FINAL DRAFT

R-700

MIT's ROLE IN PROJECT APOLLO

FINAL REPORT ON CONTRACTS
NAS 9-153 and NAS 9-4065

VOLUME V

THE SOFTWARE EFFORT

by

Madeline S. Johnson

with

Donald R. Giller

March 1971

CHARLES STARK DRAPER LABORATORY
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
CAMBRIDGE, MASSACHUSETTS 02139

FINAL DRAFT

ACKNOWLEDGMENTS

This report was prepared under DSR Project 55-23890, sponsored by the Manned Spacecraft Center of the National Aeronautics and Space Administration through Contract NAS 9-4065.

The authors express appreciation to their many colleagues, both within and without the Draper Laboratory, who gave of their time and contributed essential information to this Final Report. These persons are cited at the end of this Volume in a section entitled, "A Note on Sources". We offer special thanks to Joseph A. Klawsnik, who, though with us only briefly, left considerable impact on the format and contents of Sections II and III herein.

REPORT R-700

MIT's ROLE IN PROJECT APOLLO

Final Report on Contracts
NAS 9-153 and NAS 9-4065

VOLUME V
THE SOFTWARE EFFORT

ABSTRACT

Seventy-six days after the President of the United States committed the nation to a massive lunar-landing program, the Charles Stark Draper (formerly Instrumentation) Laboratory of the Massachusetts Institute of Technology received the first major contract of the Apollo program. This volume of the Final Report discusses the efforts of Laboratory personnel in developing the specialized software for the Guidance, Navigation and Control System. Section I presents the historical background of the software effort. Section II discusses the software architecture developed for the Apollo Guidance Computer. Section III treats the methods of testing and verification of the flight programs, and the Laboratory's mission-support activities. Four appendices present functional descriptions of some major program capabilities—coasting-flight navigation, targeting, powered-flight navigation and guidance, and the digital autopilots.

PREFACE

Rarely has mankind been so united as in its awe at one man's step onto the lunar surface. When Neil Armstrong placed his left foot in the dust of the moon, engineers and scientists at the Massachusetts of Technology Instrumentation Laboratory felt a special pride for their significant contribution to this accomplishment in the design of the Primary Guidance, Navigation and Control System for the Apollo spacecrafts.

This report discusses the efforts of Instrumentation Laboratory personnel in developing the special software for the Guidance, Navigation and Control System. Although it is part of a multi-volume series documenting the total Project Apollo efforts of the Instrumentation Laboratory, this section may be read independently of the other volumes; the authors intend it to be meaningful to the general reader who may or may not have read the preceding volumes.

In January 1970, this facility became the Charles Stark Draper Laboratory, named in honor of its founder and current President. Throughout this report, "MIT" and "Draper Laboratory" are used interchangeably, in reference to the former Instrumentation Laboratory.

CONTENTS

ILLUSTRATIONS

SECTION I

HISTORY OF THE SOFTWARE EFFORT

1.1 Introduction

Seventy-six days after John Fitzgerald Kennedy committed the United States to participation in a massive lunar-landing program, the Instrumentation Laboratory* of the Massachusetts Institute of Technology received the first major contract of the Apollo program. Steps leading to this award, however, did not begin 25 May 1961—the day of the President's special message to Congress; the footprints of this history trace back at least several years earlier.

In the Fall of 1957, a group of scientists and engineers at MIT began the investigation of a recoverable interplanetary space vehicle. Under contract to the U.S. Air Force, the MIT group collaborated with AVCO Corporation, the Reaction Motors Division of Thiokol Chemical Corporation, and MIT's Lincoln Laboratory. As reported in the MIT/IL document R-235, "A Recoverable Interplanetary Space Probe", this investigation established the feasibility of designing a vehicle which would journey to a neighboring planet, take a high-resolution photograph there, and return for recovery on earth. The investigators studied the navigational techniques and interplanetary orbits which would be required for a variety of such missions. This study served to bring the engineering problems of interplanetary navigation, attitude control, communications, reentry, and space exploration into sharp focus. R-235 argued that the "early execution of a recoverable interplanetary space probe is an effective means for advancing the state-of-the-art in self-contained interplanetary navigation and control needed for later scientific and military achievements". Furthermore, the report stressed that the "successful physical recovery of a small vehicle which has navigated itself around the solar system and which brings back

---

*As explained in the Preface, the Laboratory was renamed the Charles Stark Draper Laboratory in January 1970.

photographic evidence of its close and well-controlled passage by another planet is certain to enhance the prestige of this nation".

Following the publication of the study in July 1959, the newly established National Aeronautics and Space Administration undertook its first contract with the Draper Laboratory. In September 1959, MIT agreed to investigate guidance and navigation concepts for a variety of interplanetary missions. Placing emphasis on unmanned missions, the Draper Laboratory devised a system for automatic guidance, including the design of an automatic sextant. Upon completion of this contract in March 1960, several months of discussion ensued between representatives of MIT and NASA's Space Task Group, headquartered at Langley Field, Virginia. A second study contract resulted, this one for another six-month effort: MIT was to present a preliminary guidance and navigation design for a manned lunar-landing mission. This study ran concurrently with several industry investigations of the overall Apollo spacecraft mission.

Although work on the preliminary guidance and navigation design for a manned mission began in late 1960, the actual contract was not announced until 7 February 1961. Midway through the contractual period, President Kennedy declared that a manned lunar landing and return would be a national goal for the 1960s. The President's decision opened the way for formal contractual designations by NASA for design, development and manufacture of the various Apollo spacecraft systems.

Thus, by the time of the Presidential message to Congress, the Draper Laboratory had demonstrated scientific and engineering competence in three space studies: the early recoverable space-vehicle investigation; the six-month unmanned guidance and navigation study; and the preliminary manned guidance and navigation examination. Another factor which proved influential in NASA's assessment of MIT's capabilities was the Laboratory's responsibility for the design and development of guidance and navigation systems for the Polaris guided missile. MIT's experience with the U.S. Navy's Polaris project included engineering and managerial techniques which, it appeared, might be implemented during Project Apollo. Indeed, during the month of July 1961, representatives of NASA and the Laboratory studied the development and scheduling of the Polaris guidance and navigation system, from original conception through production. The group plotted a rough schedule for a similar program on Apollo. NASA representatives also expressed interest in MIT's subcontractor philosophy on Polaris: through significant support by subcontractors, the Draper Laboratory had been able to build up a working force and achieve substantial results in a relatively short period of time. Thus, though Project Apollo would undoubtedly prove to be a much larger and more complex task than Polaris, MIT had demonstrated achievement on a qualitatively similar project.

As a result of the preliminary manned guidance and navigation study, NASA's Space Task Group recommended that the guidance and navigation portion of the Apollo spacecraft mission be negotiated as a contract separate from the development of the Apollo spacecraft. Shortly after this decision was made, and following a noncompetitive, sole-source procurement procedure, the Space Task Group designated MIT to implement the guidance and navigation system of the Apollo spacecraft. Announced 9 August 1961, the first major Apollo contract awarded by NASA called on MIT to conduct a Navigation and Guidance System Development Program which would "meet the intermediate as well as the ultimate objectives of Project Apollo", and which would "provide a general on-board guidance capability for the various earth-orbital and cislunar missions".

Although, by the end of 1961, a great deal of theorizing and experimenting had already been accomplished, and the major Apollo spacecraft contractors had been chosen, a significant unknown remained to be answered: how would men actually land on the moon—and equally important, how would they return to earth? The time had come to forecast the amount of rocket power that could be achieved by the end of the decade, to estimate how much weight the lunar surface could actually support, and to devise a means for leaving the moon after a safe landing.

By early 1962, three types of mission plans were being discussed by NASA planners. These methods were called direct ascent, earth-orbit rendezvous (EOR) and lunar-orbit rendezvous (LOR)

The direct-ascent scheme would place a 150,000-lb manned spaceship directly into lunar trajectory, using the boosting power of a still-to-be developed rocket with an initial thrust of about 12 million lb. From lunar trajectory, the spacecraft would enter lunar orbit; braking rockets would fire and the vehicle would back down toward the lunar surface. The same vehicle would later blast off the surface and land back on earth from an earth orbit. But two problems faced this type of mission.

First, there was considerable doubt that the necessary rocket power could be harnessed by 1970. The so-called "Nova" would have required about twice as much power as any rocket then being discussed. Second, planners were concerned that so large a spacecraft might break through the lunar crust—or, indeed, that its high center of gravity (the spacecraft itself would have measured about 90 ft) would cause it to topple upon landing.

A second method of lunar landing and earth return avoided the requirement of so massive an initial rocket thrust. Earth-orbit rendezvous would have placed two payloads in orbit around the earth. First, a "tanker" rocket would be launched, containing fuel that would eventually be fed into the second payload. After the tanker had achieved its requisite orbit, the second payload would be launched; this would be the manned Apollo spacecraft, propelled by a "Saturn V" rocket whose third stage lacked the liquid-oxygen fuel necessary for the lunar trip. After the payloads had rendezvoused, the spacecraft would dock with the tanker, and the fuel delivery would be accomplished. The advantage of this method was that it involved rocket power then considered likely by the end of the decade. But the same problems of landing on the lunar surface as faced the direct-ascent method still remained.

The third method of lunar landing at first appeared the least likely, probably because it intuitively seemed the most risky. A Saturn V rocket would propel an Apollo vehicle containing three astronauts, plus something new—a detachable craft designed specifically for landing on the moon (e.g., it would possess a low center of gravity and special landing "legs"). After stabilizing in earth orbit, the combined spacecraft and landing vehicle would enter a lunar trajectory and finally stabilize into a lunar orbit. At that point, two astronauts would move into the lunar landing craft, detach it from the mother ship, and descend toward the moon's surface. To rejoin the orbiting Apollo vehicle, the two astronauts would fire rockets for the lunar craft to reinsert into lunar orbit. After the two vehicles had rendezvoused and docked, the astronauts would reenter the main Apollo spacecraft, the landing vehicle would be scuttled, and the Apollo ship would fire its rockets for a return to earth.

The differences between earth-orbit and lunar-orbit rendezvous were immense. EOR plotted a rendezvous in earth orbit before embarking onto a lunar trajectory; LOR involved rendezvous in lunar orbit after the actual landing. The idea of doing a rendezvous (which itself at the time seemed a hazardous maneuver) so far away from earth as planned in the LOR method was initially a frightening proposition. Eventually, however, a team of Langley scientists and engineers demonstrated that, despite outward appearances, LOR would result in substantial savings in earth boost requirements. In addition, it would offer substantial simplification in all phases of a mission—development, testing, manufacture, erection, countdown, launch and flight operations.

With the selection of the lunar-orbit rendezvous method in July 1962, NASA filled in the most significant void then facing the major Apollo contractors. The myriad of scientists and engineers planning for man's eventual landing on the moon could now follow a specific plan. More specifically, the software effort ongoing at MIT at last was able to proceed toward a specific goal. For the most part, conception and development of the Guidance, Navigation and Control hardware did not depend upon the specific mission plan chosen; software, on the other hand, most assuredly had been hampered by the lack of a definitive goal. Landing on the moon and returning via lunar-orbit rendezvous—this was the Apollo mission; the software effort could now begin in earnest.

1.2 Software Programs for the Apollo Missions

The Draper Laboratory's software efforts culminated in a series of flight programs for the Apollo Primary Guidance, Navigation and Control System. Each flight required its own set of software, defined by the mission objectives and constraints. In general, however, the flight programs were comprised of mission programs and routines which remained rather fixed in approach and technique. Thus, such mission programs as rendezvous, targeting and landing are now part of every lunar-landing flight; their underlying techniques are relatively constant, but, in general, control data change with each mission.

Before work could begin on the first flight program—indeed, even before the Apollo mission had been finalized—basic software techniques had to be developed. Many of these early software efforts are briefly discussed in Section 2.2.1. A completed flight program represents the assembly of mission programs and routines. In common parlance, the completed assembly of hard-wire fixed and erasable memory is known as a "rope", a name taken from the weaving process by which the fixed memory is manufactured; the result of this weaving process actually resembles a rope.

