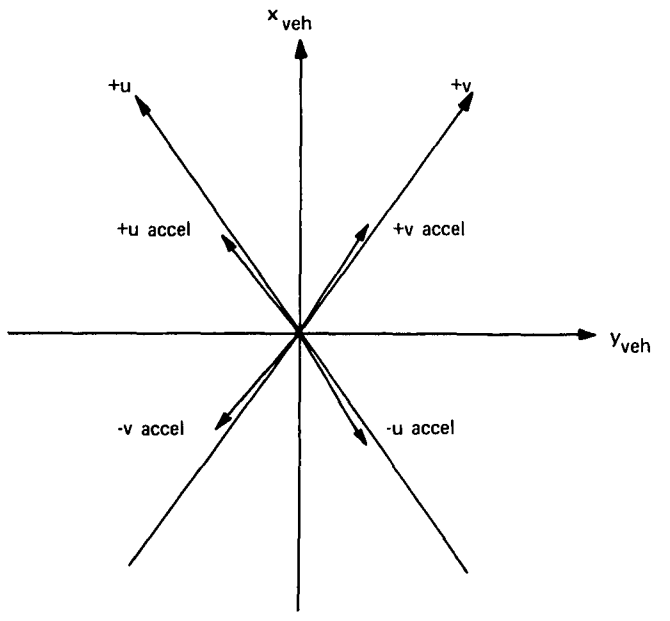
The attitude and rate errors for the autopilot are derived in vehicle axes. In order to utilize the decoupling technique these must be transformed from vehicle axes to control axes, i.e.,

$$\begin{pmatrix} \theta_{eu} \\ \theta_{ev} \\ \theta_{ew} \end{pmatrix} = \overset{*}{M}_{PC} \begin{pmatrix} \theta_{ex} \\ \theta_{ey} \\ \theta_{ez} \end{pmatrix}$$

$$\begin{pmatrix} \omega_{eu} \\ \omega_{ev} \\ \omega_{ew} \end{pmatrix} = \overset{*}{M}_{PC} \begin{pmatrix} \omega_{ex} \\ \omega_{ey} \\ \omega_{ez} \end{pmatrix}$$

### 6.2.2 State Estimator for the ACPS On-Orbit Phase

Control of the shuttle vehicle in orbit requires estimates of vehicle angular rate, $\omega$, and vehicle angular disturbance acceleration, $\alpha_d$. These quantities are



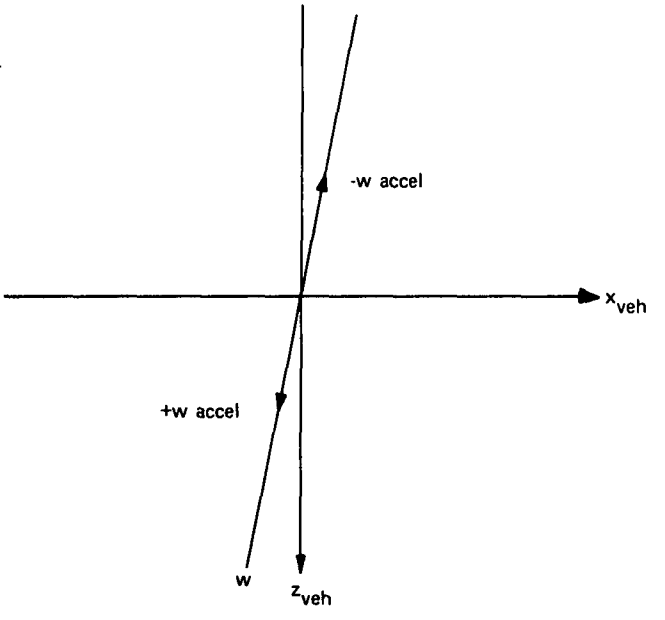




Fig. 6-2 u, v, w Control Axes

determined from measurements of the IMU gimbal angles. The procedure used is based on Kalman filter theory in which angular measurements are mixed with extrapolated state estimates to form a filtered estimate of the state. The process of extrapolation and measurement incorporation is described first in terms of a single axis. The complete three axis filter as implemented in the DFCS is discussed at the end of this subsection.

### 6.2.2.1 Vehicle Model

In estimation theory it is necessary to define a model of the process being observed. In the case of the shuttle vehicle, we will assume that it is a rigid body with the following equations of motion for one axis:

$$\begin{bmatrix} \dot{\theta}_B \\ \dot{\omega} \\ \dot{\alpha}_d \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \theta_B \\ \omega \\ \alpha_d \end{bmatrix} + \begin{bmatrix} 0 \\ \alpha_C \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \alpha_m \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ n_\omega \\ n_\alpha \end{bmatrix} \tag{6-1}$$

where

$\theta_B$ = vehicle altitude

$\omega$ = vehicle angular velocity

$\alpha_d$ = angular disturbance acceleration

$\alpha_C$ = control acceleration (produced by ACPS jets)

$\alpha_m$ = known bias acceleration

$n_\omega, n_\alpha$ = driving noise terms

### 6.2.2.2 Extrapolation

The discrete filtering process can be divided into two parts: extrapolation and measurement incorporation. In the extrapolation process the estimate of the vehicle state, $\hat{\underline{x}}(t_0)$, given at some time, $t_0$, is integrated forward to the next measurement incorporation time, $t_1$. This integration includes the effect of any known biases or control inputs during the interval $t_0$, $t_1$. The result of the extrapolation process is a predicted estimate $\hat{\underline{x}}'(t_1)$. In our case $\hat{\underline{x}}'(t_1)$ is determined by integrating Eq.(6-1), i.e.,

$$\hat{\underline{x}}'(t_1) = \overset{*}{\Phi}(t_1, t_0)\hat{\underline{x}}(t_0) + \int_{t_0}^{t_1} \overset{*}{\Phi}(t_1, \tau) \begin{bmatrix} 0 \\ \alpha_C(\tau) \\ 0 \end{bmatrix} d\tau + \int_{t_0}^{t_1} \overset{*}{\Phi}(t_1, \tau) \begin{bmatrix} 0 \\ \alpha_m(\tau) \\ 0 \end{bmatrix} d\tau \tag{6-2}$$

where

$$\underline{\hat{x}}(t_0) = \begin{bmatrix} \hat{\theta}_B(t_0) \\ \hat{\omega}(t_0) \\ \hat{\alpha}_d(t_0) \end{bmatrix} = \text{state estimate at } t_0 \qquad (6\text{-}3)$$

and

$$\overset{*}{\Phi}(t_1, \tau) = \begin{bmatrix} 1 & (t_1 - \tau) & \frac{(t_1 - \tau)^2}{2} \\ 0 & 1 & (t_1 - \tau) \\ 0 & 0 & 1 \end{bmatrix} = \text{transition matrix} \qquad (6\text{-}4)$$

$\alpha_C(\tau)$ is the control acceleration applied by the ACPS jets. A typical plot of $\alpha_C(\tau)$ is shown in Fig. 6-3. It is assumed that the control acceleration can have one of three values $+|\alpha_C|$, 0, $-|\alpha_C|$. It is also assumed that if a jet goes on, it turns at the beginning of the interval, $t_0$. It is further assumed that when the jet goes off it remains off until the end of the interval. With this jet model we can now integrate the second term in Eq. (6-2)·

$$\int_{t_0}^{t_1} \overset{*}{\Phi}(t_1, \tau) \begin{bmatrix} 0 \\ \alpha_C(\tau) \\ 0 \end{bmatrix} d\tau = \begin{bmatrix} |\alpha_C| \, \mathrm{SIGN}(\alpha_C) \, t_{JET} \left(\Delta T - \frac{t_{JET}}{2}\right) \\ |\alpha_C| \, \mathrm{SIGN}(\alpha_C) \, t_{JET} \\ 0 \end{bmatrix} \qquad (6\text{-}5)$$

where

$$\Delta T \overset{\Delta}{=} t_1 - t_0 = t_{fast}$$

$\alpha_m(\tau)$ is a known bias acceleration which is assumed to be constant over the over the integration interval:

$$\alpha_m(\tau) = \alpha_m(t_0)$$

The third term in Eq. (6-2) then reduces to

$$\int \overset{*}{\Phi}(t_1, \tau) \begin{bmatrix} 0 \\ \alpha_m^o(\tau) \\ 0 \end{bmatrix} d\tau = \begin{bmatrix} \alpha_m(t_0) \Delta T \left(\frac{\Delta T}{2}\right) \\ \alpha_m(t_0) \Delta T \\ 0 \end{bmatrix} \qquad (6\text{-}6)$$

Before grouping terms, define

$$\Delta\omega_{dm}(t_0) = [\hat{\alpha}_d(t_0) + \alpha_m(t_0)] \; \Delta T$$

$$\Delta\omega_C(t_0) = |\alpha_C| \; \mathrm{SIGN}(\alpha_C(t_0)) \, t_{JET}$$

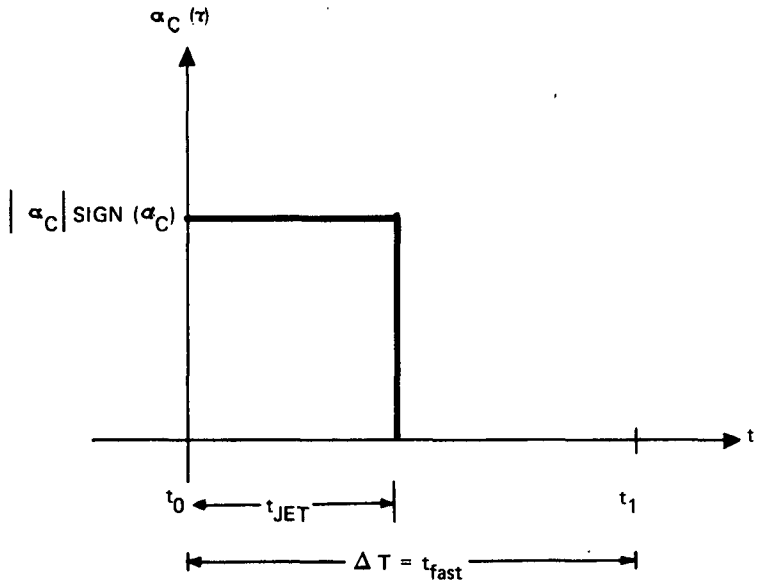



Fig. 6-3 ACPS Jet Control Acceleration Model

Combining results, we find

$$\hat{\theta}_B'(t_1) = \hat{\theta}_B(t_0) + \hat{\omega}(t_0)\,\Delta T + \Delta\omega_{dm}(t_0)\,\frac{\Delta T}{2} + \Delta\omega_C(t_0)\,(\Delta T - \frac{t_{JET}}{2}) \tag{6-7a}$$

$$\hat{\omega}'(t_1) = \hat{\omega}(t_0) + \Delta\omega_{dm}(t_0) + \Delta\omega_C(t_0) \tag{6-7b}$$

$$\hat{\alpha}_d'(t_1) = \hat{\alpha}_d(t_0) \tag{6-7c}$$

6.2.2.3 Measurement Incorporation

For the measurement incorporation process, the extrapolated state is combined with the measurement at time $t_n$ in the following manner:

$$\hat{\theta}_B(t_n) = \hat{\theta}_B'(t_n) + K_\theta\,(\tilde{\theta}(t_n) - \hat{\theta}_B(t_n)) \tag{6-8a}$$

$$\hat{\omega}(t_n) = \hat{\omega}'(t_n) + K_\omega(\tilde{\theta}(t_n) - \hat{\theta}_B'(t_n))\,/\,(t_n - t_\ell) \tag{6-8b}$$

$$\hat{\alpha}_d(t_n) = \hat{\alpha}_d'(t_n) + K_\alpha(\tilde{\theta}(t_n) - \hat{\theta}_B'(t_n))\,/\,(t_n - t_\ell)^2 \tag{6-8c}$$

$$t_\ell \leftarrow t_n$$

where

$K_\theta$ = position gain (constant)

$K_\omega$ = velocity gain (constant)

$K_\alpha$ = acceleration gain (constant)

$\tilde{\theta}(t_n)$ = angle measurement at time $t_n$

$t_\ell$ = time of the last measurement incorporation

This measurement incorporation process is similar to the linear Kalman filter if the measurement interval, $t_n - t_\ell$, is fixed. However, for the Shuttle filter, measurements will be incorporated only if

$$|\tilde{\theta}(t_n) - \hat{\theta}_B'(t_n)| > \theta_{BT} \tag{6-9}$$

where

$\theta_{BT}$ = "trap" angle

If this condition is not true, then no measurement is processed and

$$\theta_B(t_n) = \theta_B'(t_n) \tag{6-10a}$$

$$\hat{\omega}(t_n) = \hat{\omega}'(t_n) \tag{6-10b}$$

$$\hat{\alpha}_d(t_n) = \hat{\alpha}_d'(t_n) \tag{6-10c}$$

This additional logic helps to reduce the effect of quantization of the gimbal angle measurements by "trapping" small-amplitude noise before it reaches the estimate.

6.2.2.4 The 3-Axis Filter as Implemented in the DFCS

The 3-axis state estimator is basically a simple extension of the single axis filter described above. Due to the kinematics of Euler rotations, however, the gimbal angle measurements are not simply related to the vehicle body rates. We can circumvent this problem by defining a set of "body angle" measurements. The body angles are defined as the integral of the body rates:

$$\theta_{B_1} = \int \omega_1 \, dt \tag{6-11a}$$

$$\theta_{B_2} = \int \omega_2 \, dt \tag{6-11b}$$

$$\theta_{B_3} = \int \omega_3 \, dt \tag{6-11c}$$

To a good approximation we can calculate the body angles in recursive fashion as:

$$\begin{bmatrix} \tilde{\theta}_{B_1}(n) \\ \tilde{\theta}_{B_2}(n) \\ \tilde{\theta}_{B_3}(n) \end{bmatrix} = \begin{bmatrix} \tilde{\theta}_{B_1}(n-1) \\ \tilde{\theta}_{B_2}(n-1) \\ \tilde{\theta}_{B_3}(n-1) \end{bmatrix} \oplus \begin{bmatrix} 1 & \mathrm{SIN}(\theta_{GA_3}) & 0 \\ 0 & \mathrm{COS}(\theta_{GA_1})\mathrm{COS}(\theta_{GA_3}) & \mathrm{SIN}(\theta_{GA_1}) \\ 0 & -\mathrm{SIN}(\theta_{GA_1})\mathrm{COS}(\theta_{GA_3}) & \mathrm{COS}(\theta_{GA_1}) \end{bmatrix} \begin{bmatrix} \theta_{GA_1}(n) \ominus \theta_{GA_1}(n-1) \\ \theta_{GA_2}(n) \ominus \theta_{GA_2}(n-1) \\ \theta_{GA_3}(n) \ominus \theta_{GA_3}(n-1) \end{bmatrix} \tag{6-12}$$

where

$\theta_{GA_1}$ = outer gimbal angle

$\theta_{GA_2}$ = inner gimbal angle

$\theta_{GA_3}$ = middle gimbal angle

and

$\oplus$ indicates modular addition

$\ominus$ indicates modular subtraction

(Modular addition and subtraction of two angles produces a result which lies in the range: $-180° < \theta \leq 180°$.) Measurement incorporation is performed as before (Eq. (6-8)) except that the measurements are now the computed body angles (Eq. (6-12)). For the x axis,

if

$$|\tilde{\theta}_{B_x}(t_n) - \hat{\theta}'_{B_x}(t_n)| > \theta_{BT} \tag{6-13}$$

then $$\hat{\theta}_{B_x}(t_n) = \hat{\theta}'_{B_x}(t_n) \oplus K_\theta\left[\tilde{\theta}_{B_x}(t_n) \ominus \hat{\theta}'_{B_x}(t_n)\right] \tag{6-13a}$$

$$\hat{\omega}_x(t_n) = \hat{\omega}'_x(t_n) + K_\omega\left[\tilde{\theta}_{B_x}(t_n) \ominus \hat{\theta}'_{B_x}(t_n)\right]/(t_n - t_{\ell_x}) \tag{6-13b}$$

$$\hat{\alpha}_{dx}(t_n) = \hat{\alpha}_{dx}(t_n) + K_\alpha\left[\tilde{\theta}_{B_x}(t_n) \ominus \hat{\theta}_{B_x}(t_n)\right]/(t_n - t_{\ell_x})^2 \tag{6-13c}$$

$$t_{\ell_x} \leftarrow t_n \tag{6-13d}$$

otherwise,

$$\hat{\theta}_{B_x}(t_n) = \theta'_{B_x}(t_n) \tag{6-14a}$$

$$\hat{\omega}_x(t_n) = \omega'_x(t_n) \tag{6-14b}$$

$$\hat{\alpha}_{dx}(t_n) = \alpha'_{dx}(t_n) \tag{6-14c}$$

In the above equations

$\hat{\theta}_{B_x}$ = estimated value of the x body angle

$\hat{\theta}'_{B_x}$ = extrapolated value of the x body angle

$\tilde{\theta}_{B_x}$ = "measured" value of the x body angle

$\hat{\omega}_x$ = estimated x-axis body rate

$\hat{\omega}'_x$ = extrapolated x-axis body rate

$\hat{\alpha}_{dx}$ = estimated x-axis component of disturbing acceleration

$\hat{\alpha}'_{dx}$ = extrapolated x-axis component of disturbing acceleration

$t_n$ = current time

$t_{\ell_x}$ = time of the last measurement incorporation for the x-axis

The y and z axis computations are similar. Note that a separate "trap" is required for each axis.

After the appropriate jet control is exercised the extrapolation of the vehicle state, which depends to a large extent on the ACPS jet selection and timing logic used in the Shuttle autopilot, is performed. It is currently assumed that the jet firings produce accelerations about the u, v and w axes defined in Subsection 6.2.1. That is, the angular acceleration produced about the vehicle axes by the jets is the sum of the three components:

$$\underline{\alpha}_{CU} = \overset{*}{J}^{-1} (\underline{M}_{P2U} - \underline{M}_{M2U})/2 \tag{6-15a}$$

$$\underline{\alpha}_{CV} = \overset{*}{J}^{-1} (\underline{M}_{P2V} - \underline{M}_{M2V})/2 \tag{6-15b}$$

$$\underline{\alpha}_{CW} = \overset{*}{J}^{-1} (\underline{M}_{P2W} - \underline{M}_{M2W})/2 \tag{6-15c}$$

where $\overset{*}{J}$ is the inertia matrix and the $\underline{M}$'s are the torque vectors defined in Subsection 6.2.1. It is also assumed that the restrictions on the timing of jet firings which were given for the single-axis filter apply. If we define

$t_{JET_1}$ = time of firing of the u jets during the interval $t_0 < t < t_1$

$t_{JET_2}$ = time of firing of the v jets during the interval $t_0 < t < t_1$

$t_{JET_3}$ = time of firing of the w jets during the interval $t_0 < t < t_1$

and

$SGNC_1$ = sign of the desired u control acceleration

$SGNC_2$ = sign of the desired v control acceleration

$SGNC_3$ = sign of the desired w control acceleration

then by analogy with the results of the single axis filter:

$$\Delta\underline{\omega}_u(t_0) = |\underline{\alpha}_{CU}|\ SGNC_1\ t_{JET_1} \tag{6-16a}$$

$$\Delta\underline{\omega}_v(t_0) = |\underline{\alpha}_{CV}|\ SGNC_2\ t_{JET_2} \tag{6-16b}$$

$$\Delta\underline{\omega}_w(t_0) = |\underline{\alpha}_{CW}|\ SGNC_3\ t_{JET_3} \tag{6-16c}$$

and

$$\Delta\underline{\omega}_{dm}(t_0) = (\hat{\underline{\alpha}}_d(t_0) + \underline{\alpha}_m(t_0))\Delta T \tag{6-16d}$$

Note that these are vector quantities. The bias acceleration, $\underline{\alpha}_m$, is computed by the attitude maneuver routine as

$$\underline{\alpha}_m = -\overset{*}{J}^{-1}(\underline{\omega}_d x \overset{*}{J}\underline{\omega}_d) \tag{6-17}$$

where $\underline{\omega}_d$ is the desired maneuver rate. With these definitions, the complete three axis extrapolation equations can be written as

$$\begin{aligned}\hat{\underline{\theta}}'_B(t_1) = \hat{\underline{\theta}}_B(t_0) \oplus \Bigg[ & \hat{\underline{\omega}}(t_0)\Delta T + \Delta\underline{\omega}_{dm}(t_0)\frac{\Delta T}{2} \\ & + \Delta\underline{\omega}_u(t_0)\left(\Delta T - \frac{t_{JET_1}}{2}\right) + \Delta\underline{\omega}_v(t_0)\left(\Delta T - \frac{t_{JET_2}}{2}\right) \\ & + \Delta\underline{\omega}_w(t_0)\left(\Delta T - \frac{t_{JET_3}}{2}\right)\Bigg]\end{aligned} \tag{6-18a}$$

$$\hat{\underline{\omega}}'(t_1) = \hat{\underline{\omega}}(t_0) + \Delta\,\underline{\omega}_{dm}(t_0) + \Delta\underline{\omega}_u(t_0) + \Delta\underline{\omega}_v(t_0) + \Delta\underline{\omega}_w(t_0) \tag{6-18b}$$

$$\hat{\underline{\alpha}}'_d(t_1) = \hat{\underline{\alpha}}_d(t_0) \tag{6-18c}$$

Note that these quantities are vectors with components in the x, y, z vehicle (body) axes.

The estimate of disturbance acceleration, $\hat{\underline{\alpha}}_d$, may be accurate enough for the powered flight phases, but it may be too noisy for use during drifting flight. If an estimate is needed during coasting flight to counteract such disturbances as gravity gradient and aerodynamic torques, then rate gyro measurements may be required. If not, the $\hat{\underline{\alpha}}_d$ estimation process may be disabled by using appropriate filter gains. Alternatively, the gain, $K_\alpha$, simply can be reduced during coasting flight. In any event, the bias acceleration estimate, $\underline{\alpha}_m$, should be used during coasting flight maneuvers to reduce fuel consumption.

### 6.2.3 Control Laws for the On-Orbit ACPS Phase

The object of the phase plane switching logic is to compute changes in angular velocity so that by turning on or off the appropriate jets to achieve this velocity change the vehicle limit cycles about a desired attitude and attitude rate. The two dimensional phase plane is applied to the decoupled control axes described in Subsection 6.2.1. Limit cycling is desirable since it would be extremely costly to attempt to maintain the desired attitude and rate exactly. By designing appropriate switching lines into the phase plane, the vehicle can be driven into the desired limit cycle. Two separate logics are used depending on the size of the position and rate errors. For large errors, a switching logic has been designed to drive the state into the small-error control region. The small-error switching logic then drives the state into a limit cycle about the desired attitude and attitude rate. The distinction between large and small errors will be quantified later in this section.

The small-error switching logic has the following features:

1) The design is based upon the optimal switch curves for minimizing a weighted combination of fuel and response time. The optimal curves are separated by a deadband in order to allow for limit cycling.

2) The appropriate deadband can be chosen in order to
   a) fix the limit cycle period
   b) fix the limit cycle maximum rate
   c) fix the maximum attitude excursion while limit cycling

3) The average attitude error is minimized by biasing the placement of the switch curves.

4) Parabolic switch curves approximate the trajectories which the state follows. However, hysteresis logic prevents performance degradation due to errors in these approximations.

The design incorporating these features is described in the following subsections.

In Subsection 6.2.3.1 two idealized limit cycles are presented. One limit cycle is in the presence of positive disturbing accelerations while the other is in the presence of zero disturbing accelerations. For each limit cycle the average attitude error will be derived. In Subsection 6.2.3.2 the design of switching curves to drive the state into a limit cycle with zero average attitude error is described. This design is referred to as the small-error control logic. The large-error control logic is described in Subsection 6.2.3.3.

### 6.2.3.1 Idealized Limit Cycles

An idealized limit cycle for a positive disturbing acceleration is illustrated in Fig. 6-4. The coordinates $\theta_e$ and $\omega_e$ are chosen to be the state, where $\theta_e$ represents the angular position error and $\omega_e$ represents the angular rate error. $X_{DB}$ is the attitude variation in the course of limit cycling. In the presence of a positive disturbance acceleration, $U_D$, the state is driven along the trajectory from A to B. At point B a negative control acceleration, $-U_C$, is applied to return the state to point A.

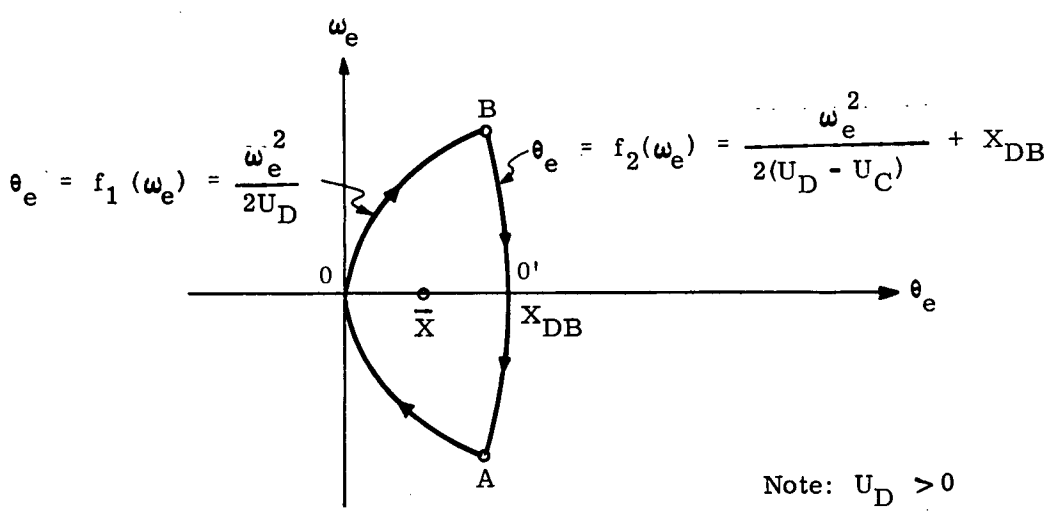



Fig. 6-4 Idealized Limit Cycle for a Positive Disturbing Acceleration

The average attitude error, $\overline{X}$, of the limit cycle in Fig. 6-4 is found by determining the average attitude error for each of the trajectories OB and BO′ (called $\overline{X}_1$ and $\overline{X}_2$) and then "weighting" them according to time to calculate the net average, $\overline{X}$.

Determination of $\overline{X}_1$:

$\overline{X}_1$ is, by definition,

$$\overline{X}_1 = \frac{1}{t_1} \int_0^{t_1} \frac{1}{2} U_D t^2 \, dt \tag{6-19}$$

$t_1$, the time to travel from point O to point B, can be calculated from the value of the $\theta_e$ abscissa of point B, $B_{\theta_e}$, since

$$B_{\theta_e} = \frac{1}{2} U_D t_1^2 \tag{6-20}$$

$B_{\theta_e}$ is found by first rearranging $\theta_e = f_1(\omega_e)$ and $\theta_e = f_2(\omega_e)$ such that,

$$\omega_e = g_1(\theta_e), \quad \omega_e = g_2(\theta_e) \tag{6-21}$$

Then, at point B, $g_1(\theta_e) = g_2(\theta_e)$, and,

$$B_{\theta_e} = X_{DB}\left(\frac{U_C - U_D}{U_C}\right) \tag{6-22}$$

and from Eq(6-20)

$$t_1 = \pm\sqrt{\frac{2X_{DB}(U_C - U_D)}{U_C U_D}} \tag{6-23}$$

Using $t_1$ in Eq(6-19)

$$\overline{X}_1 = \frac{1}{3} X_{DB}\left(\frac{U_C - U_D}{U_C}\right) \tag{6-24}$$

Determination of $\overline{X}_2$:

The time average attitude error for the trajectory segment BO′ can be found by setting t = 0, at point B. Then,

$$\overline{X}_2 = \frac{1}{t_2} \int_0^{t_2} [X_o + \dot{X}_o t + \frac{1}{2}(U_D - U_C) t^2] \, dt$$

or,

$$\overline{X}_2 = X_o + \frac{\dot{X}_o t_2}{2} + \frac{1}{6}(U_D - U_C) t_2^2 \tag{6-25}$$

$X_o$ is defined in Eq (6-22) and $\dot{X}_o$ can be determined from,

$$\dot{X}_o = U_D t_1 = +\sqrt{\frac{2X_{DB}U_D(U_C - U_D)}{U_C}} \tag{6-26}$$

To find $t_2$ one can use the following relation,

$$\dot{X}_o + (U_D - U_C)t_2 = 0$$

from which it follows,

$$t_2 = \sqrt{\frac{2X_{DB}U_D}{U_C(U_C - U_D)}} \tag{6-27}$$

Substituting the results of Eq (6-22), (6-26) and (6-27) into Eq (6-25), yields:

$$\overline{X}_2 = X_{DB}\left(1 - \frac{U_D}{3U_C}\right) \tag{6-28}$$

Determination of $\overline{X}$:

Finally, the total average attitude error, $\overline{X}$, can be found from:

$$\overline{X} = \frac{\overline{X}_1 t_1 + \overline{X}_2 t_2}{t_1 + t_2} \tag{6-29}$$

Substituting the results of Eqs (6-23), (6-24), (6-27), (6-28) into Eq (6-29), yields:

$$\overline{X} = \frac{X_{DB}}{3}\left(1 + \frac{U_D}{U_C}\right) \tag{6-30}$$

This is the average attitude error for the limit cycle in Fig. 6-4. Thus, the average attitude error would be zero if the intercept of the curve on the right, which is the trajectory along which the control is applied, were shifted from point O′ to have an abscissa of

$$DR = X_{DB} - \overline{X} \tag{6-31}$$

A similar result can be obtained for the case of a negative disturbance acceleration. Equations (6-30) and (6-31) are used in the DFCS in order to minimize the average attitude error.

The attitude variation, $X_{DB}$, could be chosen as follows:

1) To achieve the cycle period, T:

$$\text{set } X_{DB} = \left[1 - \frac{|U_D|}{|U_C|}\right] \frac{|U_D|}{8} T^2 \tag{6-32}$$

2) To achieve the limit cycle maximum rate, $\dot{X}_{max}$:

$$\text{set } X_{DB} = \frac{|U_C|(\dot{X}_{max})^2}{2|U_D|(|U_C| - |U_D|)} \tag{6-33}$$

3) By simply setting $X_{DB}$ to the allowable attitude variation.

Currently, the final option is employed in the DFCS.

An idealized limit cycle in the presence of no disturbing accelerations is illustrated in Fig. 6-5. From point A the state is allowed to drift

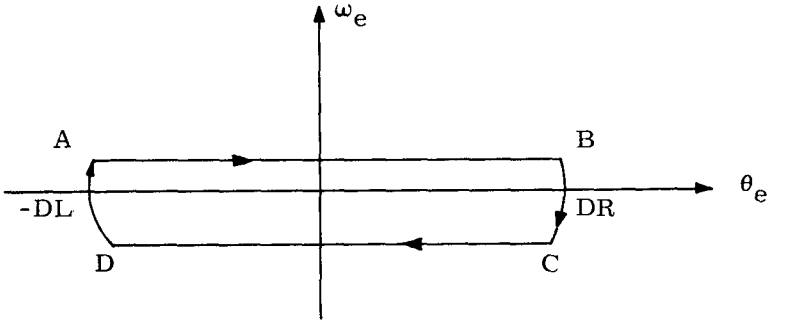



Fig. 6-5 Limit Cycle in Presence of No Disturbing Accelerations

to point B where a negative control acceleration, $-U_C$, (in the form of a minimum impulse jet firing) is applied to drive the state to point C. From point C the state is then allowed to drift to point D. At point D a positive control acceleration, $+U_C$, is applied to return the state to point A. By choosing DL = DR the state limit cycles with zero average attitude error. The generalized limit cycle for positive disturbing accelerations is made to reduce to the limit cycle for zero disturbing accelerations by selecting

$$DL = \overline{X} + \frac{1}{3} X_{DB} \tag{6-34}$$

6.2.3.2 Small-Error Control Law

Control regions for relatively small errors are illustrated in Fig. 6-6, definitions of curves are summarized in Table 6-1, and corresponding control actions are given in Table 6-2. All logic in this subsection is formulated in terms of positive disturbing accelerations. In the actual implementation, (see Subsection 6.4), negative disturbances are treated by inverting the signs of the state variables $\theta_e$ and $\omega_e$. Biasing the right and left deadbands, DR and DL, to null the average attitude error in accordance with the equations derived in the preceeding subsection, defines the placement of the switching curves BOUNDARY_II_TO_COAST and BOUNDARY_III_TO_COAST. These curves approximate the trajectory of the state during a firing. When the state lies in Region II, control is applied to drive the state to the curve TARGET_DISTURB. The vehicle is then allowed to coast.

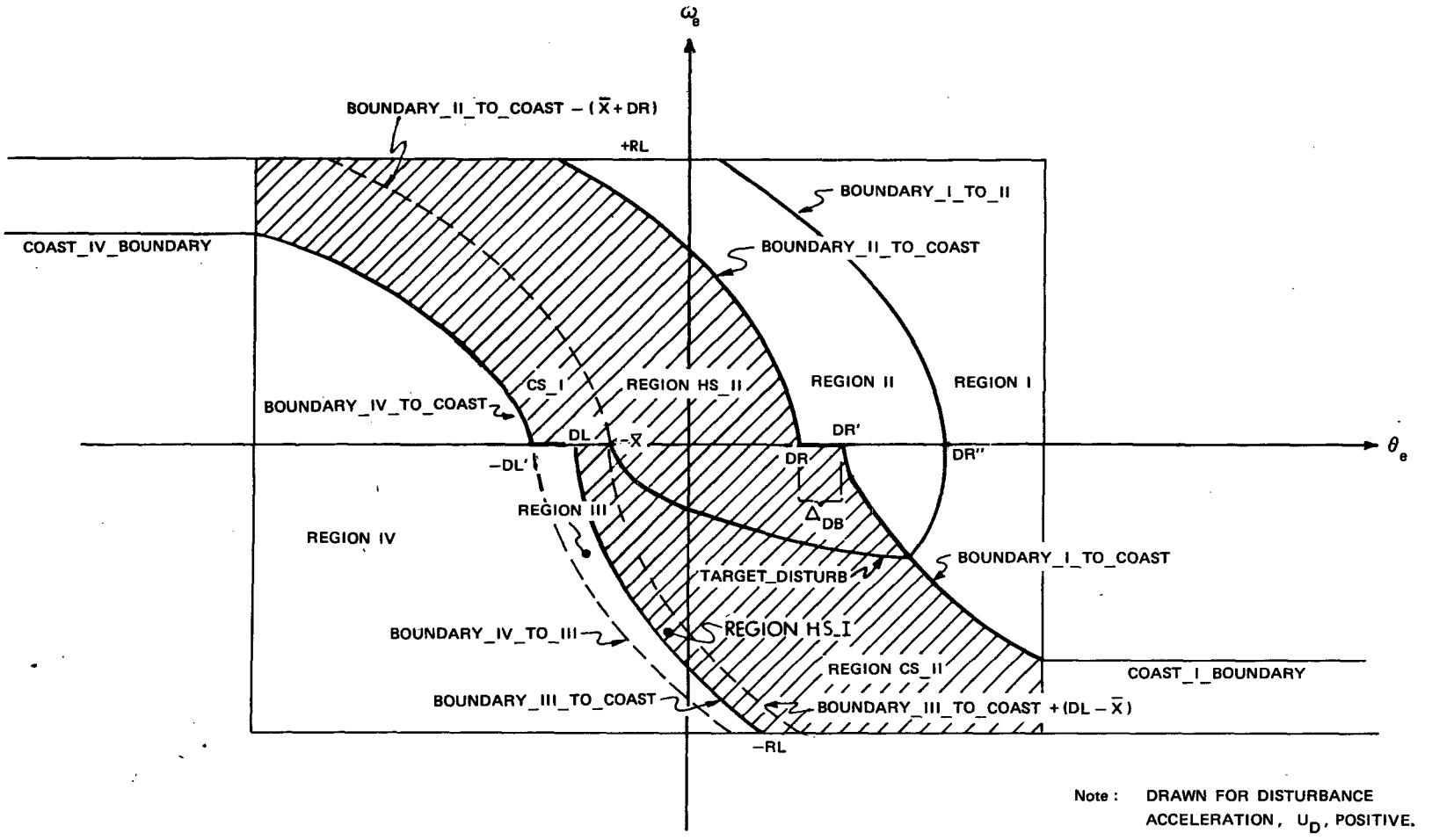



Fig. 6-6 Small-Error Disturbance Logic

Table 6-1 Definitions of Boundaries of Small-Error Control Regions

| Boundary/Parameter | Definition |
|---|---|
| BOUNDARY_I_TO_II | $\theta_e = -\dfrac{\omega_e^2}{2(\lvert U_C\rvert - \lvert U_D\rvert)} + DR''$ |
| BOUNDARY_II_TO_COAST | $\theta_e = -\dfrac{\omega_e^2}{2(\lvert U_C\rvert - \lvert U_D\rvert)} + DR$ |
| BOUNDARY_I_TO_COAST | $\theta_e = +\dfrac{\omega_e^2}{2CD(\lvert U_C\rvert + \lvert U_D\rvert)} + DR'$ |
| BOUNDARY_III_TO_COAST | $\theta_e = +\dfrac{\omega_e^2}{2(\lvert U_C\rvert + \lvert U_D\rvert)} - DL$ |
| BOUNDARY_IV_TO_III | $\theta_e = +\dfrac{\omega_e^2}{2(\lvert U_C\rvert + \lvert U_D\rvert)} - DL'$ |
| BOUNDARY_IV_TO_COAST | $\theta_e = -\dfrac{\omega_e^2}{2CD(\lvert U_C\rvert - \lvert U_D\rvert)} - DL'$ |
| TARGET_DISTURB | $\theta_e = +\dfrac{\omega_e^2}{2\lvert U_D\rvert} - \overline{X}$ |

WHERE

$\Delta_{DB} = 0.2\, X_{DB}$

$DR = X_{DB} - \overline{X}$ $\qquad DL = \overline{X} + \frac{1}{3} X_{DB}$

$DR' = DR + \Delta_{DB}$ $\qquad DL' = DL + \Delta_{DB}$

$$DR'' = (\overline{X} + DR')(\lvert U_D\rvert)\left(\frac{(\lvert U_C\rvert - \lvert U_D\rvert) + CD(\lvert U_C\rvert - \lvert U_D\rvert)}{(CD(\lvert U_C\rvert + \lvert U_D\rvert) - \lvert U_D\rvert)(\lvert U_C\rvert - \lvert U_D\rvert}\right) + DR'$$

Table 6-2 Control Actions in Small-Error Control Regions

| Region | Control Action |
|---|---|
| I | Drive state to BOUNDARY_I_TO_COAST |
| II | Drive state to TARGET_DISTURB |
| III | Drive state to zero rate |
| IV | Drive state to BOUNDARY_IV_TO_COAST |
| HS_I | If coming from LARGE ERROR CONTROL LOGIC or REGION III then drive state to zero rate, else coast |
| HS_II | If coming from LARGE ERROR CONTROL LOGIC or REGION II then drive state to TARGET_DISTURB, else coast |
| CS_I | coast |
| CS_II | coast |

The deadband increment $\Delta_{DB}$ is added to prevent fuel-consuming overshoot when the disturbance acceleration is zero or very small. The points

$$DR' = DR + \Delta_{DB} \tag{6-35}$$

$$DL' = DL + \Delta_{DB} \tag{6-36}$$

define the placement of the curves BOUNDARY_I_TO_COAST and BOUNDARY_IV_TO_COAST.