An intriguing aspect of the rope developmental history is the means by which the ropes acquired their given names. At first, virtually all of the rope names derived from their association with the name given the entire lunar-landing mission —Apollo: Greek god of the sun*. Those early ropes without "SUN" in their name generally related to astronomical phenomena: thus, ECLIPSE (developed at the time of a major solar eclipse, in 1963), CORONA and AURORA. (RETREAD was an extensively revised version of SUNRISE.) Assigning the early rope names was the treasured prerogative of those most intimately concerned with each rope's development. After the succession of the "SUN" names given the next ropes—SUNDIAL, SUNSPOT, SUNBURST, SUNDISK and SUNDANCE (and SOLARIUM, with its direct sun association, as well)—it became somewhat difficult to differentiate which of the ropes were for the Command Module and which were for the Lunar Module. Accordingly, NASA requested, and MIT agreed, that all Command Module ropes begin with a "C", and all ropes for the Lunar Module with an "L". After a lively intramural competition, the names finally chosen for the LM and CM series were LUMINARY and COLOSSUS, respectively (but not until such names as "Lewis" and "Clark" and "Lemon" and "Coughdrop" had been, for more or less obvious reasons, disqualified).

The following sections summarize the development of flight programs for the Apollo Guidance Computer (AGC). As the result of a NASA decision emanating from a Guidance and Navigation System Implementation Meeting (see Section 1.3.1.2), MIT began digital-autopilot design in late 1964. Two decisions—to integrate an autopilot function into the Guidance, Navigation and Control System, and to enlarge and redesign the AGC—occurred at about the same time, requiring software to fit that computer. Thus, two basic designs of the AGC evolved. Ropes for the earlier, Block I computer, are discussed in Section 1.2.1. The next section discusses the programs developed for the Block II AGC. Section 1.2.3 presents a summary of the Apollo flights, including the names of the flight programs, the launch dates and

*No satisfactory explanation has yet been offered for naming a project aimed at landing on the moon after the sun god. Apollo's sister, Diana (also called Artemis), goddess of the moon, might well feel offended.

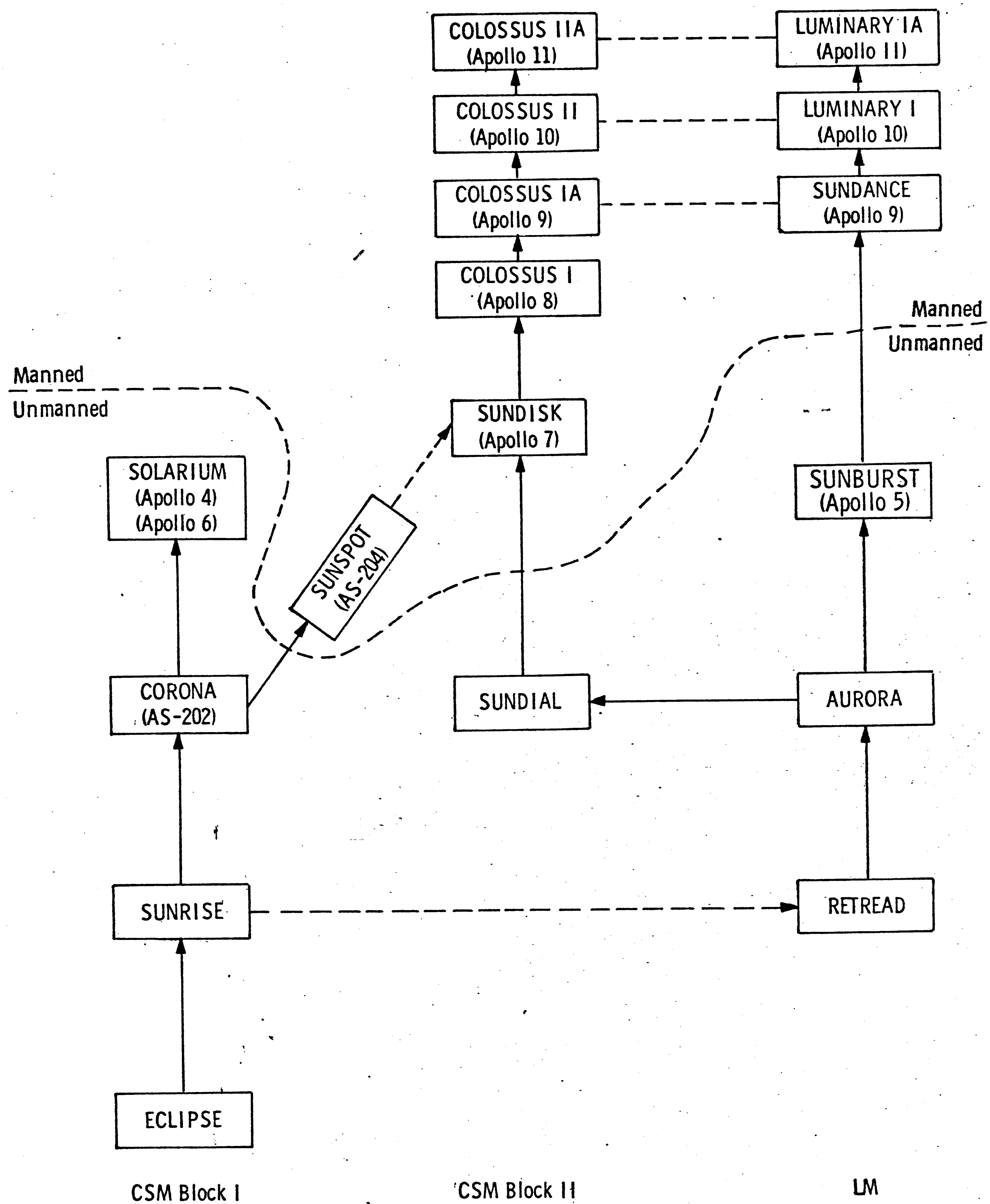


Figure 1.2-1 Rope Tree

crews, and flight descriptions. Figure 1.2-1 depicts the interrelationship of the Block I and Block II ropes discussed in Section 1.2.1 and 1.2.2.

1.2.1 Block I Rope Summary

ECLIPSE is generally ascribed as the first test program designed for use in an early Block I Apollo Guidance Computer. ECLIPSE was, in fact, an assembly of many fundamental routines. It brought together such routines as the Executive, Interpreter and Waitlist. (See Section II for a description of the AGC computer architecture.) In addition, ECLIPSE included Program PINBALL GAME BUTTONS AND LIGHTS, which processes the buttons and illuminates the lights of the spacecraft's Display and Keyboard. Because ECLIPSE was intended only as a test of the Block I AGC, it contained no routines to exercise the Guidance, Navigation and Control System (GN&CS) hardware.

By adding fundamental guidance and navigation functions to ECLIPSE, MIT engineers designed and developed SUNRISE, the first G&N systems-test program for the Block I computer. SUNRISE was the first Block I program suitable for operation in a laboratory-based guidance system. Included in SUNRISE were such G&N-specific routines as an IMU mode-switching program, interface-monitoring programs, down telemetry, and routines to measure gyro-drift coefficients and the bias and scale factors of the three accelerometers. SUNRISE also contained a program for prelaunch alignment. Although not destined for an actual mission, SUNRISE served as a building block for the first flight programs that followed. Programs under development could be interfaced with SUNRISE, and thus tested and changed in a working computer environment.

The program designed for the first Apollo flight was known as CORONA; it was used on the unmanned mission, AS-202. CORONA interacted with an onboard Mission Control Programmer, a series of relays connected to the computer interface to simulate certain astronaut functions. Also, CORONA included an earth-orbital reentry program which served as the model for all future such programs.

Two developmental extensions of CORONA occurred at about the same time. The more straightforward evolution led to SOLARIUM, the flight program for the unmanned missions, Apollo 4 and Apollo 6. SOLARIUM contained few major changes from CORONA, except that rescaling occurred to replace the elliptical trajectory of CORONA with parabolic and hyperbolic reentry trajectories for SOLARIUM.

The second evolution from CORONA led to SUNSPOT, the program intended for what would have been the first manned mission, AS-204. The major change represented by SUNSPOT allowed for elaborate astronaut-interface display programs. Whereas programs were sequenced automatically in the previous unmanned missions, allowing only a certain preordained series of events, SUNSPOT introduced the flexibility of astronaut selection of programs. Most of the automatic sequences provided in CORONA were removed in SUNSPOT.

1.2.2 Block II Rope Summary

To a great extent, the development of programs for the Block II Apollo Guidance Computer resembled the path taken in developing the Block I programs (Fig. 1.2-1). The most obvious differences resulted from the added presence of the Lunar Module (LM), which was to contain an Apollo Guidance Computer identical to that in the Command Module (CM). Following the testing of a Block II program which contained basic guidance and navigation functions, programs for the CM and LM computers evolved simultaneously.

For the initial development of a Block II program, the basic Block I systems-test programs were adapted and assembled into the rope appropriately known as RETREAD. Because the Block II computer contained a larger and more powerful instruction repertoire than that of the Block I AGC, recoding of the basic Block I programs resulted in increased speed and efficiency. Analogous to Block I's ECLIPSE, Block II's RETREAD contained the system-software programs required to test the potential of the computer—Executive, Waitlist, Interpreter. As in ECLIPSE, no provision for mission- or spacecraft-specific programs was included in RETREAD.

From RETREAD evolved the main on-line ropes, beginning with AURORA. In many ways equivalent in purpose to the Block I SUNRISE, AURORA included programs which interfaced with LM GN&CS hardware. AURORA included the monitoring routines for the Inertial Measurement Unit, prelaunch alignment programs, radar-manipulation routines, and various means to control the Display and

Keyboard logic, altitude and altitude-rate meters, and the turn-on and turn-off processes. Like the Block I SUNRISE, AURORA provided a software environment for testing and development of future ropes.

As an offshoot from AURORA, a rope called SUNDIAL tested the GN&C System for the Command Module. SUNDIAL naturally resembled AURORA, except that the LM-specific functions of AURORA were replaced with the CM-specific functions of SUNDIAL. SUNDIAL and AURORA both grew out of RETREAD and they "fathered" two lines of flight programs specific, respectively, to the Command and Lunar Module computers.

The first rope for a manned mission using the Block II AGC was SUNDISK, developed for Apollo 7. Although this program was developed for an earth-orbital flight, it contained many translunar programs in their formative stages. COLOSSUS I, the rope for Apollo 8, the first mission to orbit the moon, included operational cislunar and return-to-earth targeting and navigation programs. Apollo 8 orbited the moon without a Lunar Module, however. CSM/LM rendezvous programs were exercised in earth orbit in COLOSSUS IA, the rope developed for the Apollo 9 mission. COLOSSUS II, developed for the Apollo 10 mission, allowed for the first CSM/LM rendezvous in lunar orbit and included a revised model of the lunar-gravity potential. COLOSSUS IIA, flown on Apollo 11, was virtually the same as COLOSSUS II.

Programs for the LM Apollo Guidance Computer evolved from the early AURORA assembly. SUNBURST was developed for Apollo 5, an unmanned flight test of the Lunar Module and its flight rope. The SUNDANCE rope was developed for the first manned Lunar-Module flight, Apollo 9. Although the Apollo 9 mission was strictly earth-orbital, SUNDANCE exercised lunar-landing, lunar-ascent and rendezvous routines for the first time.

Employing the rope LUMINARY I, Apollo 10 marked the first low pass (to 50,000 ft) over the lunar surface by a solo LM. LUMINARY I represented a refinement of SUNDANCE, and included scaling for the lunar descent. On 20 July 1969, LUMINARY IA finally guided the Lunar Module to its safe touchdown on the moon's surface, thus fulfilling the nation's commitment to a lunar landing in the 1960s.

1.2.3 Overview of the Apollo Flights

Figure 1.2-2 is a summary of the missions flown during Project Apollo, through the flight of Apollo 11. Included are the flight name, the flight program(s) employed, a description of the objectives, the launch date and the crew for each flight. For those flights where two ropes are listed, the first is for the Command Module and the second for the Lunar Module*.

1.3 Control of the Software Effort

This section describes the various means by which MIT's software activities were monitored—internally, through several operating committees; and externally, through formal contact with the customer, NASA. Linking these types of control was the Guidance System Operations Plan—a multi-volumed document that served several functions, including specification control of each succeeding mission flight plan. This document was prepared by the Draper Laboratory for NASA approval, and reflected the technical decisions emanating from internal and external monitoring operations. Section 1.3.1 below discusses external control; Section 1.3.2 describes the Guidance System Operations Plan; and Section 1.3.3 comments upon other types of control, including internal control.

1.3.1 Control by NASA

Much of NASA's control of MIT's software activities occurred in the form of regular series of meetings conducted among representatives of NASA, MIT, North American Rockwell, Grumman Aircraft Engineering Corporation, and other relevant contractors and subcontractors. These meetings served as a vehicle for communications among the prime contractors and the customer, and apparent conflicts were often settled through unhampered discourse. When contractors were unable to agree on technical issues or future directions, NASA would often use the forum of these meetings to issue its decisions on such matters.

---

*For an insight into all of the phases which comprise a lunar-landing mission, the reader may choose at this time to continue with Section 2.2.2.

Flight	Flight Program Name	Description	Launch Date	Crew
AS-202	CORONA	Suborbital; supercircular entry with high heat load	8-25-66	unmanned
Apollo 4	SOLARIUM	High apogee; suborbital; supercircular entry at lunar return velocity	11-9-67	unmanned
Apollo 5	SUNBURST	First Lunar Module flight; earth orbital	1-22-68	unmanned
Apollo 6	SOLARIUM	High apogee; suborbital; supercircular entry at lunar return velocity; verification of closed-loop emergency detection system	3-4-68	unmanned
Apollo 7	SUNDISK	First manned flight; earth orbital	10-11-68	Schirra Eisele Cunningham
Apollo 8	COLOSSUS I	First manned lunar-orbital flight; first manned Saturn V launch	12-21-68	Borman Lovell Anders
Apollo 9	COLOSSUS IA SUNDANCE	First manned Lunar Module flight; exercise of lunar landing, ascent and rendezvous techniques in earth orbit; EVA (Extra Vehicular Activity)	3-3-69	McDivitt Scott Schweikart
Apollo 10	COLOSSUS II LUMINARY I	First lunar-orbit rendezvous; Lunar descent to 50,000 ft	5-18-69	Stafford Young Cernan
Apollo 11	COLOSSUS IIA LUMINARY IA	First lunar landing (7-20-69)	7-16-69	Armstrong Aldrin Collins

Figure 1.2-2 The Apollo Flights

1.3.1.1 G&N System Panel Meetings

The earliest series of discussions was known as G&N System Panel Meetings. This series occurred from August 1962 through February 1964, under the direction of the Apollo Systems Project Office of NASA/MSC. Participants represented NASA, MIT, North American Rockwell, Grumman and Bellcomm. Throughout this period, three subseries of Panel Meetings met regularly, each focusing on a separate issue: lunar-orbit operations of the Lunar and Command Modules; earth-orbit and cislunar activities of both vehicles; and the reentry activities of the CSM. Through the medium of vigorous discussion and debate, these meetings collated the technical decisions being made in the design and development of the Guidance, Navigation and Control System.

1.3.1.2 G&N System Implementation Meetings

The next set of meetings served to define the required interfaces between the GN&C System and the spacecraft. The Guidance and Navigation System Implementation Meetings were a means of negotiating the Interface Control Documents (ICDs) which were binding upon all contractors. Implementation Meetings focusing on interfaces for the CSM occurred from June 1964 through February 1965. Implementation Meetings responsible for LM interfaces occurred from September 1964 through April 1966. In addition to physical interfaces, among the topics discussed were kinds of data being sent across the interfaces; the formating of data transmission; data rates; and accuracies of data.

The Implementation Meetings monitored the integration of guidance, navigation and control. Out of these discussions came a decision which had a major impact on MIT's Apollo responsibilities. Originally, the Apollo autopilot function had been the responsibility of the Honeywell Corporation, under subcontract to North American Rockwell. The Honeywell autopilot was analog and was deemed by the NASA monitors to lack the flexibility and versatility required for the complex Apollo mission plan. Consequently, NASA directed the Draper Laboratory to develop a digital autopilot which would have none of these limitations. The existing Block I computer hardware did not have sufficient storage capacity to accommodate an addition of such import; however, at about this same period, another significant decision was made to enlarge the computer capacity and at the same time make its computer architecture more powerful than had heretofore been possible. Therefore, through the forum of the G&N System Implementation Meetings, the Block II Apollo Guidance Computer and the digital autopilots were conceived.

1.3.1.3 Data Priority Meetings

As a means of relieving the problem of customer-contractor and inter-contractor communications, the concept of Data Priority Meetings emerged in 1967. The Planning and Analysis Division of NASA/MSC regularly gathers together the flight crews, flight directors, flight controllers, various MSC software, hardware and analytical specialists, and appropriate contractor representatives. There are thus brought into a single room three significant components: those with questions; those with answers; and those with authority to render decisions.

The group meticulously reviews the guidance and control details for each succeeding mission. Data Priority Meetings define how the various data can be used and the priority which can be imposed to effect the nominal and backup execution of each mission phase.

MIT's role is restricted to the Guidance, Navigation and Control System, but this is one of the most complex subsystems in the Apollo spacecraft. MIT's representatives to the Data Priority Meetings oversee the Laboratory's follow-up to each meeting. Questions arising from these meetings elicit formal responses, usually in the form of Mission Techniques Memoes.

1.3.1.4 "Tiger" Teams

A fourth type of NASA control of MIT's software activities occurred thorugh a means less formal than that of an organized meeting. In late 1967, the Flight Operations Directorate of NASA/MSC organized so-called "Tiger" Teams to hasten technical decisions on MIT's rendezvous and display techniques. The Tiger Teams were aptly named, for despite their relatively informal approach, they were extremely effective. The first Tiger Team spent several days in Cambridge in a successful attempt to clarify the rendezvous displays and operations. Display interfaces between the crew and the landing and rendezvous maneuvers were determined, and rendezvous-display compatibility (e.g., scaling, polarity) between the LM and the CM were established. Targeting programs were made consistent from one program to the next. The second Tiger Team addressed itself to the same issues, but since the decisions of greatest import had already been made, its impact was less pervasive —hence, this Tiger Team was dubbed the "Pussycat" Team.

1.3.1.5 "Black Friday" Meetings

Shortly after MIT evidenced its dismay over the rapidly-saturating fixed-memory storage capacity of the AGC, joint MIT/NASA meetings were held to purge the mission programs then under development of any routines deemed "nonessential". Three such meetings took place—on 13 May 1966, 13 January 1967 and 28 August 1967. These meetings became emotional because of disagreement about what was, in fact, nonessential. Nonetheless, difficult compromises resulted in the current fixed-storage capacity being reduced sufficiently to allow inclusion of every essential routine.

1.3.2 GSOP Concept and History

Beginning with CORONA, the computer program for the AS-202 mission, a document known as the Guidance System Operations Plan (GSOP) served as the specification toward which the software efforts were directed. Development and control of the GSOP were important activities in planning the release of a flight program. The format for the GSOP evolved through a series of discussions among key personnel at NASA and the Draper Laboratory. During preparation of the CORONA rope, several alternative mission profiles had been considered: orbital, short-ranged suborbital, and long-ranged suborbital. MIT provided NASA with estimates of navigational difficulty that might be encountered on each type of mission, whereupon NASA chose the long-ranged suborbital trajectory. The CORONA GSOP represented an integration of inputs from MIT, NASA and North American Rockwell (the manufacturer of the CSM spacecraft), further defining the mechanics of achieving such a trajectory. NASA reviewed the document, modified it where necessary, and finally approved it as the specification for MIT's software effort.

In comparison to the GSOP format which would follow, the AS-202 document was relatively informal, encompassing in one small volume the same type of information which would later require six separate volumes for each rope. The

CORONA GSOP discussed the general description of the mission, the logic diagrams defining the operation of the Apollo Guidance Computer, the uplink and downlink that would interface with the guidance system, and the guidance equations and routines which MIT considered of potential interest to NASA.

Further evolution of the GSOP structure resulted from the additional requirements, constraints and capabilities of later missions. For instance, the SUNSPOT rope developed for the AS-204 mission was the first to allow for manned Apollo flight. With astronauts involved for the first time, more time was required for the GSOP discussions, and more personnel participated in the GSOP development. SUNBURST, the rope for the Apollo 5 flight, contained the first routines developed specifically for the Lunar Module, and thus the GSOP for SUNBURST was the first of the LM GSOPs. Beginning with SUNDISK (Apollo 7, CSM) and SUNDANCE (Apollo 9, LM), successive GSOPs generally represented merely changes from the preceding version, and did not require a completely fresh start. Most of the effort in Guidance System Operations Plans currently involves accounting for changes, with relatively little rewriting.

As mentioned above, the GSOP is published separately for the Lunar Module and the Command Module, and is updated with each new program release, thus providing NASA with current and accurate control over the software and system operations. In addition to these functions, the GSOP has served as an internal working document to coordinate the efforts of the various MIT groups, and as a testing guide for simulation personnel. Finally, the GSOP serves as a GN&C software description and a crew training aid for MSC personnel and contractors. A more detailed description of the GSOP is contained in Section III.

1.3.3 Additional Software Control

The Draper Laboratory monitored the incorporation of mission-program requirements into the mission programs through the actions of a Mission Design Review Board (MDRB), a formally-constituted group comprised of the directors of all software groups. Under the direction of each rope's Project Manager, the MDRB approved, internally, all mission-related documentation. The Project Manager was charged with the responsibility for MDRB coordination and participation to ensure proper processing of control documentation. The specific function of the MDRB was to provide a mechanism for internal control and coordination of mission-related activities. Program Change Requests (PCRs) and Program Change Notices (PCNs) were used as interim revisions of the GSOP, and to document departures from the published GSOP until such a time as MSC-approved changes were incorporated in official GSOP revisions. A NASA-comprised group known as the Software Control Board (with representatives of MIT) initiated and approved specific change concepts, whereupon the MDRB would monitor MIT compliance with these changes.

1.4 Man and Machine Loading Requirements

The story of Project Apollo's successful completion represents, in the end, a myriad of individual successes, most of which are based upon an intricately-tuned interaction among men and machines. For its own part, the story of MIT's software-development effort demonstrates the essential interdependence of talented scientists, engineers, mathematicians and technicians with increasingly complex, versatile and powerful computing equipment. As the tempo of the Laboratory's involvement in software tasks changed, these changes were reflected in the number and types of personnel participating in the effort, and in the power and speed of computers which the Laboratory acquired. This section discusses, in general terms, the history and philosophy of MIT's personnel and computing requirements.

1.4.1 History of Man Loading

1.4.1.1 Initial Philosophy

At the beginning of the Laboratory's participation in Project Apollo, a simple philosophy guided the staffing of the software-development group. Essentially, this philosophy placed a premium on engineers and scientists who, in addition to original, conceptual work, would put their own ideas into a form which machines could understand. Thus, in the early days of the Apollo work, there were no "programmers", as such. Instead, engineers and scientists learned the techniques of programming. At this stage, a relatively small group was thought capable of handling what was then considered a practicable task. It was believed that competent engineers with a credible, solid mathematical background could learn computer programming much more easily than programmers could learn the engineering aspects of the effort. The small size of the initial staff dictated that integration of engineering and programming talents in a few individuals would be preferable to attempts at intercommunication by individual engineers and programmers. Thus, the original intent was to have the project's basic core of engineers follow the program through, from conception to actual flight support.