The factor CD determines the weight given to response time as opposed to fuel consumption, by adjusting the steepness of the switch curves that are normally crossed when entering the coast region. To insure that BOUNDARY_IV_TO_COAST is always less than BOUNDARY_II_TO_COAST and BOUNDARY_I_TO_COAST is always greater than BOUNDARY_III_TO_COAST, CD must be between 0 and 1. Currently the value of CD is 0.8.

Hysteresis has been incorporated to make the most efficient use of disturbance accelerations, to prevent switching line chatter, and to prevent tight limit cycling about the point DR. The dashed curves within the shaded area and parallel to BOUNDARY_II_TO_COAST provide hysteresis in the regions HS_I and HS_II. If the state lies inside either Region HS_I or HS_II and the sign of the newly computed change in rate is not the same as the sign of the last computed change in rate, then no control is applied and the vehicle is allowed to coast. However if the signs are the same, then in

Region HS_I the state is driven to zero and in Region HS-II the state is driven to TARGET_DISTURB. Whenever the state lies inside either Region CS_I or CS_II the vehicle is simply allowed to coast.

When the state lies to the right of BOUNDARY_I_TO_II position error is reduced more quickly by driving the state to BOUNDARY_I_TO_COAST and then coasting instead of driving the state to TARGET_DISTURB and then coasting. This is particularly important as the disturbance acceleration approaches zero. The curve BOUNDARY_I_TO_II is defined by the intersection of TARGET_DISTURB and BOUNDARY_I_TO_COAST.

When in Region III the rate error is driven to zero, and when in Region IV the state is driven to BOUNDARY_IV_TO_COAST to more quickly reduce position and rate errors.

6.2.3.3 Large-Error Control Law

Control regions for relatively large errors are illustrated in Fig. 6- 7, definitions of the boundaries are summarized in Table 6-3, and corresponding control actions are given in Table 6-4. As in the preceeding subsection, all logic is formulated in terms of positive disturbing accelerations. In the actual implementation (see Subsection 6.4), negative disturbances are treated by inverting the signs of the state variables, $\theta_e$ and $\omega_e$.

The purpose of the large-error control logic essentially is to drive the state into the small-error control region. Bounds between the small-error and large-error regions are set by the values of the parameters RL and RL′ (currently 4.0 and 3.0 deg/sec respectively). The rate error bounds are equal to ± RL. The left position error limit, -PLL, is defined by the intersection of BOUNDARY_IV_TO_COAST and COAST_IV_BOUNDARY. Similarly, the right position error limit, +PLR, is defined by the intersection of BOUNDARY_I_TO_COAST and COAST_I_BOUNDARY. The boundaries COAST_IV_BOUNDARY and COAST_I_BOUNDARY are rate errors equal to ± RL′.

TARGET_ARL is the average of RL and RL′. TARGET_ARL and -(TARGET_ARL) are taken to be the target rate when the state is in Regions II and I respectively; i.e., when the firing is against the disturbance acceleration. When the state is in Regions III and IV the targets are the nearest boundary; i.e. COAST_IV_BOUNDARY and -RL respectively.

Hysteresis is incorporated into the large-error coast regions to prevent chatter about +RL and COAST_I_BOUNDARY. If the state is driven into Region HL_I from Region II, then it is further driven to TARGET_ARL. If the state otherwise appears in Region HL_I the vehicle is allowed to coast. Similarly, if the state is driven into Region HL_II from Region I then it is further driven to -(TARGET_ARL); otherwise the vehicle is allowed to coast. Should the state appear in either Region CL_I or Region CL_II the vehicle is simply allowed to coast.

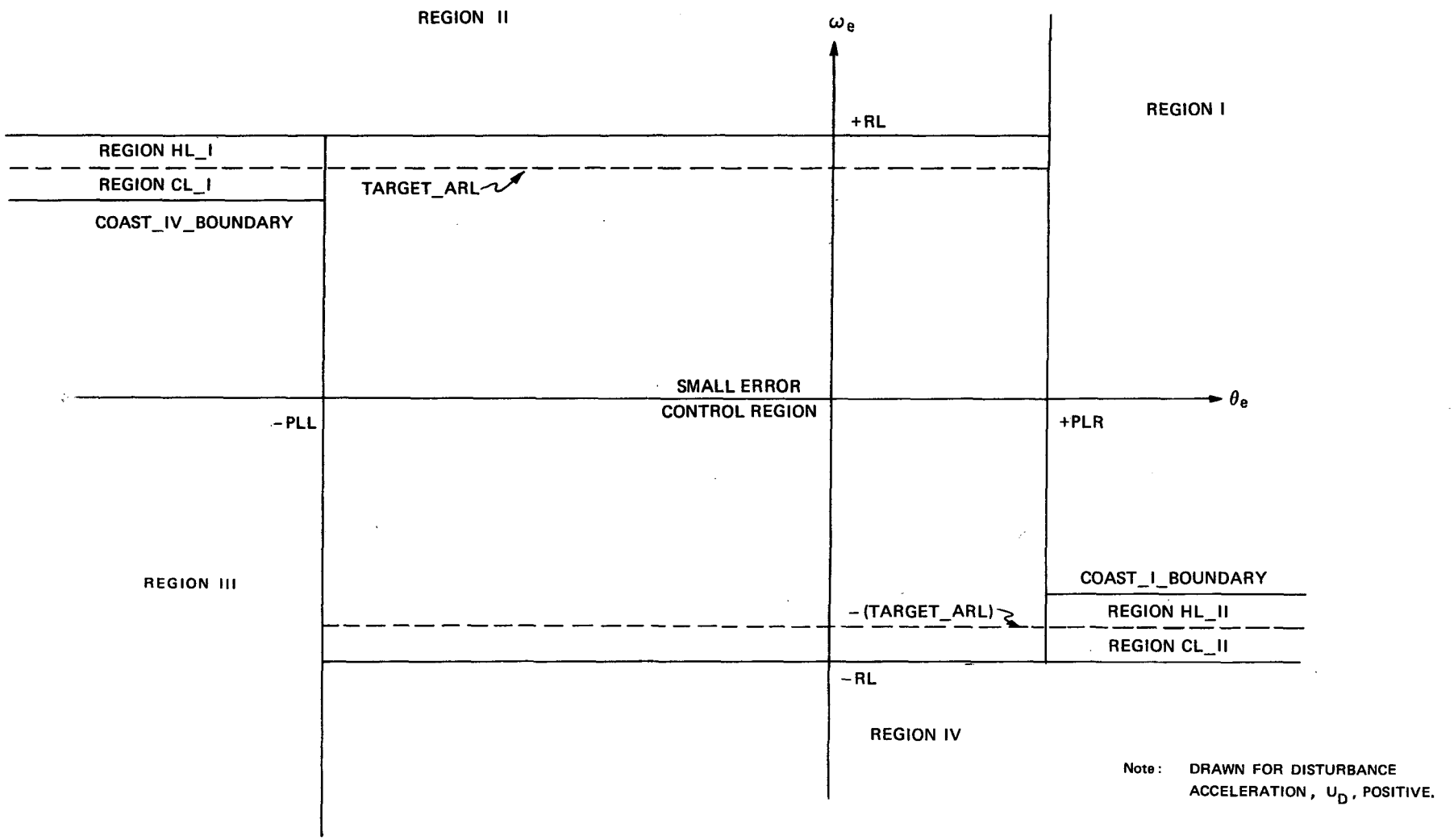



Fig. 6-7 Large Error Control Region

Table 6-3 Definitions of Boundaries for the Large-Error Control Regions

| Boundary/Parameter | Definition |
|---|---|
| COAST_IV_BOUNDARY | $\omega_e = RL'$ |
| COAST_I_BOUNDARY | $\omega_e = -RL'$ |
| TARGET_ARL | $\omega_e = (RL + RL')/2$ |
| PLR | $\theta_e = + \dfrac{(RL')^2}{2CD(\lvert U_C \rvert + \lvert U_D \rvert)} + DR'$ |
| PLL | $\theta_e = - \dfrac{(RL')^2}{2CD(\lvert U_C \rvert - \lvert U_D \rvert)} - DL'$ |

Table 6-4 Control Actions in Large-Error Control Regions

| REGION | CONTROL ACTION |
|---|---|
| I | Drive state to -(TARGET_ARL) |
| II | Drive state to TARGET_ARL |
| III | Drive state to COAST_IV_BOUNDARY |
| IV | Drive state to -RL |
| HL_I | If coming from region II then drive state to TARGET_ARL, else coast |
| HL_II | If coming from region I then drive state to -(TARGET_ARL), else coast |
| CL_I | coast |
| CL_II | coast |

A final feature of the small-error control logic can be described, now that TARGET_ARL has been introduced. If the disturbance acceleration is very large, part of the curve TARGET_DISTURB may lie below - (TARGET_ARL). In this situation a check is made to see if a rate change to drive the state to TARGET_DISTURB is greater than a rate change to drive the state to -(TARGET_ARL). If it is, the state is driven to -(TARGET_ARL), which means less fuel consumption with the result of finishing in a limit cycle about the origin.

## 6.3 On-Orbit TVC Phase: Preliminary Functional Summary

The Orbiter Maneuvering Subsystem (OMS) is used for major on-orbit maneuvers and to establish the deorbit entry trajectory. During powered flight the roll, pitch and yaw of the space-shuttle orbiter is achieved through deflection of the two gimbaled engines. Computation of gimbal servo commands in response to computed errors between commanded and measured attitudes is the function of the on-orbit TVC (Thrust Vector Control) "phase" of the DFCS. The following functional summary is an adaptation of the Apollo Command Module TVC control; it is illustrated in Fig. 6-8.

1) The steering-loop computations generate incremental attitude commands in inertial coordinates and transform them into body coordinates.

2) The DFCS back-differences the gimbal angles to obtain incremental variations over each sampling interval.

3) The gimbal increments over each sampling interval are then transformed into body coordinates and subtracted from the commanded increments which are generated by the steering program.

4) The respective attitude errors are fed to the roll, pitch and yaw compensation filters, whose outputs contribute to the commands to the engine gimbal servos. These signals bear the burden of controlling rapid transients, determining the system bandwidth and the stability of the vehicle slosh and bending modes.

5) As shown in Fig. 6-8b, the total command signal to each engine-gimbal servo is made up of the component from the associated compensation filter plus another component from a Thrust Misalignment Correction (TMC) loop. The latter component serves to bias the total command so that a zero output from the compensation filter will cause the thrust vector to pass exactly through the center of gravity (cg) when there is no cg motion and no motion of the thrust vector relative to the commanded angle.

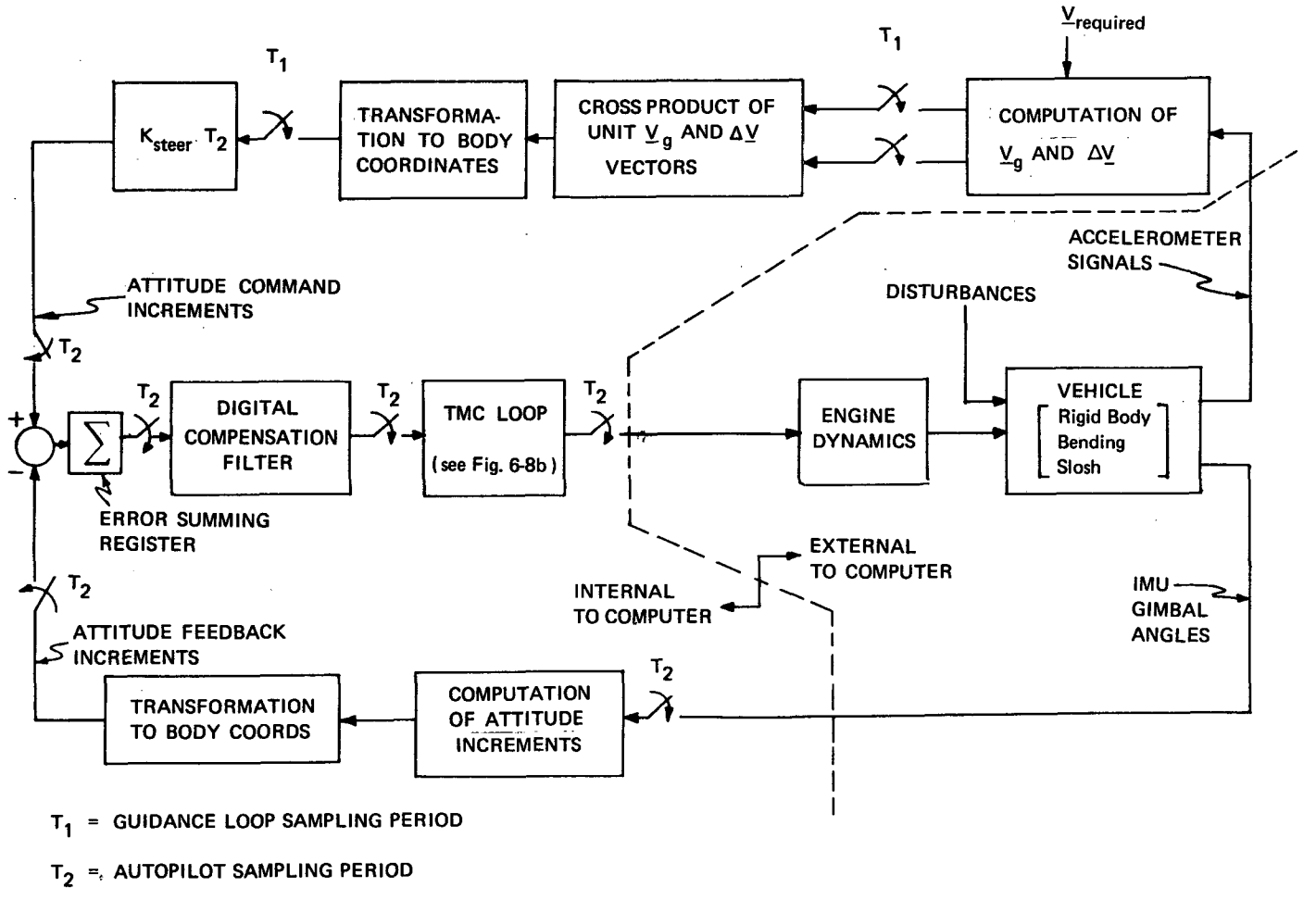



Fig. 6-8a Functional Block Diagram of the TVC DAP

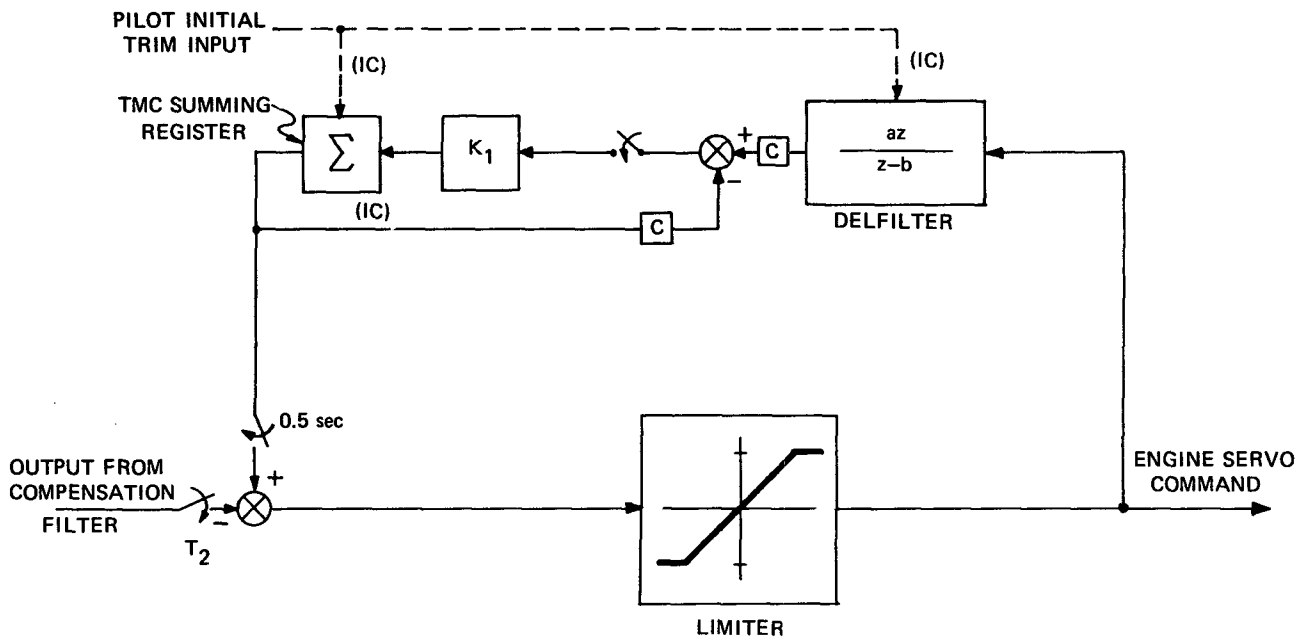



Fig. 6-8b Block Diagram of the Thrust Misalignment Correction (TMC) Loop

6) The two major elements of the TMC loop are its summing register, which supplies the bias, and its low-pass filter, DELFILTER, which tracks the total command signal at autopilot sampling frequencies. The difference between the bias and the DELFILTER output is used to increment the summing register so as to slowly correct for the offset caused by thrust-to-cg misalignment. This slow reset action is roughly equivalent to inserting a proportional-plus-integral transfer function between the compensation filter and the total command signal. The TMC loop is designed so that its dynamics have no effect on the vehicle bending and slosh modes and very little effect on the rigid-body stability.

7) The operation of the TVC digital autopilot cannot be considered separately from that of the steering loop which interacts with the autopilot. The steering loop operations are shown in Fig. 6-8 for the case of "External $\Delta V$" guidance. This form of guidance is based on achieving a desired accelerometer-measured velocity change, $\Delta \underline{V}_C$, which is inserted into the steering program as an initial value of the computed velocity-to-be-gained vector, $\underline{V}_g$. The $\underline{V}_g$ vector is periodically reduced during the course of the burn by subtracting the accelerometer-measured increments, $\Delta \underline{V}$.

8) The purpose of the steering loop is to align the vehicle thrust with the velocity-to-be-gained. This is achieved by commanding a vehicle turning-rate which is proportional to the normalized cross-product between these two vectors.

9) The pitch- and yaw-axis components of the vector cross-product are multiplied by the proportionality constant $K_{steer}$ to obtain the commanded rates about these axes - and multiplied by the autopilot sampling period, $T_2$, to obtain attitude-command increments, which are supplied to the autopilot every $T_2$ seconds.

10) The roll control channel is designed to provide independent attitude and rate control about the roll axis by differential control of the OMS engines.

## 6.4 Digital Implementation (ACPS phase)

The overall composition of the DFCS as it applies to the on-orbit phase is reiterated in Fig. 6-9 in structured flow-chart form. The procedure MAJOR_TRANSIENT provides initial mode and flag information to be used by all mission phases. Currently, no initial mode or flag information unique to the on-orbit ACPS phase has been implemented in MAJOR_TRANSIENT. The procedure INPUT_INTERFACE contains, in general, information that is explicitly read by the DFCS from external sources. More specifically, when the on-orbit phase is selected, the gimbal angles are read and the time of their reading, $t_n$, is recorded. On the first pass of the DFCS for a given phase FILTER_AND_PARAM_INIT and PARAM are both called. FILTER_AND_PARAM_INIT currently does nothing in this phase. PARAM calls MASS_CALC (See Fig. 6-10)

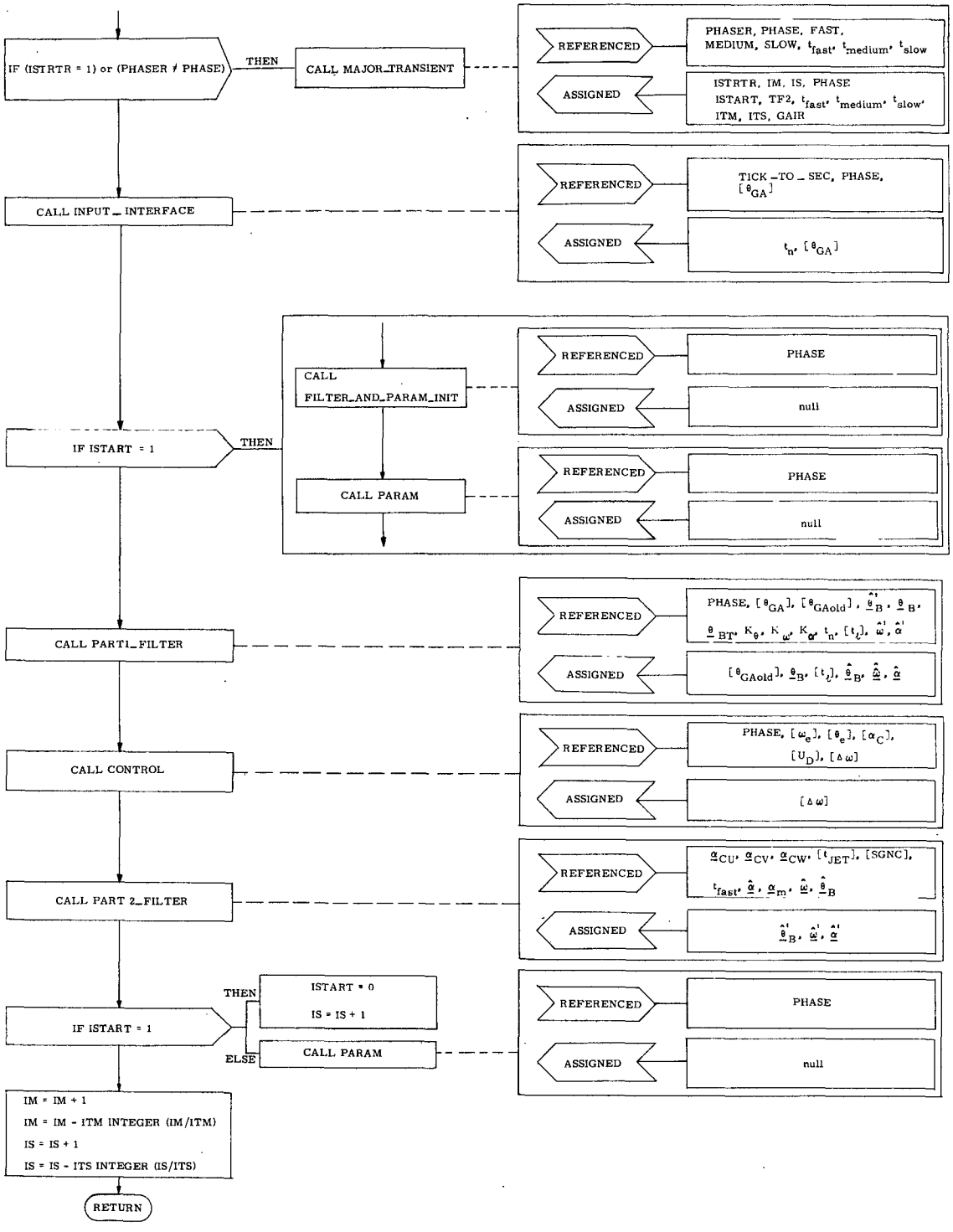



Fig. 6-9 Logic Flow of DFCS

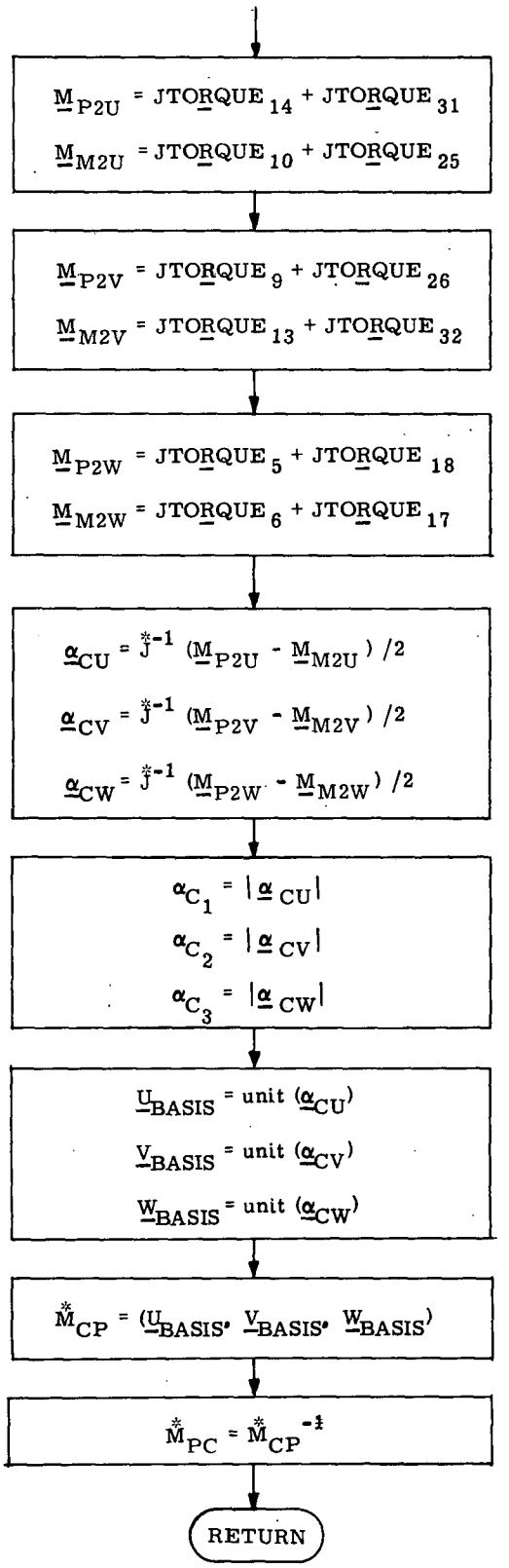



Fig. 6-10 MASS_CALC

which then takes as an input the torques supplied by each ACPS thrust group and computes three more or less independent control axes and a pilot-to-control transformation matrix. The procedure PART1_FILTER shown in Fig. 6-11 is then entered which takes as an input the extrapolated estimates of angular position, rate, and disturbance acceleration for each axis and incorporates the measurement of gimbal angles taken at time $t_n$. From these updated estimates the procedure CONTROL (Fig. 6-12) calls upon:

1) ERRORS (Fig. 6-13) to compute the errors in attitude and angular rate from the desired
2) PHASE_PLANE (Fig. 6-14) to compute the desired change in angular rate to correct these errors
3) JET_CONTROL (Fig. 6-15) to select the appropriate jets and control their firings, according to provisional logic suggested in Subsection 6.2.1.

The digital implementation of the phase plane switching logic is composed of three principle routines:

PHASE_PLANE

The procedure PHASE_PLANE selects the appropriate sign to be attached to the computed change in angular rate, computes position error intercepts, and also computes the boundaries between the small error control logic and the large error control logic, as illustrated in the logic flow diagram in Fig. 6-14a. PHASE_PLANE then selects either the small-error control logic or the large-error control logic to compute the desired change in rate.

LARGE_ERROR_CONTROL_LOGIC

The procedure LARGE_ERROR_CONTROL_LOGIC computes the change in rate to drive the state into the small error control region. As illustrated in Fig. 6-14b, the region in which the state lies is singled out and then the appropriate control action is computed.

SMALL_ERROR_CONTROL_LOGIC

The procedure SMALL_ERROR_CONTROL_LOGIC computes the change in rate to drive the state into a limit cycle with specified deadband about the desired position and rate. Current testing has used deadbands of 0.5 and 5.0 degrees. As illustrated in Figs. 6-14c and 6-14d, the region in which the state lies is singled out and then the appropriate control action is computed.

The function BOUNDARY, illustrated in Fig. 6-14e, is implemented to compute the various switching curves.

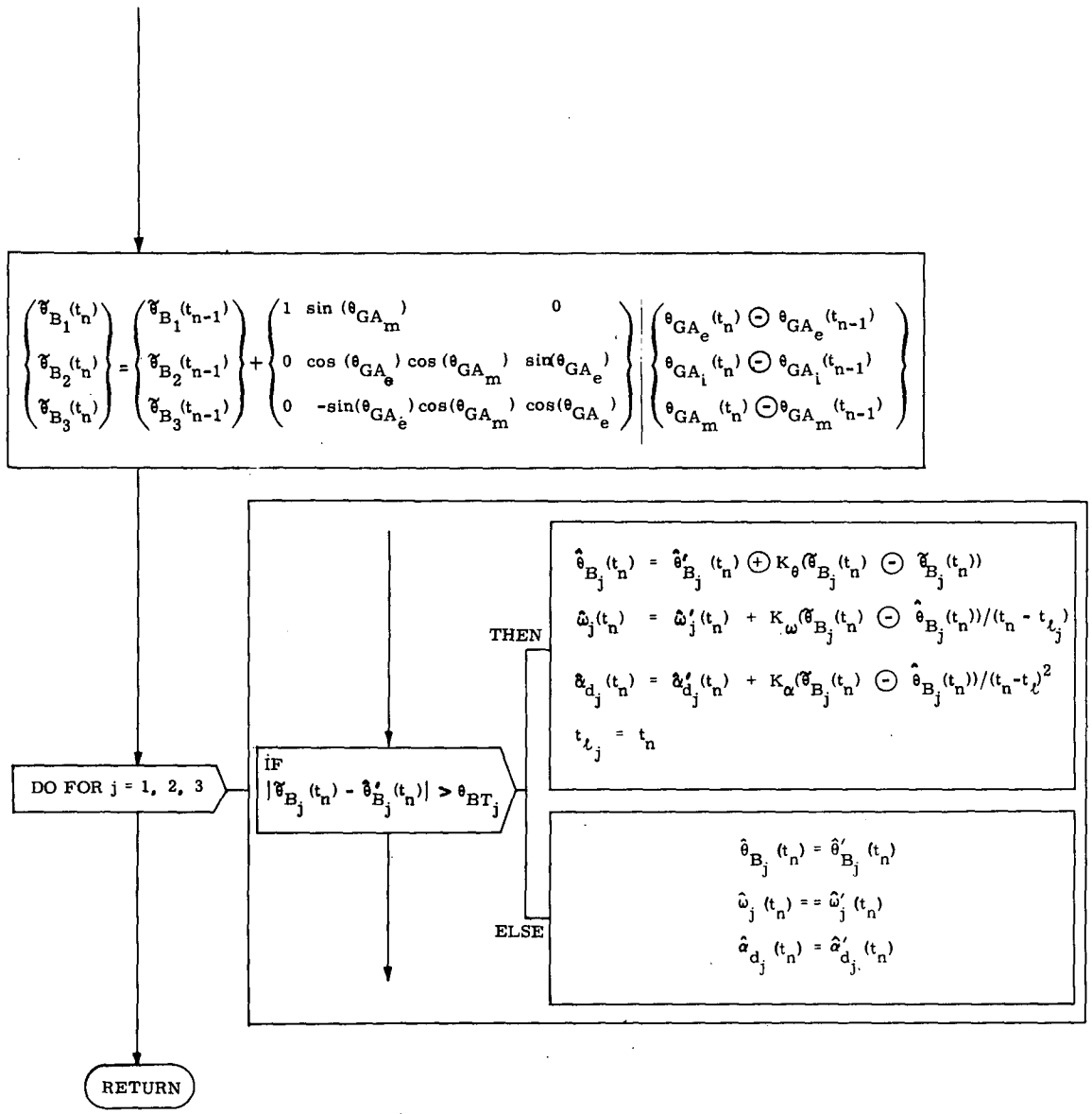



Fig. 6-11 On-orbit Segment of PART1-FILTER Procedure

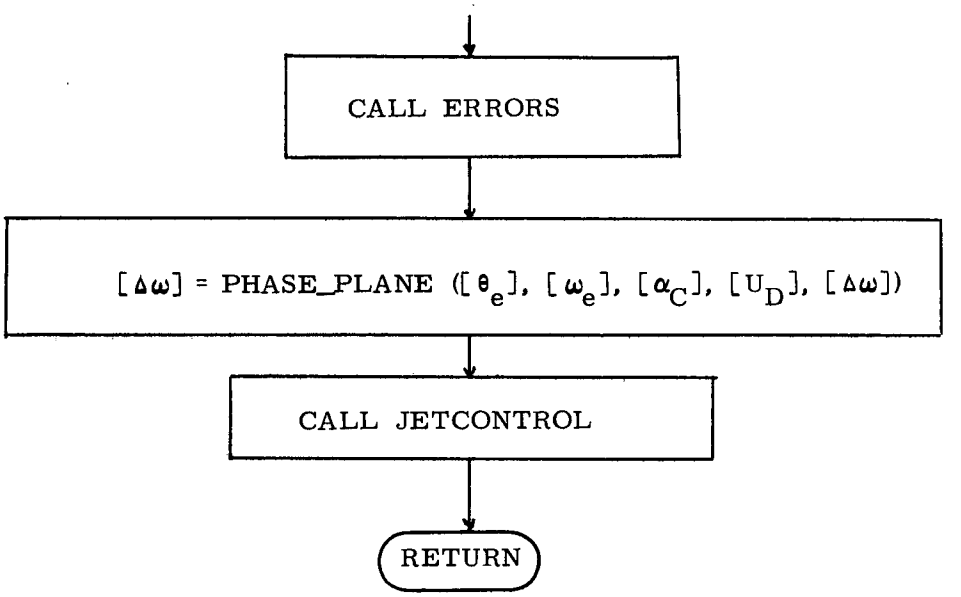



Fig. 6-12 On-orbit Segment of CONTROL Procedure

$$[\theta_D] = [\theta_D] \oplus [\Delta\theta_D]$$

$$[\theta_{GA_e}] = [\theta_{GA}] \ominus [\theta_D]$$

$$\begin{Bmatrix} \theta_{Be_1} \\ \theta_{Be_2} \\ \theta_{Be_3} \end{Bmatrix} = \begin{Bmatrix} 1 & \sin(\theta_{GA_m}) & 0 \\ 0 & \cos(\theta_{GA_m})\cos(\theta_{GA_o}) & \sin(\theta_{GA_o}) \\ 0 & -\cos(\theta_{GA_m})\sin(\theta_{GA_o}) & \cos(\theta_{GA_o}) \end{Bmatrix} \begin{Bmatrix} \theta_{GA_{e_1}} \\ \theta_{GA_{e_2}} \\ \theta_{GA_{e_3}} \end{Bmatrix}$$

$$\underline{TEMP1} = \overset{*}{M}_{PC}\ \underline{\theta}_{Be}$$

$$\underline{TEMP2} = \overset{*}{M}_{PC}\ (\underline{\omega} - \underline{\omega}_d)$$

$$\underline{TEMP3} = \overset{*}{M}_{PC}\ (\underline{\alpha}_d + \underline{\alpha}_m)$$

DO FOR i = 1, 2, 3

$$\theta_{e_i} = TEMP1_i + \beta_{BIAS_i}$$

$$\omega_{e_i} = TEMP2_i$$

$$U_{D_i} = TEMP3_i$$

RETURN

Fig. 6-13 ERRORS

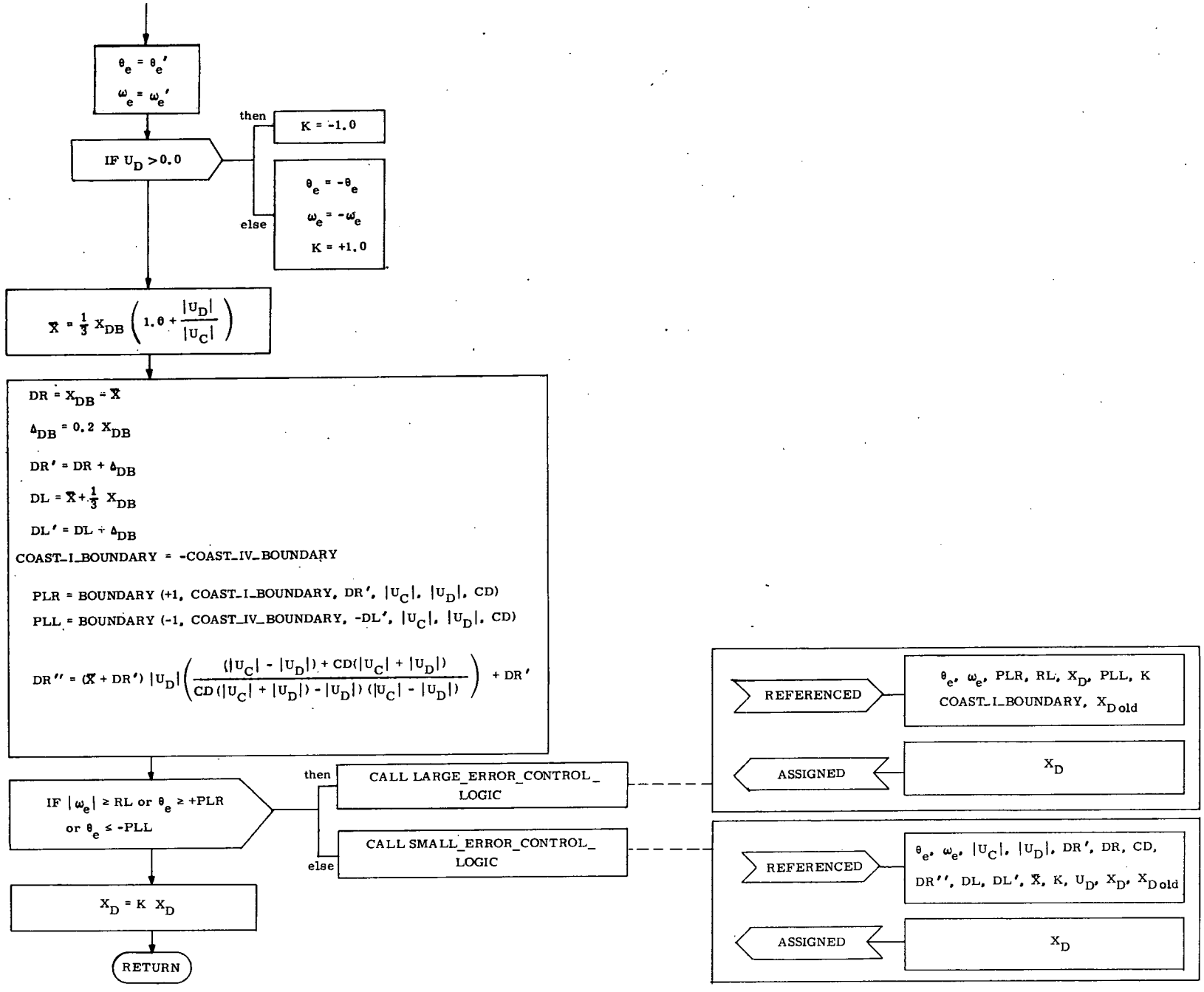



Fig. 6-14a PHASE_PLANE: FUNCTION ($\theta_e'$, $\omega_e'$, $|U_C|$, $U_D$, $X_{D\,old}$)