With the passage of time, however, it became clear that the philosophy could best be followed in spirit, rather than in letter. As desirable as it might be to have a staff composed solely of multidisciplinary personnel, it was clearly impossible to shape such a staff beyond a certain size. Individuals talented in both engineering and in computer programming were not readily available. Also, as the software tasks became better delineated, it was apparent that a major underestimation of program-testing requirements had initially occurred. Because the Apollo Guidance Computer has a comparatively small erasable memory, the problem of having various people using the same registers for different tasks, the problem of overlaying memory —these all required extensive precautionary measures to avoid conflict. Optimally, one dedicated engineer/programmer assumed responsibility for ensuring that no erasable-memory conflict occurred, and for integrating the individual flight programs.

1.4.1.2 Creative Use of Subcontractors

Part of the solution to the problems discussed in the preceding section developed through the extensive use of subcontracted personnel. From the very beginning of MIT's participation in Project Apollo, the Laboratory had stressed that its frequent and extensive use of subcontractors would allow it manpower leverage essential to its responsibilities under the Apollo contract. Through the use of subcontracted personnel, the Laboratory would not be required to assemble and disassemble its own staff to meet the time-varying responsibilities of the Apollo program. Subcontractors would serve as a buffer for the Laboratory's staffing requirements. Importantly, Draper Laboratory personnel have traditionally enjoyed the benefits of long-term employment, so the use of subcontractors would permit Laboratory management to carry a mainline staff of a size that would assure maximum security to all personnel. As detailed in Section 1.1, MIT's extensive hiring of subcontractors during the Polaris project had been a strong point in its presentation to NASA in advance of the Apollo program. Thus, when it became apparent that work loads were greater than initially estimated—especially in the areas of testing and verification—subcontracted personnel were made available for virtually immediate deployment.

Throughout MIT's participation in Project Apollo, subcontractors have served in a variety of roles. They have provided a complement to the talents of the Laboratory's own staff. Except in the area of direct administration, subcontractors have played parts in virtually every phase of the software effort, including design, analysis, testing, verification and simulation. Perhaps most significantly, the ready availability of subcontracted personnel facilitated quick solutions to unexpected personnel requirements, since the Laboratory could hire such personnel without necessarily promising any long-term commitment. The costs—direct and indirect —relating to in-house staffing levels were therefore kept to a mimimum throughout.

1.4.1.3 Review of Man Loading

Figure 1.4-1 depicts the man-loading history of the Apollo program at MIT from September 1961 through March 1970. As well as containing a curve for the total personnel levels, the figure shows separate breakdowns for subcontracted hardware and software and total hardware and software levels.

Inclusion of the hardware-personnel figures demonstrates the relative personnel requirements for the hardware and software tasks under MIT's Apollo contract. Thus, the project manpower resources were concentrated on developing system hardware from 1961 through 1965. In 1966, this hardware-development effort rapidly tapered off, and the requirements for designing and developing the mission computer programs increased. In November 1966, the software effort captured precedence as the primary task of the Laboratory's Apollo division.

Figure 1.4-2 demonstrates some of the reasons for the rapid buildup of software personnel. In 1966, no fewer than five ropes were being developed at one time. During the following year, when much of the software buildup had already occurred, six ropes were worked on simultaneously. Figure 1.4-2 is also a milestone chart of the many decisions and events germane to the Apollo software efforts of MIT.

1.4.2 History of Digital Machine Loading

Digital-computation facilities have played a significant role in MIT's development of software for the Primary Guidance, Navigation and Control System of the Apollo spacecraft. As will be discussed in Sections II and III, digital computers

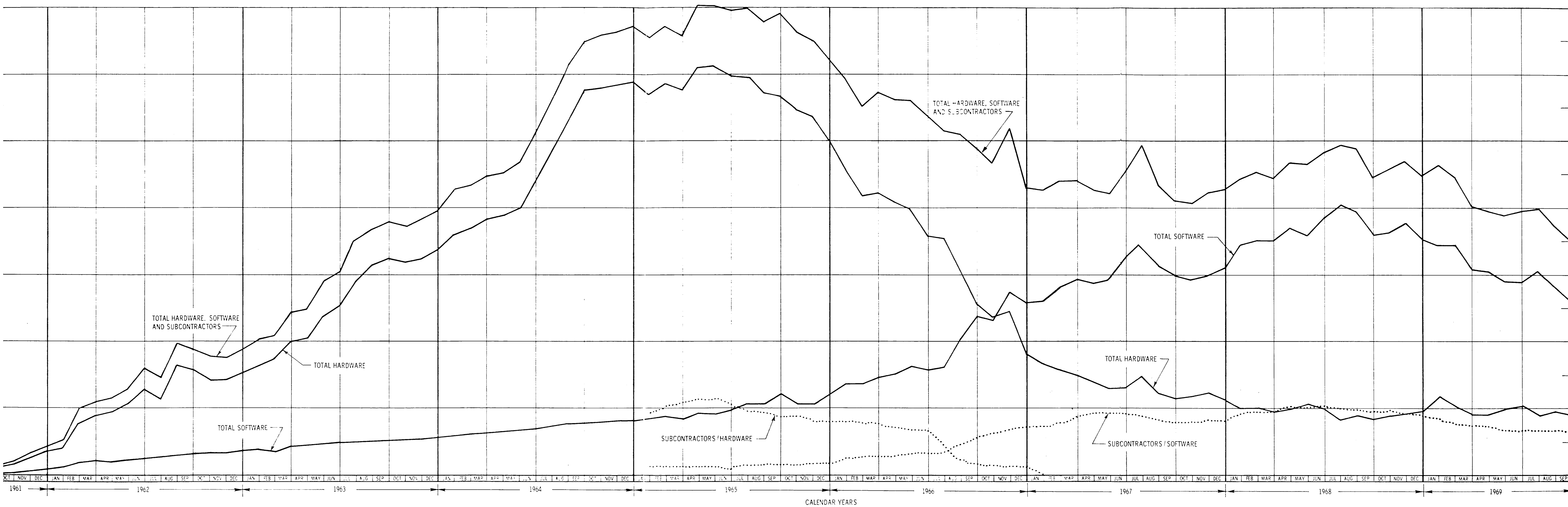


Figure 1.4-1 APOLLO Manloading - Charles Stark Draper Laboratory

are used in simulating the Apollo Guidance Computer during design, verification and testing of software. (In addition to this so-called "All-Digital Simulator" function of the digital computer, it serves as a basis for the Engineering Simulator, also described in Section III. A Hybrid Simulator and a Systems Test Laboratory also assisted in the test and verification of computer software and are also discussed in Section III.) This section discusses, in chronological sequence, the four types of digital computers around which the Draper Laboratory fashioned its digital-computation facilities. These computers are the IBM 650, the Honeywell 800, two Honeywell 1800s, and two IBM 360/75s.

During the period in which MIT has participated in the Apollo program, the computing facilities described in this section have served other Draper Laboratory groups in addition to the Apollo division. However, Apollo activities have accounted for about 90 percent of the total use of these facilities.

1.4.2.1 IBM 650

When the Draper Laboratory received its first contract from NASA, in September 1959, an IBM 650 provided the Laboratory with its in-house computing capability. The IBM 650 was a 2000-word-drum central processor, with 60 words of core storage. One tape drive and a disc bar were the only pieces of peripheral equipment. Programmers would write in MAC*, and the IBM 650 was used primarily to compile these programs for computation on much faster and more powerful outside equipment, such as the IBM 704, 709 and 7090. Toward the end of 1959, the burden of the NASA pre-Apollo workload, added to the much larger workload of the Laboratory's Polaris project, stimulated investigation into the possibility of providing additional in-house equipment to accommodate all the work then done by the IBM 650 and the outside rented machines.

---

*MAC is a high-level programming language for general-purpose computers, developed at MIT for scientific application. It is not to be confused with MIT's Project MAC. The latter was named independently some years later and is unrelated to the MAC language.

1.4.2.2 Honeywell 800

The Honeywell 800 was ordered during Summer 1960, with delivery occurring in December 1961. Based upon the workload of mid-1960, the H-800 was predicted to run about 4 hours per day and to cost no more than the previous total of in-house IBM 650 and outside rented time. By the time the H-800 was placed in production* —in May 1962—it was apparent that even greater speed and power were necessary. Rather than the expected four hours per day, two operator shifts (16 hours/day) were required for the initial H-800 workload. Despite the unexpected demands which the H-800 faced immediately upon being placed in production, it represented approximately a threefold increase over the capabilities of the IBM 650.

To overcome the inadequacy of the Honeywell 800, two approaches were undertaken simultaneously in mid-1962. First, additional memory and peripheral equipment were acquired for the H-800; second, an order was placed for the Honeywell 1800, with expected delivery 18 months later.

The Honeywell 800 had been delivered with a 16,000-word memory, each word having 48 bits. It included a printer, six tape drives and a card reader/punch. To upgrade the H-800 while awaiting delivery of the H-1800, the memory was doubled, additional tape drives and a printer were acquired, and a disc file and a graphic plotter were added.

1.4.2.3 Honeywell 1800

Honeywell's 1800 possesses a 2-μsec access-to-memory, while the H-800's access was on the order of 6 μsec. The H-1800's delivered memory size was 32,000 words, double that of the H-800. These capabilities rendered the delivered H-1800 roughly three times as powerful as the H-800.

Although the Laboratory's H-1800 was delivered in January 1964, it was not until the following May that the system was in total production. In the meantime, failures in hardware necessitated total replacement of the machine's memory. As

---

* "In production" implies that the equipment is capable of a complete AGC simulation. Computers were "in operation" before they could be "in production".

a result of these difficulties with the new system, between the months of January and May, no in-house digital-computing facilities were available, since the H-800 had been removed to provide space for the H-1800. Consequently, time was rented on outside equipment during this period.

By October 1964, it was becoming apparent that the H-1800 was computing much more rapidly than its peripheral equipment could provide input and output. At that time, Honeywell announced its Model 200 computer, a small machine that could do much of its own computation, could provide its own input and output, and could serve as a buffer for the much more powerful H-1800. MIT ordered a Model 200 for delivery in October 1965.

Two decisions were reached in Summer 1965 regarding the need for additional computing facilities: a second Honeywell 1800 was ordered in June; and a study was begun of the potential advantages offered by even more powerful computers. The second H-1800 was delivered and placed in production in March 1966. The investigation into other computers resulted in the Laboratory's decision to order an IBM 360, Model 75.

The original H-1800's memory had been increased in size from 32,000 to 48,000 words. The second H-1800 was delivered with the larger memory. By the time of the second H-1800's acceptance, a second Model 200 had also been acquired. The final upgrading of the H-1800 facilities occurred with the delivery of a Honeywell Model 2200, a system approximately equivalent to two Model 200s. It was estimated that the addition of the Model 2200 increased the capability of the H-1800 facilities by about 20 percent.

1.4.2.4 IBM 360/75

When the Summer-1965 study of large computing systems began, several systems were under consideration. One was highly valued, but doubts existed that it would ever be manufactured. Another system was by far the fastest machine under consideration, but Laboratory officials were concerned that internal parity checking would not reach the standard necessary to ensure the safety of astronauts —the ultimate customers of the Laboratory's services. Still another system was rejected primarily because it did not allow eventual expansion into an even larger system. Finally, the IBM 360, Model 75 (360/75) was chosen because of its relatively high speed, its degree of internal error checking, and the availability of the more powerful Model 91, should the need for expansion occur. It was estimated that a single IBM 360/75 would be roughly equivalent to four Honeywell 1800s.

The IBM 360/75 was delivered in October 1966, and it became operational two months later. During the first eight months of operation, three basic activities consumed most of the machine's availability: MAC language was adapted for the 360/75, system software was developed, and simulation software was implemented. During these first months of IBM 360/75 operation, it was concluded that the CPU's 512,000-byte memory would not suffice for simulation purposes; memory size was thereafter doubled. Not until September 1967, about ten months after delivery, was the IBM 360/75 in total production for general simulations.