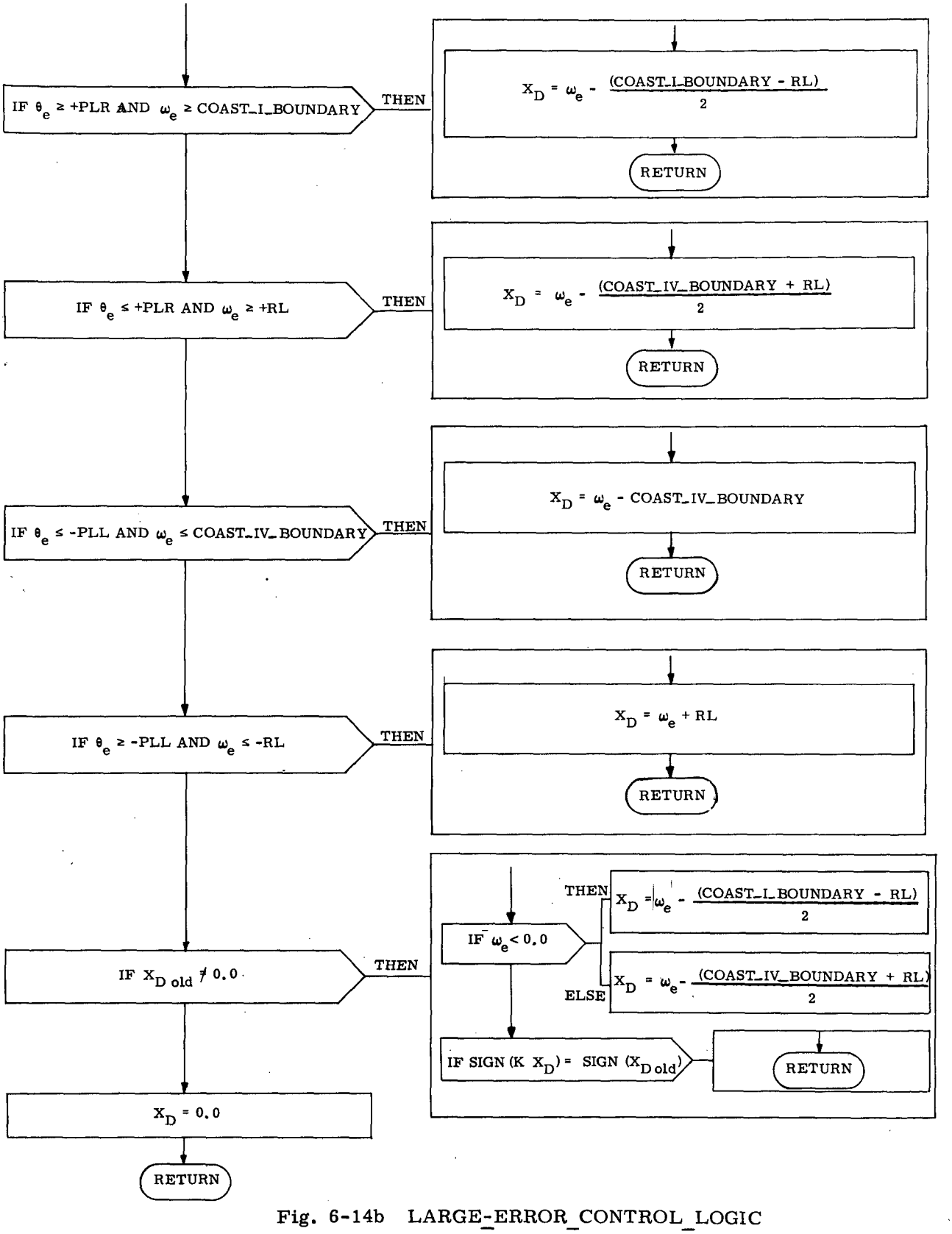



Fig. 6-14b LARGE_ERROR_CONTROL_LOGIC

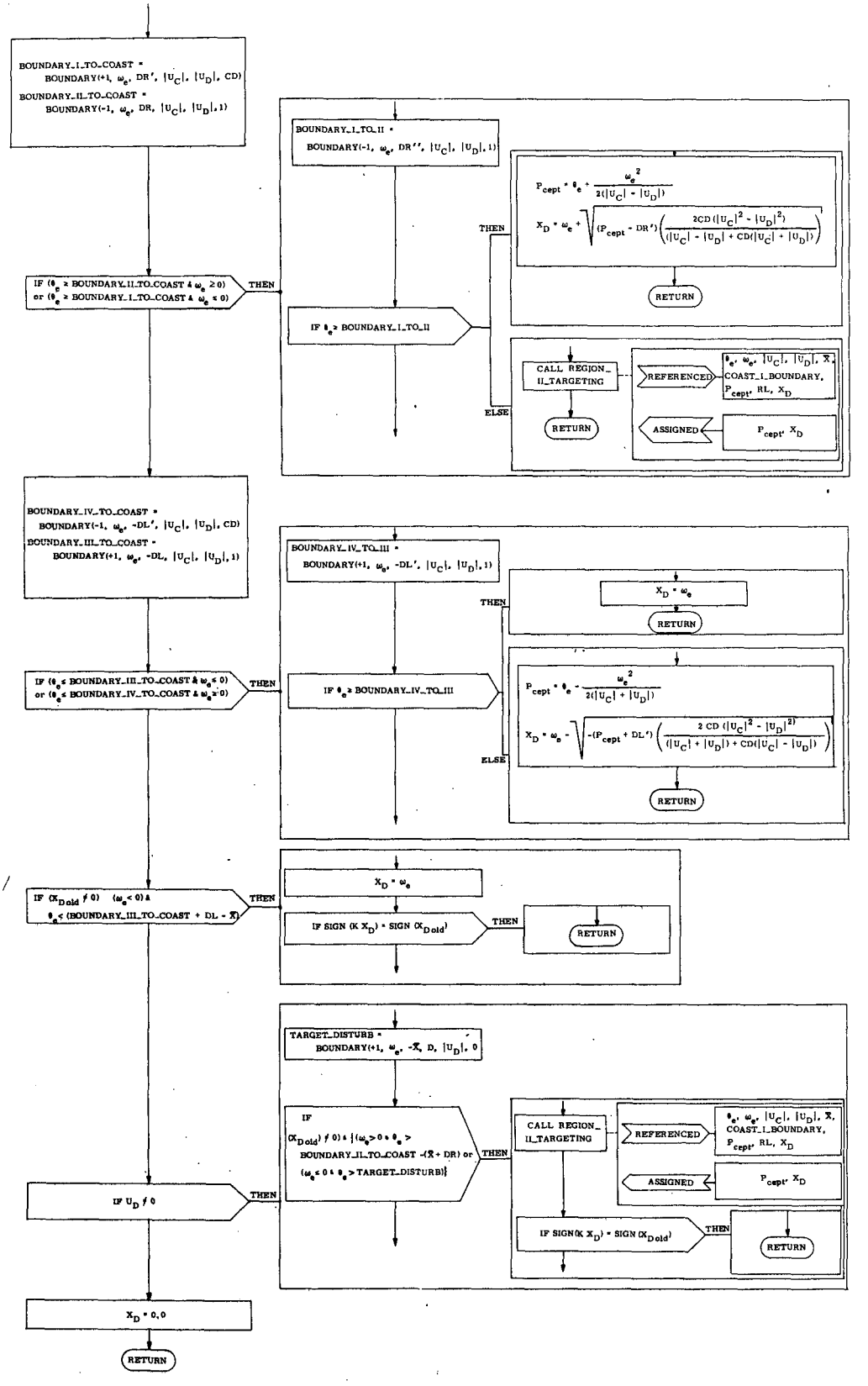



Fig. 6-14c SMALL-ERROR_CONTROL_LOGIC

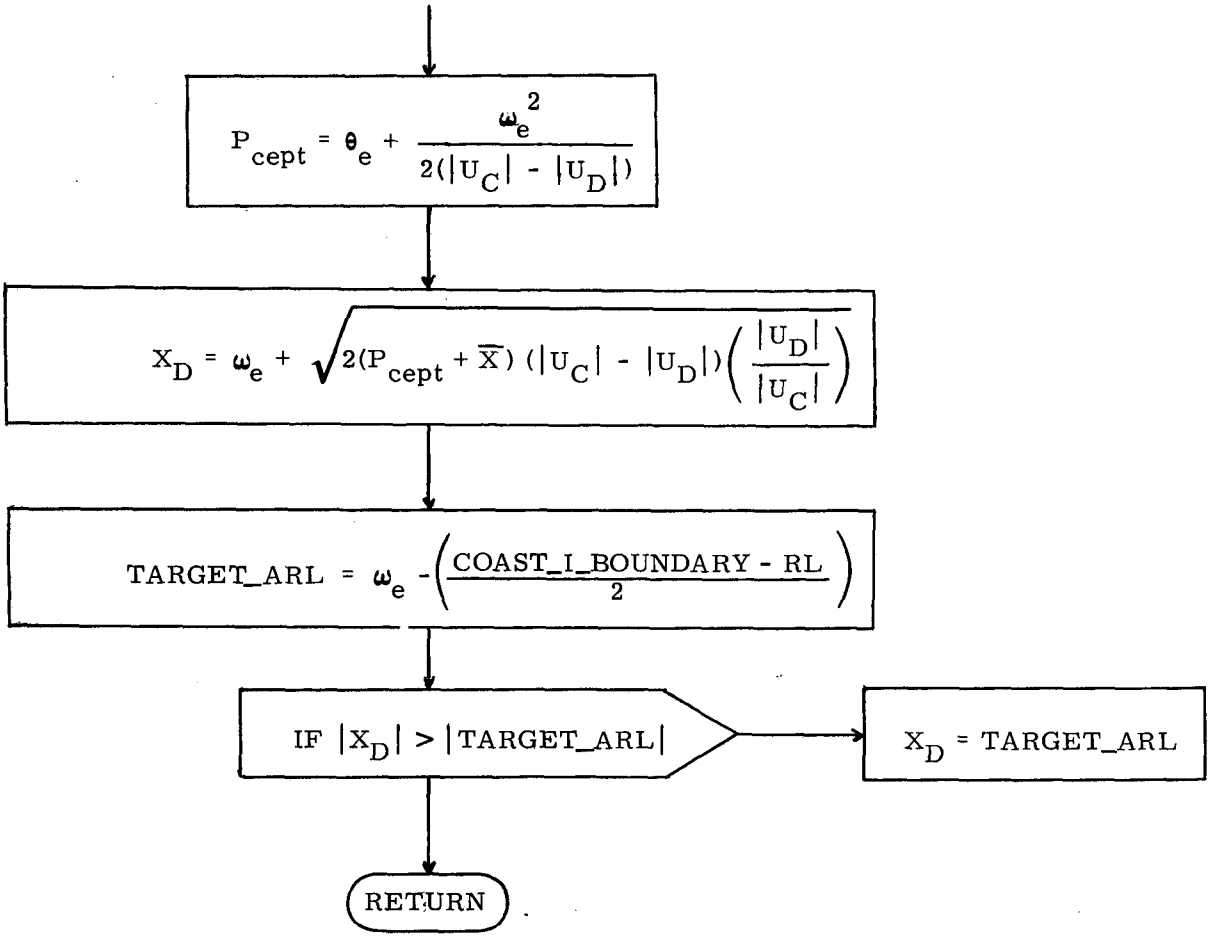



Fig. 6-14d. REGION_II_TARGETTING

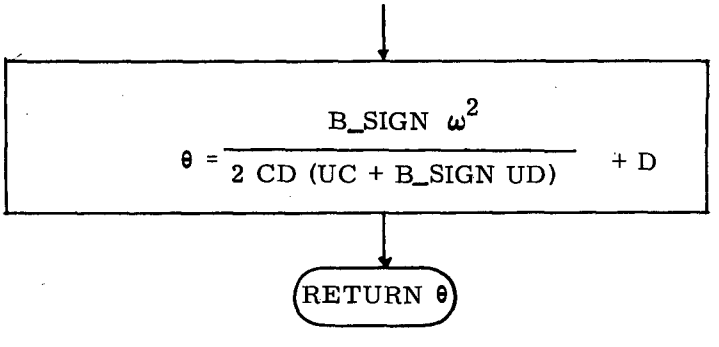



Fig. 6-14e BOUNDARY: FUNCTION (B_SIGN, , D, UC, UD, CD)

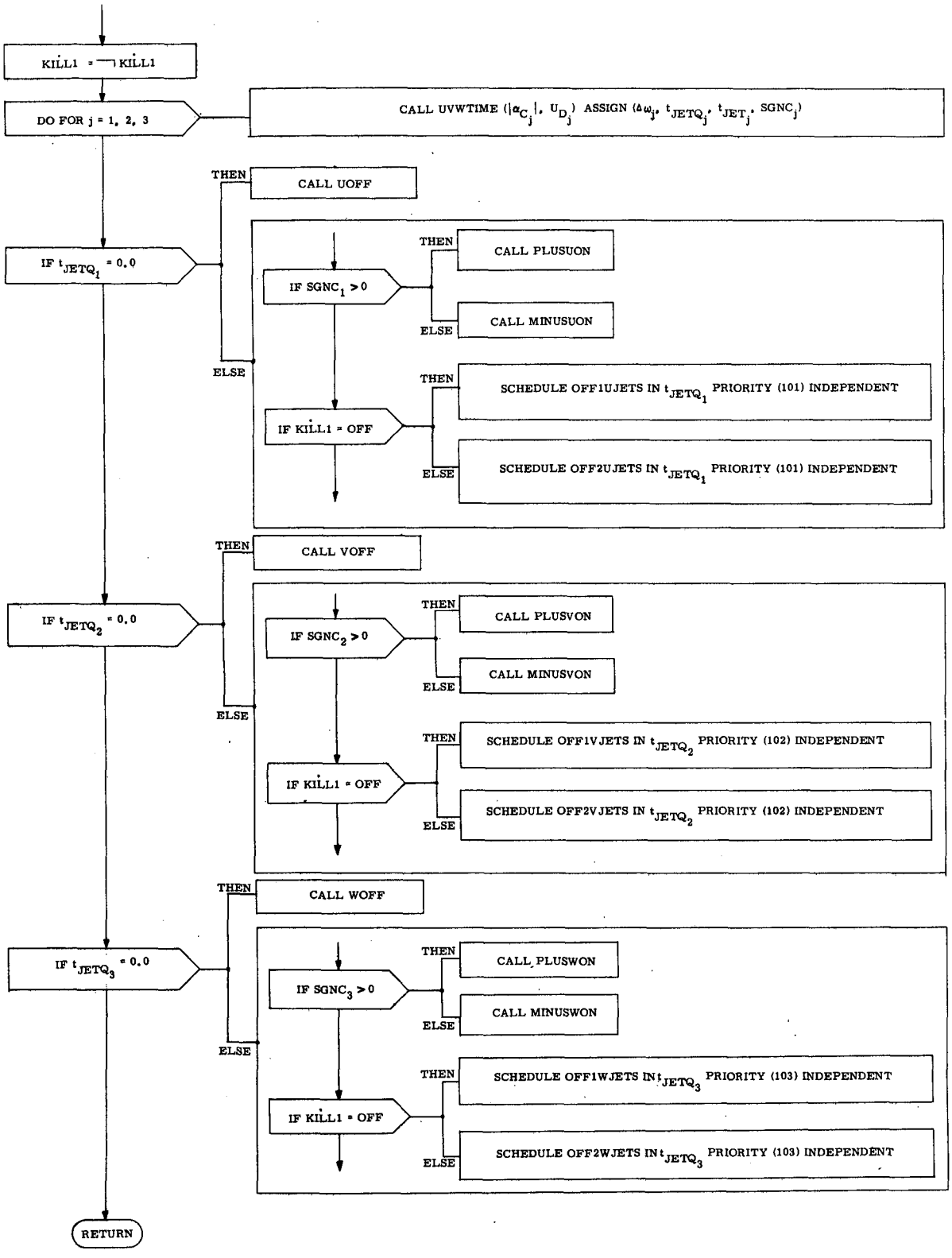



Fig. 6-15a JET_CONTROL Procedure

Inputs: $\alpha_C$, $U_D$

Outputs: $\Delta\omega$, $t_{JETQ}$, $t_{JET}$, SGNC

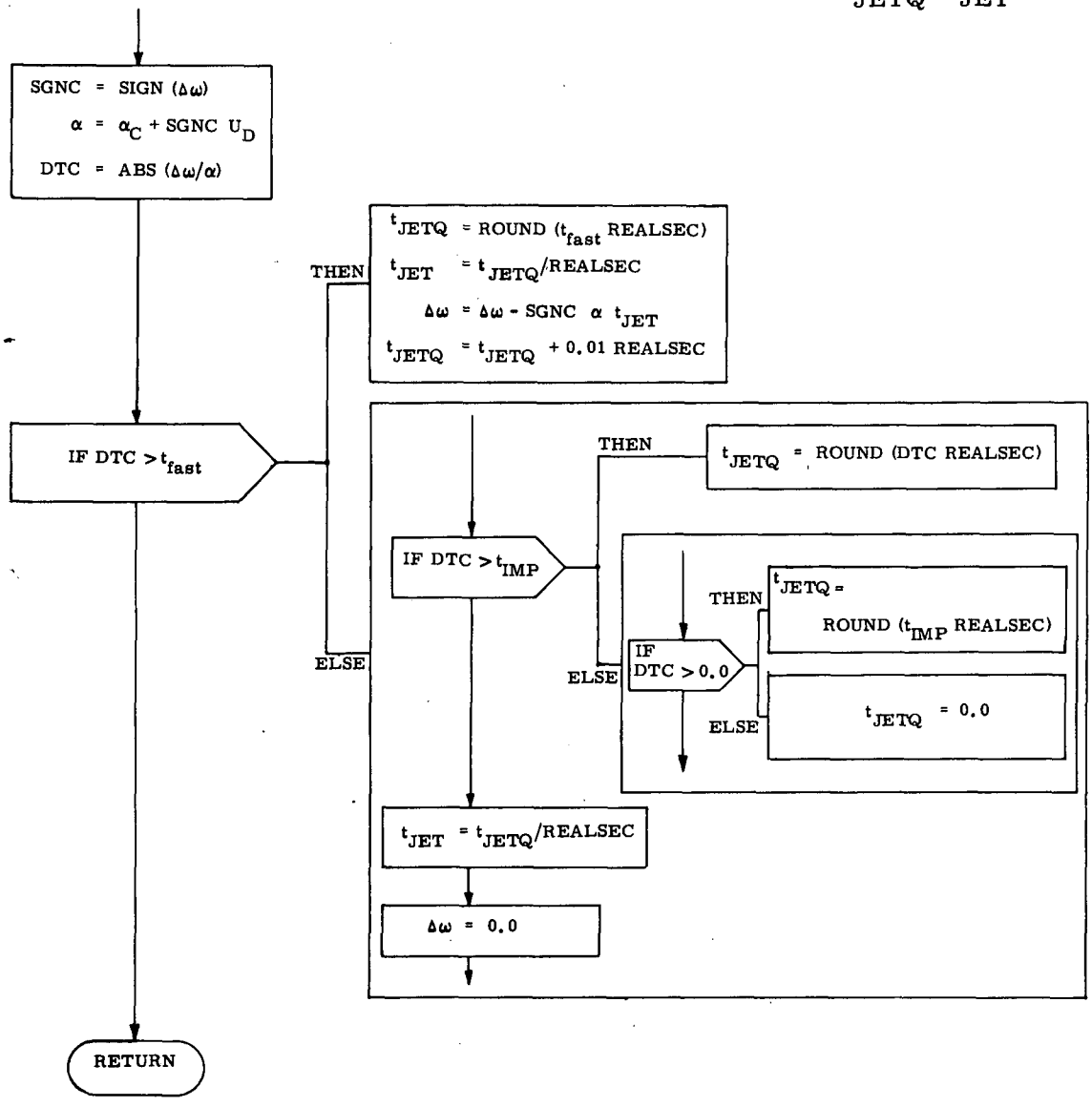



Fig. 6-15b UVWTIME Procedure

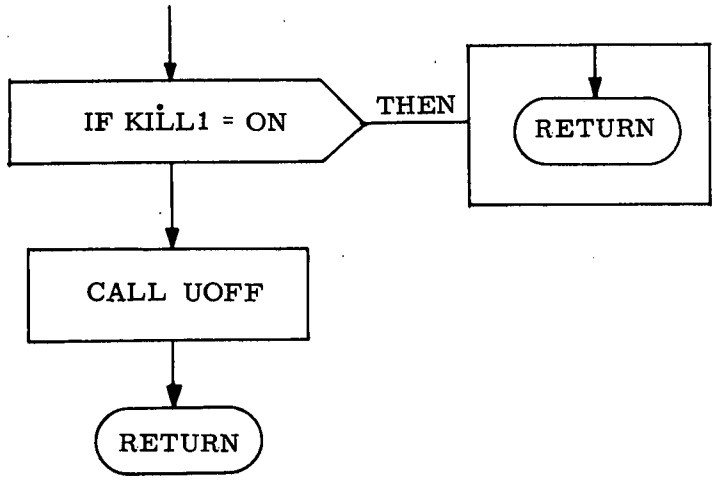



Fig. 6-15c OFF1JETS Procedure

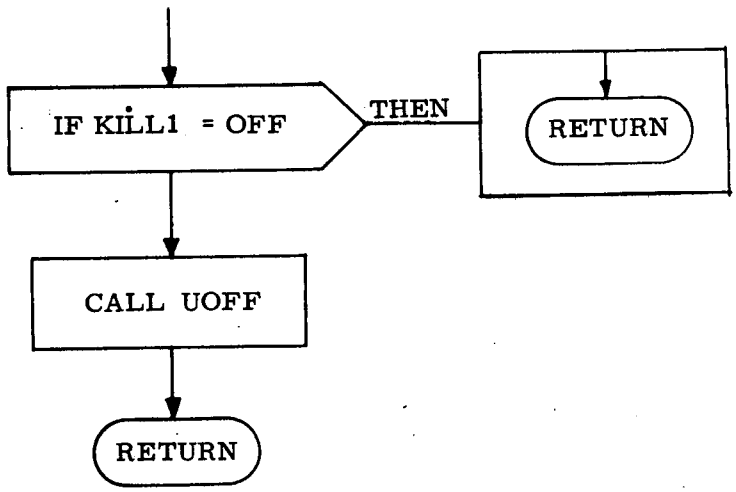



Fig. 6-15d OFF2UJETS Procedure

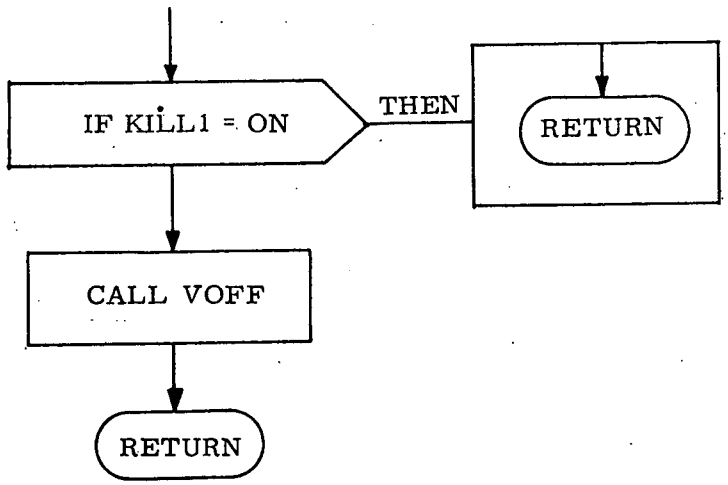



Fig. 6-15e OFF1VJETS Procedure

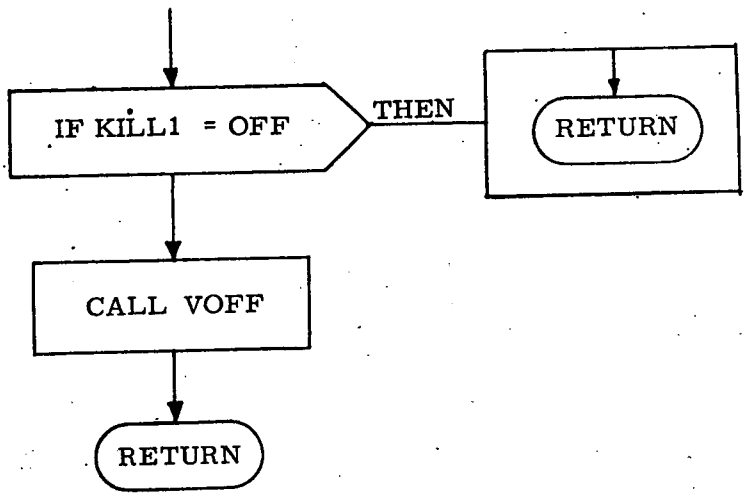



Fig. 6-15f OFF2VJETS Procedure

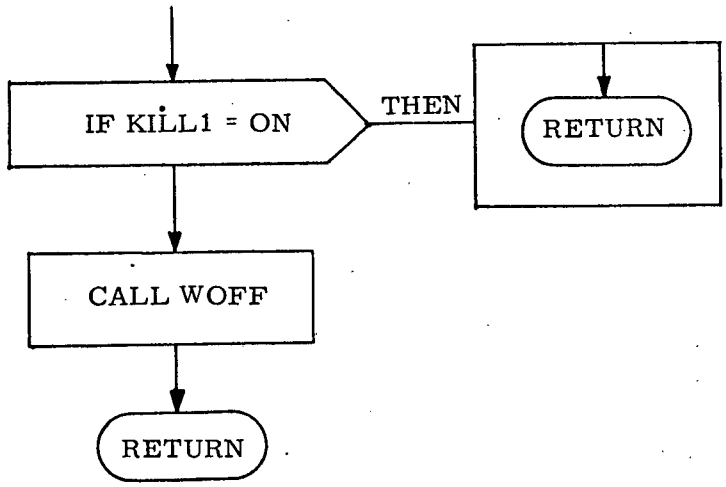



Fig. 6-15g OFF1WJETS Procedure

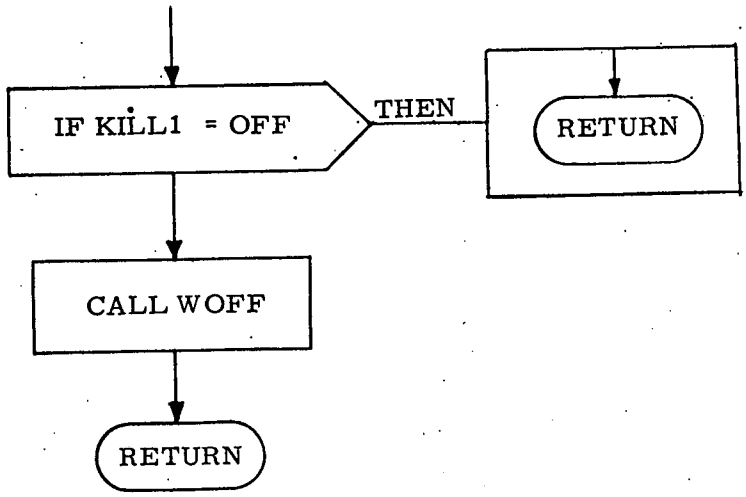



Fig. 6-15h OFF2WJETS Procedure

After CONTROL has called JETCONTROL to implement the firing of various jets, the estimate of the state is extrapolated for the next pass of the DFCS by the procedure PART2_FILTER (see Fig. 6-16). Finally, PARAM is called on passes other than initialization passes, and the DFCS pass is terminated.

$$\Delta \underline{\omega}_u = SGNC_1 \; \underline{\alpha}_{CU} \; t_{JET_1}$$

$$\Delta \underline{\omega}_v = SGNC_2 \; \underline{\alpha}_{CV} \; t_{JET_2}$$

$$\Delta \underline{\omega}_w = SGNC_3 \; \underline{\alpha}_{CW} \; t_{JET_3}$$

$$\Delta \underline{\omega}_{dm} = (\underline{\alpha}_d + \underline{\alpha}_m) \, t_{fast}$$

$$\hat{\underline{\theta}}'_B = \hat{\underline{\theta}}_B + \left( \hat{\underline{\omega}} \, t_{fast} + \Delta \underline{\omega}_d \frac{t_{fast}}{2} + \Delta \underline{\omega}_u \left( t_{fast} - \frac{t_{JET_1}}{2} \right) + \Delta \underline{\omega}_v \left( t_{fast} - \frac{t_{JET_2}}{2} \right) + \Delta \underline{\omega}_w \left( t_{fast} - \frac{t_{JET_3}}{2} \right) \right)$$

$$\hat{\underline{\omega}}' = \hat{\underline{\omega}} + \Delta \underline{\omega}_u + \Delta \underline{\omega}_v + \Delta \underline{\omega}_w + \Delta \underline{\omega}_{dm}$$

$$\hat{\underline{\alpha}}' = \hat{\underline{\alpha}}$$

RETURN

Fig. 6-16 ORBIT_ACPS_FILTER2; On-orbit Segment of PART2_FILTER Procedure

Table 6-5 Glossary for On-Orbit Phase

| MNEMONIC | ENGINEERING TERM | DEFINITION |
|---|---|---|
| $AC\overline{C}2U$ | $\underline{\alpha}_{CU}$ | Angular acceleration about u-axis produced by $\underline{M}_{P2U}$ or $\underline{M}_{M2U}$ |
| $AC\overline{C}2V$ | $\underline{\alpha}_{CV}$ | Angular acceleration about v-axis produced by $\underline{M}_{P2V}$ or $\underline{M}_{M2V}$ |
| $AC\overline{C}2W$ | $\underline{\alpha}_{CW}$ | Angular acceleration about w-axis produced by $\underline{M}_{P2W}$ or $\underline{M}_{M2W}$ |
| $ALPH\overline{A}DBAR$ | $\hat{\underline{\alpha}}'_d$ | Extrapolated disturbance acceleration |
| $ALPH\overline{A}DHAT$ | $\hat{\underline{\alpha}}_d$ | Estimated disturbance acceleration |
| $AL\overline{PH}AM$ | $\underline{\alpha}_m$ | Bias acceleration |
| $BAS\overline{IS}U$ | $\underline{u}_{BASIS}$ | Unit vectors along u, v, w control axes |
| $BAS\overline{IS}V$ | $\underline{v}_{BASIS}$ | |
| $BAS\overline{IS}W$ | $\underline{w}_{BASIS}$ | |
| [BIAS_BETA] | $[\beta_{BIAS}]$ | Bias body angles |
| CD | CD | Scale factor used in phase plane switching logic |
| [CDU] | $[\theta_{GA}]$ | Gimbal angles |
| [CDUERROR] | $[\theta_{GA_e}]$ | Error in gimbal angles from the reference desired |
| DELTA | $\Delta_{DB}$ | Deadband factor |
| DELTA_POSITION | $\theta_e'$ | Angular position error (within PHASE_PLANE) |
| DELTA_RATE | $\omega_e'$ | Angular rate error (within PHASE_PLANE) |
| [DELTA_THETA_D] | $[\Delta\theta_D]$ | Incremental desired reference gimbal angles |
| $DEL\overline{W}DM$ | $\Delta\underline{\omega}_{dm}$ | Change in angular velocity vector due to disturbance and bias accelerations |
| $DEL\overline{W}U$ | $\Delta\underline{\omega}_u$ | Change in angular rate due to u-control jets |
| $DEL\overline{W}V$ | $\Delta\underline{\omega}_v$ | Change in angular rate due to v-control jets |
| $DEL\overline{W}W$ | $\Delta\underline{\omega}_w$ | Change in angular rate due to w-control jets |
| DL, DL_PRIME1, | DL, DL′ | Switching line angular position error intercepts |
| DR, DR_PRIME1, | DR, DR′ | |
| DR_PRIME2 | DR″ | |
| $[\overline{DRATE}]$ | $[\Delta\omega]$ | Components of changes of angular rates |

Table 6-5 Glossary for On-Orbit Phase (Cont)

| MNEMONIC | ENGINEERING TERM | DEFINITION |
|---|---|---|
| DTC | DTC | Total firing time of an ACPS control jet for a maneuver |
| ISTRTR | ISTRTR | Restart required flag (external) |
| PHASER | PHASER | Mission phase flag (external) |
| [FAST] | [FAST] | Fast, medium, and slow sampling periods for each mission phase |
| [MEDIUM] | [MEDIUM] | |
| [SLOW] | [SLOW] | Slow sampling period for each mission phase |
| GAIR | $G_{AIR}$ | Filter gain for air data |
| INERTIA | $\overset{*}{J}$ | Inertia matrix |
| IM | IM | medium rate sampling index |
| IS | IS | Slow rate sampling index |
| ISTART | ISTART | Initial pass flag for mission phase |
| ITM | ITM | Ratio of medium to fast sampling periods |
| ITS | ITS | Ratio of slow to fast sampling periods |
| [JTORQUE] | [JTORQUE] | Array of ACPS jet torque vectors |
| K | K | Sign of rate change |
| KALPHA | $K_{\alpha}$ | Filter gains |
| KOMEGA | $K_{\omega}$ | |
| KTHETA | $K_{\theta}$ | |
| KILL1 | KILL1 | ACPS jet continuity indicator |
| [MAGACC] | $[\alpha_C]$ | Array of magnitudes of control accelerations |
| MAG_U_D | $\lvert U_D \rvert$ | Magnitude of disturbance acceleration |
| MAG_U_C | $\lvert U_C \rvert$ | Magnitude of control acceleration |
| MP2U | $\underline{M}_{P2U}$ | Torque in the +u direction by 2 u-jets |
| MM2U | $\underline{M}_{M2U}$ | Torque in the -u direction by 2 u-jets |
| MP2V | $\underline{M}_{P2V}$ | Torque in the +v direction by 2 v-jets |
| MM2V | $\underline{M}_{M2V}$ | Torque in the -v direction by 2 v-jets |

Table 6-5 Glossary for On-Orbit Phase (Cont)

| MNEMONIC | ENGINEERING TERM | DEFINITION |
|---|---|---|
| MP2W | $\underline{M}_{P2W}$ | Torque in the +w direction by 2 w-jets |
| MM2W | $\underline{M}_{M2W}$ | Torque in the -w direction by 2 w-jets |
| MCP | $\overset{*}{M}_{CP}$ | Control to pilot transformation matrix |
| MPC | $\overset{*}{M}_{PC}$ | Pilot to control transformation matrix |
| MINIMP | $t_{IMP}$ | Time of a minimum impulse ACPS jet firing |
| NETACC | $[\alpha]$ | Net acceleration |
| OMEGABAR | $\hat{\underline{\omega}}'$ | Extrapolated angular velocity |
| OMEGAHAT | $\hat{\underline{\omega}}$ | Estimated angular velocity |
| -PLL | -PLL | Left angular position error limit |
| +PLR | +PLR | Right angular position error limit |
| P_CEPT | $P_{cept}$ | Intermediate position error axis intercept |
| PHASE | PHASE | Phase of the mission |
| ±RL | ±RL | Upper and lower rate error limits |
| POSITION_ERROR | $[\theta_e]$ | Angular position error |
| RATE_ERROR | $[\omega_e]$ | Angular rate error |
| REALSEC | REALSEC | Scale factor for scaling computer time $\left(= \frac{1}{\text{TICK_TO_SEC}}\right)$ |
| RUNTIME | RUNTIME | Runtime of the computer in computer ticks |
| [SGNC] | [SGNC] | Signs of the angular rate components |
| TARGET_ARL | TARGET_ARL | Lower rate limit average |
| TF | $t_{fast}$ | Fast sampling period |
| TM | $t_{medium}$ | Medium sampling period |
| TS | $t_{slow}$ | Slow sampling period |
| TEMP1 | TEMP1 | Temporary vector of body angle errors in control coordinates |
| TEMP2 | TEMP2 | Temporary vector of body rate errors in control coordinates |
| TEMP3 | TEMP3 | Temporary vector of acceleration in control coordinates |
| [TJET] | $t_{JET}$ | Time of an ACPS jet firing in seconds |

Table 6-5 Glossary for On-Orbit Phase (Cont)

| MNEMONIC | ENGINEERING TERM | DEFINITION |
|---|---|---|
| THETĀBAR | $\hat{\underline{\theta}}'_B$ | Extrapolated body angles |
| THETĀHAT | $\hat{\underline{\theta}}_B$ | Estimated body angles |
| TNOW | $t_n$ | Time of present gimbal angle measurement |
| [TJETQ] | $t_{JETQ}$ | Time of an ACPS jet firing scaled to the computer |
| RATĒ_INT | $\tilde{\underline{\theta}}_B$ | "Measured" body angles |
| [TLAST] | $t_{\ell_j}$ | Time of the last gimbal angle measurement |
| [THETA_D] | $[\theta_D]$ | Desired set of gimbal angles |
| TICK_TO_SEC | TICK_TO_SEC | Scale factor for scaling computer time |
| [U_D] | $[U_D]$ | Array of disturbance accelerations |
| W̄_D | $\underline{\omega}_d$ | Desired angular velocity |
| X_BAR | $\overline{X}$ | Average angular position error |
| X_DB | $X_{DB}$ | Deadband of limit cycle |
| X_D | $X_D$ | One component of $\Delta\underline{\omega}$ the desired change in angular velocity |
| NO_CHANGE | $X_{D\,old}$ | One component of the last desired change in angular velocity |
| XYZĒRROR | $\underline{\theta}_{Be}$ | Errors in x, y, z axis rotation angles |

NOTATION CONVENTIONS

| | |
|---|---|
| $\oplus$ | Modular addition |
| $\ominus$ | Modular subtraction |
| ~ | Measured quantity |
| ^ | Estimated quantity |
| ′ | Extrapolated quantity |
| _ | Vector |
| * | Matrix |
| [ ] | Array |

## REFERENCES FOR SECTION 6


1. Keene, D., "Control Axes for the Shuttle On-Orbit Autopilot", SSV MEMO No. 73-23C-6, April 1973.

2. Keene, D., "State Estimator for the Shuttle On-Orbit Autopilot", SSV MEMO No. 73-23C-7, April 1973.

3. Keene, D. and Kubiac, E., Section 6, "On-Orbit Phase", Integrated Digital Flight Control System for the Space Shuttle Orbiter, E-2736, March 1973.

# SECTION 7

# BOOST PHASE

by

Bernard Marcantel (JSC/EJ3)

## 7.1 Introduction

The baseline launch configuration control system software described in this section satisfies the requirement to control the booster from lift-off through main engine cutoff at external tank separation. The system is adaptable to various control laws, including attitude control, drift minimum control (with and without accelerometer feedback) and/or load relief control through proper choice of gain schedules. It also contains the option of using rudder and thrust vector control combined in the roll channel.

The following paragraphs contain: A description of the control system with functional block diagrams of the pitch, yaw, and roll channels; an annotated flow diagram; and a discussion of the control gains.

## 7.2 Functional Description

The baseline launch control system is a digital autopilot receiving body attitude from an inertial reference system, body rates from three sets of body mounted rate gyros, and translational accelerations from two sets of body mounted accelerometers. The current baseline includes rate gyros at the aft end of each solid rocket booster and the orbiter. Normal and lateral accelerometers are located in the forward avionics bay and the aft end of the orbiter. The autopilot also utilizes prestored engine deflection commands and acceleration commands. The former are the nominal trim values for the expected center-of-gravity and aerodynamic histories, and the latter are the normal linear accelerations in the nominal trajectory. [7-1] The final inputs to the control loop are the attitude commands, which may either be prestored or come from a guidance system.

From the inputs, the control system determines and sends out deflection commands for the engines and the rudder. A general optimal mixing logic has been designed and may be installed at a later date. It has been designed to distribute the desired torques efficiently among the control actuators while accounting for failures and saturation. [7-2]

Table 7-1 summarizes the control system inputs and outputs.

Table 7-1 Input/Output Summary

INPUTS

- Inertial attitude angles
- Body rotational rates
- Y & Z translational accelerations
- Prestored engine deflection commands
- Prestored accelerations commands
- Attitude commands from guidance system

OUTPUTS

- Deflection commands to orbiter engines
- Deflection commands to SRM engines
- Deflection commands to aerosurfaces (rudder)

Variable control gains have been calculated to satisfy the rigid body response requirements. They are stored as a function of time from lift-off; the numerical values are presented in Subsection 7.4. In the future, the design will be extended to include consideration of slosh and bending, and these prestored gains will be replaced by filters. All filters will be implemented as second order over second order. Filter inclusion in two places in each sensor loop recognizes common filtering requirements and reduces the order of the filter.

Some aspects of the design presented in this section are specific to the first stage of boost (prior to SRB separation). However, control of the orbiter during the second stage will utilize basically the same control law. The major differences are that the rudder will not be used and there will be no acceleration feedback in the second phase. Furthermore, there will only be one set of body mounted rate gyros available. These gyros will be located in the aft section of the orbiter. All filter coefficients and gains will be changed at separation and vary as the mass properties, flexible body and fuel slosh characteristics vary.

### 7.2.1 Roll Channel Description

Referring to Fig. 7-1, the block in the lower right hand corner labeled "Vehicle and Subsystem Dynamics" includes actuator and sensor dynamics, possible analog prefiltering of sensor outputs, input/output voting, redundancy management, A/D and D/A converters. Inputs to the control system from "Vehicle and Subsystem Dynamics" are the inertial platform attitude angle, and rate measurements from three sets of body mounted rate gyros. Outputs from the control system to the vehicle are deflection commands for the engines and the rudder.

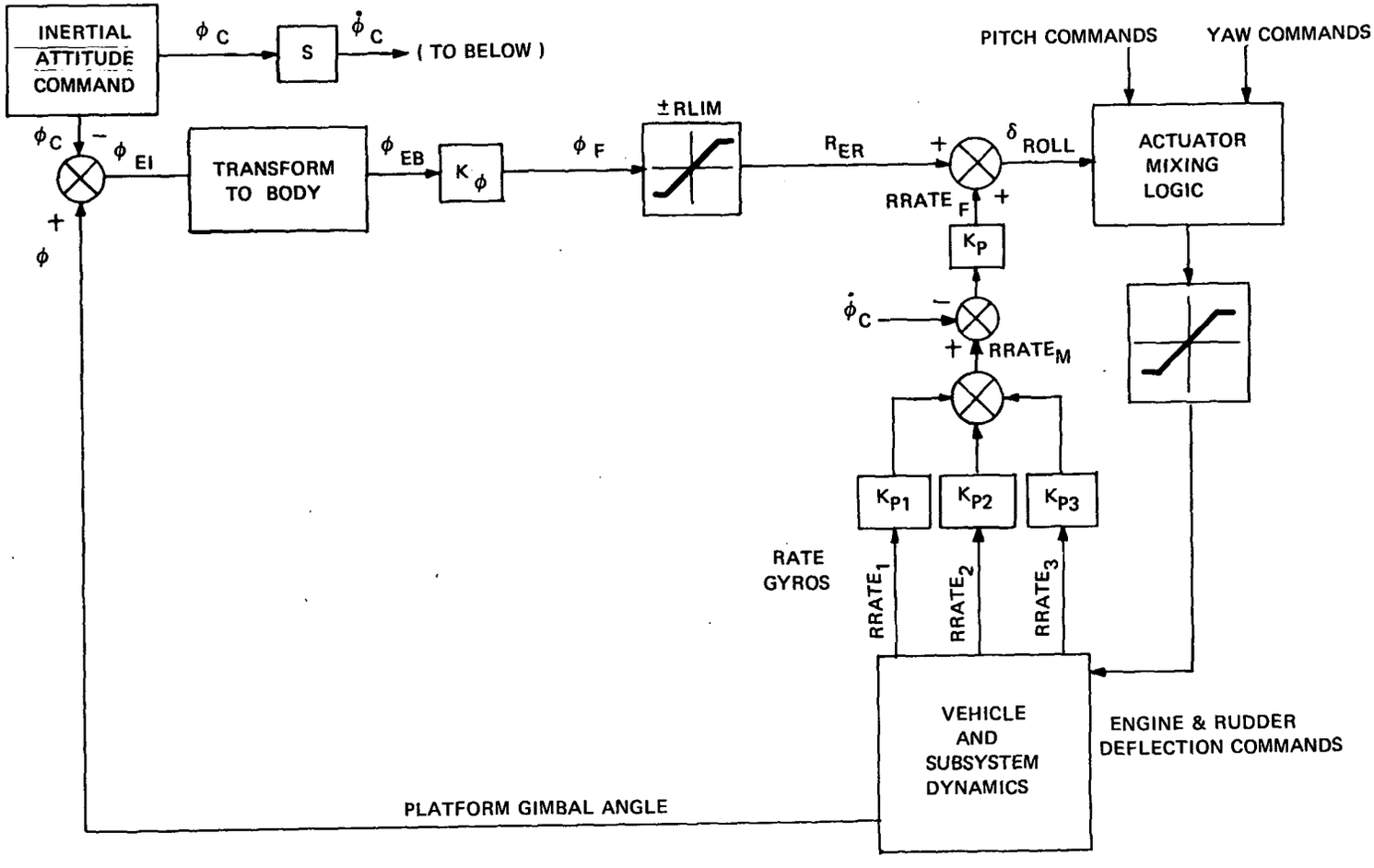



Fig. 7-1 Baseline Launch Configuration Control System Software Roll Channel

Following the attitude loop in Fig. 7-1, the platform gimbal angle, $\phi$, is differenced with the inertial attitude command, $\phi_C$, to produce the inertial attitude error, $\phi_{EI}$. This inertial attitude error is transformed to a body roll error, $\phi_{EB}$, by adding the inner gimbal error times the sine of the middle gimbal angle ($\phi_{EB} = \phi_{EI} + \theta_{EI} \sin \psi$).

It should be noted that a limiter has been included in the attitude error loop. This limiter reserves some TVC authority for rate damping (even though the attitude error signal may be saturated) thus preventing oscillation and complete loss of control because of large signal instability. The limiter is set to a value less than eight degrees.

Returning to the "Vehicle and Subsystem Dynamics" block in Fig. 7-1, three sets of roll rate gyro signals are filtered then blended and again scaled and filtered prior to being summed with the limited attitude error signals. The resulting signal is sent to the rudder and engine mixing logic for distribution to the individual engine and rudder actuators. Then each actuator command is limited to eight degrees to prevent the roll channel from usurping all pitch and yaw controllability. The rudder command is limited to 15 degrees.

### 7.2.2 Yaw Channel Description

The yaw channel block diagram in Fig. 7-2 is laid out in the same manner as the roll channel with the "Vehicle and Subsystem Dynamics" block in the lower right hand corner, the "Inertial Attitude Command" in the upper left hand corner and the "Actuator Mixing Logic" on the right. The rudder is not used for yaw control, because there is adequate engine control authority to handle yaw-plane torque.

Three features which were not included in the roll channel appear in the yaw channel. Two accelerometer inputs have been added to the attitude loop, a forward loop integrator follows the limiter, and a rate command signal is differenced with the input from the rate gyros.

The platform gimbal angle, $\psi$, is differenced with the inertial attitude command, $\psi_C$, transformed to body attitude error($\psi_{EB} = \psi_{EI} \cos \phi - \theta_{EI} \sin \phi \cos \psi$) and scaled. The attitude error signal is then modified by summing with an acceleration error signal and limiting. The acceleration error is obtained by blending appropriately filtered signals from two body mounted accelerometers and then applying gain and filtering compensation according to the control law or bending requirements. The accelerometer gain is varied with time to meet the drift minimum or load relief requirements. The forward loop integrator trims out steady state errors resulting from biases and thrust misalignments.

The rate command signal is introduced because the normal attitude command changes at a predictable rate. The rate command signal minimizes the tendency for vehicle attitude to lag behind the attitude command due to rate feedback. If the

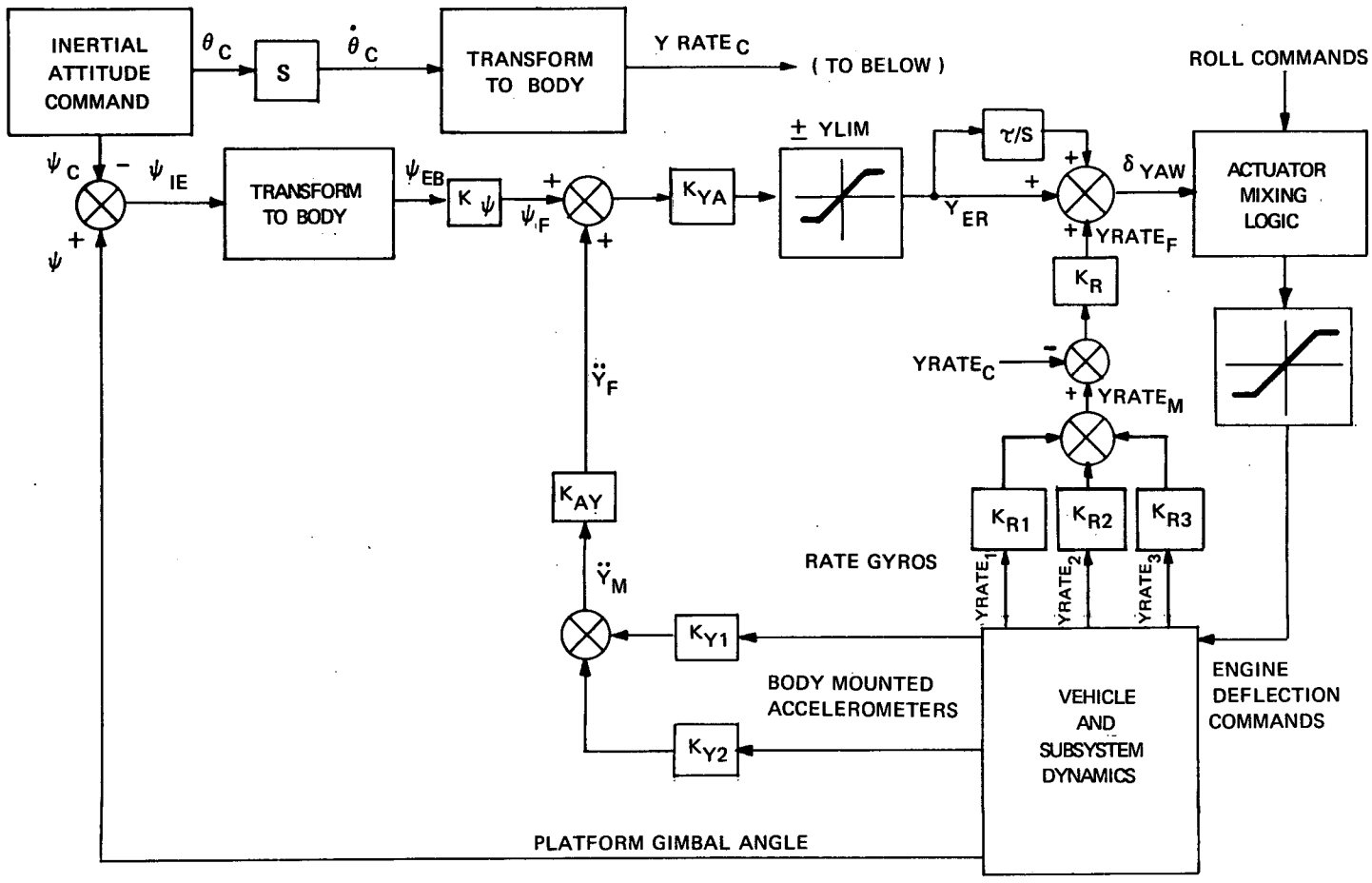



Fig. 7-2 Baseline Launch Configuration Control System Yaw Channel

vehicle rolls appreciably the rate command must be applied to the yaw channel. In operation, the attitude command is differentiated, multiplied by the sine of $\phi$, and differenced with the rate feedback signal. The rate error signal is then added to the attitude error signal and sent to the actuator mixing logic routine. Actuator command limiting is performed for the same reason stated for roll control.

### 7.2.3 Pitch Channel Description

Comparison of Fig. 7-3, which is the pitch channel block diagram, with Fig. 7-2 (yaw channel) reveals that the only difference is the addition of prestored engine deflection and acceleration commands. Both of these command schedules are the nominal values from a reference trajectory.

The addition of the prestored acceleration command permits use of accelerometer feedback with any feasible trajectory to modify the attitude error signal to conform to the desired control law.

The prestored engine deflection command schedule provides coarse control of the engine gimbal trim in the pitch plane. Center of gravity travel alone could be handled by the integral compensation, but aerodynamic pitching moment increases sharply in the vicinity of Mach one and decreases again thereafter. A high integrator gain would be required to follow these rapidly changing deflection requirements. This is undesirable because it degrades the rigid body stability margin; therefore, the prestored commands are used to control the nominal part of this variation.

The rest of the pitch channel is the same as the yaw channel. All engine commands are sent to the actuator mixing logic routine.

## 7.3 Logic Flow Description

A flow chart showing the logic flow of the control system is presented in Fig. 7-4.

## 7.4 Values of Control Gains

The basic control system is independent of configuration, assuming that thrust vector control is primary for pitch and yaw, and that the principal guidance commands will be in the pitch plane. It includes no provision for use of elevons during boost.

The control gain and command schedules are, of course, configuration dependent. They will be affected by modifications to either the vehicle configuration or the reference trajectory. This will not impact the design, however, because all control gains and commands are continuously variable input functions of time. Some of them may subsequently be modified to functions of other trajectory variables, such as altitude and velocity, but this will involve only minor software modifications.

Table 7-2 presents the current numerical values of the control gains.

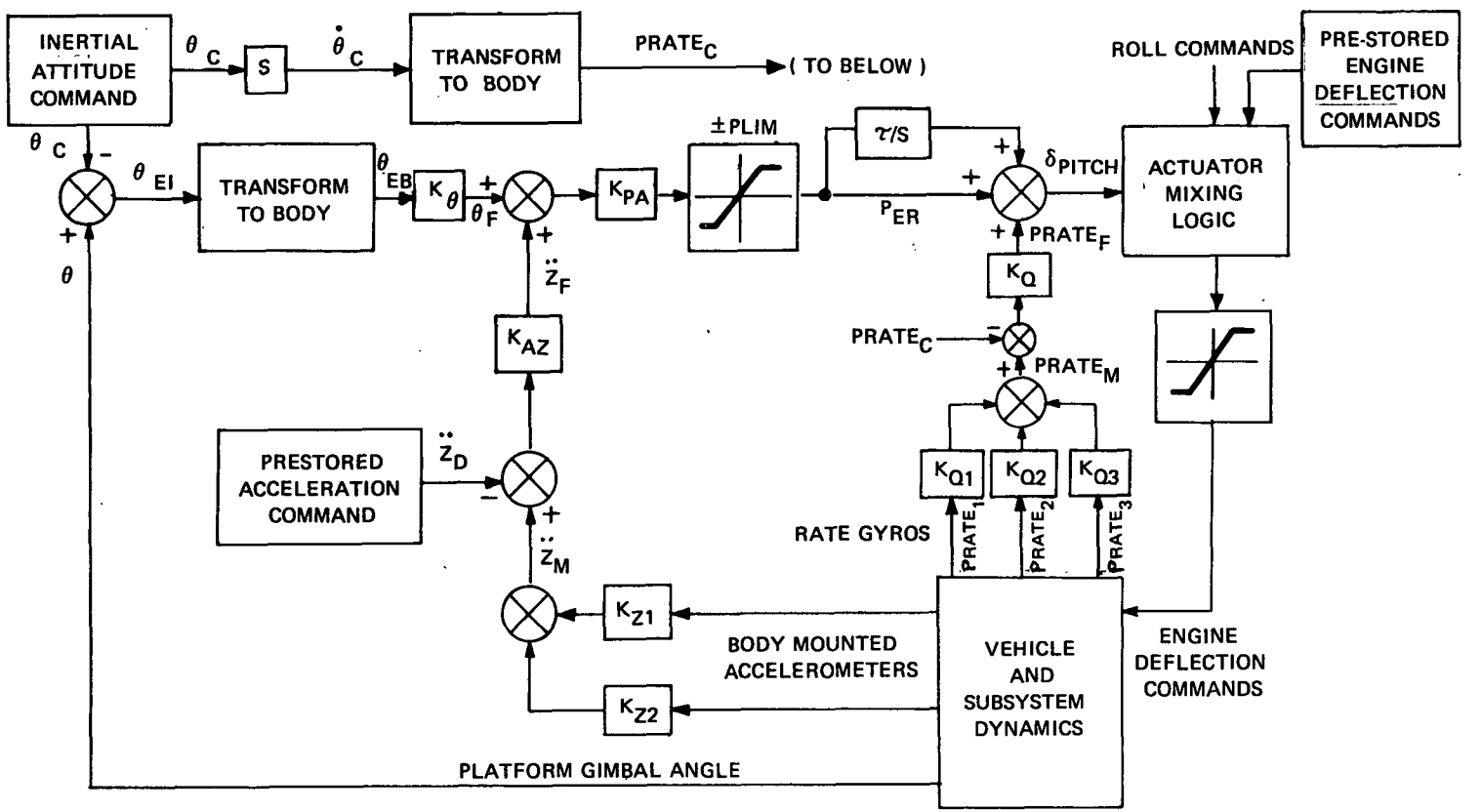



Fig. 7-3 Baseline Launch Configuration Control System Software Pitch Channel

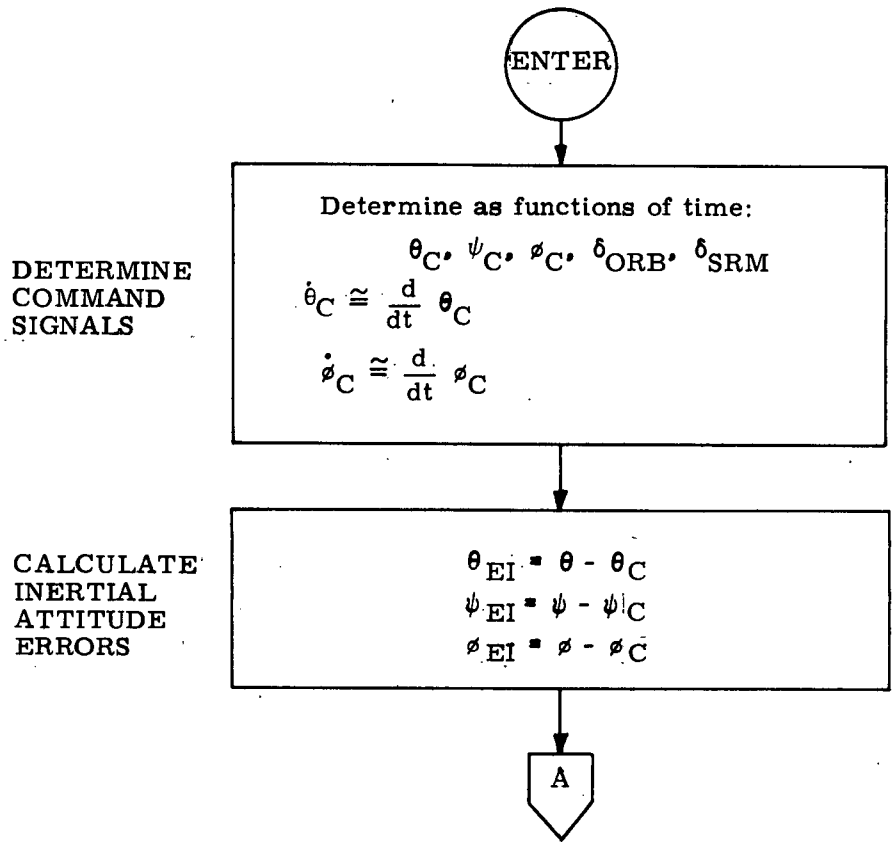



Fig. 7-4 Logic Flow

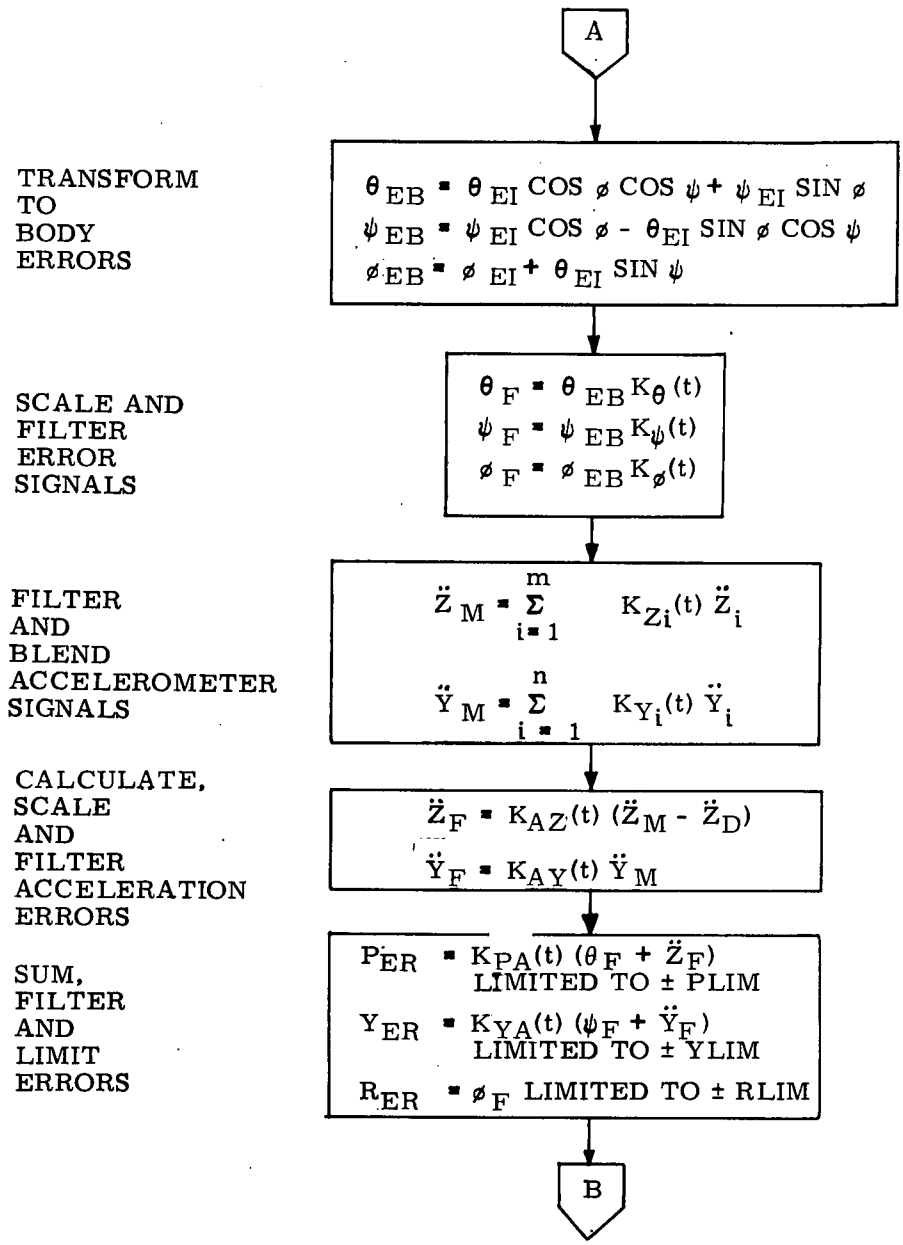



Fig. 7-4 Logic Flow (Cont)

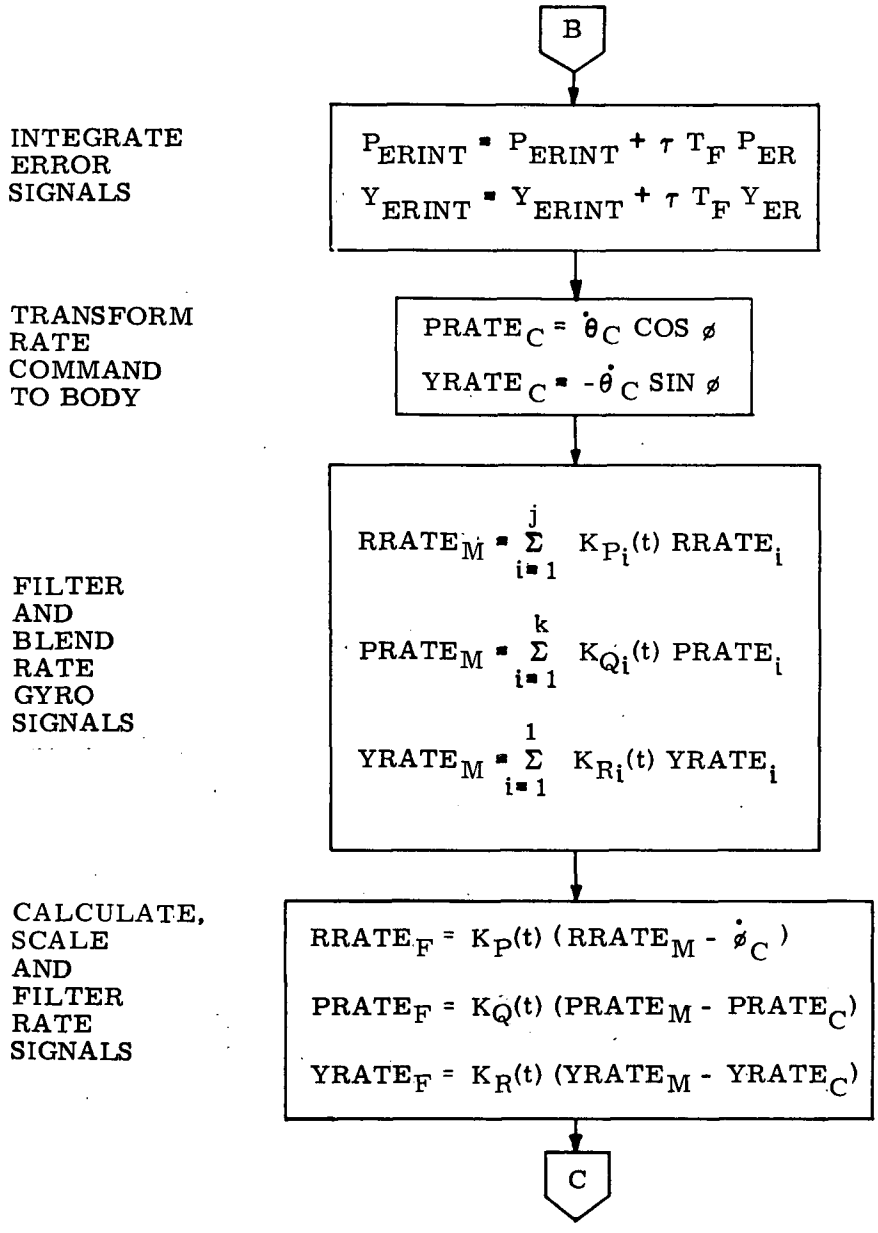



Fig. 7-4 Logic Flow (Cont)

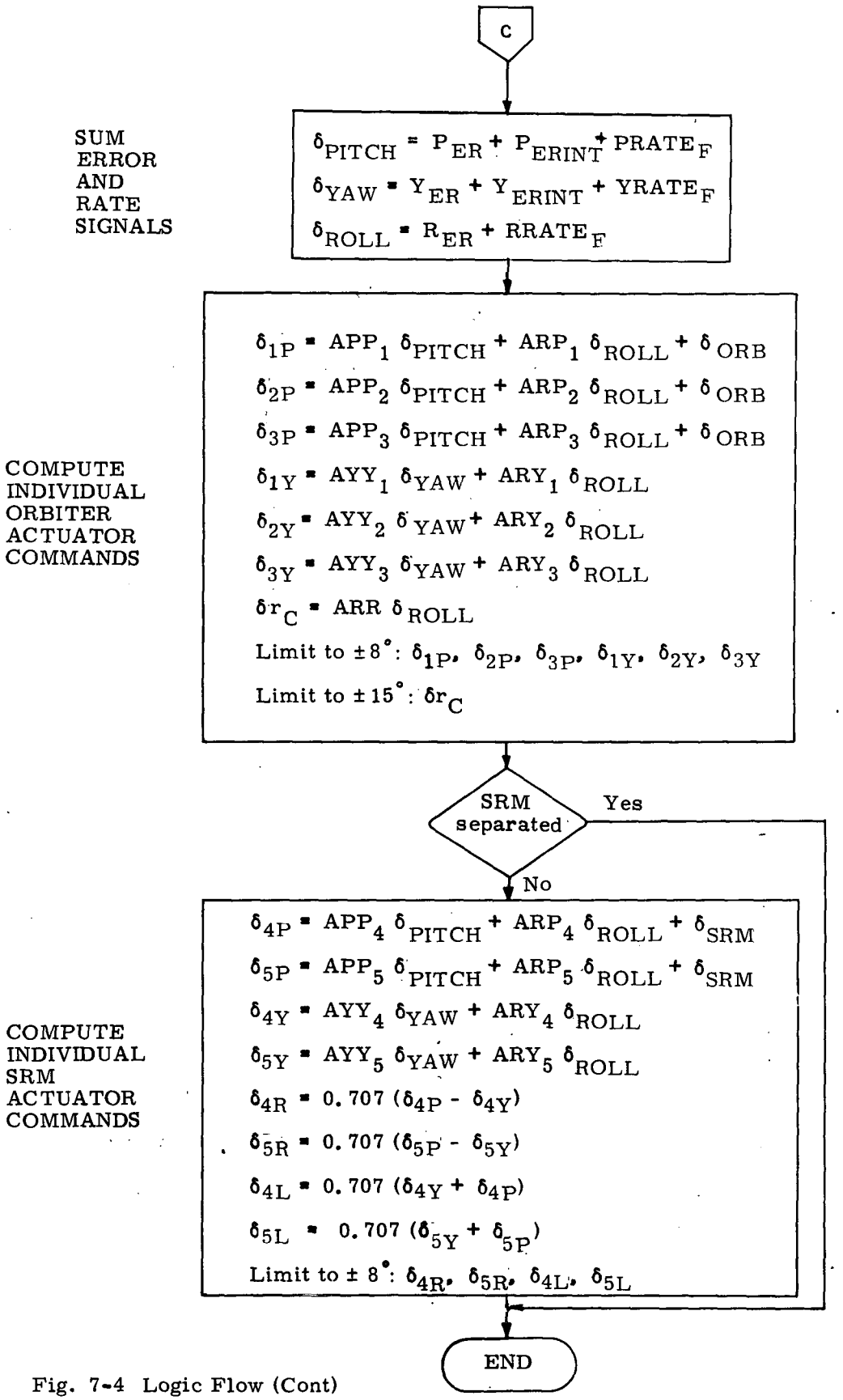



Fig. 7-4 Logic Flow (Cont)

Table 7-2 Gains for Boost Phase

Control Gains vs Time (sec)

| | 0. | 25 | 30 | 75 | 95 | 120 | 150 |
|---|---|---|---|---|---|---|---|
| $K_\theta$ | 1.1 | 1.1 | .88 | .88 | .66 | .66 | .66 |
| $K_\psi$ | 1.4 | 1.4 | .56 | .56 | 1.4 | | 1.4 |
| $K_\phi$ | .707 | | | .707 | .424 | | .424 |
| $K_{AZ}$ | 0 | 0 | -.109 | -.109 | 0 | | 0 |
| $K_{AY}$ | 0 | 0 | .137 | .137 | 0 | | 0 |
| $K_P$ | .707 | | | .707 | .424 | | .424 |
| $K_Q$ | 1.1 | | | 1.1 | .66 | | .66 |
| $K_R$ | 1.4 | | | 1.4 | .84 | | .84 |

$K_{P_i}$, $K_{Q_i}$, $K_{R_i}$, $K_{Y_i}$, $K_{Z_i}$, $K_{PA}$ $K_{YA}$ } Currently unity for all values of time.

Mixing Logic Gains

| Orbiter Engines | APP | AYY | ARP | ARY |
|---|---|---|---|---|
| 1 | 1.4 | 0 | 0 | -3.5 |
| 2 | 1.4 | 1.4 | -3.5 | 1.8 |
| 3 | 1.4 | 1.4 | 3.5 | 1.8 |
| SRM Engines | | | | |
| 4 | 1.4 | 1.4 | -1.4 | 0 |
| 5 | 1.4 | 1.4 | 1.4 | 0 |

ARR = -7.0

Miscellaneous Fixed Gains

| | |
|---|---|
| PLIM | 7.00 degrees |
| YLIM | 7.00 degrees |
| RLIM | 5.00 degrees |
| $\tau$ | 0.1 |

## References


7-1 Space Division, Rockwell International (W. T. Schleich), Space Shuttle Guidance and Control Data Book, SD73-SH-0097, March 1973 (reference trajectory)

7-2 W. Peters, NASA/MSC/EG2, "Control Effector Mixing Logic for Parallel Burn Shuttle Ascent", 24 May 1972.