By the time the IBM 360/75 came into total production, the need for a second IBM 360/75 was already recognized. Accordingly, the Honeywell 1800s would be removed. Removal of the second-delivered H-1800 occurred in December 1967, and the original H-1800 was removed in April 1968 to make way for the second IBM 360/75, to be delivered the following month. Thus, during the last quarter of 1967, three complete systems were in operation— the IBM 360/75 and the two H-1800s.

The second IBM 360/75 was placed in total production a mere two weeks after delivery, primarily as a result of the experience gained through the lengthy break-in procedures on the first IBM 360/75. By the time the second system was placed in production, the peripheral equipment originally delivered had also been expanded in power and capacity. For instance, the six original IBM 2311 disc packs were increased to ten. Two printers were added to the original two, and additional tape drives and a card reader were acquired. Finally, three IBM 2314 disc packs were gained, each of which was roughly equivalent to four IBM 2311s.

In August 1969, following Apollo 11's successful lunar mission, the second-delivered IBM 360/75 was removed, thus leaving the original IBM 360/75 and the systems's peripheral equipment as the remaining digital-computing facility of the Draper Laboratory. Although the remaining IBM 360/75 was deemed adequate for the needs after the lunar landing, within seven months it also reached saturation.

1.4.2.5 Loading of the Digital Computing Facilities

Figure 1.4-3 charts the monthly load which was logged on the Laboratory's digital-computing facilities, expressed in equivalent Honeywell-1800 hours. In this figure, monthly saturation of a Honeywell 1800 is 660 hours (=22 hours/day x 30 days). Since the Honeywell 800 was roughly a third as powerful as the H-1800, saturation of the H-800 occurs at 220 hours/month. The IBM 650 was, in turn, about one third as powerful as the H-800—or a ninth as powerful as the H-1800; thus on this graph its saturation is 73.3 hours/month. The IBM 360/75 is roughly four times as powerful as the H-1800, and thus its saturation occurs at 2640 hours/month. This figure also indicates the dates of computer acquisitions and removals.

1.5 Major Recurrent Problems

With the manned lunar landing and return accomplished in July 1969, Project Apollo met the national goal enunciated eight years earlier. Through its design, development and implementation of the Primary Guidance, Navigation and Control System for the Apollo spacecrafts, MIT's Draper Laboratory shared in that eminent success. Along the course of its participation in the Apollo adventure, MIT experienced the kinds of technical and managerial difficulties that can only be expected in undertaking so massive a program—but that nevertheless create uneasiness at the time of their occurrence. This section focuses on the two problems which caused the greatest difficulty in the software effort. Difficulties were encountered in the estimate of time and manpower schedules and in the control of accurate, up-to-date spacecraft data. Both of these problems continually plagued MIT's software efforts, since neither their cause nor their solutions could be found within the Laboratory, alone; ultimate solution would require an extraordinarily well-tempered orchestration among NASA and all of its contractors and subcontractors.

1.5.1 Difficulty in Estimating Time and Manpower Schedules

Throughout much of the Apollo software effort at MIT, managers have experienced difficulty in estimating the time and manpower requirements to design, test and verify the successive mission-flight programs. At the commencement of

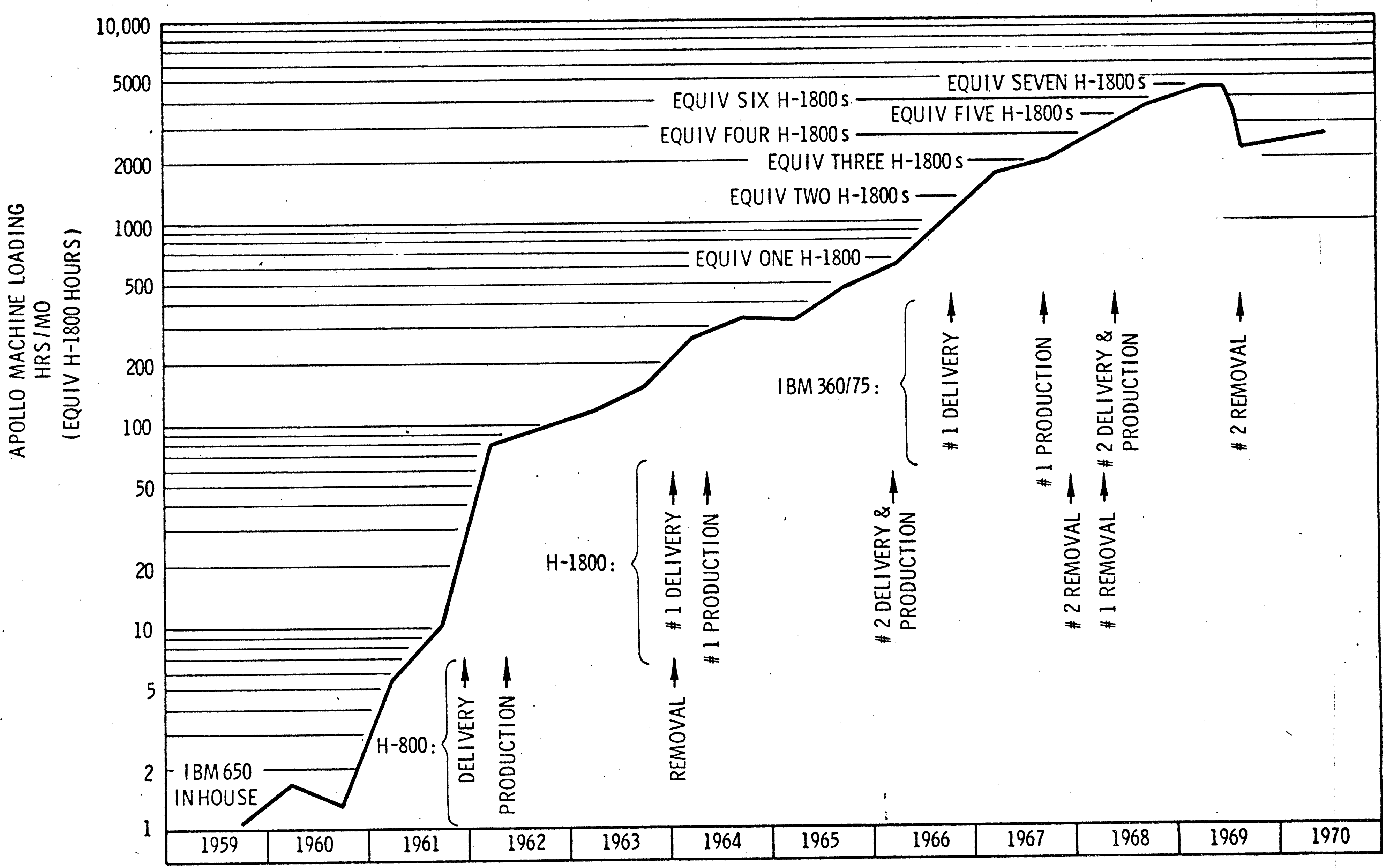


Figure 1. 4-3 Apollo Machine Loading

work on a new flight program, it is advantageous—perhaps even essential—to break down the total required effort into a series of smaller tasks, each fitting into a preplanned sequence of steps leading to the required whole. Specialists in each of the subdivided tasks can then be assigned stated responsibilities within a specified time constraint. This description fits the optimal situation—the situation in which the Draper Laboratory more nearly finds itself today than it has in the past.

It is more likely that at the commencement of work on an entirely new mission-flight program, the separate tasks required to lead to the assembled program cannot be known in advance. Indeed, this was the case with virtually every program up to the revisions in COLOSSUS and LUMINARY which currently suffice in the planning of new missions. Part of the development process includes the understanding of what these basic steps should be. In brief, at the beginning no one can forecast all the little pieces which will eventually be required, and thus predicting accurate work schedules is almost a priori an impossible task.

Another probable cause of this overall scheduling problem is that subtasks required an ordered interrelationship. Not all of the tasks could occur simultaneously; some took precedence over others, and certain later tasks could not proceed until the completion of earlier tasks. In other words, the entire sequence of tasks could be completed no sooner than the time required to complete perhaps a certain few "pacemaker" tasks. Perhaps the most difficult estimate to be made in advance is the amount of time required for iteration and retests. Thus, to adequately forecast accurate work schedules, the manager would have had to predict not only all the necessary subtasks, but, in addition, which few of these subtasks would be the pacemakers and which would later be redesigned and require further testing.

Another cause of the work-schedule problem relates to the vagaries of personal dynamics. Throughout much of the software effort, management encountered a problem of deadline definition; that is, when a deadline for rope release became known, a number of intermediate deadlines or goals had to be established, particularly for pacemaker tasks, to ensure that the final deadline be met. After all of the deadlines had been assigned, it was sometimes difficult to convince software personnel of the importance of meeting the earlier deadlines; the tendency was strong for those with the earlier tasks to aim toward the deadline for the completed flight program. Consequently, management was continually required to reemphasize the importance of meeting each assigned deadline.

A final cause of the work-schedule problem also relates to the area of human dynamics. The communication of "bad news"—e.g., news of imminent delays—slows as it goes up the line of management. This difficulty derives from the basic human drive to prefer the communication of positive tidings to that of negative findings. Both the bearer and the receiver of bad news feel uneasy with the experience, but management must encourage its personnel to communicate the bad with the good. When the person responsible for one of the subtasks recognizes that his schedule must slip, it is human nature to defer passing along word of the delay. As this one piece of bad news progresses up the ladder of administrative responsibility, communication of the bad news is further impeded. As the initial step in rationalization, each person along the line attempts to discover for himself whether the bad news is as bad as anticipated—or if, perhaps, some degree of overstatement has occurred. Only through conscious recognition of this process by all personnel can this problem be alleviated.

Thus, four separate causes combined to render the estimation of work schedules an especially vexing problem:

a. the difficulty of predicting all of the required subtasks;
b. the difficulty of pinpointing and hastening pacemaker subtasks;
c. the difficulty of meeting deadlines for individual subtasks;
d. the difficulty of communicating "bad news" quickly through the line of management.

As MIT gained experience through its successive responsibilities in the Apollo program, the work-schedule problem became increasingly more routine—and less annoying. Nevertheless, small remnants of this problem continue to cause occasional difficulties in the scheduling of current ropes.

1.5.2 Control of Timely Spacecraft Data

The second major problem encountered by MIT software planners relates to the acquisition of complete and up-to-date data on spacecraft parameters. In the design, verification and testing of guidance, navigation and control software, it was essential that the responsible MIT engineers possess the most current data obtained by other NASA contractors in the development of the spacecraft components. From the beginning, it was clear that a mechanism for such data exchange was of prime importance.

One of the initial responsibilities of liaison personnel was the development of a data-exchange mechanism. For instance, North American Rockwell's liaison with MIT was to record the most up-to-date information on the Command and Service Modules, and the liaison from Grumman was to do the same for the Lunar Module. In practice, however, this official mechanism broke down quickly, since spacecraft engineers were reluctant to formally release data on parameters still undergoing development, measurement or testing. Such virtually universal reluctance to commit preliminary data, even to discretionary use, rendered the officially-recognized channels rather dinosauric in current-information content. During years of effort to establish a smoothly-functioning, up-to-date data-exchange program, MIT software personnel resorted to other means for learning the parameters and tolerances to which they should design their software and simulations.

As MIT software personnel became acquainted with their peers at the other relevant spacecraft contractors, an informal network of data exchange developed. Rather than relying upon the official mechanism of liaison contact, the engineers responsible for the development of software would place strategic telephone calls to learn up-to-the-minute data being used in the development of the spacecraft systems. Although this informal method of data exchange possessed the disadvantage of consuming much valuable time, it produced the distinct benefit of collecting the most timely information available.

In an attempt to formalize the person-to-person method of data control, a "Data Book" which listed current data was organized at MIT. There were two sections within this document: class A data, which were official and verified by an authority at the originating contractor; and Class B data, the type generally received through telephone and person-to-person communication, but which lacked official verification. But the Data-Book mechanism required personal enthusiasm for the task of collecting data—enthusiasm which virtually all dedicated software engineers feel should better be devoted to the task of designing software.

All of the parameters and tolerances to which the software and simulations were designed were published in Chapter 6 of the GSOP. (See Section 3.2.1 of this report.) In this fashion, the Laboratory kept NASA continuously and officially apprised of MIT's current information—information which could be approved along with NASA's general approval of GSOP revisions.

By no means has the problem of timely data control been solved, but solely because of MIT's increased familiarity with the spacecraft components, it has become somewhat less of a problem. Just as there were elements of human dynamics in the problem of time and manpower scheduling above, so, too, did personal vagaries play a role in this difficulty: people are unwilling to divest themselves of data which they consider not yet final. And the very qualities of technical competence and conscientiousness which one needs to invest in the area of data exchange are difficult to come by, since individuals so endowed generally prefer to apply these qualities in the actual software development.

SECTION II

AGC SOFTWARE

This section describes the software which controls the present LM and CM guidance computers. The computer is the heart of the Apollo Guidance, Navigation and Control System. The software maintains positional knowledge of the vehicle in space, determines the path to a desired destination, and steers the spacecraft along that path by sending commands to the engines. It communicates with the astronauts and the ground, and monitors the performance of the GN&C System.

Mission programs, such as rendezvous, targeting and landing, control some of the phases of an Apollo flight. However, before these can be discussed, it is necessary to examine the underlying computer organization which allows the mission program to operate. Thus, Section 2.1 describes the basic machine architecture, the Executive and service programs which control AGC operations, and the input/output functions which allow the computer to monitor the GN&CS and to communicate with the astronauts and the ground. Although the CM and LM computers satisfy different mission requirements, the underlying system software is quite similar for the two vehicles. Hence Section 2.1 presents a generalized Apollo Guidance Computer, and specific differences are noted when they apply.

Section 2.2 includes a general description of all the phases of the Apollo mission and of the major flight tasks required for that mission. The design effort which produced these mission programs has been a long and challenging task. This report will not attempt to give a complete discussion of this effort, since it has been documented in other sources; however, the rope design philosophy and the problems encountered as it finally evolved are discussed in Section 2.2.4, and the major program capabilities are described in somewhat greater detail in the appendices to this report.

2.1 Computer Capabilities

2.1.1 Storage and Manipulation of Computer Instructions

The AGC contains two distinct memories, fixed and erasable, as well as various computer hardware. The fixed memory is stored in a wire braid which is manufactured and installed in the computer. This memory cannot be changed after manufacture and it can only be read by the computer. Fixed memory contains 36,864 "words" of memory grouped into 36 banks. Each word contains 15 bits of information (a sixteenth bit is used as a parity check). The word may contain either a piece of data, or an instruction which tells the computer to perform an operation. A series of instructions forms a routine or a program. In addition to storing programs, the fixed memory stores data such as constants and tables which will not change during a mission.

The erasable memory makes use of ferrite cores which can be both read and changed. It consists of 2048 words divided into 8 banks. Erasable memory is used to store such data as may change up to or during a mission, and is also used for temporary storage by the programs operating in the computer.

Included in the hardware is a Central Processing Unit (CPU). The CPU performs all the actual manipulation of data, according to the instructions designated by a program. The 34 possible machine instructions include arithmetic operations (add, multiply, etc.) as well as logical operations, sequence control, and input/output operations. Also included are a limited number of "double-precision" instructions which permit two words of data to be processed as a single "word" of greater precision.

The memory cycle time (MCT) in the AGC is 11.7 μsec. Most single-precision instructions (e.g., addition) are completed in two MCTs; most double-precision machine instructions are completed in three MCTs. The unconditional transfer-control instructions, however, operate in one MCT.

To be used as an instruction, a computer word must specify the operation to be performed and give the location of the data to be operated on. However, a 15-bit word does not contain enough information to specify 34 operations and 38,912 fixed and erasable locations. In fact, 15 bits cannot even specify 38,912 locations unambiguously. It is for this reason that both the fixed and the erasable memories are grouped into banks. An instruction may specify any address within its own bank, and may also address the first four banks of erasable and the first two banks of fixed memory. Access to other banks is accomplished using bank-selection registers in the CPU. In many cases a program exists entirely within one bank memory, in which case bank switching is not required.

Many of the tasks the AGC performs can be adequately carried out by machine instructions. However, for extensive mathematical calculations—in such areas as navigation—the short word length of the AGC presents difficulties. It limits the number of instructions available, the range of memory that can be addressed without switching banks, and the precision with which arithmetic data can be stored and manipulated. To alleviate these problems, nontime-critical mathematical calculations are coded in "interpretive language" and are processed by a software system known as Interpreter. Each Interpreter instruction is contained in two or more consecutive computer words. The increased information available allows more possible instructions and a greater range of memory addressable without bank switching. In fact, with some exceptions, all of erasable memory and fixed memory may be addressed directly. Among the available Interpreter instructions are a full set of operations on double-precision quantities, including square root and trigonometric functions, some triple-precision instructions, and a set of vector instructions such as cross product, dot product, matrix multiply, and vector magnitude. Interpreter routines translate an Interpreter instruction into an equivalent series of machine instructions to be performed by the CPU. Thus, one Interpreter instruction may be equivalent to many machine instructions, and much storage space is saved in the computer. The Interpreter also contains software routines for the manipulation and temporary storage of double- and triple-precision quantities and vectors.

Interpreter expands the processing capabilities of the CPU hardware. However, its operation is quite slow, since the CPU must perform all the actual operations, and much time is spent in the translation of instructions and the manipulation of data. Although processing time is slower, much storage space is saved in fixed memory by the more powerful Interpreter instructions; thus, the vast majority of nontime-critical mathematical computations are coded using interpretive language.

2.1.2 Timing and Control of the Computer

Two of the more stringent requirements placed upon the AGC are the need for real-time operations and the necessity for time-sharing of multiple tasks.

Certain computer functions must occur in real time. For example, certain input data must be stored or processed immediately upon receipt; and outputs, such as those which turn the jets on and off, must occur at precisely the correct time. An interrupt system causes normal computer operation to be suspended while performing such time-critical tasks.

Several programs, which are less time-critical, may all be required during a phase of the mission. Time sharing between these programs is controlled by a software executive system which monitors the programs and processes them in order of priority. The Executive can stop one job when a higher priority job is necessary, then resume the low-priority job when time is available.

2.1.2.1 Interrupt System

To permit quick response to time-dependent requests, the AGC has a complex interrupt structure. There are two classes of interrupts, counter interrupts and program interrupts. Counter interrupts have the highest priority of all AGC operations. Counters are locations in erasable memory which can be modified by inputs originating outside the CPU. Some counters are used as clocks, while others interface with spacecraft systems to receive or transmit sequences of data pulses. The counters respond to a set of involuntary instructions called counter interrupts, which may increment, decrement, or shift the contents of the counters. A counter interrupt suspends the normal operation of the CPU for one MCT, while the instruction is being processed. Except for the short time loss, the ongoing program is not affected by the counter interrupt; in fact, it is not aware that the interrupt has occurred. These inferrupts are used solely for counter update and maintenance; their priority assures that no information will be lost in the counters.

The use of counters as input/output devices will be described in Section 2.1.3.1; it is appropriate now, however, to discuss the six counters which are used for timing purposes. Two counters, designated TIME1 and TIME2, form a double-precision master clock in the AGC. TIME1 is incremented at the rate of 100 counter interrupts per second. Overflow of TIME1 triggers a counter interrupt to increment TIME2. Since total time that must elapse before TIME2 overflows exceeds 31 days, TIME1 and TIME2 are thus able to keep track of total elapsed mission time.

The remaining clock-counters, designated TIME3 through TIME6, measure time intervals needed by the AGC hardware and software. For example, autopilot computations must be processed periodically whenever the autopilot is in use. Before reaching completion, these computations preset the TIME5 counter so that it will overflow at a specified time in the future. TIME5 is incremented at the rate of 100 counter interrupts per second. When TIME5 overflows, a signal sent to the CPU causes a "program interrupt" which interrupts the program in process and begins the autopilot computations once again.

Program interrupts have lower priority than counter interrupts, but greater priority than normal program operation. Unlike counter interrupts, the purpose of program interrupts is to alter the normal processing sequence. There are 11 program interrupts; they may be triggered by a clock-counter overflow, as in the example given above, or by externally generated signals, such as the depression of a key on the Display and Keyboard (DSKY) by an astronaut. The occurrence of a program interrupt causes the computer to suspend normal operation at the end of the current instruction. The current CPU data are saved, the computer is placed in interrupt mode, and control is passed to a preassigned location in fixed memory. This preassigned location is the beginning of a program which performs the action appropriate to the interrupt. While the interrupt program is running, the computer remains in interrupt mode, and no additional program interrupts will be accepted, although counter interrupts can still occur. (Requests for other program interrupts are stored by the hardware and processed before returning to normal operation.) At the conclusion of the interrupt program, a "resume" instruction is executed. If there are no other program interrupts, the CPU is taken out of interrupt mode, the original contents are restored, and the program returns to the point at which it was interrupted. One program interrupt (restart) takes precedence over all the others, and can even interrupt an interrupt. It results from various kinds of computer malfunctions. (This interrupt will be discussed in Section 2.1.4.)

A computation which takes place by means of a program interrupt is called a task. Since tasks may not be interrupted, they must be short to avoid delaying other tasks. This speed requirement precludes the use of interpretive language.

One class of tasks is initiated by overflow of time counters TIME3, TIME4, TIME5, and TIME6. These are considered time-dependent tasks. The TIME5 interrupt, described above, initiates autopilot computations at precise periodic intervals. TIME6 controls the timing of the autopilot RCS jet firings. TIME4 initiates a series of routines which periodically monitor the IMU, radar, etc., and process input/output commands. The TIME3 counter is under the control of the software executive system (described below). It is available for general use by any program needing to schedule a task for a specific time.

A second class of tasks is initiated by interrupts caused by external action. For example, depressing a DSKY key initiates a task that begins processing DSKY readings and storing the information for later processing. Telemetry and the radar also cause interrupts that initiate tasks to receive or transmit the next data word.

2.1.2.2 Software Executive System

Computation in the AGC is managed by a software executive system comprised of two groups of routines, Executive and Waitlist. This system controls two distinct types of computational units, jobs and tasks. In its normal operating mode, the computer processes jobs. These are scheduled by the Executive, according to a priority system. The Waitlist uses the TIME3 interrupt to schedule tasks for a specific time in the future. (Tasks originated by the other program interrupts take place independently of the software executive system.)

Most AGC computations are processed as jobs. Division of a program into discrete jobs is at the discretion of the programmer, who also assigns a priority to each job indicative of its importance. The Executive can manage up to seven jobs (eight in the LM program) simultaneously.

To schedule a job, the Executive places the job's priority and beginning location on a list, assigning the job a set of working storage locations called a core set. In addition, if a job requires a larger working storage, as in the use of interpretive language, a second area, called a VAC area, may be assigned. The Executive is capable of maintaining seven core sets (eight in the LM program) and five VAC areas as each is assigned to a job, and of redesignating them as available when the job is finished.

A job in process must periodically call Executive to scan the list of waiting jobs, thus determining if any scheduled job has a priority higher than itself. If so, the job currently active is suspended and the higher priority job is initiated. To permit suspension of a job and subsequent resumption at a point other than its beginning, the working storage associated with the job is saved when the job is suspended and restored when the job is reinstated. A suspended job is returned to the job list and is not reinstated until it has the highest priority on the list. Eventually, a given job will run to completion, at which time it is removed entirely from consideration. When all jobs on the list have run to completion, a "DUMMYJOB" with zero priority constantly checks to see if new jobs have appeared. (The computer also performs a self-check, as described in Section 2.1.4.)