SECTION 8

## DFCS GLOSSARY

| Phase | Mnemonic | Symbol in Text | Definition |
|---|---|---|---|
| 4 | A 1,2,3,4 | $a_1, a_2, a_3, a_4$ | Gains for aileron control |
| 4 | A 11,12,21,22 | $a_{11}, a_{12}, a_{21}, a_{22}$ | Lateral fixed gains |
| 8 | ABC(1) | $\delta_{ROLL}$ | Total roll channel command |
| 8 | ABC(2) | $\delta_{PITCH}$ | Total pitch channel command |
| 8 | ABC(3) | $\delta_{YAW}$ | Total yaw channel command |
| 8 | ABE(1) | $\phi_{EB}$ | Body roll attitude error |
| 8 | ABE(2) | $\theta_{EB}$ | Body pitch attitude error |
| 8 | ABE(3) | $\psi_{EB}$ | Body yaw attitude error |
| 8 | ABEF(1) | $\phi_F$ | Scaled roll error |
| 8 | ABEF(2) | $\theta_F$ | Scaled pitch error |
| 8 | ABEF(3) | $\psi_F$ | Scaled yaw error |
| 2,4 | ACCBOD | $\underline{\tilde{a}}^b$ | Linear accel measurement (body coord) |
| 4 | ACCG | $g$ | Acceleration of gravity |
| 4 | ACCINT(1) | $\Delta p^j$ | Roll integrated angular acceleration |
| 4 | ACCINT(2) | $\Delta q^j$ | Pitch integrated angular acceleration |
| 4 | ACCINT(3) | $\Delta r^j$ | Yaw integrated angular acceleration |
| 4 | ACCL | $\tilde{a}_L$ | Measured lift acceleration |
| 4 | ACCLMN | $a_{LM}$ | Mean lift acceleration |
| 4 | ADOT | $\dot{\alpha}$ | Derivative of angle-of-attack wrt time |
| 4 | ADO(1) | $\Delta\hat{\dot{\alpha}}$ | Angle-of-attack rate filter output |
| 4 | ADO(2) | $\Delta\hat{\dot{\alpha}}'$ | Angle-of-attack rate filter extrapolated value |
| 2,4 | AG | $\alpha_G$ | Angle-of-attack guidance command |
| 4 | AGOLD | $\alpha_G^{old}$ | Previous-pass value of AG |
| 4 | AI(1) | $\Delta\tilde{\alpha}$ | Angle-of-attack filter input |
| 2 | AINT | $a_{INT}$ | Aileron integrator output |

DFCS GLOSSARY (Cont)

| Phase | Mnemonic | Symbol in Text | Definition |
|---|---|---|---|
| 2 | AINTI | | Aileron integrator input |
| 2 | AINT1 | | Aileron integrator node |
| 2 | AINTMX | $a_{INT}^{max}$ | Aileron integrator output limits(+-max) |
| 8 | AIV(1) | $\phi$ | Platform outer gimbal angle |
| 8 | AIV(2) | $\theta$ | Platform inner gimbal angle |
| 8 | AIV(3) | $\psi$ | Platform middle gimbal angle |
| 8 | AIVC(1) | $\phi_C$ | Platform outer gimbal angle command |
| 8 | AIVC(2) | $\theta_C$ | Platform inner gimbal angle command |
| 8 | AIVC(3) | $\psi_C$ | Platform middle gimbal angle command |
| 8 | AIVE(1) | $\phi_{EI}$ | Platform outer gimbal angle error |
| 8 | AIVE(2) | $\theta_{EI}$ | Platform inner gimbal angle error |
| 8 | AIVE(3) | $\psi_{EI}$ | Platform middle gimbal angle error |
| 4 | ALIM | $\alpha_{lim}$ | Limited value of angle-of-attack filter output |
| 2 | AM | $\tilde{\alpha}$ | Measured angle-of-attack |
| 4 | AMEAN | $\alpha_T$ | Mean (trim) angle of attack |
| 2 | AN | an | Normal acceleration (-zbody, with gravity) |
| 2 | ANF | $an_F$ | Normal acceleration filter output |
| 2 | ANGRV | $an_{GRV}$ | Gravity corrections for AN |
| 2 | ANN(I) | | Normal acceleration filter nodes |
| 4 | AO(1) | $\Delta\hat{\alpha}$ | Angle-of-attack filter output |
| 4 | AO(2) | $\Delta\hat{\alpha}'$ | Angle-of-attack filter extrapolated value |
| 8 | APP(I) | $APP_i$ | Gains for scaling pitch input to the pitch actuators |
| 8 | ARP(I) | $ARP_i$ | Gains for scaling roll input to the pitch actuators |
| 8 | ARR | ARR | Gain for scaling the roll input to the rudder |
| 8 | ARY(I) | $ARY_i$ | Gains for scaling roll input to the yaw actuators |

DFCS GLOSSARY (Cont)

| Phase | Mnemonic | Symbol in Text | Definition |
|---|---|---|---|
| 4 | AX,AY,AZ | $U_x, U_y, U_z$ | Jet acceleration in roll, pitch, yaw |
| 2 | AYM | $\tilde{ay}$ | Measured lateral acceleration |
| 2 | AYMF | $\tilde{ay}_F$ | Lateral acceleration filter output |
| 2 | AYMN(I) | | Lateral acceleration filter nodes |
| 8 | AYY(I) | $AYY_i$ | Gains for scaling yaw input to the yaw actuators |
| 2 | AZM | $\tilde{az}$ | Measured normal acceleration (+zbody) |
| 4 | B | b | Fixed estimator parameter |
| 4 | B 11,12,21,22 | $b_{11}, b_{12}, b_{21}, b22$ | Lateral fixed gains |
| 4 | BANK | $\phi_V$ | Roll (bank) about air-relative velocity vector |
| 4 | BANKDO(1) | $\Delta\hat{\dot{\phi}}_V$ | Bank rate filter output |
| 4 | BANKDO(2) | $\Delta\hat{\dot{\phi}}_V'$ | Bank rate filter extrapolated value |
| 4 | BANKG | $\phi_{VG}$ | Bank guidance command |
| 4 | BANKI(1) | $\Delta\tilde{\phi}_V$ | Bank filter input |
| 4 | BANKM | $\phi_{VM}$ | Mean bank angle |
| 4 | BANKO(1) | $\Delta\hat{\phi}_V$ | Bank filter output |
| 4 | BANKO(2) | $\Delta\hat{\phi}_V'$ | Bank filter extrapolated value |
| 4 | BDOT | $\dot{\beta}$ | Derivative of sideslip wrt time |
| 4 | BDO(1) | $\Delta\hat{\dot{\beta}}$ | Sideslip rate filter output |
| 4 | BDO(2) | $\Delta\hat{\dot{\beta}}$ | Sideslip rate filter extrapolated value |
| 4 | BDTM | $\dot{\beta}_M$ | Rate limit in roll jet logic |
| 2 | BE | $\beta_E$ | Sideslip error filter input |
| 2 | BEF | $\beta_{EF}$ | Sideslip error filter output |
| 2 | BEN(I) | | Sideslip error filter nodes |
| 2 | BG | $\beta_G$ | Sideslip guidance command |
| 4 | BI(1) | $\Delta\tilde{\beta}$ | Sideslip angle filter input |

DFCS GLOSSARY (Cont)

| Phase | Mnemonic | Symbol in Text | Definition |
|---|---|---|---|
| 4 | BKGOLD | $\phi_{VG}^{old}$ | Previous-pass value of BANKG |
| 2 | BM | $\tilde{\beta}$ | Measured sideslip |
| 4 | BO(1) | $\Delta\hat{\beta}$ | Sideslip angle filter output |
| 4 | BO(2) | $\Delta\hat{\beta}'$ | Sideslip angle filter extrapolated value |
| 4 | BRGOLD | $\dot{\phi}_{VG}^{old}$ | Previous-pass value of bank rate command |
| 4 | CAT | $\cos\alpha_T$ | Cos (AMEAN) |
| 2 | CC | $c_C$ | CAS command (QC, AC, or CSTARC) |
| 2 | CCAMN | | Angle-of-attack values for CCMN |
| 2 | CCAMX | | Angle-of-attack values for CCMX |
| 2 | CCCMN | | CAS values (C* or q) for CCMN |
| 2 | CCCMX | | CAS values (C* or q) for CCMX |
| 2 | CCMN | $c_C^{min}$ | CAS command limit, min (working) |
| 2 | CCMX | $c_C^{max}$ | CAS command limit, max (working) |
| 2 | CE | $c_E$ | CAS error filter input |
| 2 | CEF | $c_{EF}$ | CAS error filter output |
| 2 | CEN(I) | | CAS error filter nodes |
| 2 | CF | $c_F$ | CAS feedback (QF, AF, or CSTARF) |
| 2 | CG | $c_G$ | CAS guidance command (QG or CSTARG) |
| 4 | CLBP | $C'_{l_\beta}$ | Scheduled estimator parameter |
| 4 | CLDAP | $C'_{l_{\delta a}}$ | Scheduled estimator parameter |
| 4 | CNBP | $C'_{n_\beta}$ | Scheduled estimator parameter |
| 4 | CNDAP | $C'_{n_{\beta a}}$ | Scheduled estimator parameter |
| 2,4 | CRTB | $C_r^b$ | Reference-to-body direction cosine matrix |
| 4 | CRUD | $C_{RUD}$ | Rudder crossfeed gain |
| 2 | CSTARG | $c_G^*$ | CSTAR guidance command |
| 4 | DA1,DA2 | $\delta_{\alpha 1}$, $\delta_{\alpha 2}$ | Deadbands for pitch jet logic |
| - | DAC | $\delta a_C$ | Aileron command |

DFCS GLOSSARY (Cont)

| Phase | Mnemonic | Symbol in Text | Definition |
|---|---|---|---|
| 2 | DACF | $\delta a_{CF}$ | Aileron command filter output |
| 2,4 | DACMX | $\delta a_C^{max}$ | Aileron command limit(+-max) |
| 2 | DACN(I) | | Aileron command filter nodes |
| 2 | DAG | $\delta a_G$ | Aileron guidance command |
| 8 | DAIVC | $\dot{\theta}_C$ | Platform inner gimbal rate command |
| - | DAMAX | $\delta a^{max}$ | Physical aileron limit (+-max) |
| 4 | DAT | $\delta a_t$ | Aileron trim estimate |
| 2 | DATRIM | | Effective aileron trim, for mode change |
| 4 | DB1,DB2 | $\delta_{R1}$, $\delta_{R2}$ | Deadbands for roll jet logic |
| 2 | DDG | $\dot{\delta}_G$ | TAEM guidance interface variable |
| - | DEC | $\delta e_C$ | Elevator command |
| 2,4 | DECMN | $\delta e_C^{min}$ | Elevator command limit, min |
| 2,4 | DECMX | $\delta e_C^{max}$ | Elevator command limit, max |
| 2 | DECU | | Elevator command from speed loop crossfeeds |
| 2 | DECUB | | BCS version of DECU |
| 2 | DEG | $\delta e_G$ | Elevator guidance command |
| - | DELC | $\delta EL_C$ | Left elevon command |
| 8 | DELL 4,5 | $\delta_{4L}$, $\delta_{5L}$ | Commands to SRM left gimbals |
| 8 | DELORB | $\delta_{ORB}$ | Prestored orbiter engine deflection command |
| 8 | DELP(I) | $\delta_{iP}$ | Commands to orbiter pitch gimbals (i=1,2,3) |
| 8 | DELP(I) | $\delta_{iP}$ | Commands to SRM pseudo pitch gimbals (i=4,5) |
| 8 | DELR 4,5 | $\delta_{4R}$, $\delta_{5R}$ | Commands to SRM right gimbals |
| 8 | DELSRM | $\delta_{SRM}$ | Prestored SRM engine deflection command |
| 8 | DELY(I) | $\delta_{iY}$ | Commands to orbiter yaw gimbals (i=1,2,3) |
| 8 | DELY(I) | $\delta_{iY}$ | Commands to SRM pseudo yaw gimbals (i=4,5) |
| - | DEMIN,DEMAX | $\delta_e^{min}$, $\delta_e^{max}$ | Physical elevator lower (upper) limit |
| - | DERC | $\delta ER_C$ | Right elevon command |

DFCS GLOSSARY (Cont)

| Phase | Mnemonic | Symbol in Text | Definition |
|---|---|---|---|
| 4 | DET | $\delta e_t$ | Elevator trim estimate |
| 2 | DETRIM | | Effective elevator trim, for mode change |
| 2 | DFC | $\delta f_C$ | Body-flap command |
| 2 | DFCF | $\delta f_{CF}$ | Flap command filter output |
| 2 | DFCM | $\delta f_{CM}$ | Flap command from Mach profile |
| 2 | DFCMCH(I,J) | | 2xN matrix of (DFCM,MACH) pairs |
| 2 | DFCMN | $\delta f_C^{min}$ | Flap command limit, min |
| 2 | DFCMNB | | BCS version of DFCMN |
| 2 | DFCMX | $\delta f_C^{max}$ | Flap command limit, max |
| - | DFMIN,DFMAX | $\delta f^{min}$, $\delta f^{max}$ | Physical flap lower (upper) limit |
| 4 | DP1STR | $\bar{q}_1^*$ | Blending parameter for elevator use |
| 4 | DP2STR | $\bar{q}_2^*$ | Blending parameter for aileron use |
| 4 | DP3STR | $\bar{q}_3^*$ | Blending parameter for rudder use |
| 4 | DR1,DR2 | $\delta_{\phi 1}$, $\delta_{\phi 2}$ | Deadbands for yaw jet logic |
| - | DRC | $\delta r_C$ | Rudder command |
| 2 | DRCMX | $\delta r_C^{max}$ | Rudder command limit(+-max) |
| 2 | DRF | $\delta r_F$ | Rudder loop feedback |
| 2 | DRG | $\delta r_G$ | Rudder guidance command |
| - | DRMAX | $\delta r^{max}$ | Rudder physical limit (+-max) |
| 8 | DROLL | $\dot{\phi}_C$ | Platform outer gimbal rate command |
| 2 | DRTRIM | | Rudder effective trim, for mode change |
| - | DSC | $\delta s_C$ | Speed brake command |
| 2 | DSCF | $\delta s_{CF}$ | Speed brake command filter output |
| 2 | DSCMN | $\delta s_C^{min}$ | Speed brake command limit, min |
| 2 | DSCMNB | | BCS version of DSCMN |
| 2 | DSCMX | $\delta s_C^{max}$ | Speed brake command limit, max |
| 2 | DSCP | | Past value of DSC |

DFCS GLOSSARY (Cont)

| Phase | Mnemonic | Symbol in Text | Definition |
|---|---|---|---|
| 2 | DSDMX | $\dot{\delta s}^{max}$ | Speed brake (command) rate limit |
| 2 | DSG | $\delta s_G$ | Speed brake guidance command |
| - | DSMIN,DSMAX | $\delta s^{min}$, $\delta s^{max}$ | Speedbrake physical upper (lower) limit |
| 2 | DTC | $\delta t_C$ | Throttle command |
| 2 | DTCF | $\delta t_{CF}$ | Throttle command filter output |
| 2 | DTCMN | $\delta t_C^{min}$ | Throttle command limit, min |
| 2 | DTCMNB | | BCS version of DTCMN |
| 2 | DTCMX | $\delta t_C^{max}$ | Throttle command limit, max |
| 2 | DTCP | | Past value of DTC |
| 2 | DTDMX | $\dot{\delta t}^{max}$ | Throttle (command) rate limit |
| 2 | DTDMXB | | BCS version of DTDMX |
| 2 | DTG | $\delta t_G$ | Throttle guidance command |
| - | DTMIN,DTMAX | | Physical throttle lower (upper) limit |
| 2,4 | DYNP | $\bar{q}$ | Dynamic pressure estimate |
| 4 | DYNP 1,2,3 | $\bar{q}_1$, $\bar{q}_2$, $\bar{q}_3$ | Dynamic pressure switch levels for elevator, aileron, roll jet mode III |
| 2 | EINT | $e_{INT}$ | Elevator integrator output |
| 2 | EINTI | | Elevator integrator input |
| 2 | EINTMN | $e_{INT}^{min}$ | Elevator integrator output limit, min |
| 2 | EINTMX | $e_{INT}^{max}$ | Elevator integrator output limit, max |
| 2 | EINT1 | | Elevator integrator node |
| 4 | EL 1,2,3,4 | $\bar{\ell}_1$, $\bar{\ell}_2$, $\bar{\ell}_3$, $\bar{\ell}_4$ | Temporary blending parameters |
| 2 | ET | $e_T$ | Elevator trim tracker output |
| 4 | ETA | $\eta$ | Lateral scheduled gain (late entry) |
| 2 | ETDB | $e_{TDB}$ | Elevator trim tracker input deadband |
| 2 | ETDMX | $\dot{e}_T^{max}$ | Elevator trim tracker rate limit |
| 2 | ETF | $e_{TF}$ | Elevator trim tracker filter output |

DFCS GLOSSARY (Cont)

| Phase | Mnemonic | Symbol in Text | Definition |
|---|---|---|---|
| 2 | ETMN | $e_T^{min}$ | Elevator trim tracker input limit, min |
| 2 | ETMX | $e_T^{max}$ | Elevator trim tracker input limit, max |
| 4 | F1,F2 | $f_1$, $f_2$ | Longitudinal blending parameters |
| 4 | F3,F4 | $f_3$, $f_4$ | Lateral blending parameters |
| 4 | FILTER | FILTER | State estimation flag: 0=no filter(use measured values of state) 1=filter |
| 4 | FKLAT | $K_{lat}$ | Scheduled estimator gains: lateral channel |
| 4 | FKLON | $K_{lon}$ | Scheduled estimator gains: longitudinal channel |
| 4 | G1,G2 | $G_1$, $G_2$ | Lateral fixed gains |
| 2 | GAINT | | CAS aileron integral gain |
| 2 | GAIP | | CAS aileron integrator gain (GAP*GAINT) |
| 4 | GAIR | $G_A$ | Filter gain for air data (=TF/TS) |
| 4 | GAM11,12,21,2 | $\gamma_{11}$, $\gamma_{12}$, $\gamma_{21}$, $\gamma_{22}$ | Lateral scheduled gains (late entry) |
| 4 | GAMMA | $\gamma$ | Flight path angle |
| 2 | GAN(I) | | Normal acceleration filter coefficients |
| 2 | GAP | | Roll rate to aileron CAS loop gain |
| 2 | GAPM | | Roll rate to aileron SAS gain (working) |
| 2 | GAPMB | | BCS version of GAPM |
| 2 | GAPMS(I) | | Gain-switch-sensitive SAS values for GAPM |
| 2 | GAX | | SAS roll stick gearing gain |
| 2 | GAXB | | BCS version of GAX |
| 2 | GAYM(I) | | Lateral acceleration filter coefficients |
| 2 | GBE(I) | | Sideslip error filter coefficients |
| 2 | GCE(I) | | CAS error filter coefficients (working) |
| 2 | GCEA(I) | | Angle-of-attack values for GCE(I) |
| 2 | GCEC(I) | | CAS values (C* or q) for GCE(I) |

DFCS GLOSSARY (Cont)

| Phase | Mnemonic | Symbol in Text | Definition |
|---|---|---|---|
| 2 | GCFQM | | CSTAR q feedback gain |
| 2 | GCH | | Altitude loop gain (H to C) |
| 2 | GCTH | | Theta loop gain (TH to C) |
| 2 | GCX | | CAS pitch stick gearing gain |
| 2 | GDAC(I) | | Aileron cmnd filter coef. (A to R crossfeed) |
| 2 | GDFC | | Flap command filter coefficient |
| 2 | GDSC | | Speed brake to elevator crossfeed filter coef. |
| 2 | GDTC | | Throttle to elevator crossfeed coefficient |
| 2 | GEC | | CAS loop gain |
| 2 | GECA | | Angle-of-attack value for GEC |
| 2 | GECC | | CAS value (C* or q) for GEC |
| 2 | GEF | | Flap to elevator crossfeed gain |
| 2 | GEFB | | BCS version of GEF |
| 2 | GEIC | | CAS elevator integration gain (GEC*GEINT) |
| 2 | GEICA | | Angle-of-attack value for GEIC |
| 2 | GEICC | | CAS value (C* or q) for GEIC |
| 2 | GEINT | | CAS elevator integral gain (working) |
| 2 | GEINTA | | Angle-of-attack value for GEINT |
| 2 | GEINTC | | CAS value (C* or q) for GEINT |
| 2 | GEQM | | Elevator damper loop gain (working) |
| 2 | GEQMA | | Angle-of-attack value for GEQM |
| 2 | GEQMB | | BCS version of GEQM |
| 2 | GEQMC | | CAS value (C* or q) for GEQM |
| 2 | GEQMS(I) | | Gain-switch-sensitive SAS values for GEQM |
| 2 | GES | | Speed brake to elevator crossfeed gain |
| 2 | GESB | | BCS version of GES |
| 2 | GET | | Throttle to elevator crossfeed gain |

DFCS GLOSSARY (Cont)

| Phase | Mnemonic | Symbol in Text | Definition |
|---|---|---|---|
| 2 | GETB | | BCS version of GET |
| 2 | GETF | | Elevator trim tracker filter coefficient |
| 2 | GEX | | SAS pitch stick gearing gain |
| 2 | GEXB | | BCS version of GEX |
| 2 | GFX | | Flap controller gearing gain |
| 2 | GFXB | | BCS version of GFX |
| 2 | GHE(I) | | Altitude error filter coefficients |
| 4 | GNU1,GNU2 | | Lateral scheduled gains (late entry) |
| 2 | GPE(I) | | Roll rate error filter coefficients |
| 2 | GPHE(I) | | Bank angle error filter coefficients |
| 2 | GPHPS | | Psi-loop gain (psi to phi) |
| 2 | GPHT(I) | | Phi-turn filter coefficients |
| 2 | GPM(I) | | Roll rate feedback filter coefficients |
| 2 | GPMB(I) | | BCS version of GPM(I) |
| 2 | GPPH | | Phi-loop gain (phi to p) |
| 2 | GPSE(I) | | Psi error filter coefficients |
| 2 | GPX | | CAS roll stick gearing gain |
| 2 | GQM(I) | | Pitch rate feedback filter coefficients |
| 2 | GQMB(I) | | BCS version of GQM(I) |
| 2 | GQW | | Pitch rate washout filter coefficient |
| 2 | GRAY | | Lateral acceleration feedback gain |
| 2 | GRB | | Sideslip feedback gain |
| 2 | GRCPH | | Phi to RC gain, turn coordination |
| 2 | GRDAC | | Aileron to rudder crossfeed gain |
| 2 | GRDACB | | BCS version of GRDAC |
| 2 | GRE(I) | | Yaw rate error filter coefficients |
| 2 | GREB(I) | | BCS version of GRE(I) |



C-4

DFCS GLOSSARY (Cont)

| Phase | Mnemonic | Symbol in Text | Definition |
|---|---|---|---|
| 2 | GRM | | Yaw rate filter coef. (elevator loops) |
| 2 | GRRE | | Yaw rate feedback gain |
| 2 | GRRW | | Yaw rate washout feedback gain (working) |
| 2 | GRRWA | | AUTO version of GRRW |
| 2 | GRRWB | | BCS version of GRRW |
| 2 | GRRWS(I) | | Gain-switch-sensitive SAS values for GRRW |
| 2 | GRW | | Yaw rate washout filter coefficient |
| 2 | GRWB | | BCS version of GRW |
| 2 | GRX | | Pedal gearing gain (working) |
| 2 | GRXAUT | | RAUT mode value for GRX |
| 2 | GRXB | | BCS version of GRX |
| 2 | GRXSAS | | RSAS mode value for GRX |
| 2 | GSDDG | | TAEM guidance DDG to speed brake gain |
| 2 | GSU | | Speed brake velocity loop gain |
| 2 | GSUE(I) | | Speed brake values for GUE(I) |
| 2 | GSX | | Speed brake manual controller gearing |
| 2 | GSXB | | BCS version of GSX |
| 2 | GTDDG | | TAEM guidance DDG to throttle loop gain |
| 2 | GTHE(I) | | Theta error filter coefficients |
| 2 | GTU | | Throttle velocity loop gain |
| 2 | GTUE(I) | | Throttle values for GUE(I) |
| 2 | GTX | | Throttle manual controller gearing |
| 2 | GTXB | | BCS version of GTX |
| 2 | GUE(I) | | Velocity error filter coefficents (working) |
| 2 | GXA | | Roll stick filter coefficient |
| 2 | GXAX | | Roll stick parabolic shaping gain |
| 2 | GXE | | Pitch stick filter coefficient |

| Phase | Mnemonic | Symbol in Text | Definition |
|---|---|---|---|
| 2 | GXEX | | Pitch stick parabolic shaping gain |
| 2 | GXR | | Pedal filter coefficient |
| 2 | GXRX | | Pedal parabolic shaping gain |
| 2 | GXSX | | Speedbrake controller parabolic shaping gain |
| 2 | GXSXB | | BCS version of GXSX |
| 2 | GXTX | | Throttle controller parabolic shaping gain |
| 2 | GXTXB | | BCS version of GXTX |
| 2 | HE | $H_E$ | Altitude error |
| 2 | HEF | $H_{EF}$ | Altitude error filter output |
| 2 | HEN(I) | | Altitude error filter nodes |
| 2 | HG | $H_G$ | Altitude guidance command |
| 2 | HM | $H$ | Measured altitude |
| 2 | HREF | $H_{REF}$ | EALT mode reference altitude command |
| 2 | IABES | | ABES present (1); ABES absent (0) |
| 2 | IACSS | | Roll cntl stk steer active (1); inact (0) |
| 2 | IAGAIN | | AGAIN switch position (1 thru 4) |
| 2 | IAINT | | Aileron integrator active (1); inactive (0) |
| 2 | IANGRV | | Gravity corrections: UNITY(1), COSTH(2), COSTH * COSPH(3) |
| 2 | IAINOK | | Pilot requested values for IAINT |
| 2 | IATRIM | | Roll stick trim activity: 0,+1(rgt),-1(left) |
| 2 | IBFDBK | | Beta feedback (1); lat. accel. feedback (0) |
| 2 | ICC1 | | CAS command switch #1 options (0=off):<br>1 = ECAS mode or control stick steering<br>2 = EATT or EALT mode<br>3 = THG or HG mode<br>4 = CG mode (QG or CSTARG)<br>5 = AG mode |

| Phase | Mnemonic | Symbol in Text | Definition |
|---|---|---|---|
| 2 | ICC2 | | CAS command switch #2 options (0=off):<br>1 = theta loop (THREF or THG)<br>2 = altitude loop (HREF or HG) |
| 2 | ICF | | CAS feedback: Q(1), CSTAR(2), ALPHA(3) |
| 2 | ICSTAR | | CAS mode options: CSTAR(1), RCAH(0) |
| 2 | IECSS | | Pitch cntl stk steer active (1); inact (0) |
| 2 | IEGAIN | | EGAIN switch position (1 thru 4) |
| 2 | IEINOK | | Pilot-requested values for IEINT |
| 2 | IEINT | | Elevator integrator active (1); inactive (0) |
| 2 | IETRIM | | Pitch stick trim activity: 0,+1(fwd),-1(aft) |
| 2 | IGEARX | | Extend gear (1), retract gear (0) |
| 2 | IHC | | Altitude command: HREF (1), HG (2) |
| 4 | ILAT | $I_{LAT}$ | Mode flag for entry phase lateral control<br>0 = early entry, "reverse" aileron<br>1 = late entry, "proverse" aileron |
| - | IM | IM | Medium-sampling-rate index:<br>0 = perform calculations this pass |
| - | INGUID | INGUID | Flag specifies guidance command set<br>neg=commands not updated this pass<br>+/- 1=BANKG,ALPHG specified<br>+/- 2=ROLLG,QG specified |
| - | INSENS | INSENS | Flag specifies source of measurements<br>0=provided by simulator (hybrid sim.)<br>1=sensors or environment (SSFS)<br>2=G&N and sensors or environment (SSFS) |
| 2 | IPC | | Roll rate command options (0=off):<br>1 = ACAS mode or control stick steering<br>2 = AATT or AHDG mode<br>3 = PHG or PSG mode<br>4 = PG mode |

| Phase | Mnemonic | Symbol in Text | Definition |
|---|---|---|---|
| | | | 2 = AATT or AHDG mode |
| | | | 3 = PHG or PSG mode |
| | | | 4 = PG mode |
| 2 | IPHC | | Bank angle command options (0=off): |
| | | | 1 = AATT mode |
| | | | 2 = AHDG mode |
| | | | 3 = PSG mode |
| | | | 4 = PHG mode |
| 2 | IPHM | | PHM valid (1), invalid (0) |
| 2 | IPHRC | | Phi source for yaw rate command options: |
| | | | 1 = zero (beta/r-wash. fdbk) |
| | | | 2 = PHM (measured) |
| | | | 3 = PHC (commanded, if available) |
| 2 | IPSC | | Psi command: PSREF(1), PSG(2) |
| 2 | IQTRN | | Pitch rate turn coordination option: |
| | | | 1 = off, use Q feedback |
| | | | 2 = use Q washout feedback |
| | | | 3 = use Q=R*TANPHI feedback |
| 2 | IRCHAI | | Request change IAINT state (1), no request(0) |
| 2 | IRCHEI | | Request change IEINT state (1), no request(0) |
| 2 | IRGAIN | | RGAIN switch position (1 thru 4) |
| 2 | IRTRIM | | Pedal trim activity: 0,+1(lft),-1(rgt) |
| 2 | IRW | | Yaw rate washout active(1),inactive(0) |
| 2 | IRWP | | Pad load indicator for IRW |
| - | IS | IS | Slow-sampling-rate index: |
| | | | 0 = perform calculations this pass |
| - | ISTART | ISTART | Initialization pass(1), not initialization(0) |

| Phase | Mnemonic | Symbol in Text | Definition |
|---|---|---|---|
| - | ISTRTR | ISTRTR | Request initialization(1), no request(0) |
| 2 | ITHC | | Theta command: THREF(1), THG(2) |
| - | ITM, ITS | ITM, ITS | Number of fast passes per medium (slow) pass |
| 4 | IXX | $I_{xx}$ | Fixed estimator parameter |
| 4 | IZZ | $I_{zz}$ | Fixed estimator parameter |
| 2 | JAGAIN | | Local past-value copy of IAGAIN |
| 2 | JEGAIN | | Local past-value copy of IEGAIN |
| 2 | JGEARX | | Local copy of IGEARX |
| 2 | JRCHAI | | Local copy of IRCHAI |
| 2 | JRCHEI | | Local copy of IRCHEI |
| 2 | JRGAIN | | Local past-value copy of IRGAIN |
| 4 | KA,KE | $K_a$, $K_e$ | Aileron (elevator) deflection limit parameter |
| 4 | KALF1 | $K_{\alpha 1}$ | Elevator trim gain (early entry) |
| 4 | KALF2 | $K_{\alpha 2}$ | Elevator trim gain (late entry) |
| 4 | KC1,KC2 | KC1, KC2 | Low-dynamic-pressure constant filter gains |
| 4 | KROLL | $K_{\phi}$ | Aileron trim gain |
| 8 | KY(I) | $K_{Yi}$ | Individual y accelerometer gain |
| 8 | KZ(I) | $K_{Zi}$ | Individual z accelerometer gain |
| 4 | LAMDAA,LAMDAE | $\lambda_a$, $\lambda_e$ | Switch set to 1 if pitch (lateral) ACPS jets are needed |
| 4 | MA,ME | $m_a$, $m_e$ | Slope of KA(KE) curve |
| 2,4 | MACH | $\mathcal{M}$ | Mach number |
| 2 | MODA | | Current aileron loop mode:<br>1 = ABCS 2 = ASAS 3 = ACAS<br>4 = AAUT 5 = AATT 6 = AHDG |
| 2 | MODACH | | Indicate aileron loop mode change<br>1 = no change 2 = MODAGR change |

| Phase | Mnemonic | Symbol in Text | Definition |
|---|---|---|---|
| | | | 3 = MODAR change 4 = first pass |
| 2 | MODAG | | Current aileron guidance mode (MODA=4): |
| | | | 1 = PG 2 = PHG 3 = PSG |
| 2 | MODAGR | | Requested values for MODAG |
| 2 | MODAR | | Requested values for MODA |
| 2 | MODB | | BCS mode (1), not BCS mode (0) |
| 2 | MODBR | | Requested values for MODB |
| 2 | MODE | | Current elevator loop mode: |
| | | | 1 = EBCS 2 = ESAS 3 = ECAS |
| | | | 4 = EAUT 5 = EATT 6 = EALT |
| 2 | MODECH | | Indicate elevator loop mode change |
| | | | 1 = no change 2 = MODEGR change |
| | | | 3 = MODER change 4 = first pass |
| 2 | MODEG | | Current elevator guidance mode (MODE=4): |
| | | | 1 = QG 2 = THG 3 = AG 4 = HG |
| 2 | MODEGR | | Requested values for MODEG |
| 2 | MODER | | Requested values for MODE |
| 2 | MODR | | Current rudder loop mode: |
| | | | 1 = RBCS 2 = RSAS |
| | | | 3 = RCAS (=2=RSAS) 4 = RAUT |
| 2 | MODRCH | | Indicate rudder loop mode change: |
| | | | 1 = no change 2 = MODRGR change |
| | | | 3 = MODRR change 4 = first pass |
| 2 | MODRG | | Current rudder guidance mode (MODR=4): |
| | | | 1 = yaw rate feedback |
| | | | 2 = yaw rate washout feedback |
| 2 | MODRGR | | Requested values for MODRG |

| Phase | Mnemonic | Symbol in Text | Definition |
|---|---|---|---|
| 2 | MODRR | | Requested values for MODR |
| 2 | MODS | | Current speed brake mode:<br>1 = BCSSB 2 = MANSB 3 = IASSB<br>4 = AUTOSB 5 = DSGSB 6 = TAEMS |
| 2 | MODT | | Current throttle mode:<br>1 = BCSTH 2 = MANTH 3 = IASTH<br>4 = AUTOTH 5 = DTGTH 6 = TAEMTH |
| 2 | MODU | | Current speed control mode:<br>1 = UBCS 2 = UMAN (IAS-man)<br>3 = UASB (IAS-sb) 4 = UAUT<br>5 = UATH (IAS-th) |
| 2 | MODUCH | | Indicate speed control loop mode change:<br>1 = no change 2 = MODUGR change<br>3 = MODUR change 4 = first pass |
| 2 | MODUFP | | Flag for FILUP subroutine speed control<br>1 = MANUAL or AUTO-dir 2 = IAS-hold<br>3 = AUTO(UG-guid) 4 = AUTO(TAEM-guid) |
| 2 | MODUFT | | Flag for velocity filter moding:<br>0 = UFILT inactive 1 = SB-ufilt<br>2 = TH-ufilt |
| 2 | MODUG | | Current guidance speed control mode:<br>(1-10, see speed control flag table) |
| 2 | MODUGR | | Requested values for MODUG |
| 2 | MODUR | | Requested values for MODU |
| 4 | MU | | Roll jet switch parameter |
| 4 | P 11,12,21,22 | $P_{11}$, $P_{12}$, $P_{21}$, $P_{22}$ | Lateral derived gains (late entry) |
| 2 | PC | $P_C$ | Roll rate command |

DFCS GLOSSARY (Cont)

| Phase | Mnemonic | Symbol in Text | Definition |
|---|---|---|---|
| 8 | PCEGK | $K_{PA}$ | Pitch channel error gain |
| 2 | PCMX | $p_C^{max}$ | Roll rate command limit, +-maximum |
| 2 | PE | $p_E$ | Roll rate error / error filter input |
| 2 | PEF | $p_{EF}$ | Roll rate error filter output |
| 8 | PEGK | $K_\theta$ | Pitch error gain |
| 2 | PEN(I) | | Roll rate error filter nodes |
| 8 | PER | $P_{ER}$ | Limited pitch channel error |
| 8 | PERINT | $P_{ERINT}$ | Weighted integral of PER |
| 2 | PG | $p_G$ | Roll rate guidance command |
| - | PHASE | PHASE | Flight phase flag (internal copy of PHASER) |
| - | PHASER | PHASER | Flight phase flag (external): 1=rollout phase<br>2=cruising phase<br>4=entry phase<br>5=orbit TVC phase<br>6=orbit RCS phase<br>7=insertion TVC phase<br>8=booster TVC phase |
| 2 | PHC | $\phi_C$ | Bank angle command |
| 2 | PHCMX | $\phi_C^{max}$ | Bank angle command limit,+-maximum |
| 2 | PHE | $\phi_E$ | Bank angle error / error filter input |
| 2 | PHEF | $\phi_{EF}$ | Bank angle error filter output |
| 2 | PHEN(I) | | Bank angle error filter nodes |
| 2 | PHG | $\phi_G$ | Bank angle guidance command |
| 4 | PHILAT | $\Phi_{lat}^{CL}$ | Estimator state transition matrix: lateral channel |
| 4 | PHILIM | $\phi_{lim}$ | Limited value of bank filter output |
| 4 | PHILON | $\Phi_{lon}^{CL}$ | Estimator state transition matrix: longitudinal channel |

DFCS GLOSSARY (Cont)

| Phase | Mnemonic | Symbol in Text | Definition |
|---|---|---|---|
| 2 | PHM | $\tilde{\phi}$ | Measured bank angle |
| 2 | PHREF | $\phi_{REF}$ | AATT mode reference bank angle command |
| 2 | PHSEL | $\phi_{SEL}$ | Pilot-selected values for PHREF |
| 2 | PHTF | $\phi_{TF}$ | Phi-turn filter output and/or bank angle for turn-coordination ($\epsilon \delta r$) |
| 2 | PHTN(I) | | Phi-turn filter nodes |
| 4 | PI(1) | $\Delta\tilde{p}$ | Roll rate filter input |
| 8 | PLIM | PLIM | Pitch error software limit |
| 2,4 | PM | $\tilde{p}$ | Measured roll rate |
| 2 | PMDIS | $\tilde{p}_{DIS}$ | Roll rate for AATT or AHDG reengage after ACSS |
| 2 | PMF | $\tilde{p}_F$ | Roll rate filter output |
| 2 | PMFB | | BCS version of PMF |
| 2 | PMNB(I) | | BCS version of PMN(I) |
| 2 | PMN(I) | | Roll rate filter nodes |
| 4 | PN | $p_N$ | Nominal roll rate |
| 8 | PRATEC | $PRATE_C$ | Pitch rate command |
| 8 | PRATEF | $PRATE_F$ | Scaled pitch rate error |
| 8 | PRATEM | $PRATE_M$ | Blended pitch rate |
| 8 | PRGG(I) | $K_{Qi}$ | Individual pitch rate gyro gain |
| 8 | PRGK | $K_Q$ | Pitch rate gain |
| 2 | PSE | $\Psi_E$ | Heading error |
| 2 | PSEF | $\bar{\Psi}_{EF}$ | Heading error filter output |
| 2 | PSEN(I) | | Heading error filter nodes |
| 2 | PSG | $\Psi_G$ | Heading guidance command |
| 2 | PSM | $\tilde{\Psi}$ | Measured heading |
| 2 | PSREF | $\Psi_{REF}$ | AHDG mode reference heading command |

DFCS GLOSSARY (Cont)

| Phase | Mnemonic | Symbol in Text | Definition |
|---|---|---|---|
| 2 | PSSEL | $\Psi_{SEL}$ | Pilot-selected values for PSREF |
| 2 | QCT | $q_{CT}$ | Pitch turn coordination q-command |
| 2 | QCTMX | $q_{CT}^{max}$ | QCT maximum (0. = min) |
| 2 | QG | $q_G$ | Pitch rate guidance command |
| 4 | QI(1) | $\Delta\tilde{q}$ | Pitch rate filter input |
| 2,4 | QM | $\tilde{q}$ | Measured pitch rate |
| 2 | QMDIS | $\tilde{q}_{DIS}$ | Pitch rate for EATT or EALT reengage after ECSS |
| 2 | QMF | $\tilde{q}_F$ | Pitch rate filter output |
| 2 | QMFB | | BCS version of QMF |
| 2 | QMN(I) | | Pitch rate filter nodes |
| 2 | QMNB(I) | | BCS version of QMN(I) |
| 4 | QN | $q_N$ | Nominal pitch rate |
| 2 | QWF | $q_{WF}$ | Pitch rate washout filter output |
| 2 | QW1 | | Pitch rate washout filter node |
| 4 | R1 | $r_1$ | Angle-of-attack error limits |
| 4 | R8 | $r_8$ | Roll angle error limits |
| 4 | R 11,12,21,22 | $r_{11}$, $r_{12}$, $r_{21}$, $r_{22}$ | Lateral derived gains (late entry) |
| 8 | RAGTVK | $K_\phi$ | Roll error gain |
| 2 | RC | $r_C$ | Yaw rate command |
| 4 | | | Derivative of bank angle |
| 4 | RDOTC | $\dot{\phi}_C(\dot{\phi}^C)$ | Bank rate command |
| 2 | RE | $r_E$ | Yaw rate error |
| 2 | REF | $r_{EF}$ | Yaw rate error filter output |
| 2 | REN(I) | | Yaw rate error filter nodes |
| 2 | RENB(I) | | BCS version of REN(2) |
| 8 | RER | $R_{ER}$ | Limited roll error for TVC |