The relative importance of a job may change for various reasons. When this is the case, Executive changes the priority list and rechecks the list for the job of highest priority. Many times it is desirable to purposely suspend the execution of a job, but not to terminate it completely. Temporary suspension is desirable to await an event such as the input or output of data, or for the availability of a nonreenterable subroutine currently in use. To accomplish temporary suspension, Executive saves the job's interrupted registers and sets its priority to a negative value. Because the interrupted job has a negative priority, DUMMYJOB has priority over it. As a result, the job is, in effect, suspended indefinitely. Eventually, Executive is called to restore the job, usually by the event for which the job is waiting. Executive restores the original priority and again checks the list for the highest priority job.

Waitlist allows any program to schedule a task to occur at a specified time in the future. The TIME3 clock interrupts the job in process at the correct time and initiates the task. (As mentioned before, tasks initiated by the other program interrupts are not controlled by the Executive.)

To schedule a new task, Waitlist requires the starting address of the task and the amount of time which must elapse before execution. Waitlist maintains a list of tasks waiting to run in the order in which they will be performed and a list of time differences between adjacent items on the task list. It determines when the new task will run in relation to others on the list, placing it appropriately in the list.

The TIME3 counter counts the time to the first item on the list. When this time arrives, the TIME3 program interrupt occurs. TIME3 is immediately set to overflow when the time has elapsed for the next task on the list, and all tasks and times move up one position on the list. The computer remains in interrupt mode until the task is completed. It is then free to process other interrupts or return to the original job.

Since TIME3 is a single precision AGC word (15 bits) that is incremented 100 times a second, Waitlist can process tasks up to 162.5 sec in the future. For longer delays, a routine called LONGCALL processes a single task—the repeated calling of Waitlist. LONGCALL can schedule tasks for as long as 745 hours in the future, a time span larger than an entire Apollo mission.

2.1.2.3 Sequence Control

In normal AGC operation, the Executive maintains a constant background of activity, while program interrupts break in for short, time-critical bursts. The execution of a job is subject to numerous interruptions. A counter interrupt may occur after the completion of any instruction. Program interrupts stop the job in process. While the computer is in interrupt mode, any further program interrupts are saved by the hardware and processed one at a time before returning to the job. Under control of the Executive, high-priority jobs also steal time from a job in process. This control system of interrupts and priorities ensures that in times of heavy load, the most critical computations for the mission will be processed first.

Normally, the CPU does not stop during periods of low activity. If no jobs or tasks are being executed, the CPU executes a short loop of instructions (DUMMYJOB) which continually looks for jobs to initiate. Periodically, TIME4 overflows, initiating a task to monitor various GN&C subsystems. If an autopilot is in operation, TIME5 triggers other interrupts for autopilot functions. In addition, periodic counter interrupts will occur as counter input is received and clock counters are updated. More extensive computer activity awaits action by the astronaut, as described in the following section.

2.1.3 Computer Interfaces

To perform its various functions, the AGC must interact with the other spacecraft systems, the astronaut, and the ground. External to the AGC are the various sensors and controls which provide inputs, and the spacecraft systems and displays which receive outputs. Figures 2.1-1 and 2.1-2 illustrate the signal interconnections between the computer and the external hardware for the CM and LM systems, respectively. This report will not, in general, discuss these external equipments, except as they apply to specific AGC programs. (See functional description treated in Part 1, Chapter II, and Part 2, Chapter I of this Apollo Final Report.)

Within the AGC, the actual transmission of data is accomplished through special registers known as counters and channels, as discussed below. Various AGC programs process the input and output data. A mission program such as rendezvous will interrogate selected counters and channels for the specific input data it requires. The program will, in turn, issue commands by means of these interfaces. The operation of the mission programs is discussed in Section 2.2 and in the appendices. In addition to the mission programs, there are also special programs designed to process input/output information for purposes of telemetry and communication with the astronauts. These interfaces are discussed in the present section.

2.1.3.1 Counters and Channels

All AGC input/output takes place through counters and channels. Counters are used for the transmission and reception of numeric data; channels are used for the communication of discrete* data.

Channels are solid-state registers in the CPU that do not form part of memory. They cannot be referenced by most machine-language instructions, but are read and in some cases written into by means of special channel instructions. Each channel can consist of up to 15 separate bits or discretes. For input channels, the

---

*The AGC has 15 input and output channels whose bits are individually distinct (i.e., discrete). Each bit either causes or indicates a change of state, e.g., liftoff, zero optics, SPS-engine on, RCS-jet on, etc.

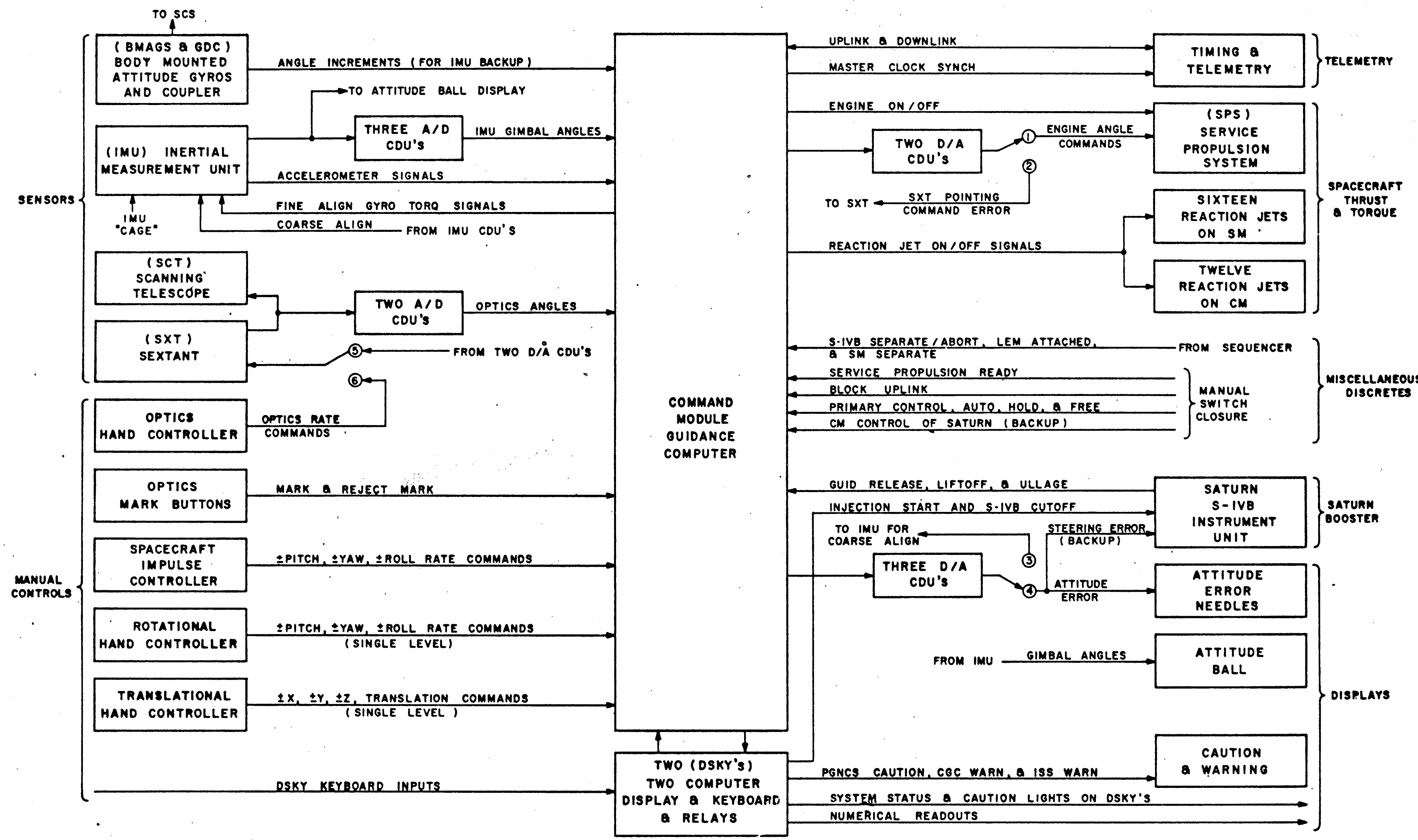


Figure 2.1-1 Guidance, Navigation and Control Interconnections in the Command Module

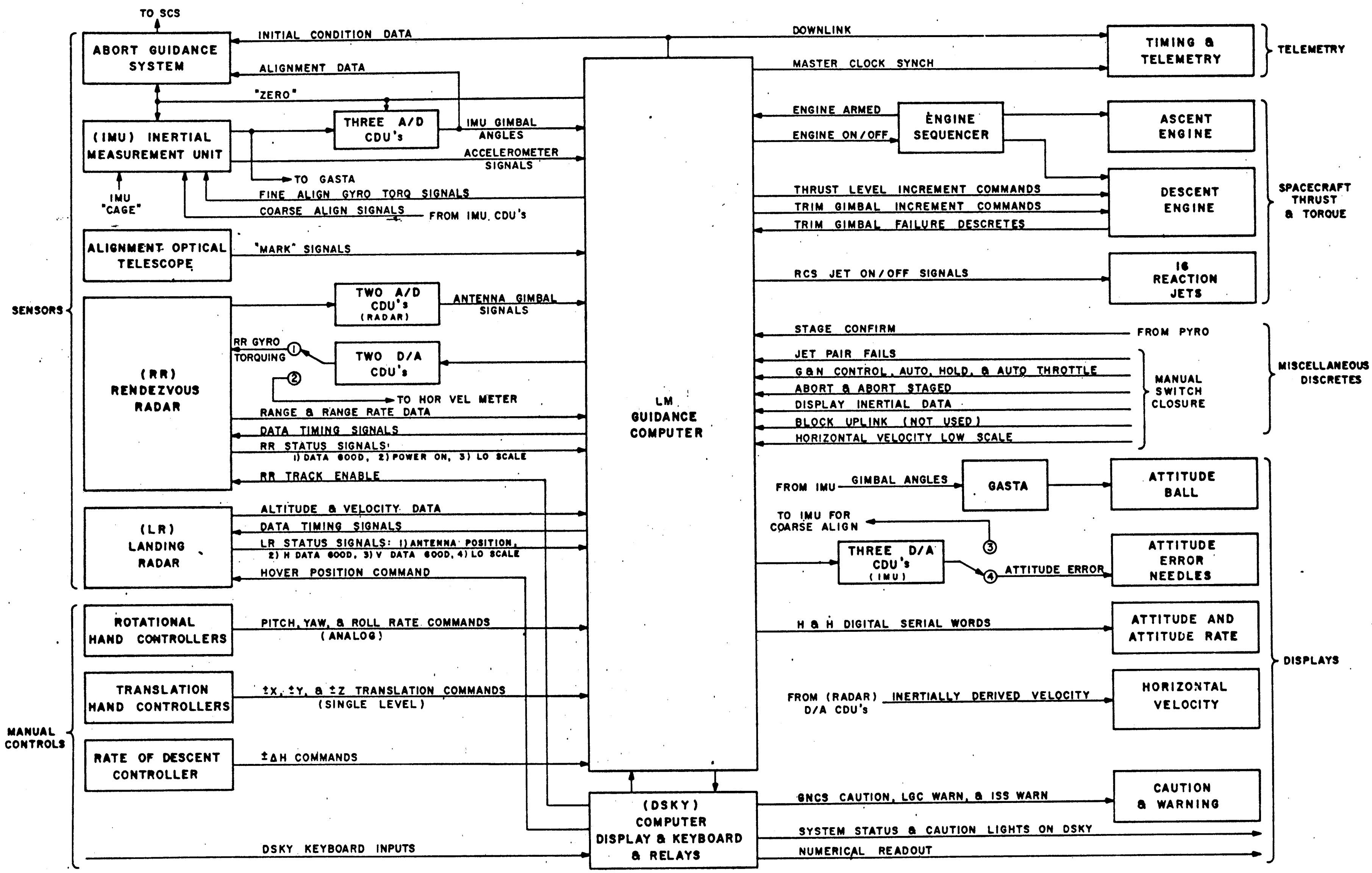


Figure 2.1-2 Guidance, Navigation and Control Interconnections in the Lunar Module

discretes are set by external G&N hardware and may be read by the computer. The input channels inform the computer of the state of the hardware, such as a hand controller out of detent, or the last key depressed on the DSKY. Output channels are written into by the computer to command external hardware functions, such as turning jets on or off, changing the DSKY display, or turning on panel lights. The AGC reads or writes into channels only when instructed to do so—either by the ongoing program or by a program interrupt. For example, pressing a key on the DSKY changes the information in channel 15; it also initiates the KEYRUPT1 program interrupt which causes the computer to read channel 15.