DFCS GLOSSARY (Cont)

| Phase | Mnemonic | Symbol in Text | Definition |
|---|---|---|---|
| 2 | RG | $r_G$ | Yaw rate guidance command |
| 2,4 | RHO | | Reference air density |
| 4 | RI(1) | $\Delta\tilde{r}$ | Yaw rate filter input |
| 8 | RLIM | RLIM | Roll error software limit |
| 2,4 | RM | $\tilde{r}$ | Measured yaw rate |
| 2 | RMF | $\tilde{r}_F$ | Yaw rate filter output (elevator loops) |
| 4 | RN | $r_N$ | Nominal yaw rate |
| 2,4 | ROLL | $\phi$ | Computed roll angle |
| 2 | ROLLG | $\phi_G$ | Guidance commanded roll angle |
| 4 | RPHI 1,2,3 | $r_{\phi 1}, r_{\phi 2}, r_{\phi 3}$ | Parameters for maneuver rate calculation |
| 4 | RR | $\underline{R}_{rel}$ | Position vector in reference coordinates |
| 8 | RRATEF | $RRATE_F$ | Scaled roll rate for TVC |
| 8 | RRATEM | $RRATE_M$ | Blended roll rate signals |
| 8 | RRGG(I) | $K_{Pi}$ | Individual roll rate gyro gain |
| 8 | RRGTVK | $K_P$ | TVC roll rate gain |
| 2 | RWFB | | BCS version of RWF |
| 2 | RW1B | | BCS version of RW1 |
| 2 | RWF | $r_{WF}$ | Yaw rate washout filter output |
| 2 | RW1 | | Yaw rate washout filter node |
| 4 | S | S | Fixed estimator parameter |
| 4 | SAT | $\sin\alpha_T$ | Sin (AMEAN) |
| 4 | SIGMA A,B,R | $\sigma_\alpha, \sigma_\beta, \sigma_\varphi$ | Gains for jet logic - pitch, roll, yaw |
| 8 | TAUINT | $\tau$ | Integrator gain |
| 2 | TDSDMX | | TF * speed brake rate limit |
| 2 | TDTDMX | | TF * throttle rate limit |
| 2 | TEMP | | Temporary local variable(s) |
| 2,4 | TEMPER | $T_A$ | Air temperature, deg F and dimensionless |
| 2,4 | TEMPRK | $TK_A$ | Input air temperature in deg. Kelvin |

DFCS GLOSSARY (Cont)

| Phase | Mnemonic | Symbol in Text | Definition |
|---|---|---|---|
| 2 | TETDMX | | TF * elevator trim tracker rate limit |
| - | TF | $T_F$ | Fast sampling interval |
| 2 | TF2 | | TF / 2. |
| 2 | TFB | | BCS version of TF2 |
| 2 | TGSDDG | | TF * TAEM/SB loop gain |
| 2 | TGTDDG | | TF * TAEM/TH loop gain |
| 2 | THE | $\theta_E$ | Theta error |
| 2 | THEF | $\theta_{EF}$ | Theta error filter output |
| 2 | THEN(I) | | Theta error filter nodes |
| 2 | THG | $\theta_G$ | Theta guidance command |
| 2 | THM | $\tilde{\theta}$ | Measured theta |
| 2 | THREF | $\theta_{REF}$ | EATT mode reference pitch attitude command |
| 2 | THSEL | $\theta_{SEL}$ | Pilot-selected values for THREF |
| - | TM | $T_M$ | Medium sampling interval |
| - | TNEXT | TNEXT | Next DFCS entry time |
| - | TS | $T_S$ | Slow sampling interval |
| 2 | TXADMX | | TF * roll stick rate limit |
| 2 | TXATR | | TF * roll stick trim rate |
| 2 | TXATRB | | BCS version of TXATR |
| 2 | TXEDMX | | TF * pitch stick rate limit |
| 2 | TXETR | | TF * pitch stick trim rate |
| 2 | TXETRB | | BCS version of TXETR |
| 2 | TXRDMX | | TF * pedal rate limit |
| 2 | TXRTR | | TF * pedal trim rate |
| 2 | TXRTRB | | BCS version of TXRTR |
| 2 | TXTDMX | | TF * throttle controller rate limit |
| 2 | T2GAIP | | TF2 * roll integrator gain |

DFCS GLOSSARY (Cont)

| Phase | Mnemonic | Symbol in Text | Definition |
|---|---|---|---|
| 2 | T2GEIC | | TF2 * pitch integrator gain |
| 4 | U1,U2 | $U_1$, $U_2$ | Roll jet logic gains |
| 2 | UE | $u_E$ | Velocity error |
| 2 | UEF | $u_{EF}$ | Velocity error filter output |
| 2 | UEN(I) | | Velocity error filter nodes |
| 2 | UG | $u_G$ | Velocity guidance command |
| 4 | UJ (1),(2),(3) | $u_x$, $u_y$, $u_z$ | ACPS torque commands |
| 4 | UL(I) | $\underline{u}_L$ | Vector in longitudinal plane normal to velocity |
| 2 | UM | $\tilde{u}$ | Measured velocity |
| 2 | UREF | $u_{REF}$ | IAS-hold mode reference velocity command |
| 2 | USEL | $u_{SEL}$ | Pilot-selected values for UREF |
| 2,4 | UXV,UYV,UZV | $\underline{UV}_x$, $\underline{UV}_y$, $\underline{UV}_z$ | Unit vectors along the velocity coordinate system, in reference coordinates |
| 4 | VBODY | $\underline{V}_{body}$ | Air-relative velocity in body coordinates |
| 4 | VMACH1 | $M_1$ | Lateral switch parameter |
| 4 | VR | $\underline{V}_{rel}$ | Air-relative velocity in reference coordinates |
| 4 | W1,W2 | $\omega_1$, $\omega_2$ | Gains for aileron control logic |
| 8 | WB(1,I) | $RRATE_i$ | Individual roll rate gyro signal |
| 8 | WB(2,I) | $PRATE_i$ | Individual pitch rate gyro signal |
| 8 | WB(3,I) | $YRATE_i$ | Individual yaw rate gyro signal |
| 4 | WD | $\omega_d$ | Gain for elevator control |
| 2 | XA | xa | Local variable for XAM processing |
| 2 | XAB | $xa_B$ | Roll stick detent bias |
| 2 | XADB | $xa_{DB}$ | Roll stick deadband |
| 2 | XADMX | $\dot{xa}^{max}$ | Roll stick rate limit |
| 2 | XAF | $xa_F$ | XA filter output |
| 2 | XAM | $\tilde{xa}$ | Measured roll stick deflection |

DFCS GLOSSARY (Cont)

| Phase | Mnemonic | Symbol in Text | Definition |
|---|---|---|---|
| 2 | XAMDIS | $\tilde{xa}_{DIS}$ | Roll stick disengage level for ACSS |
| 2 | XAMX | $xa^{max}$ | Roll stick limits (+-max) |
| 2 | XAP | $xa_P$ | Processed roll stick |
| 2 | XAT | $xa_T$ | Roll stick trim |
| 2 | XATB | | BCS version of XAT |
| 2 | XATMX | $xa_T^{max}$ | Roll stick trim limit (+-max) |
| 2 | XATMXB | | BCS version of XATMX |
| 2 | XATR | $\dot{xa}_T$ | Roll stick trim rate |
| 2 | XATRB | | BCS version of XATR |
| 2 | XE | $xe$ | Local variable for XEM processing |
| 2 | XEB | $xe_B$ | Pitch stick detent bias |
| 2 | XEDB | $xe_{DB}$ | Pitch stick deadband |
| 2 | XEDMX | $\dot{xe}^{max}$ | Pitch stick rate limit |
| 2 | XEF | $xe_F$ | XE filter output |
| 2 | XEM | $\tilde{xe}$ | Measured pitch stick deflection |
| 2 | XEMDIS | $\tilde{xe}_{DIS}$ | Pitch stick disengage level for ECSS |
| 2 | XEMN | $xe^{min}$ | Pitch stick minimum (aft) |
| 2 | XEMX | $xe^{max}$ | Pitch stick maximum (fwd) |
| 2 | XEP | $xe_P$ | Processed pitch stick |
| 2 | XET | $xe_T$ | Pitch stick trim |
| 2 | XETB | | BCS version of XET |
| 2 | XETMN | $xe_T^{min}$ | Pitch stick trim minimum (aft) |
| 2 | XETMNB | | BCS version of XETMN |
| 2 | XETMX | $xe_T^{max}$ | Pitch stick trim maximum (fwd) |
| 2 | XETMXB | | BCS version of XETMX |
| 2 | XETR | $\dot{xe}_T$ | Pitch stick trim rate |
| 2 | XETRB | | BCS version of XETR |

DFCS GLOSSARY (Cont)

| Phase | Mnemonic | Symbol in Text | Definition |
|---|---|---|---|
| 2 | XFB | | Flap controller bias |
| 2 | XFM | $\tilde{xf}$ | Measured flap controller deflection |
| 2 | XFMMN | $\tilde{xf}^{min}$ | Flap controller minimum |
| 4 | XI 1 | $\xi_1$ | Scheduled control gain: longitudinal |
| 4 | XI 2,3,4,5,6 | $\xi_2, \xi_3, \xi_4, \xi_5, \xi_6$ | Scheduled control gains:lateral, early entry |
| 2 | XR | xr | Local pedal variable |
| 2 | XRB | $xr_b$ | Pedal detent bias |
| 2 | XRDB | $xr_{DB}$ | Pedal deadband |
| 2 | XRDMX | $\dot{xr}^{max}$ | Pedal rate limit |
| 2 | XRF | $xr_F$ | XR filter output |
| 2 | XRM | $\tilde{xr}$ | Measured pedal deflection |
| 2 | XRMX | $xr^{max}$ | Pedal limits (+-max) |
| 2 | XRP | $xr_P$ | Processed pedal |
| 2 | XRT | $xr_T$ | Pedal trim |
| 2 | XRTB | | BCS version of XRT |
| 2 | XRTMX | $xr_T^{max}$ | Pedal trim limits (+-max) |
| 2 | XRTMXB | | BCS version of XRTMX |
| 2 | XRTR | $\dot{xr}_T$ | Pedal trim rate |
| 2 | XRTRB | | BCS version of XRTR |
| 2 | XS | xs | Local variable for XSM processing |
| 2 | XSM | $\tilde{xs}$ | Measured speed brake controller deflection |
| 2 | XSMDIS | $\tilde{xs}_{DIS}$ | XSM disengage level for IAS-sb |
| 2 | XSMMN | $\tilde{xs}^{min}$ | XSM minimum (retract) |
| 2 | XSMREF | $\tilde{xs}_{REF}$ | XSM reference for IAS-sb disengage |
| 2 | XSP | $xs_P$ | Processed XSM |
| 2 | XT | xt | Local variable for XTM processing |
| 2 | XTM | $\tilde{xt}$ | Measured throttle controller deflection |

DFCS GLOSSARY (Cont)

| Phase | Mnemonic | Symbol in Text | Definition |
|---|---|---|---|
| 2 | XTMDIS | $\tilde{xt}_{DIS}$ | XTM disengage level for IAS-th |
| 2 | XTMMN | $\tilde{xt}^{min}$ | XTM minimum (idle) |
| 2 | XTMREF | $\tilde{xt}_{REF}$ | XTM reference for IAS-th disengage |
| 2 | XTP | $xt_P$ | Processed XTM |
| 2 | XTPB | | BCS version of XTP |
| 2 | XTPP | | Past value of XTP |
| 2 | XTPPB | | BCS version of XTPP |
| 8 | YACCER | $\ddot{Y}_F$ | Scaled y acceleration error |
| 8 | YACIN(I) | $\ddot{Y}_i$ | Individual y accelerometer signal |
| 8 | YACSIG | $\ddot{Y}_M$ | Blended y accelerometer signals |
| 8 | YAGK | $K_{AY}$ | Y acceleration gain |
| 8 | YCEGK | $K_{YA}$ | Yaw channel error gain |
| 8 | YEGK | $K_{\Psi}$ | Yaw error gain |
| 8 | YER | $Y_{ER}$ | Limited yaw channel error |
| 8 | YERINT | $Y_{ERINT}$ | Weighted integral of YER |
| 8 | YLIM | YLIM | Yaw error software limit |
| 8 | YRATEC | $YRATE_C$ | Yaw rate command |
| 8 | YRATEF | $YRATE_F$ | Scaled yaw rate error |
| 8 | YRATEM | $YRATE_M$ | Blended rate gyro signals |
| 8 | YRGG(I) | $K_{Ri}$ | Individual yaw rate gyro gain |
| 8 | YRGK | $K_R$ | Yaw rate gain |
| 8 | ZACCER | $\ddot{Z}_F$ | Scaled z acceleration error |
| 8 | ZACD | $\ddot{Z}_D$ | Z acceleration command |
| 8 | ZACIN(I) | $\ddot{Z}_i$ | Individual z accelerometer signal |
| 8 | ZACSIG | $\ddot{Z}_M$ | Blended z accelerometer signals |
| 8 | ZAGK | $K_{AZ}$ | Z acceleration gain |
| 4 | ZETAD | $\xi_d$ | Gain for elevator control |
| 4 | ZETA 1,2 | $\xi_1, \xi_2$ | Gains for aileron control logic |

# PART 2

# PERIPHERAL ROUTINES

# SECTION 1

# AUTOMATIC TRACKING

The designs presented in this section were part of the Apollo/Skylab flight programs and are reprinted here with only minor modification. They are a baseline for the Space Shuttle pointing routines.

The automatic tracking function applies to any mission situation requiring the orbiter to track a designated body (or target vehicle) by pointing a specified body fixed vector at the target or to any mission situation requiring the orbiter to rotate at a constant rate around a specified axis. Typically, automatic tracking can be utilized for rendezvous navigation, local vertical tracking, landmark tracking, and passive thermal control. In this mode of operation, the autopilot (DAP) is utilized as a rate command system. Attitude control, if required, is maintained by the Tracking Attitude Routine (R61) with attitude feedback from the navigation functions. R61 operates at a variable computation period, $\Delta t_c$, which is generally much longer than the autopilot sampling period of 0.1 seconds. Initial alignment of the vehicle to the tracking attitude is generally performed by R61 using the automatic maneuver routine (R60). In the following TV denotes the target vehicle and ORB denotes the orbiter vehicle.

## 1.1 Tracking Options

There are five automatic tracking options as defined below:

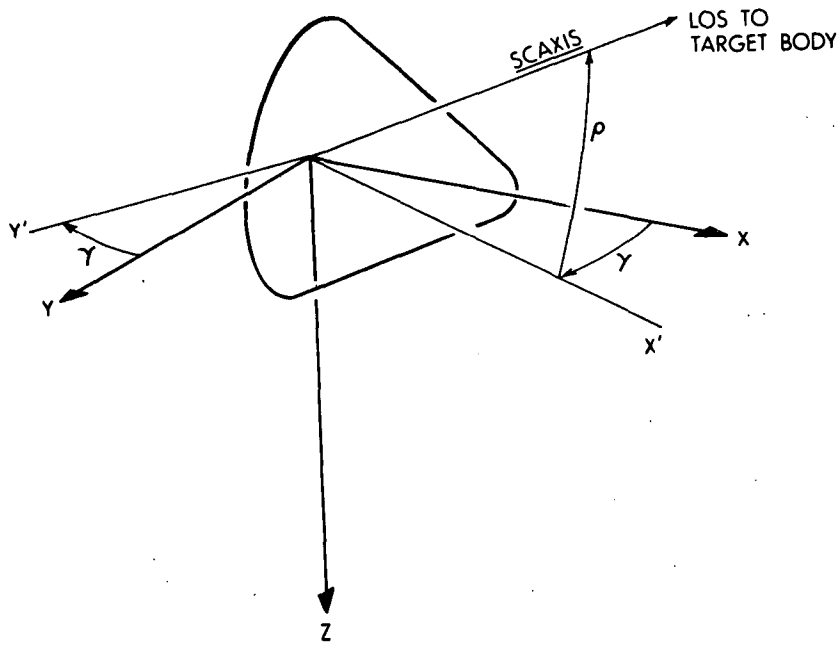
| Option | Purpose |
|---|---|
| 0 | Point a specified spacecraft vector at the TV. Do not constrain rotational attitude about this pointing vector. (Use VECPOINT) |
| 1 | Point a specified spacecraft vector at a celestial body. Do not constrain attitude about this vector (Use VECPOINT) |
| 2 | Rotate at a specified rate about a specified spacecraft vector. |
| 4 | Same as Option 0, but constraining attitude about the pointing vector to a specified value (OMICRON). |
| 5 | Same as Option 1, but constraining attitude about pointing vector to a specified value (OMICRON). |

1.2 Option 0: Orient the Spacecraft Such that a Specified Body-Fixed Pointing Vector Aligns with the Estimated Line of Sight to the TV.

The pointing vector (SCAXIS) is defined by two angles:

1) A specified positive angle ($\gamma$) from the spacecraft X axis about the spacecraft Z axis ($0 \leqq \gamma < 360$ deg)

2) A specified positive angle ($\rho$) about the Y'-axis produced in the XY plane as a result of $\gamma$ ($-90 \leqq \rho \leqq 90$ deg).



The LOS is determined by advancing the estimated ORB and TV state vectors to the current time. Once the coincidence of SCAXIS and LOS has been established, R61 establishes a spacecraft angular rate to maintain the coincidence. For Option 0, R61 provides the RCS DAP with the following quantities:

1) $\underline{\omega}_{CA}$.....desired LOS rate plus LOS correction rate in control-axis coordinates

2) $IGA_D$, $MGA_D$, $OGA_D$.....DAP reference angles, i.e., center of the deadband.

3) $\Delta\underline{G}A$.....a vector defining the desired incremental changes in the IMU gimbal angles every 0.1 second.

The desired LOS rate ($\underline{\omega}_{CA}$) is computed as follows:

$$\underline{u}_{LOS} = \underline{r}_W - \underline{r}_C$$

$$\underline{\omega}_{correction} = ([NBSM]\ \underline{SCAXIS} \times [REFSMMAT]\ UNIT\ (\underline{u}_{LOS}))/\Delta t_C$$

where NBSM uses the DAP reference angles THETADX, Y, and Z for the transformation from navigation-base to stable-member coordinates.

If $|\underline{\omega}_{correction}|$ exceeds MAXRATE, distribute as follows:

$$\underline{\omega}_{correction} = MAXRATE\ [UNIT\ (\underline{\omega}_{correction})]$$

$$\underline{\omega}_{LOS} = [REFSMMAT] \left\{ \frac{UNIT\ (\underline{u}_{LOS}) \times (\underline{v}_W - \underline{v}_C)}{|\underline{u}_{LOS}|} \right\} + \underline{\omega}_{correction}$$

$$\underline{\omega}_{CA} = [NBCA]\ [SMNB]\ \underline{\omega}_{LOS},$$

where $\underline{\omega}_{LOS}$ is the LOS rate in stable member coordinates. [SMNB] is the transformation from stable member axes to navigation base coordinates. [NBSM] is the transformation from navigation base coordinates to stable member coordinates. [REFSMMAT] is the transformation from reference inertial coordinates to stable member axes. [NBCA] is the matrix for transforming a vector from navigation base to control-axis coordinates. $\underline{r}_W$, $\underline{v}_W$, $\underline{r}_C$, and $\underline{v}_C$ are the position and velocity vectors of the TV and ORB in basic-reference coordinates. SCAXIS is the specified axis ($\gamma, \rho$). MAXRATE is the maximum correction rate and $\Delta t_C$ is a conservative estimate of the R61 computation interval (thus preventing overshoot in vehicle attitude by the time of the next pass).

The vector $\Delta\underline{G}A$ is computed as follows:

$$\Delta\underline{G}A = \begin{bmatrix} \cos IGA \sec MGA & 0 & -\sin IGA \sec MGA \\ -\cos IGA \tan MGA & 1 & \sin IGA \tan MGA \\ \sin IGA & 0 & \cos IGA \end{bmatrix} (0.1\ \underline{\omega}_{LOS}$$

where

IGA = inner gimbal angle
MGA = middle gimbal angle
OGA = outer gimbal angle

where $\underline{\omega}_{LOS}$ is the LOS rate in stable-member coordinates computed previously.

## 1.3 Option 1: Orient the Spacecraft Such that SCAXIS Aligns with the LOS to a Specified Celestial Body.

For Option 1, the LOS is determined by advancing the estimated ORB state vector to the current time. Once the coincidence to SCAXIS and LOS has been established, R61 establishes a spacecraft angular rate to maintain the coincidence. R61 quantities provided to the RCS DAP under tracking-option 1 are as described for Option 0 with the following exceptions:

$\underline{u}_{LOS}$ = line of sight direction, including aberration correction

and,

if the target is a star or planet,

$\underline{\omega}_{LOS} = \underline{\omega}_{correction}$;

if the target is the center of the earth,

$$\underline{\omega}_{LOS} = [REFSMMAT] \left\{ \frac{UNIT\ (\underline{r}_C) \times (\underline{v}_C)}{|\underline{r}_C|} \right\} + \underline{\omega}_{correction}$$

## 1.4 Option 2: Rotate the Spacecraft About SCAXIS

Option 2 is effected by a special routine known as the Automatic Rotation Routine (R67). SCAXIS is as defined in Option 0. R67 initializes COF, BRATE, [DEL], [MIS], and ADB in KALCMANU and then executes the KALCMANU coding to issue the following quantities to the RCS DAP:

1) $\underline{\omega}_{CA}$ .... desired rotation rate

2) $IGA_D$, $MGA_D$, $OGA_D$, .... DAP reference angles, i.e., center of deadband.

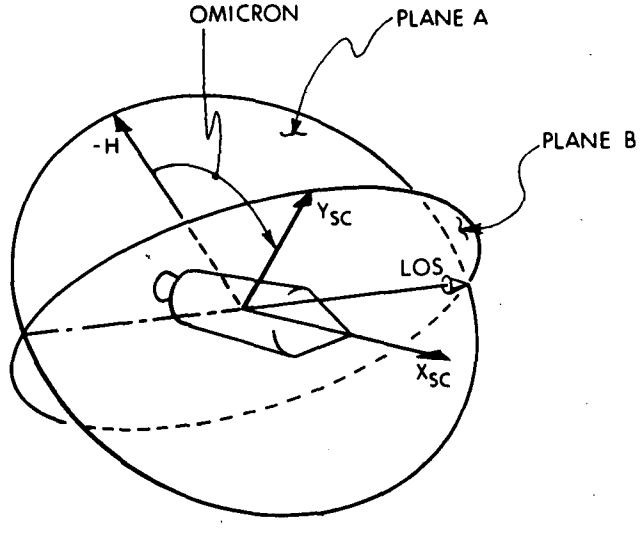
3) $\underline{\Delta}GA$ .... a vector defining the desired incremental changes in the IMU gimbal every 0.1 seconds.

These calculations are repeated once per second.

## 1.5 Option 4: Orient the Spacecraft as in Option 0, but Further Specify an Angle (OMICRON) of Rotation About SCAXIS

SCAXIS is as defined in Option 0; OMICRON is the positive-sense angle between planes A and B - where plane A is defined by the negative orbital angular momentum vector (-H) and the LOS, and plane B is defined by the spacecraft +Y-axis and the LOS.

Alignment of the LOS with either the spacecraft Y axis or the momentum vector is proscribed since the necessary two-plane condition for specifying OMICRON would not exist.



For Option 4, R61 provides the following quantities to the RCS DAP:

1) $\underline{\omega}_{CA}$ .... desired LOS rate in control-axis coordinates

2) $IGA_D$, $MGA_D$, $OGA_D$, .... DAP reference angles, i.e., center of the deadband.

3) $\Delta\underline{G}A$ .... a vector defining the desired incremental changes in the IMU gimbal angles every 0.1 second.

The desired LOS rate $\left(\underline{\omega}_{CA}\right)$ in control-axis coordinates is obtained as follows:

$$\underline{u}_{LOS} = \underline{r}_W - \underline{r}_C$$

$$\underline{\omega}_{LOS} = [REFSMMAT] \left\{ \frac{UNIT\ (\underline{u}_{LOS}) \times (\underline{v}_W - \underline{v}_C)}{\left|\underline{u}_{LOS}\right|} \right\}$$

$$\underline{\omega}_{CA} = [NBCA]\ [SMNB]\ \omega_{LOS},$$

Where $\underline{\omega}_{LOS}$ is the LOS rate in stable member cooridnates; $\underline{r}_W$, $\underline{v}_W$, $\underline{r}_C$, and $\underline{v}_C$ are the position and velocity vectors of the TV and ORB in basic reference coordinates.

The vector $(\Delta\underline{G}A)$ is computed as follows:

$$\Delta\underline{G}A = \begin{bmatrix} \cos IGA \sec MGA & 0 & -\sin IGA \sec MGA \\ -\cos IGA \tan MGA & 1 & \sin IGA \tan MGA \\ \sin IGA & 0 & \cos IGA \end{bmatrix} (0.1)\ \underline{\omega}_{LOS} + \underline{\omega}_{correction},$$

where

$$\underline{\omega}_{correction} = \begin{bmatrix} \theta_{DX} - OGA_D \\ \theta_{DY} - IGA_D \\ \theta_{DZ} - MGA_D \end{bmatrix} \frac{0.1}{\Delta t_C}$$

$\theta_{DX}$, $\theta_{DY}$, $\theta_{DZ}$ desired gimbal angles :s

$\Delta t_C$ = conservative estimate of R61 computation interval

The desired gimbal angles $\theta_{DX}$, $\theta_{DY}$, $\theta_{DZ}$ are computed as follows: First, four unit vectors are computed:

$$\underline{s}_A = \cos\ (OMICRON)\ unit\ (\underline{u}_{LOS} \times [\underline{v}_C \times \underline{r}_C]) + \sin\ (OMICRON)\ unit\left[\underline{u}_{LOS} \times \left(\underline{u}_{LOS} \times (\underline{v}_C \times \underline{r}_C)\right)\right]$$

$$\underline{s}'_A = unit\ (\underline{SCAXIS} \times \underline{y}_{SC})$$

$\underline{s}_B = \underline{u}_{LOS}$

$\underline{s}'_B = \underline{SCAXIS}$

Given these unit vectors, the unit vectors $\underline{x}$, $\underline{y}$, and $\underline{z}$ are calculated defining the desired directions of the x, y, and z axes of the navigation base with respect to the Basic Reference Coordinate System. The unit vectors $\underline{x}$, $\underline{y}$, and $\underline{z}$ and the vectors $\underline{x}_{SM}$, $\underline{y}_{SM}$, $\underline{z}_{SM}$ from the matrix REFSMMAT are then used to compute the desired gimbal angles $\theta_{DX}$, $\theta_{DY}$, $\theta_{DZ}$.

1.6 Option 5: Orient the Spacecraft as in Option 1, but Further Specify an Angle (OMICRON) of Rotation About SCAXIS.

SCAXIS is as defined in Option 0;
OMICRON is as defined in Option 4.

For Option 5, R6 quantities provided to the RCS DAP are as in Option 4, except as follows:

$\underline{u}_{LOS}$ = line of sight direction including aberration correction

and,

if the target is a star or planet

$\underline{\omega}_{LOS} = 0$;

if the target is the center of the earth,

$$\underline{\omega}_{LOS} = [\text{REFSMMAT}] \left\{ \frac{\text{UNIT } (\underline{r}_C) \times (\underline{v}_C)}{|r_C|} \right\}$$

SECTION 2

# ON-ORBIT ATTITUDE MANEUVER ROUTINE

by

John J. Turkovich

## 2.1 Introduction

The purpose of the attitude maneuver routine (ATTITUDE_MANEUVER) is to automatically reorient the spacecraft during coasting flight. ATTITUDE_MANEUVER is an adaptation of the R60 routine from the Apollo[5] flight programs to the Space Shuttle with the assumption that a 3-gimbal IMU is used. This implementation is coded in HAL and described in accordance with preliminary structured programming techniques. The general scope of variables and structure of ATTITUDE_MANEUVER is illustrated in Fig. 2-1. Each of the modules and variables used in the routine are defined in Tables 2-1 and 2-2 respectively (Subsection 2.7).

The desired attitude to which the spacecraft is to be aligned can be specified in one of two ways:

1) By a set of terminal desired gimbal angles, $\left[\theta_c\right]$.

2) By two unit vectors, $\underline{u}_{DSM}$ and $\underline{u}_{ANB}$.

where $\left[\theta_c\right]$ is the set of desired (or commanded) gimbal angles defined by

$$\left[\theta_c\right] = \begin{bmatrix} \theta_{c_o} \\ \theta_{c_i} \\ \theta_{c_m} \end{bmatrix} = \begin{bmatrix} \text{desired outer gimbal angle} \\ \text{desired inner gimbal angle} \\ \text{desired middle gimbal angle} \end{bmatrix}$$

$\underline{u}_{ANB}$ is a body-fixed axis vector, $\underline{u}_A$, with components in navigation-base-coordinates, $\underline{u}_{DSM}$ is a desired direction vector, $\underline{u}_D$, with components in stable-member coordinates and $\underline{u}_{DSM}$ specifies the direction in inertial space to which $\underline{u}_{ANB}$ is to be aligned. A flag switch, FLAG_3AXIS is provided to specify which of the two types of inputs is to be used.

If the vector type of input is chosen, the routine known as VECPOINT is used to compute a set of desired terminal gimbal angles, $[\theta_c]$, based on the values of $\underline{u}_{DSM}$, and $\underline{u}_{ANB}$, and the current gimbal angles, $[\theta]$. Regardless of the type of

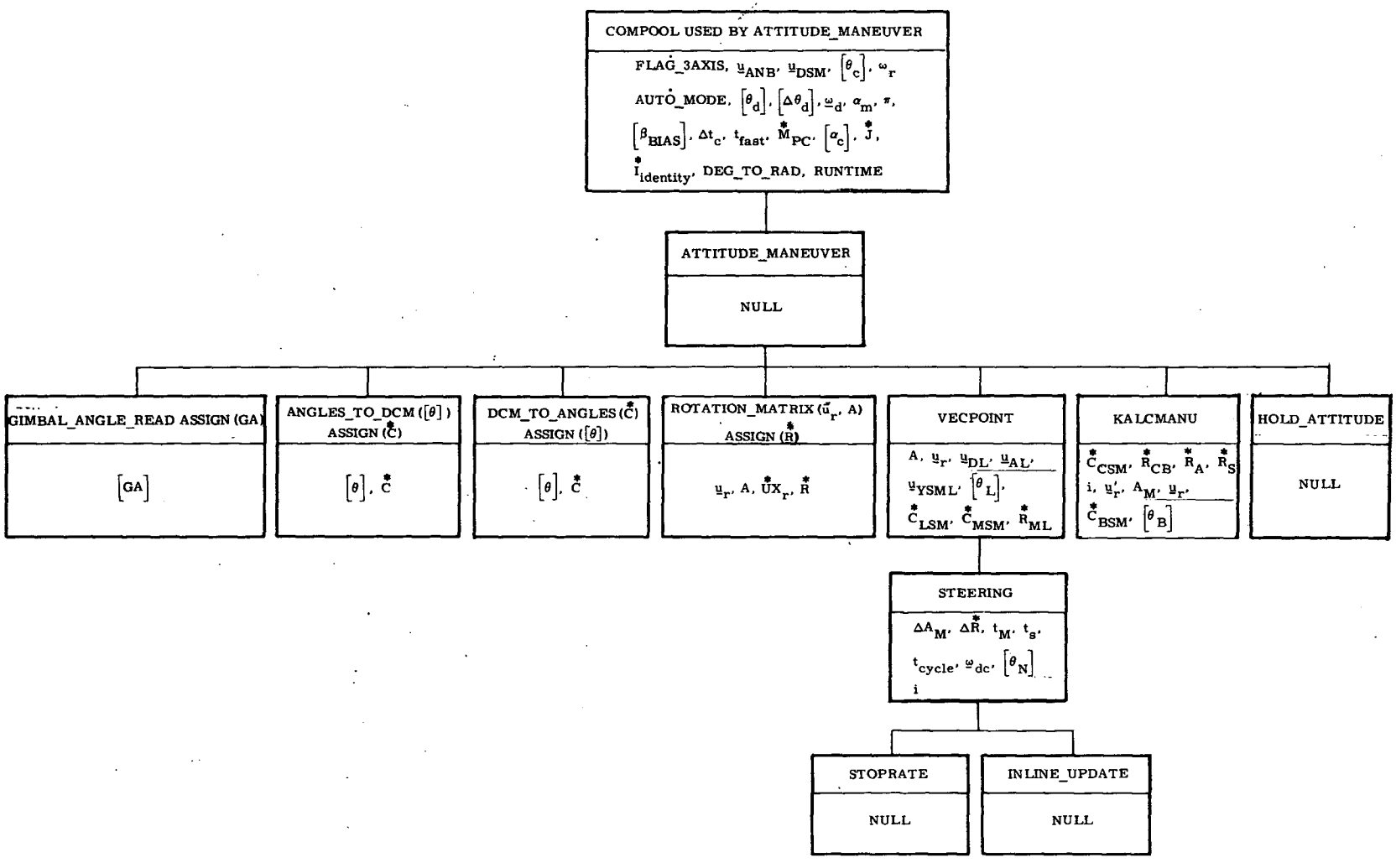



Fig. 2-1 ATTITUDE_MANEUVER Structure

input, KALCMANU is called if the automatic option is selected. Otherwise control returns to the executive system. KALCMANU computes a set of parameters ($\underline{u}_r$, A) that determine how the reorientation is to be performed. The maneuver is performed by rotating the spacecraft about the axis of single equivalent rotation, $\underline{u}_r$, by the angle $A_M$. The vector $\underline{u}_r$ and the angle $A_M$ are completely determined by the initial orientation, $[\theta]$, and the desired terminal orientation, $[\theta_c]$. Once $\underline{u}_r$ and $A_M$ are computed, KALCMANU calls the procedure STEERING.

The procedure, STEERING, cycles every $\Delta t_c$ seconds and computes, on each pass, five sets of quantities to be used by the digital autopilot in following the angular path defined by $\underline{u}_r$, A. The first of these quantities is a set of autopilot reference angles, $[\theta_d]$ (or intermediate desired CDU angles), which is updated regularly during the maneuver. To achieve a smoother sequence of commands between successive updates, the program generates a set of incremental CDU angles, $[\Delta\theta_d]$, to be added to $[\theta_d]$ by the digital autopilot. The steering routine also computes the component maneuver rates, $\underline{\omega}_d$ (in spacecraft coordinates), a set of bias accelerations, $\underline{\alpha}_m$, and a set of bias angles, $[\beta_{BIAS}]$. The bias angles, $[\beta_{BIAS}]$, are to be used by the autopilot to prevent fuel-consuming overshoot when starting and stopping an automatic maneuver.

The preceding method allows the autopilot to establish and maintain a spacecraft rotation about the vector $\underline{u}_r$ at a fixed rotation rate $\underline{\omega}_d$. The maneuvers are timed in open-loop fashion so that after a predetermined interval $[\Delta\theta_d]$, $\underline{\omega}_d$, $\underline{\alpha}_m$ and $[\beta_{BIAS}]$ are set to zero and the autopilot reference angles, $[\theta_d]$, are set equal to the terminal angles, $[\theta_c]$. Thus, upon completion of the maneuver, the spacecraft will finish up in a limit cycle about $[\theta_c]$.

## 2.2 ATTITUDE_MANEUVER

A logic flow diagram of ATTITUDE_MANEUVER is illustrated in Fig. 2-2. If the vector option is selected by the calling program, the routine VECPOINT is used to compute the terminal desired gimbal angles $[\theta_c]$ from $\underline{u}_{ANB}$ and $\underline{u}_{DSM}$. If the automatic maneuver is to proceed the automatic mode switch, AUTO_MODE, must be set "ON". If it is "ON", KALCMANU is called to calculate the rotation vector, $\underline{u}_r$, and the angle of rotation, $A_M$, and to initiate the cycling of STEERING.

## 2.3 VECPOINT

The desired gimbal angles, $[\theta_c]$, computed by VECPOINT are determined on the basis of minimizing the rotation angle required to align $\underline{u}_{ANB}$ with $\underline{u}_{DSM}$. In general, this procedure will reduce the fuel required for the maneuver by not constraining the orientation of the spacecraft about the pointing axis, defined by

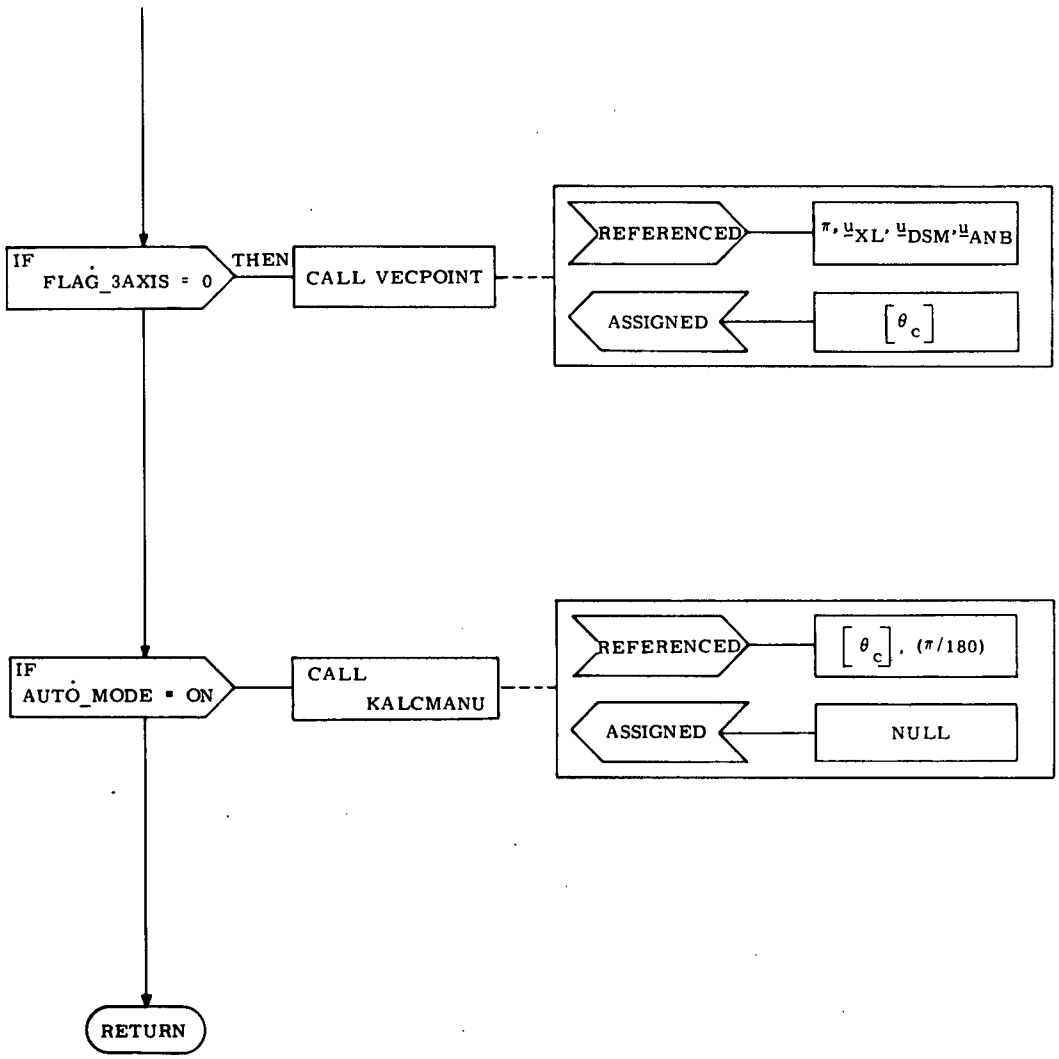



Fig. 2-2 ATTITUDE_MANEUVER : Program

$\underline{u}_{DSM}$. Additionally, the attitude about $\underline{u}_{DSM}$ will be left unchanged. Consequently, the crew may specify the attitude about the pointing axis to satisfy additional communication, visibility, or thermal constraints, if desired. This is done by manually reorienting the spacecraft about the pointing axis either prior to or after the maneuver. If this procedure is done after the automatic maneuver, a final trim will usually be required to assure that $\underline{u}_{ANB}$ is aligned with $\underline{u}_{DSM}$.

To compute the desired gimbal angles, $[\theta_c]$, from $\underline{u}_{DSM}$, $\underline{u}_{ANB}$, and the current gimbal angles, $[\theta]$, several service routines are required. These routines are described below.

### 2.3.1 ANGLES_TO_DCM

This procedure converts three gimbal angles, $[\theta]$, to a direction-cosine matrix, $\overset{*}{C}$, that relates the corresponding spacecraft navigation-base orientation to the stable-member frame. The formulas for this conversion are:

$$\overset{*}{C} = \begin{bmatrix} C_{1,1} & C_{1,2} & C_{1,3} \\ C_{2,1} & C_{2,2} & C_{2,3} \\ C_{3,1} & C_{3,2} & C_{3,3} \end{bmatrix}$$

$$C_{1,1} = \cos\theta_i \cos\theta_m$$

$$C_{1,2} = -\cos\theta_i \sin\theta_m \cos\theta_o + \sin\theta_i \sin\theta_o$$

$$C_{1,3} = \cos\theta_i \sin\theta_m \sin\theta_o + \sin\theta_i \cos\theta_o$$

$$C_{2,1} = \sin\theta_m$$

$$C2,2 = \cos\theta_m \cos\theta_o$$

$$C_{2,3} = -\cos\theta_m \sin\theta_o$$

$$C_{3,1} = -\sin\theta_i \cos\theta_m$$

$$C_{3,2} = \sin\theta_i \sin\theta_m \cos\theta_o + \cos\theta_i \sin\theta_o$$

$$C_{3,3} = -\sin\theta_i \sin\theta_m \sin\theta_o + \cos\theta_i \cos\theta_o$$

where

$$\theta_o = \text{outer gimbal angle}$$

$$\theta_i = \text{inner gimbal angle}$$

$$\theta_m = \text{middle gimbal angle}$$

Also see Fig. 2-3 for the logic flow diagram of ANGLES_TO_DCM.

The interpretation of this matrix is as follows: If $a_x$, $a_y$, $a_z$ represent the components of a vector in navigation-base axes, then the components of the same vector in stable-member axes ($b_x$, $b_y$, $b_z$) are

$$\begin{Bmatrix} b_x \\ b_y \\ b_y \end{Bmatrix} = \overset{*}{C}(\theta) \cdot \begin{Bmatrix} a_x \\ a_y \\ a_z \end{Bmatrix}$$

## 2.3.2 DCM_TO_CDU ANGLES

This procedure extracts the CDU angles from a direction-cosine matrix, $\overset{*}{C}$, that relates the navigation-base coordinates to the stable-member coordinates. The formulas for this conversion are

$$\theta_m = \arcsin C_{2,1}$$

$$\theta_i = \arcsin(-C_{3,1}/\cos\theta_m)$$

$$\left[\text{If } C_{1,1} \text{ is negative, } \theta_i \text{ is replaced by } \pi\ \mathrm{SGN}(\theta_i) - \theta_i\right]$$

$$\theta_o = \arcsin(-C_{2,3}/\cos\theta_m)$$

$$\left[\text{If } C_{2,2} \text{ is negative, } \theta_o \text{ is replaced by } \pi\ \mathrm{SGN}(\theta_o) - \theta_o\right]$$

The logic flow diagram is illustrated in Fig. 2-4.

## 2.3.3 ROTATION_MATRIX

This procedure computes the direction-cosine matrix, $\overset{*}{R}$, that relates one coordinate frame to another that is rotated with respect to the first by an angle A about a unit vector $\underset{\sim}{u}_r$. The formula for this matrix is

$$\overset{*}{R}(\underset{\sim}{u}_r, A) = \overset{*}{I}\cos A + \underset{\sim}{u}_r\,\underset{\sim}{u}_r^T(1-\cos A) + \overset{*}{U}X_r \sin A$$

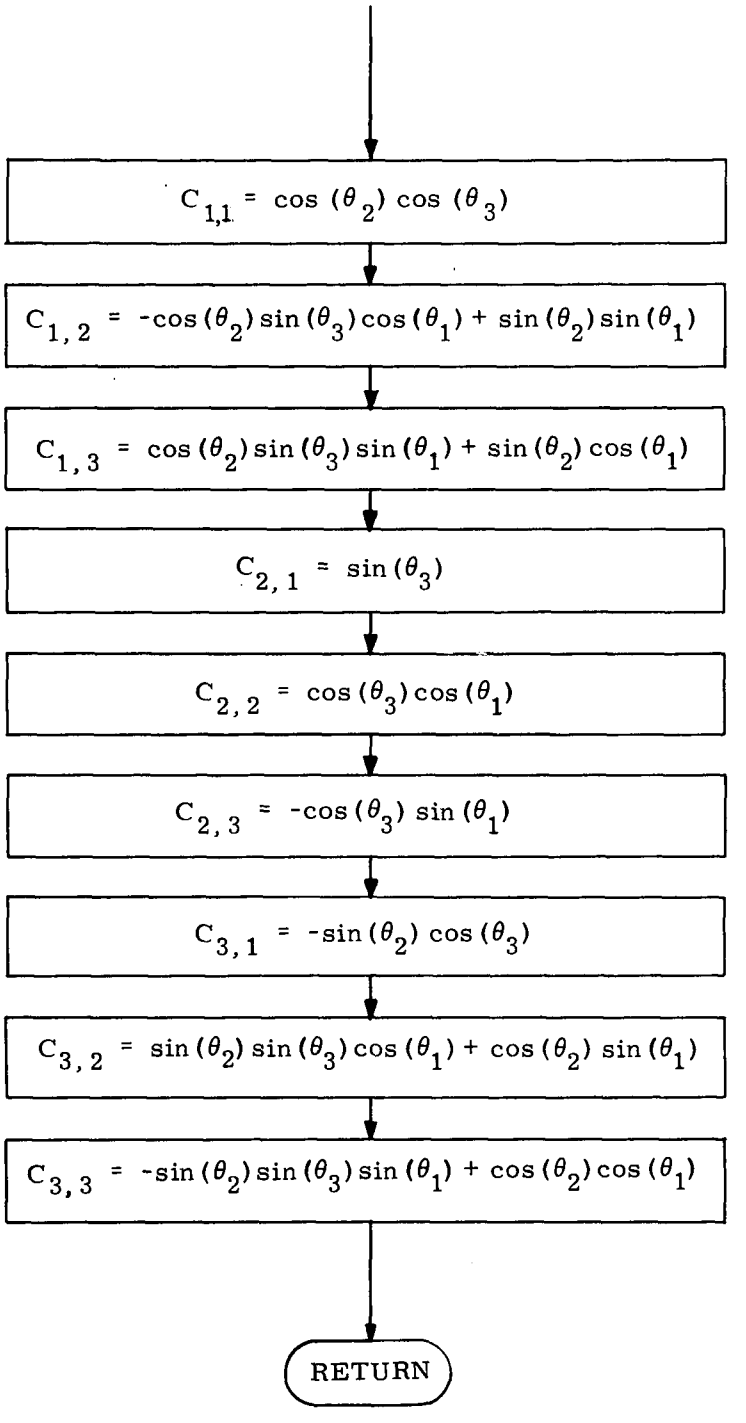



Fig. 2-3 ANGLES_TO_DCM : Procedure ($[\theta]$) Assign ($\overset{*}{C}$)

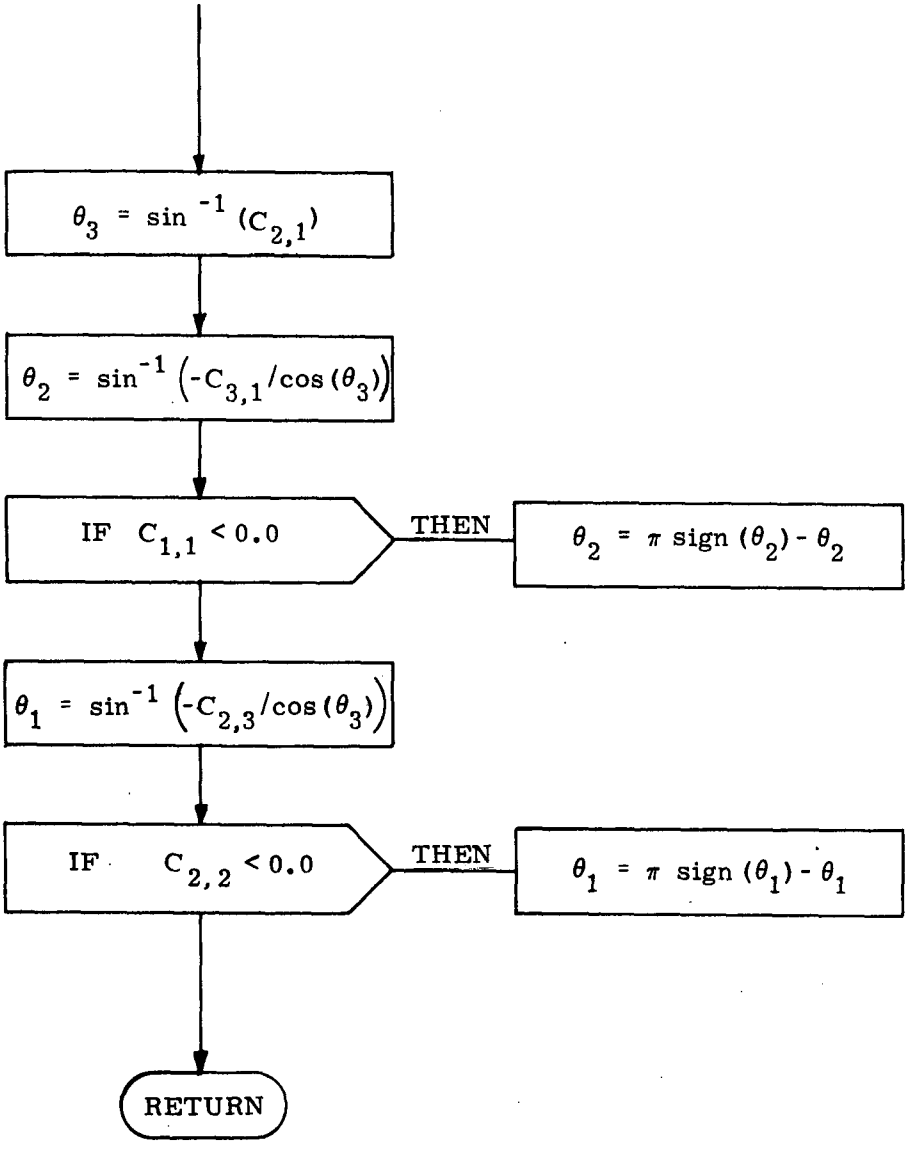



Fig. 2-4 DCM_TO_ANGLES : Procedure ($\overset{*}{C}$) Assign ([$\theta$])

where

$$\overset{*}{I} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\underline{u}_r \underline{u}_r^T = \begin{bmatrix} u_{rx}^2 & u_{rx} u_{ry} & u_{rx} u_{rz} \\ u_{ry} u_{rx} & u_{ry}^2 & u_{ry} u_{rz} \\ u_{rz} u_{rx} & u_{rz} u_{ry} & u_{rz}^2 \end{bmatrix}$$

$$\overset{*}{U}X_r = \begin{bmatrix} 0 & -u_{rx} & u_{ry} \\ u_{rz} & 0 & -u_{rx} \\ -u_{ry} & u_{rx} & 0 \end{bmatrix}$$

$\underline{u}_r$ = unit rotation vector resolved into spacecraft axes

A = rotation angle

The logic flow diagram is illustrated in Fig. 2-5.

The interpretation of $\overset{*}{R}$ is as follows: If $a_x$, $a_y$, $a_z$ represent the components of a vector in the rotated frame, then the components of the same vector in the original frame ($b_x$, $b_y$, $b_z$) are

$$\begin{Bmatrix} b_x \\ b_y \\ b_z \end{Bmatrix} = \overset{*}{R}(\underline{u}_r, A) \begin{Bmatrix} a_x \\ a_y \\ a_z \end{Bmatrix}$$

The process used by VECPOINT in determining the terminal desired angles, $[\theta_c]$, is illustrated schematically in Fig. 2-6 and in logic flow form in Fig. 2-7. VECPOINT uses two frames of reference: the L frame and the M frame - as well as the stable-member (SM) frame. The L frame is simply a base frame used as the starting point for subsequent rotations. In VECPOINT computations, the L frame is the initial navigation-base frame computed from the initial gimbal angles, $[\theta]$. The vector $\underline{u}_D$ is resolved from the stable-member frame into this frame as $\underline{u}_{DL}$ using the transformation $\overset{*}{C}{}^T_{LSM}$. The L frame is then rotated so that $\underline{u}_{AL}$ and $\underline{u}_{DL}$ are coincident. The rotation vector, $\underline{u}_r$, is defined by the cross product

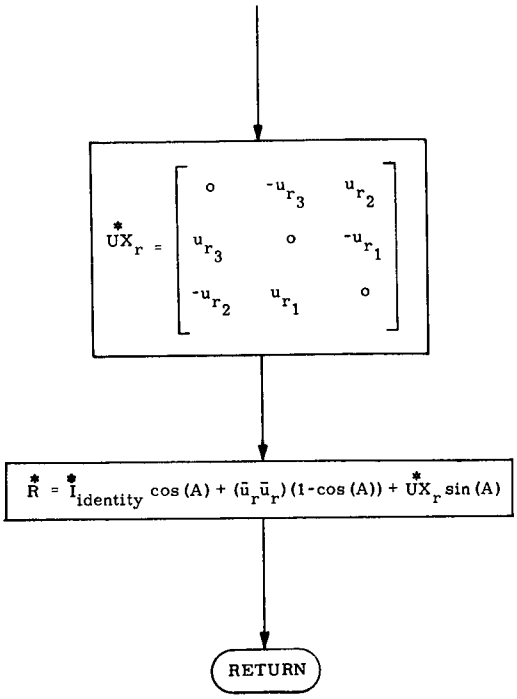



Fig. 2-5 ROTATION_MATRIX:
Procedure ($\underline{u}_r$, A) Assign ($\overset{*}{R}$)

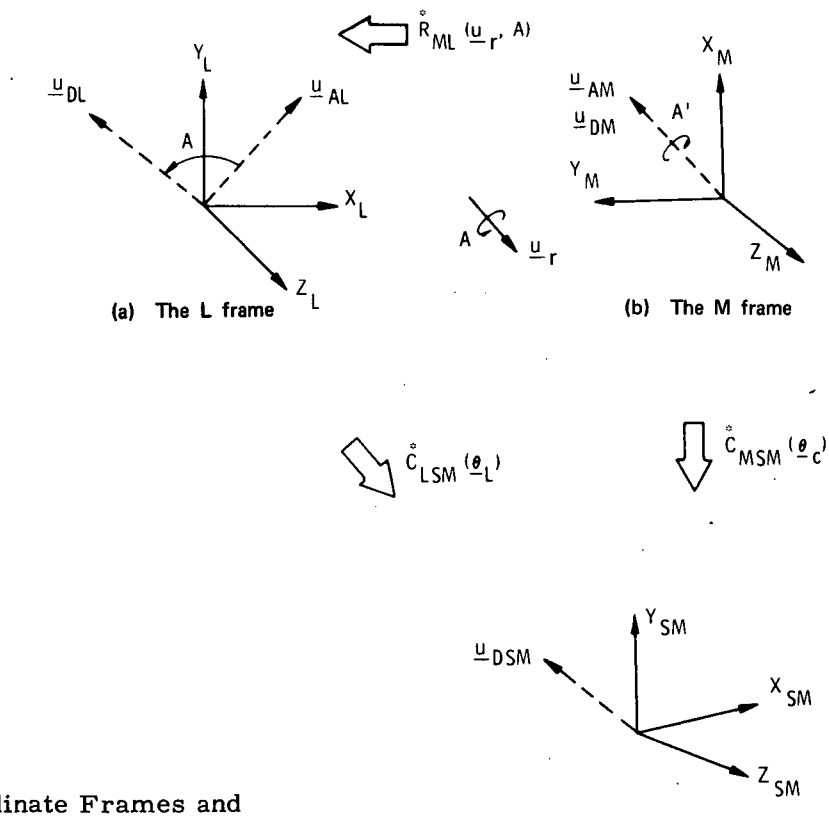



Fig. 2-6 The Coordinate Frames and
Transformations used in VECPOINT

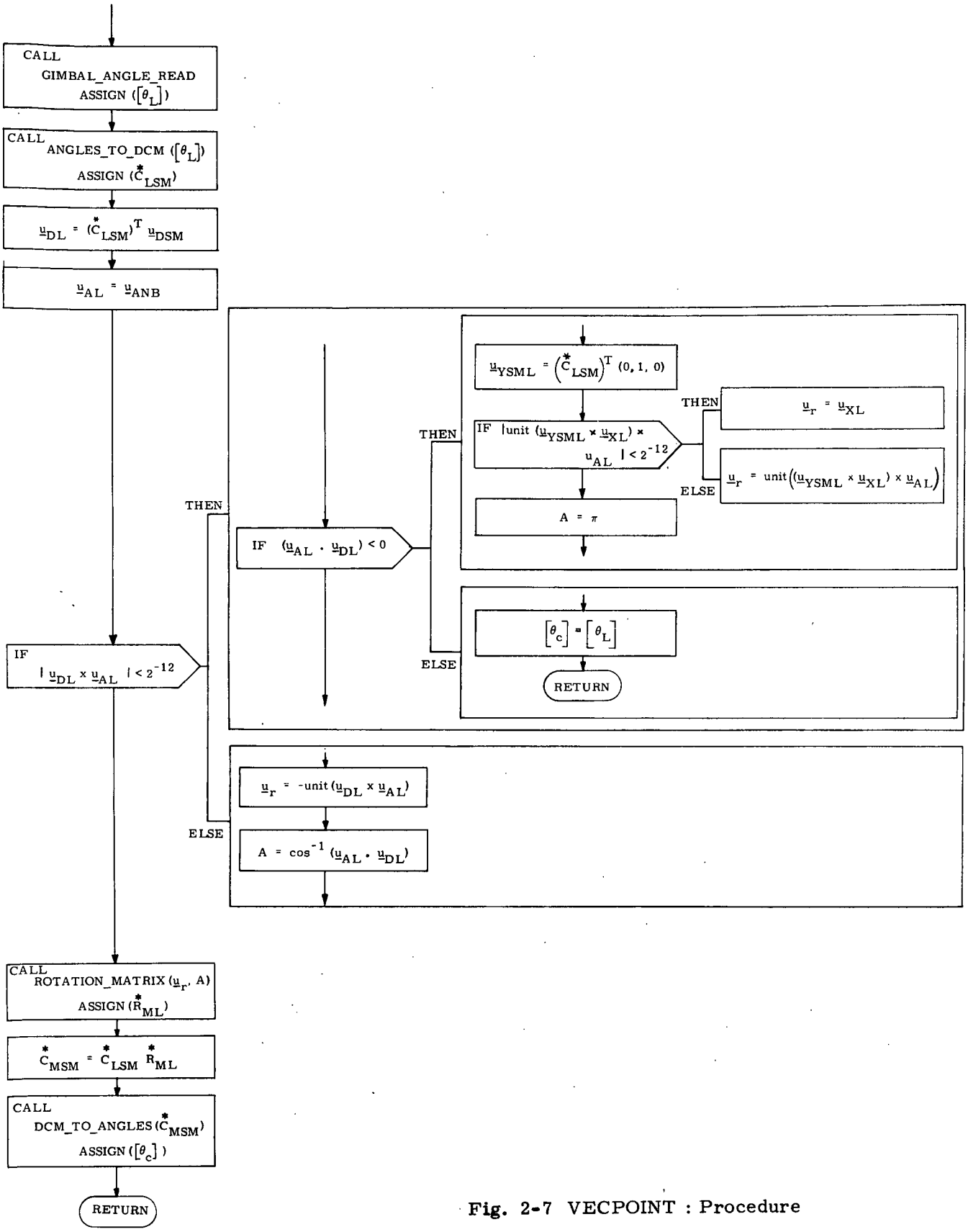



**Fig. 2-7 VECPOINT : Procedure**

between $\underline{u}_{AL}$ and $\underline{u}_{DL}$. Rotation about the cross-product vector will assure that spacecraft maneuvering will be minimized and that the rotation vector will have no component in the $\underline{u}_{AL}$ or $\underline{u}_{DL}$ direction.

If the two vectors, $\underline{u}_{AL}$ and $\underline{u}_{DL}$, are either parallel or antiparallel, a special computation of $\underline{u}_r$ is performed. When $\underline{u}_{AL}$ and $\underline{u}_{DL}$ are almost parallel then $[\theta_c]$ is set equal to $[\theta_L]$. If the two vectors are almost antiparallel, then a $180^o$ rotation is required for alignment. In this case, the rotation vector is chosen so that $\underline{u}_r$ is perpendicular to $\underline{u}_{AL}$ and lies in the plane that contains $\underline{u}_{YSM(L)}$ and $\underline{u}_{XL}$. If, however, $\underline{u}_{AL}$ happens to be perpendicular to the plane, then any vector in the plane may be used for $\underline{u}_r$. In this event the x-axis is chosen. From $\underline{u}_r$ and A the rotation matrix, $\overset{*}{R}_{ML}$, between the M- and L- frames is computed. Multiplying $\overset{*}{R}_{ML}$ by $\overset{*}{C}_{LSM}$ determines the direction - cosine matrix $\overset{*}{C}_{MSM}$ from which the terminal desired gimbal angles, $[\theta_C]$, are extracted.

## 2.4 KALCMANU Maneuver-Parameters Calculation

The procedure KALCMANU is designed to generate commands for the autopilot to reorient the spacecraft from an initial attitude to some desired attitude specified by $[\theta_c]$. Although a number of authors have discussed optimal solutions to this problem, the task is a formidable one because of the nonlinear nature of rigid-body dynamics and the kinematics of rotational motion. Optimal solutions for a particular class of spacecraft configurations are treated in Reference 1. In this reference, the criterion for optimality is based upon the product of maneuver duration and fuel expenditure. Using the same criterion, previous design studies based upon a number of suboptimal approaches[2] have shown that a reasonable design compromise is to rotate the spacecraft about a single inertially fixed axis. This technique is employed in KALCMANU.

The process used by KALCMANU in computing the rotation axis, $\underline{u}_r$, and the rotation angle, $A_M$, is illustrated in Fig. 2-8. From the current gimbal angles KALCMANU computes the transformation matrix between the initial spacecraft axes and the stable-member axes, $\overset{*}{C}_{BSM}$. This procedure also computes the matrix relating the desired terminal spacecraft attitude and the stable-member axes, $\overset{*}{C}_{CSM}$. The transformation between the desired and the initial attitude is $\overset{*}{R}_{CB}$, where where

$$\overset{*}{R}_{CB} = \overset{*}{C}_{BSM}^{T} \overset{*}{C}_{CSM}$$

$\overset{*}{R}_{CB}$ can be partitioned into its symmetric and antisymmetric components as follows. For the symmetric part,

$$\overset{*}{R}_S = \frac{1}{2} (\overset{*}{R}_{CB} + \overset{*}{R}_{CB}^{T})$$

By comparing this with the equations described in ROTATION_MATRIX it can be seen that

$$\overset{*}{R}_S = \overset{*}{I} \cos A_M + \underline{u}_r \underline{u}_r^T (1-\cos A_M)$$

For the antisymmetric part,

$$\overset{*}{R}_A = (\tfrac{1}{2})(\overset{*}{R}_{CB} - \overset{*}{R}_{CB}^T)$$

or

$$\overset{*}{R}_A = \begin{bmatrix} 0 & -u_{rz} & u_{ry} \\ u_{rz} & 0 & -u_{rx} \\ -u_{ry} & u_{rx} & 0 \end{bmatrix} \sin A_M$$

The maneuver angle can easily be obtained from $\overset{*}{R}_{CB}$ as

$$A_M = \cos^{-1}\left(\frac{\text{Trace}(\overset{*}{R}_{CB}) - 1}{2}\right)$$

The rotation vector, $\underline{u}_r$, is normally extracted from the antisymmetric part of $\overset{*}{R}_{CB}$; that is,

$$\underline{u}_r = \text{unit}\,(-R_{A_{3,3}},\ R_{A_{1,3}},\ -R_{A_{1,2}})$$

As sin $A_M$ approaches zero, however, the unit operation becomes indeterminate and an alternative calculation must be performed. If the maneuver angle is less than 0.25 deg, then no complicated automatic maneuvering is necessary and therefore one can simply set the autopilot reference angle, $[\theta_d]$, equal to the terminal desired angles $[\theta_c]$ and exit. This is done by calling HOLD_ATTITUDE. If $A_M > 170$ deg, a method employing the symmetric part of $\overset{*}{R}_{CB}$ is used to determine $\underline{u}_r$, as illustrated in Fig. 2-8. The magnitudes of the components of $\underline{u}_r$ are easily determined as

$$|u_{rx}| = \sqrt{\frac{R_{S1,1} - \cos A_M}{1 - \cos A_M}}$$

$$|u_{ry}| = \sqrt{\frac{R_{S2,2} - \cos A_M}{1 - \cos A_M}}$$

$$|u_{rz}| = \sqrt{\frac{R_{S3,3} - \cos A_M}{1 - \cos A_M}}$$

To obtain the signs of each component, it is first necessary to determine the sign of the largest component by referring to the values of $\overset{*}{R}_A$. This is done to avoid the problem caused by truncation as sin $A_M$ approaches zero. If $A_M$ is equal to 180 deg, then the sign of the largest component may be arbitrarily chosen. The signs of the remaining components may be determined by examining the off-diagonal terms in $\overset{*}{R}_S$.

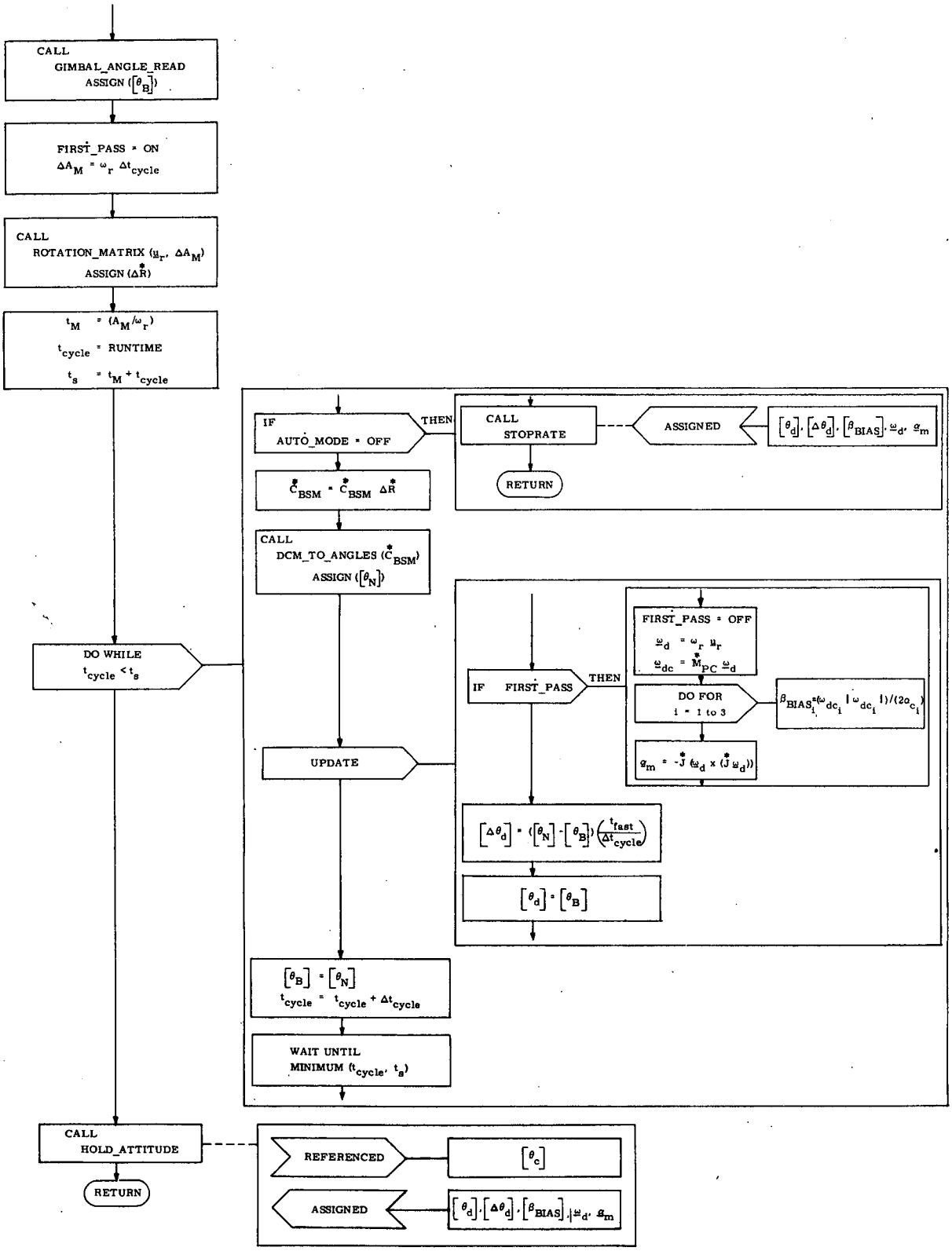



**Fig. 2-8 KALCMANU : Procedure**

## 2.5 Steering Through a Maneuver

The KALCMANU parameter calculations described in the preceding subsection solve the problem of what angular path to take in order to reorient the spacecraft without excessive fuel expenditure. However, the problem of how to steer the spacecraft along this path must now be considered. One solution is to iteratively solve for $\underline{u}_r$ and $A_M$ explicitly as a function of the current spacecraft attitude and to control the spacecraft angular velocity, $\underline{\omega}$, by firing the RCS jets so that $\underline{\omega}$ is parallel to $\underline{u}_r$. This approach, however, leads to several problems. First of all, because of the deadband requirements of the autopilot, and control inaccuracies, explicit guidance can lead to excessive changes in control commands as the spacecraft approaches the terminal attitude. This problem of terminal guidance is characteristic of explicit guidance schemes unless special consideration is given to the terminal-control problem. Secondly, there is a timing problem since the computation of $\underline{u}_r$ and $A_M$ is somewhat lengthy and would require an excessive amount of computer time. Instead, it is better to formulate the problem in terms of perturbations from a smooth reference trajectory that satisfies the end-point constraints. The method employed is to solve the global problem once via KALCMANU in order to establish a reference trajectory (defined by $\underline{u}_r$, $A_M$) and then have the autopilot solve the micro-problem of controlling the spacecraft deviations from this reference trajectory, using small-angle approximations. This approach not only avoids the terminal-guidance problem and the timing problem but also provides "tighter" path control for satisfying other boundary conditions that may be placed upon the selection of the reference trajectory.

In order to determine the required steering interface between KALCMANU and the autopilot, it is first necessary to define the nature of the autopilot control variables. To simplify the discussion, the problem of rotation about a single axis is considered first. There are two state variables of interest; namely, the attitude error, $\theta_e$, defined by

$$\theta_e = \theta - \theta_d$$

where

$\theta$ = actual vehicle attitude

$\theta_d$ = desired vehicle attitude (constant)

and the angular velocity of the vehicle, $\omega$. The "state" equations are

$$\dot{\theta}_e(t) = \omega(t)$$

and

$$\dot{\omega}(t) = f(\theta_e, \omega)$$

where f $(\theta_e, \omega)$ is the control acceleration (applied by the jets) as determined by non-linear phase-plane switching logic. The control law is defined so that $\theta_e$ and $\omega$ will be nulled to within the control-law deadbands so as to maintain attitude hold about the reference attitude, $\theta_d$.

New state variables can be defined as follows:

$$\phi_e(t) = \theta(t) - \phi_d(t) + \beta_{BIAS}$$

and

$$\omega_e(t) = \omega(t) - \omega_d$$

where

$$\phi_d(t) = \phi_d(t_o) + (\omega_d)(t - t_o)$$

$$\omega_d = \text{a constant}$$

$$\beta_{BIAS} = \text{a constant}$$

$$\phi_d(t_o) = \theta(t_o) = \text{constant}$$

The state equations for the new variables are

$$\phi_e = \omega_e$$

$$\omega_e = f(\phi_e, \omega_e)$$

With these control variables, the autopilot will attempt to null the rate error, $\omega_e$, thus forcing the spacecraft to rotate at a constant rate. Note that at the beginning of the maneuver the desired angle $\phi_d(t_o)$ is set equal to the initial angle $\theta(t_o)$. The constant, $\beta_{BIAS}$, is added to the attitude error to prevent overshoot when starting and stopping the maneuver, $\beta_{BIAS}$ is computed as

$$\beta_{BIAS} = \frac{\omega_d \, |\omega_d|}{2 \, |\alpha_c|}$$

where

$$\alpha_c = \text{the control acceleration}$$

To interpret the effect of $\beta_{BIAS}$, refer to Fig. 2-9a. This figure shows a simplified phase-plane logic and a typical phase-point trajectory. The maneuver starts at point A with $\theta_e = 0$ and $\omega = 0$. With the application of steering commands, the phase-point instantaneously jumps to a biased condition at B where $\phi_e = \beta_{BIAS}$ and $\omega_e = -\omega_d$ ($\omega_d$ is negative for the case illustrated). The phase-point drifts to point C, where the control jets are turned on to reduce the rate error toward zero

at point D. The trajectory then traverses a limit cycle DEFG until the maneuver finishes at point H, where the biases $\beta_{BIAS}$ and $\omega_d$ are removed. At this point, the trajectory jumps to an unbiased condition at point I. Note that the actual rate is approximately equal to the desired rate and the phase-point drifts to point J, where the jets decelerate the vehicle back into a limit cycle around the desired attitude. The situation without angular bias, $\beta_{BIAS}$, is illustrated in Fig. 2-9b. Note the fuel-consuming overshoot in this case. The inclusion of the bias term is based on the assumption that the initial rates are small and that the desired rates are achievable during the maneuver. For the three-dimensional problem, it must also be assumed that $\beta_{BIAS}$ is small enough so that the small-angle assumptions are not violated.

A bias acceleration, $\underline{\alpha}_m$, is also computed and sent to the autopilot to better approximate the switching curves in the phase-plane design. In the formulation of the phase-plane design a simplification is made to the equation of motion

$$\dot{\underline{\omega}}(t) = \overset{*}{J}^{-1} \underline{M} - \overset{*}{J}^{-1} (\underline{\omega} \times \overset{*}{J}\underline{\omega})$$

where

$\underline{M}$ = the torque supplied by the jets

$J^{-1}\underline{M}$ = $\underline{\alpha}_c$, the control acceleration

$-\overset{*}{J}^{-1} (\underline{\omega} \times \overset{*}{J}\underline{\omega})$ = $\underline{\alpha}_m$, the acceleration due to the non-diagonal moment of inertia $\overset{*}{J}$.

The bias acceleration, $\underline{\alpha}_m$, is assumed small and neglected to arrive at the state equations

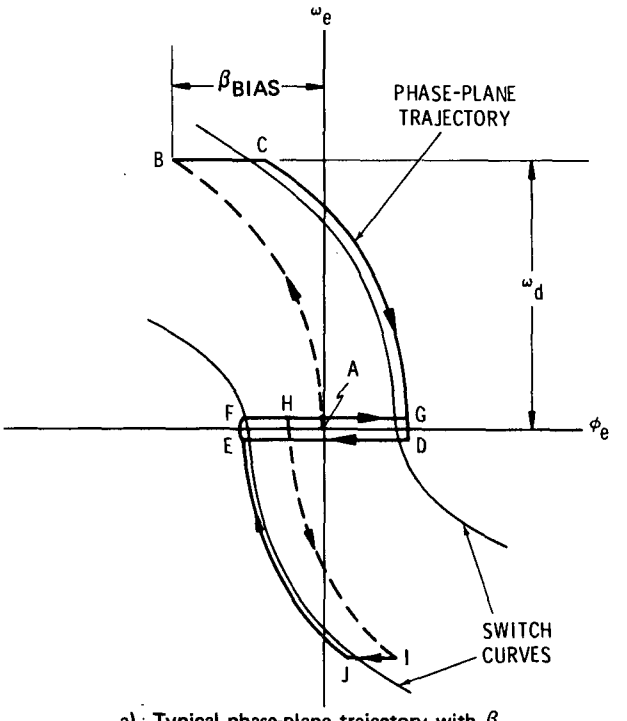
$$\begin{bmatrix} \dot{\theta}(t) - \dot{\theta}_d(t) \\ \dot{\omega}(t) - \dot{\omega}_d(t) \end{bmatrix} = \begin{bmatrix} \theta_e \\ \omega_e \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \underline{\theta}_e \\ \underline{\omega}_e \end{bmatrix} + \begin{bmatrix} 0 \\ \underline{\alpha}_c \end{bmatrix}$$

and allow the design of three independent phase-planes. To refine the determination of the switching curves, $\underline{\alpha}_m$ is sent to the autopilot and is used in estimating the net acceleration.

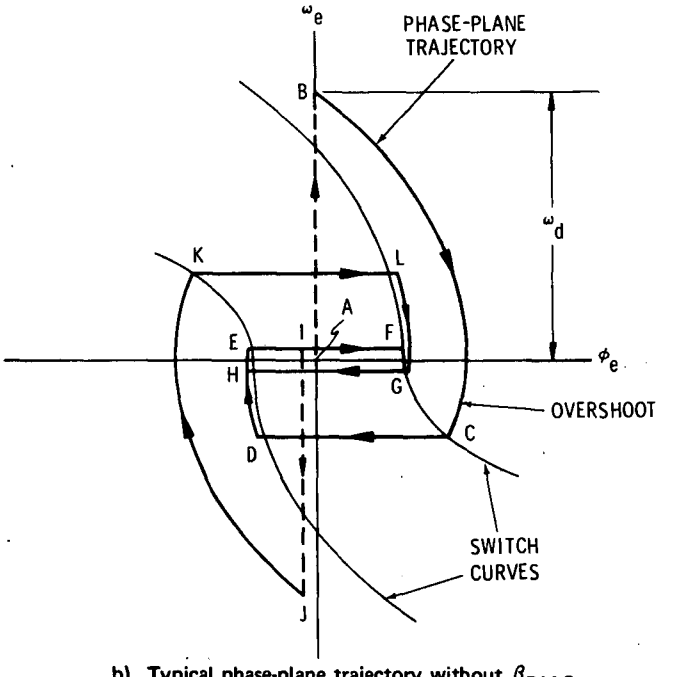
The above steering logic has been implemented in the procedure STEERING which is described below.

## 2.6 STEERING

The procedure STEERING computes five sets of steering parameters to be sent to the autopilot. These parameters are




a) Typical phase-plane trajectory with $\beta_{BIAS}$




b) Typical phase-plane trajectory without $\beta_{BIAS}$

Fig. 2-9 Maneuver Phase-Plane Trajectories

1) $[\theta_d]$, a set of intermediate desired gimbal angles

2) $[\Delta\theta_d]$, a set of incremental desired gimbal angles

3) $[\beta_{BIAS}]$, a set of bias gimbal angles

4) $\underline{\omega}_d$, the desired angular velocity vector

5) $\underline{\alpha}_m$, a set of bias accelerations

STEERING operates on a cyclic basis with period $\Delta t_c$. The parameters $[\beta_{BIAS}]$, $\underline{\omega}_d$, and $\underline{\alpha}_m$ are computed once on the first pass whereas $[\theta_d]$ and $[\Delta\theta_d]$ are computed every $\Delta t_c$ seconds until the maneuver is complete.

A logic flow diagram is illustrated in Fig. 2-10. The initial gimbal angles are read and the FIRST_PASS switch is set "ON". The incremental angle of rotation is computed from the requested magnitude of angular velocity and the time increment $\Delta t_c$. From this incremental rotation angle and the rotation vector $\underline{u}_r$, an incremental rotation matrix $\Delta\overset{*}{R}$ is determined by calling ROTATION_MATRIX. Two timing parameters are computed. These are the duration of the maneuver, $t_M$,

$$t_M = A_M/\omega_r$$

and the time at which to stop the maneuver, $t_s$,

$$t_s = t_M + t_{cycle}$$

where $t_{cycle}$ has been set equal to the current time.

Cyclic computation of the steering parameters takes place in open-loop fashion until the computed time to stop the maneuver, $t_s$, is reached (Fig. 2-10). If the AUTO_MODE switch is "OFF" then STOPRATE is called which sets the desired reference gimbal angles $[\theta_d]$ to the present gimbal angles and nulls the remaining four sets of parameters $[\Delta\theta_d]$, $[\beta_{BIAS}]$, $\underline{\omega}_d$, and $\underline{\alpha}_m$, as illustrated in Fig. 2-11. Control then returns to the executive system. However, if the AUTO_MODE switch is "ON" the direction-cosine matrix between the navigation-base and stable-member coordinate systems is determined by

$$\overset{*}{C}_{BSM_{N+1}} = \overset{*}{C}_{BSM_N} \Delta\overset{*}{R}$$

From this matrix, the gimbal angles projected to the time of the next pass, $[\theta_N]$, are calculated.

On the first pass STEERING sets the FIRST_PASS flag to "OFF" and computes $\underline{\omega}_d$, $[\beta_{BIAS}]$, and $\underline{\alpha}_m$ as illustrated in Fig. 2-10. The desired angular velocity vector, $\underline{\omega}_d$, is computed from the requested magnitude of angular velocity, $\omega_r$, and the rotation vector, $\underline{u}_r$, supplied by KALCMANU.

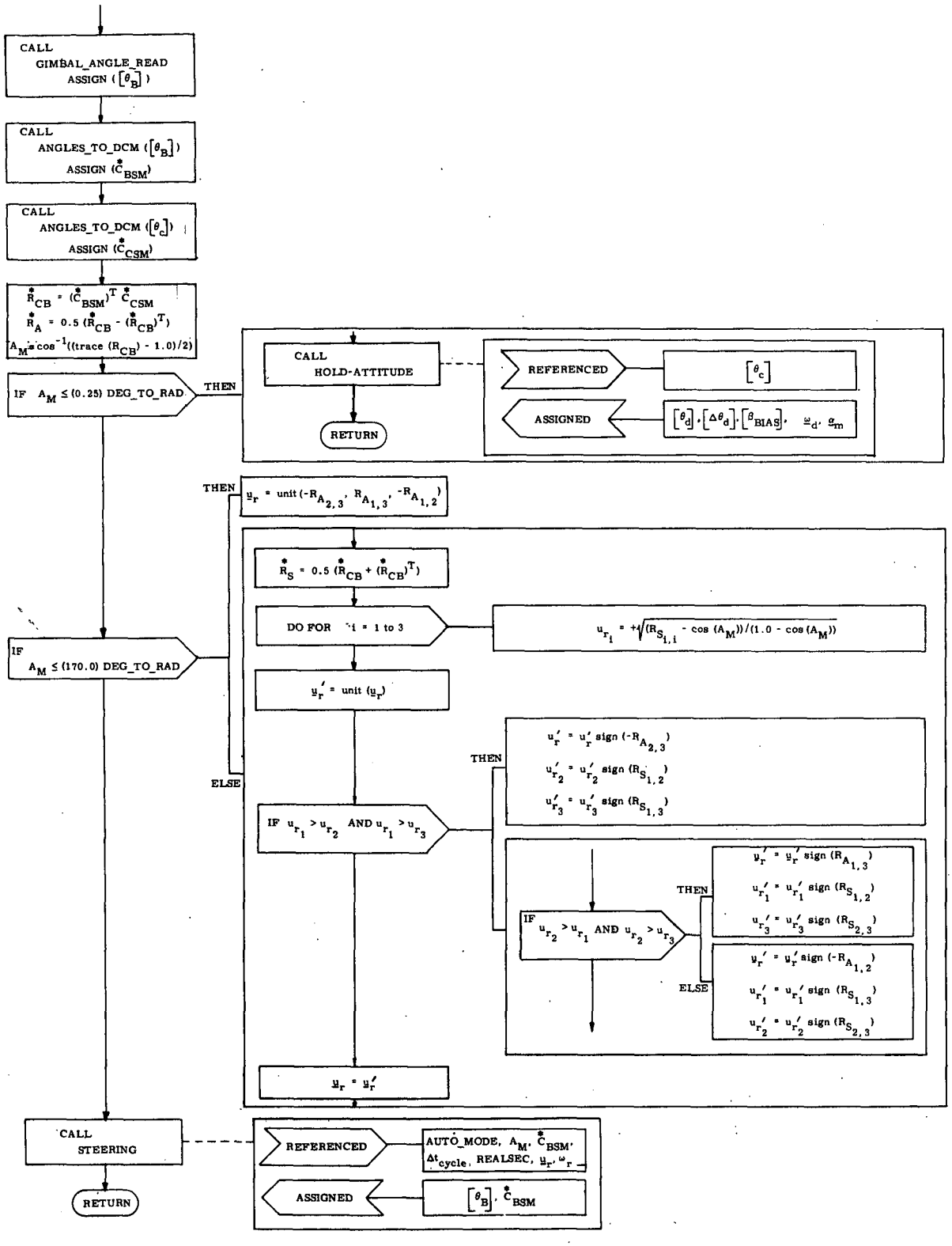



**Fig. 2-10 STEERING : Procedure**

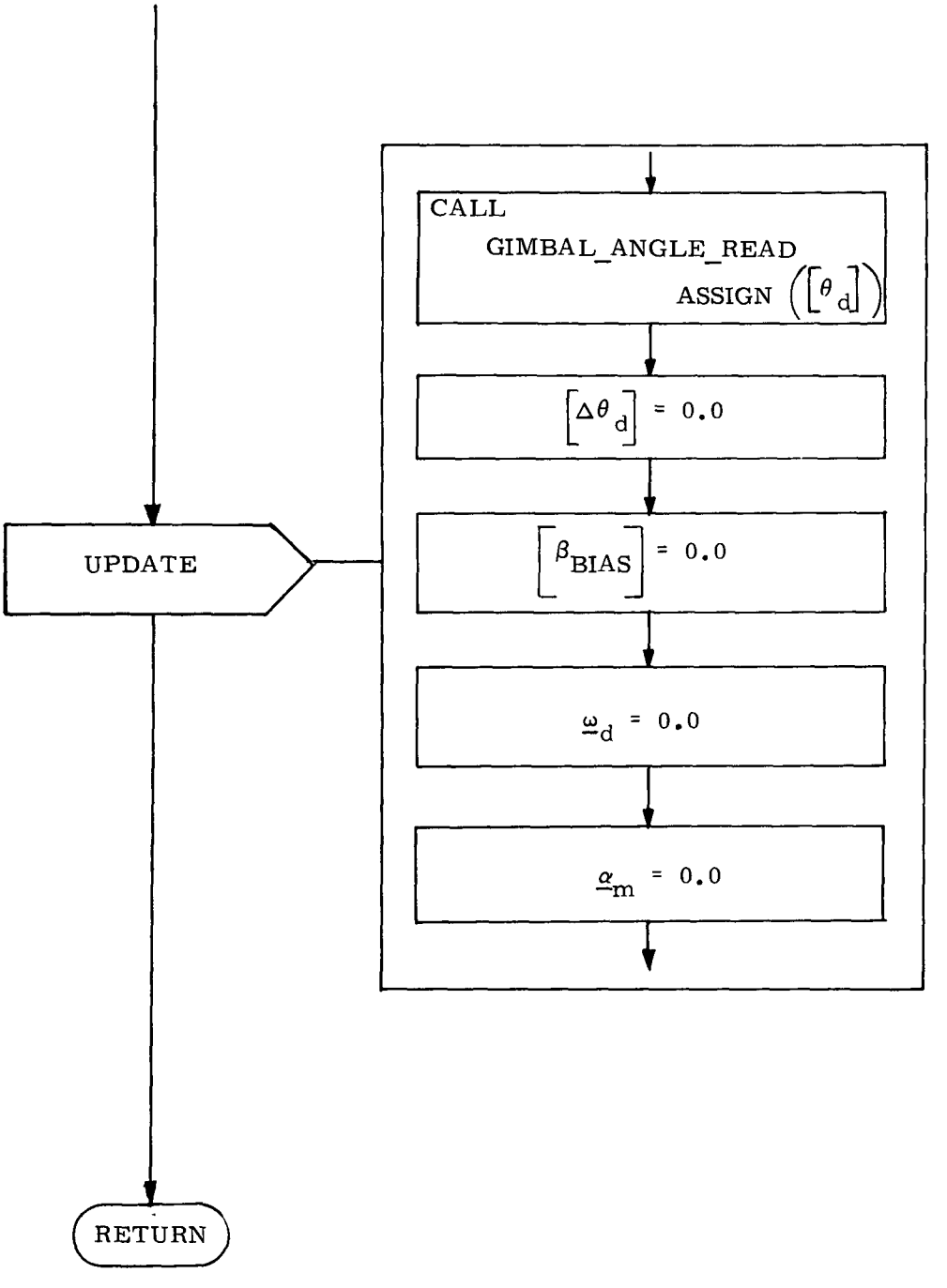



Fig. 2-11 STOPRATE : Procedure

$$\left[\beta_{BIAS}\right] = \begin{bmatrix} \beta_{BIAS(U)} \\ \beta_{BIAS(V)} \\ \beta_{BIAS(W)} \end{bmatrix} = \begin{bmatrix} \omega_{dc(U)} \; | \omega_{dc(U)} | \; / \; (2|\alpha_{c(U)}|) \\ \omega_{dc(V)} \; | \omega_{dc(V)} | \; / \; (2|\alpha_{c(V)}|) \\ \omega_{dc(W)} \; | \omega_{dc(W)} | \; / \; (2|\alpha_{c}(W)|) \end{bmatrix}$$

where $\alpha_{c(U)}$, $\alpha_{c(V)}$, $\alpha_{c(W)}$ are the control accelerations as determined by the autopilot.

On every pass STEERING also updates the incremental desired gimbal angles, $[\Delta\theta_d]$, by

$$[\Delta\theta_d] = ([\theta_N] - [\theta_B])\,(t_{fast}/\Delta t_c)$$

where

$[\theta_N]$ = desired reference gimbal angles projected $\Delta t_c$ seconds in advance

$[\theta_B]$ = present desired reference gimbal angles

$\Delta t_c$ = period of the STEERING cycle

$t_{fast}$ = sampling period of the autopilot

These incremental angles are subsequently added to $[\theta_d]$ by the autopilot at the start of each autopilot pass. $[\theta_d]$ is then updated to $[\theta_B]$ and $[\theta_B]$ is replaced by $[\theta_N]$ for use on the subsequent pass. The time of the next proposed pass, $t_{cycle}$, is computed by adding $\Delta t_c$ to the present value of $t_{cycle}$. If $t_{cycle}$ is less than the termination time of the maneuver, $t_s$, another pass is performed. Otherwise, HOLD_ATTITUDE is called to update the desired reference gimbal angles, $[\theta_d]$, to the commanded terminal set of gimbal angles $[\theta_c]$ and nulls the remaining steering parameters $[\Delta\theta_d]$, $[\beta_{BIAS}]$, $\underline{\omega}_d$, and $\underline{\alpha}_m$. This updating is illustrated in Fig. 2-12. Control is then returned to ATTITUDE_MANEUVER which in turn transfers control to the executive system.

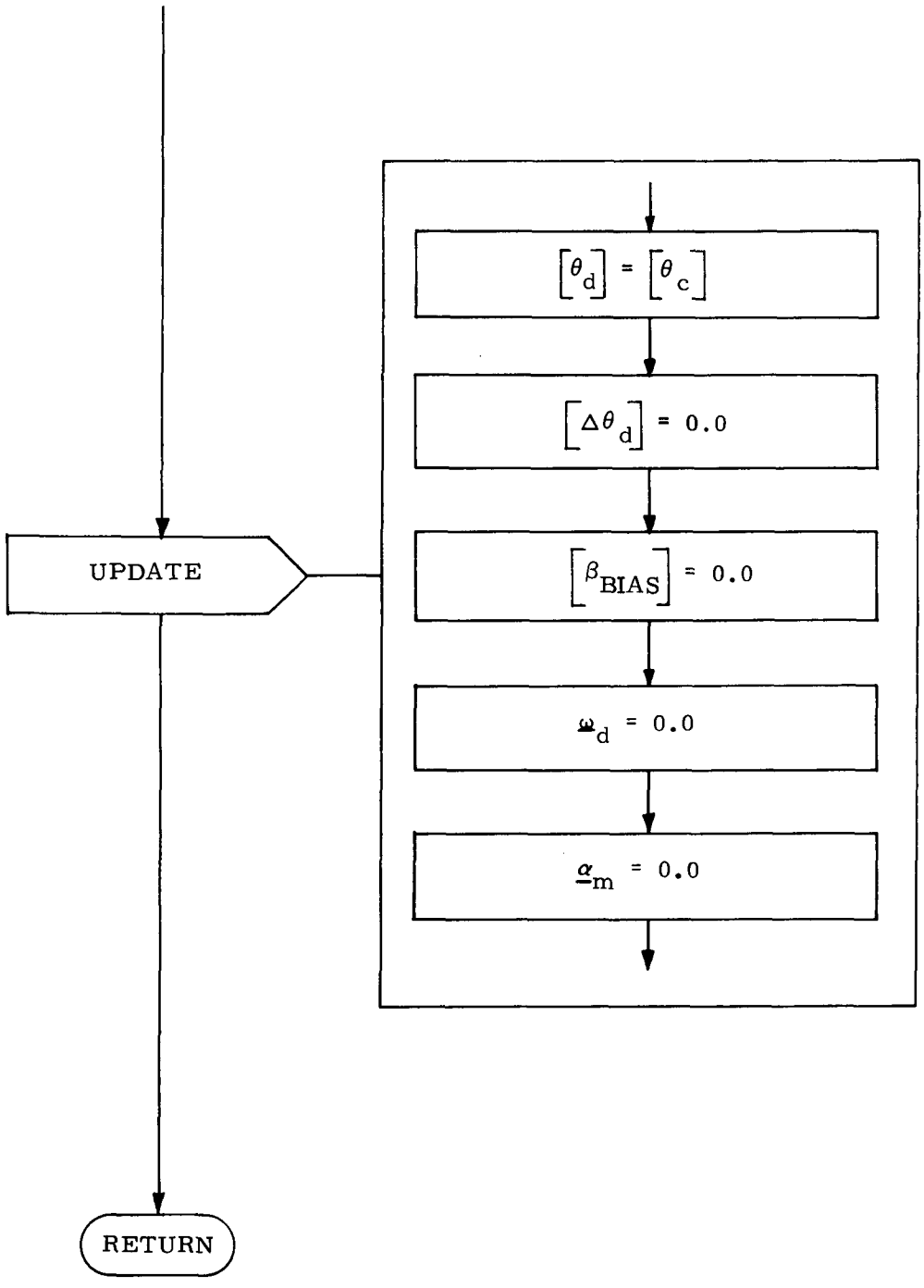



Fig. 2-12 HOLD_ATTITUDE : Procedure

# Table 2-1 ATTITUDE MANEUVER MODULE DICTIONARY

| MNEMONIC | DEFINITION | CLASS | DECLARED IN | ENTRY FORMAT | INPUTS REFERENCED | OUTPUTS ASSIGNED | LEVEL* |
|---|---|---|---|---|---|---|---|
| ANGLES_TO_DCM | Converts an array of angles to a direction-cosine matrix. | PROCEDURE | ATTITUDE_MANEUVER | ANGLES_TO_DCM: PROCEDURE ($[\theta]$) ASSIGN ($\overset{*}{C}$); | --- | --- | 2 |
| ATTITUDE_MANEUVER | Sequences an automatic reorientation during the ON-ORBIT phase. | PROGRAM | COMPOOL | ATTITUDE_MANEUVER: PROGRAM; | FLAG_3AXIS AUTO_MODE | --- | 1 |
| DCM_TO_ANGLES | Converts direction-cosine matrix to an array of angles | PROCEDURE | ATTITUDE_MANEUVER | DCM_TO_ANGLES: PROCEDURE ($\overset{*}{C}$) ASSIGN ($[\theta]$); | $\pi$ | --- | 2 |
| GIMBAL_ANGLE_READ | Reads the gimbal angles | PROCEDURE | ATTITUDE_MANEUVER | GIMBAL_ANGLE_READ: PROCEDURE ASSIGN (GA); | --- | --- | 2 |
| HOLD_ATTITUDE | Sends steering parameters to the autopilot to hold attitude at commanded gimbal angles. | PROCEDURE | ATTITUDE_MANEUVER | HOLD_ATTITUDE: PROCEDURE; | $[\theta_c]$ | $\theta_d$, $[\Delta\theta_d]$, $[\beta_{BIAS}]$, $\underline{\omega}_d$, $\alpha_m$ | 2 |
| KALCMANU | Computes angle of rotation, $A_M$, and axis of single equivalent rotation, $\underline{u}_r$ | PROCEDURE | ATTITUDE_MANEUVER | KALCMANU PROCEDURE; | $[\theta_c]$, DEG_TO_RAD | $\underline{u}_r$, $A_M$ $[\theta_B]$, $\overset{*}{C}_{BSM}$ | 2 |
| ROTATION_MATRIX | Computes rotation matrix $\overset{*}{R}$ from $\underline{u}_r$ and $A_M$. | PROCEDURE | ATTITUDE_MANEUVER | ROTATION_MATRIX: PROCEDURE ($\underline{u}_r$, A) ASSIGN ($\overset{*}{R}$); | $\overset{*}{I}_{identity}$ | --- | 2 |
| STEERING | Computes and sends to autopilot the steering parameters every $\Delta t_c$ seconds | PROCEDURE | KALCMANU | STEERING: PROCEDURE; | $A_M$, $\underline{u}_r$, $\omega_r$, $\overset{*}{M}_{PC}$, $\underline{\omega}_d$ $[\alpha_c]$, $\overset{*}{J}$, $[\theta_B]$, $t_{fast}$, $\Delta t_c$ $\overset{*}{C}_{BSM}$, REALSEC, AUTO_MODE | $[\theta_d]$, $[\Delta\theta_d]$, $[\beta_{BIAS}]$ $\underline{\omega}_d$, $\alpha_m$, $[\theta_B]$, $\overset{*}{C}_{BSM}$ | 3 |
| STOPRATE | Sends steering parameters to the autopilot to hold attitude at present gimbal angles | PROCEDURE | ATTITUDE_MANEUVER | STOPRATE: PROCEDURE; | Present gimbal angles obtained from GIMBAL_ANGLE_READ | $[\theta_d]$, $[\Delta\theta_d]$, $[\beta_{BIAS}]$, $\underline{\omega}_d$, $\alpha_m$ | 2 |
| VECPOINT | Computes $[\theta_c]$ from $\underline{u}_{ANB}$ and $\underline{u}_{DSM}$ | PROCEDURE | ATTITUDE_MANEUVER | VECPOINT: PROCEDURE; | $\underline{u}_{ANB}$, $\underline{u}_{DSM}$, $\underline{u}_{XL}$, $\pi$ | $[\theta_c]$ | 2 |

*COMPOOL being on level 0

## Table 2-2 ON-ORBIT ATTITUDE MANEUVER VARIABLE-CONSTANT DICTIONARY

| MNEMONIC | ENGINEERING TERM, RANGE | UNITS | DEFINITION | DECLARED IN | ASSIGNED IN | REFERENCED IN | TYPE | ATTRIBUTES |
|---|---|---|---|---|---|---|---|---|
| A | $0 \leq A \leq \pi$ | rad | Formal parameter. Angle of rotation about $\underline{u}_r$. | VECPOINT | VECPOINT | VECPOINT | SCALAR | DOUBLE PRECISION, STATIC |
| A | $0 \leq A \leq \pi$ | rad | Formal parameter. Angle of rotation about $\underline{u}_r$. | ROTATION_MATRIX | ROTATION_MATRIX | ROTATION_MATRIX | SCALAR | DOUBLE PRECISION, STATIC |
| A_M | $0 \leq A_M \leq \pi$ | rad | Angle of rotation about $\underline{u}_r$ | KALCMANU | KALCMANU | KALCMANU STEERING | SCALAR | DOUBLE PRECISION, STATIC |
| ALPHAM | $\underline{\alpha}_m$ | rad/sec$^2$ | Bias acceleration used by the autopilot | COMPOOL | STEERING STOPRATE HOLD_ATTITUDE | --- | VECTOR (3) | DOUBLE PRECISION, STATIC, LOCKED |
| AUTO_MODE | AUTO_MODE | --- | Automatic mode switch | COMPOOL | COMPOOL | ATTITUDE_ MANEUVER STEERING | BIT (1) | --- |
| [BIAS_BETA] | $[\beta_{BIAS}]$ | rad | Bias gimbal angles | COMPOOL | STEERING STOPRATE HOLD_ATTITUDE | --- | ARRAY (3) SCALAR | DOUBLE PRECISION, STATIC, LOCKED |
| $\overset{*}{C}$ | $\overset{*}{C}$ | --- | Formal parameter. Direction cosine matrix. | ANGLES_TO_DCM | ANGLES_TO_DCM | --- | MATRIX (3, 3) | DOUBLE PRECISION, STATIC |
| $\overset{*}{C}$ | $\overset{*}{C}$ | --- | Formal parameter. Direction cosine matrix. | DCM_TO_ANGLES | --- | DCM_TO_ANGLES | MATRIX (3, 3) | DOUBLE PRECISION, STATIC |
| $\overset{*}{C}$_BSM | $\overset{*}{C}_{BSM}$ | --- | Direction cosine matrix between present vehicle axes and stable member axes. | KALCMANU | KALCMANU STEERING | KALCMANU STEERING | MATRIX (3, 3) | DOUBLE PRECISION, STATIC |
| $\overset{*}{C}$_CSM | $\overset{*}{C}_{CSM}$ | --- | Direction cosine matrix between commanded vehicle axes and stable member axes. | KALCMANU | KALCMANU | KALCMANU | MATRIX (3, 3) | DOUBLE PRECISION, STATIC |
| $\overset{*}{C}$_LSM | $\overset{*}{C}_{LSM}$ | --- | Direct cosine matrix between L-coordinate frame and stable member axes. | VECPOINT | VECPOINT | VECPOINT | MATRIX (3, 3) | DOUBLE PRECISION, STATIC |

## Table 2-2 ON-ORBIT ATTITUDE MANEUVER VARIABLE-CONSTANT DICTIONARY (Cont)

| MNEMONIC | ENGINEERING TERM, RANGE | UNIT | DEFINITION | DECLARED IN | ASSIGNED IN | REFERENCED IN | TYPE | ATTRIBUTES |
|---|---|---|---|---|---|---|---|---|
| $\overset{*}{C}$_MSM | $\overset{*}{C}_{MSM}$ | --- | Direction cosine matrix between M-coordinate frame and stable member axes. | VECPOINT | VECPOINT | VECPOINT | MATRIX (3, 3) | DOUBLE PRECISION, STATIC |
| DEG_TO_RAD | $\pi/180$ | rad/deg | Conversion factor | COMPOOL | COMPOOL | KALCMANU | SCALAR | DOUBLE PRECISION, STATIC CONSTANT |
| [DELTA_THETA_D] | $[\Delta\theta_d]$ | rad | Incremental gimbal angles. | COMPOOL | STEERING STOPRATE HOLD_ATTITUDE | --- | ARRAY (3) SCALAR | DOUBLE PRECISION, STATIC, LOCKED |
| DELTAA_M | $\Delta A_M$ | rad | Incremental rotation angle. | STEERING | STEERING | STEERING | SCALAR | DOUBLE PRECISION, STATIC, LOCKED |
| DELT$\overset{*}{A}$R | $\Delta\overset{*}{R}$ | --- | Incremental rotation matrix. | STEERING | STEERING | STEERING | MATRIX (3, 3) | DOUBLE PRECISION, STATIC, LOCKED |
| DELTAT_C | $\Delta t_c$ | sec | Period of the cycle of STEERING | COMPOOL | COMPOOL | STEERING | SCALAR | DOUBLE PRECISION, STATIC, CONSTANT |
| FIRST_PASS | FIRST_PASS | --- | First pass switch. | STEERING | STEERING | STEERING | BIT (1) | --- |
| FLAG_3AXIS | FLAG_3AXIS | --- | Type of input flag. | COMPOOL | calling program | ATTITUDE_ MANEUVER | BIT (1) | --- |
| GA | $\theta$ | rad | Gimbal angles read from sensors | GIMBAL_ANGLE_ READ | GIMBAL_ANGLE_ READ | GIMBAL_ANGLE_ READ | ARRAY (3) SCALAR | DOUBLE PRECISION, STATIC |
| $\overset{*}{I}$D_MATRIX | $\overset{*}{I}_{identity}$ | --- | Identity matrix | COMPOOL | COMPOOL | ROTATION_MATRIX | MATRIX (3, 3) | DOUBLE PRECISION, STATIC, CONSTANT |
| INER$\overset{*}{T}$IA | $\overset{*}{J}$ | slug-ft$^2$ | Inertia matrix of the vehicle | COMPOOL | COMPOOL | STEERING | MATRIX (3, 3) | DOUBLE PRECISION, STATIC |
| [MAGACC] | $[\alpha_c]$ | rad/sec$^2$ | Magnitudes of the U, V, W control accelerations | COMPOOL | calling program | STEERING | ARRAY (3) SCALAR | DOUBLE PRECISION, STATIC |
| MAGNITUDE_W_D | $\omega_r$ | rad/sec | Requested magnitude of angular velocity | COMPOOL | calling program | STEERING | SCALAR | DOUBLE PRECISION, STATIC |
| $\overset{*}{M}$PC | $\overset{*}{M}_{PC}$ | --- | Transformation matrix from pilot to control axes | COMPOOL | calling program | STEERING | MATRIX (3, 3) | DOUBLE PRECISION, STATIC |

Table 2-2 ON-ORBIT ATTITUDE MANEUVER VARIABLE-CONSTANT DICTIONARY (Cont)

| MNEMONIC | ENGINEERING TERM, RANGE | UNITS | DEFINITION | DECLARED IN | ASSIGNED IN | REFERENCED IN | TYPE | ATTRIBUTES |
|---|---|---|---|---|---|---|---|---|
| PI | $\pi$ | rad | | COMPOOL | COMPOOL | VECPOINT DCM_TO_ANGLES | SCALAR | DOUBLE PRECISION, STATIC |
| $\overset{*}{R}$ | $\overset{*}{R}$ | --- | Formal parameter. Rotation matrix. | ROTATION_MATRIX | ROTATION_MATRIX | ROTATION_MATRIX | MATRIX (3, 3) | DOUBLE PRECISION, STATIC |
| $\overset{*}{R}$_A | $\overset{*}{R}_A$ | --- | Antisymmetric component of $\overset{*}{R}_{CB}$. | KALCMANU | KALCMANU | KALCMANU | MATRIX (3, 3) | DOUBLE PRECISION, STATIC |
| $\overset{*}{R}$_CB | $\overset{*}{R}_{CB}$ | --- | Rotation matrix between commanded and initial attitudes. | KALCMANU | KALCMANU | KALCMANU | MATRIX (3, 3) | DOUBLE PRECISION, STATIC |
| $\overset{*}{R}$_ML | $\overset{*}{R}_{ML}$ | --- | Rotation matrix between the M and L coordinate frames. | KALCMANU | KALCMANU | KALCMANU | MATRIX (3, 3) | DOUBLE PRECISION, STATIC |
| $\overset{*}{R}$_S | $\overset{*}{R}_S$ | --- | Symmetric component of $\overset{*}{R}_{CB}$. | KALCMANU | KALCMANU | KALCMANU | MATRIX (3, 3) | DOUBLE PRECISION, STATIC |
| REALSEC | REALSEC | computer ticks/sec | Scale factor to synchronize with computer clock. | COMPOOL | COMPOOL | STEERING | SCALAR | DOUBLE PRECISION, STATIC, CONSTANT |
| T_CYCLE | $t_{cycle}$ | sec | Time at which each new pass of STEERING begins. | STEERING | STEERING | STEERING | SCALAR | DOUBLE PRECISION, STATIC |
| T_M | $t_M$ | sec | Time of maneuver duration. | STEERING | STEERING | STEERING | SCALAR | DOUBLE PRECISION, STATIC |
| T_S | $t_s$ | sec | Time of maneuver completion. | STEERING | STEERING | STEERING | SCALAR | DOUBLE PRECISION, STATIC |
| TF | $t_{fast}$ | sec | Period of the autopilot cycle. | COMPOOL | autopilot | STEERING | SCALAR | DOUBLE PRECISION, STATIC |
| [THETA] | $[\theta]$ $-\pi \leq \theta_i \leq \pi$ | rad | Formal parameter. Gimbal angles. | ANGLES_TO_DCM | --- | ANGLES_TO_DCM | ARRAY (3) SCALAR | DOUBLE PRECISION, STATIC |

## Table 2-2 ON-ORBIT ATTITUDE MANEUVER VARIABLE-CONSTANT DICTIONARY (Cont)

| MNEMONIC | ENGINEERING TERM, RANGE | UNITS | DEFINITION | DECLARED IN | ASSIGNED IN | REFERENCED IN | TYPE | ATTRIBUTES |
|---|---|---|---|---|---|---|---|---|
| [THETA] | $[\theta]$ $-\pi \leq \theta_i \leq \pi$ | rad | Formal parameter. Gimbal angles. | DCM_TO_ANGLES | DCM_TO_ANGLES | DCM_TO_ANGLES | ARRAY(3) SCALAR | DOUBLE PRECISION, STATIC |
| [THETA_B] | $[\theta_B]$ $-\pi \leq \theta_{B_i} \leq \pi$ | rad | Gimbal angles at the start of KALCMANU | KALCMANU | KALCMANU STEERING | KALCMANU STEERING | ARRAY(3) SCALAR | DOUBLE PRECISION, STATIC |
| [THETA_C] | $[\theta_c]$ $-\pi \leq \theta_{c_i} \leq \pi$ | rad | Commanded terminal gimbal angles. | COMPOOL | calling program or VECPOINT | HOLD-ATTITUDE | ARRAY(3) SCALAR | DOUBLE PRECISION, STATIC |
| [THETA_D] | $[\theta_d]$ $-\pi \leq \theta_{d_i} \leq \pi$ | rad | Intermediate desired gimbal angles. | COMPOOL | STEERING STOPRATE HOLD_ATTITUDE | autopilot | ARRAY(3) SCALAR | DOUBLE PRECISION, STATIC, LOCKED |
| [THETA_L] | $[\theta_L]$ $-\pi \leq \theta_{L_i} \leq \pi$ | rad | Gimbal angles at the start of VECPOINT | VECPOINT | VECPOINT | VECPOINT | ARRAY(3) SCALAR | DOUBLE PRECISION, STATIC |
| [THETA_N] | $[\theta_N]$ $-\pi < \theta_{N_i} < \pi$ | rad | Gimbal angles predicted $\Delta t_c$ in advance | STEERING | STEERING | STEERING | ARRAY(3) SCALAR | DOUBLE PRECISION, STATIC |
| $\bar{U}$_AL | $\underline{u}_{AL}$ | ft | Prescribed unit vector $\underline{u}_A$ in the L-coordinate frame | VECPOINT | VECPOINT | VECPOINT | VECTOR(3) | DOUBLE PRECISION, STATIC |
| $\bar{U}$_ANB | $\underline{u}_{ANB}$ | ft | Prescribed unit vector $\underline{u}_A$ in navigation-base coordinates. | COMPOOL | calling program | VECPOINT | VECTOR(3) | DOUBLE PRECISION, STATIC |
| $\bar{U}$_DL | $\underline{u}_{DL}$ | ft | Desired unit vector in the L-coordinate frame | VECPOINT | VECPOINT | VECPOINT | VECTOR(3) | DOUBLE PRECISION, STATIC |
| $\bar{U}$_DSM | $\underline{u}_{DSM}$ | ft | Desired unit vector in the stable-member coordinate frame. | COMPOOL | calling program | VECPOINT | VECTOR(3) | DOUBLE PRECISION, STATIC |

Table 2-2 ON-ORBIT ATTITUDE MANEUVER VARIABLE-CONSTANT DICTIONARY (Cont)

| MNEMONIC | ENGINEERING TERM, RANGE | UNITS | DEFINITION | DECLARED IN | ASSIGNED IN | REFERENCED IN | TYPE | ATTRIBUTES |
|---|---|---|---|---|---|---|---|---|
| $\overline{U}$_R | $\underline{u}_r$ | ft | Single equivalent rotation vector. Rotation through the angle A. | VECPOINT | VECPOINT | VECPOINT | VECTOR(3) | DOUBLE PRECISION, STATIC |
| $\overline{U}$_R | $\underline{u}_r$ | ft | Formal parameter. Single equivalent rotation vector. | ROTATION_MATRIX | ROTATION_MATRIX | ROTATION_MATRIX | VECTOR(3) | DOUBLE PRECISION, STATIC |
| $\overline{U}$_R | $\underline{u}_r$ | ft | Single equivalent rotation vector. Rotation through the angle $A_M$. | KALCMANU | KALCMANU | KALCMANU STEERING | VECTOR(3) | DOUBLE PRECISION, STATIC |
| $\overline{U}$_RPRIME | $\underline{u}_r'$ | ft | Intermediate rotation vector used in determining proper signs of $\underline{u}_r$. | KALCMANU | KALCMANU | KALCMANU | VECTOR(3) | DOUBLE PRECISION, STATIC |
| $\overline{U}$_XL | $\underline{u}_{XL}$ | ft | Unit vector along the navigation-based x-axis (1, 0, 0). | COMPOOL | COMPOOL | VECPOINT | VECTOR(3) | DOUBLE PRECISION, STATIC |
| $\overline{U}$_YSML | $\underline{u}_{YSML}$ | ft | Unit vector along the y-axis of the stable-member frame projected into the L-frame. | VECPOINT | VECPOINT | VECPOINT | VECTOR(3) | DOUBLE PRECISION, STATIC, CONSTANT |
| $\overset{*}{U}$X_R | $\overset{*}{U}X_r$ | ft | Antisymmetric matrix whose components are taken from $\underline{u}_r$. | ROTATION_MATRIX | ROTATION_MATRIX | ROTATION_MATRIX | MATRIX(3, 3) | DOUBLE PRECISION, STATIC |
| $\overline{W}$_D | $\underline{\omega}_d$ | rad/sec | Desired angular velocity vector in navigation-based coordinate. | COMPOOL | STEERING STOPRATE HOLD_ATTITUDE | autopilot | VECTOR(3) | DOUBLE PRECISION, STATIC, LOCKED |
| $\overline{W}$_DC | $\underline{\omega}_{dc}$ | rad/sec | Desired angular velocity vector in control coordinates | STEERING | STEERING | STEERING | VECTOR(3) | DOUBLE PRECISION, STATIC, LOCKED |

## REFERENCES


1. Dixon, M., "Fuel-Time Optimal Spacecraft Reorientation," Ph.D. Thesis T-502, Instrumentation Laboratory, M.I.T., May 1968.

2. Crisp, R. and Keene, D., "Attitude Maneuver Optimization to Conserve Reaction Control Propellants," Report E-1832, Instrumentation Laboratory, M.I.T., August 1965.

3. Crisp, R. and Keene, D., "Apollo Command and Service Module Reaction Control by the Digital Autopilot," Report E-1964, Instrumentation Laboratory, M.I.T., May 1966.

4. Keene, D., "Block II LGC Attitude Maneuver Routine: KALCMANU For Flight 206," Group 23 Internal Memo #8, Instrumentation Laboratory, M.I.T., November 1966.

5. Keene, D., "Coasting-Flight Attitude Maneuver Routine", Guidance Systems Operations Plan for Manned LM Earth Orbital and Lunar Missions Using Program Luminary 1E, Section 3 Digital Autopilot (Rev. 8), March 1972.

6. Gorman, S. and Kimball, G., Apollo Software Information Memo 5-68-9, "VECPOINT Automatic Maneuvers for Program SUNDISK," U.S. Government Memorandum, May 1968.