Counters are used for the input and output of numerical information. As described in Section 2.1.2, counters can be changed by programs as if they were ordinary erasable locations, but the counters also respond to counter interrupts which are not under program control.

For input, a typical operation requires that a counter first be set to zero under program control. The counter may then be incremented or decremented, one count at a time, via counter interrupts triggered by an external device. Thus, a counter is able to keep track of the state of the external device. An example of this kind of counter is that used with the Coupling Data Units (CDUs), the interfaces between the Inertial Measurement Unit and the AGC. For each 39.5-arcsec change in a particular gimbal angle, the CDU generates a signal to the AGC which causes a decrement or increment counter interrupt to the appropriate counter.

The output counters function in a similar way. The program sets the counter to an initial value which is later "enabled" via a channel discrete. Following the initialization, all action is automatic and not under program control. A series of counter interrupts decrement the counter toward a value of zero. For each interrupt, a signal of appropriate sign is sent to an external device. When the counter reaches zero, another signal is generated which stops the counting process. Thus, the number of signed pulses sent out is equal to the original contents of the counter. For example, signed pulses torque the gyros or control the optics shaft and trunnion drives.

For telemetry input, counter interrupts shift a pattern of bits into the counter. Selective use of two types of interrupts achieves the desired pattern after the counter has been cleared under program control.

2.1.3.2 Cockpit Displays and Controls

The Apollo Guidance, Navigation and Control System has been designed to utilize the best features of man and machine. Many mission tasks are best left to the computer, such as those that are extremely tedious or that require accurate response too rapid to lie within man's capabilities. However, man's judgment and adaptability, his decision-making capability in reacting to unanticipated situations, and his unique ability to recognize and evaluate patterns are all necessary for mission success. The Apollo displays and controls have therefore been designed to provide the crew with the most flexibility in monitoring and controlling the spacecraft. The astronaut can choose to be directly involved in the procedures, or to allow automatic operation which he can monitor.

Displays available to the crew in both the CM and LM are the attitude ball, attitude-error needles, attitude-rate needles, caution and warning lights, and a DSKY. The LM has additional displays which give the astronaut essential information during the descent to the lunar surface; these are the altitude/altitude-rate, horizontal-velocity, and thrust-level meters and the Landing Point Designator.

Several manual controllers enable the astronaut to become directly involved in spacecraft control. Both the CM and LM have rotational and translational hand controllers. The LM has a rate-of-descent controller. In the CM, additional controllers are used in conjunction with the optics; these are the minimum-impulse and optics hand controllers and the optics mark buttons. In the LM, a DSKY command can convert the rotational hand controller to a minimum impulse controller. All of these controllers make available to the astronaut a large repertoire of manual maneuvers.

The basic man/computer interface device is the DSKY (shown in Fig. 2.1-3). Through the DSKY the astronaut can initiate, monitor, or change programs being processed by the computer. He can request the display of specific data or enter new data. Communication with the DSKY is two-way; just as the astronaut can exercise command via the DSKY, the computer can request the astronaut to monitor, approve, or enter data when necessary. There are two DSKYs available in the CM and one in the LM. Each DSKY has a keyboard, several electroluminescent displays, and activity and alarm lights. The activity lights are for the computer and the

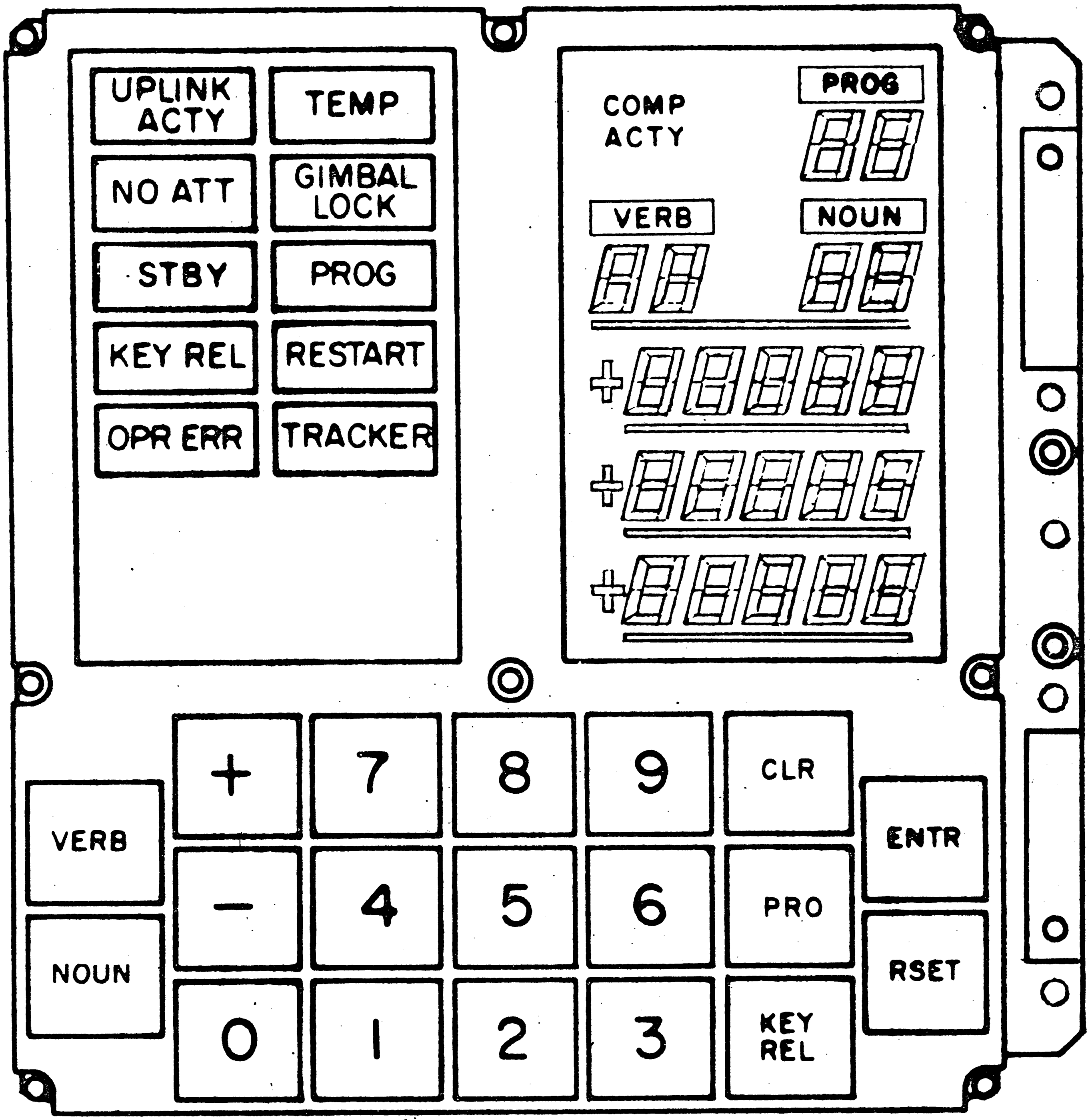


Figure 2.1-3 Display and Keyboard

telemetry uplink, and the alarm lights are for the computer and inertial subsystems. These aid the astronaut in monitoring the status of the G&N system. The alarm lights indicate equipment-failure and program alarms. There are two levels of program alarms. The more serious type of alarm either terminates all but the most necessary program activities or terminates all current program activities and requests astronaut action. The latter is accomplished by a preemptive flashing display of an error code indicating the cause of the alarm. The other type of program alarm is also indicated by the program-alarm light, but in this case the program in process continues without change. Should the astronaut wish to interrogate the cause of this alarm, he can key in a request to the computer to display the error code. The DSKY keyboard and displays are discussed in the next section.

2.1.3.3 PINBALL and DSKY Displays

The AGC program which responds to DSKY buttons and requests illumination of the DSKY lights is called PINBALL GAME BUTTONS AND LIGHTS—or PINBALL, for short. PINBALL is under Executive control and enables communication between the computer and the astronaut. As mentioned in the previous section, exchanges can be initiated by operator action or by an internal computer program. Four modes of operation are associated with PINBALL—internal data display, external data loading, systems-test usage, and initiation of large-scale mission phases. Internal data can be displayed once for verification (e.g., the ascent-injection parameters for lunar ascent) or periodically updated and displayed for monitoring (e.g., time-to-go to main-engine ignition). External data are displayed in the appropriate display-panel register as they are keyed into the DSKY. The data for the loading (external) and displaying (internal) modes can be presented in octal or decimal format; if internal data are presented in decimal format, the program supplies the appropriate scale factors for the display. PINBALL can also initiate a class of routines used for systems-test functions which might require operator interaction to determine whether to stop or continue the routine. The final mode of PINBALL is initiation of large-scale mission phases by operator action, i.e., by changing the mission program via the DSKY. (Fig. 2.1-4 lists the AGC programs for the CM and LM available during a lunar mission.)

The DSKY keyboard contains the following notations: VERB, NOUN, +, -, the numerical characters 0 through 9, CLR (clear), ENTR (enter), RSET (error reset),

Command Module AGC Programs		Lunar Module AGC Programs	
00	CMC* Idling	00	LGC** Idling
01	Prelaunch Initialization	06	GNCS Power Down
02	Gyro Compassing		
03	Verify Gyro Compassing	12	Powered Ascent Guidance
06	CMC Power Down		
07	IMU Ground Test	20	Rendezvous Navigation
		21	Ground Track Determination
11	Earth Orbit Insertion (EOI) Monitor	22	Lunar Surface Navigation
17	Transfer Phase Initiation (TPI) Search	25	Preferred Tracking Attitude
		27	LGC Update
20	Rendezvous Navigation		
21	Ground Track Determination	30	External ΔV
22	Orbital Navigation	31	Lambert Aimpoint Maneuver
23	Cislunar Midcourse Navigation	32	Coelliptic Sequence Initiation (CSI)
27	CMC Update	33	Constant Delta Height (CDH)
		34	Transfer Phase Initiation (TPI)
30	External ΔV	35	Transfer Phase Midcourse (TPM)
31	Lambert Aimpoint Maneuver	38	Stable Orbit Rendezvous (SOR)
32	Coelliptic Sequence Initiation (CSI)	39	Stable Orbit Midcourse (SOM)
33	Constant Delta Height (CDH)		
34	Transfer Phase Initiation (TPI)	40	DPS
35	Transfer Phase Midcourse (TPM)	41	RCS
37	Return to Earth (RTE)	42	APS
38	Stable Orbit Rendezvous (SOR)	47	Thrust Monitor
39	Stable Orbit Midcourse (SOM)		
		51	IMU Orientation Determination
40	SPS	52	IMU Realign
41	RCS	57	Lunar Surface Align
47	Thrust Monitor		
		63	Braking Phase
51	IMU Orientation Determination	64	Approach Phase
52	IMU Realign	65	Landing Phase (Auto)
53	Backup IMU Orientation Determination	66	Landing Phase (ROD)
54	Backup IMU Realign	67	Landing Phase (Manual)
		68	Landing Confirmation
61	Maneuver to CM/SM Separation Attitude		
62	CM/SM Separation & Preentry Maneuver	70	DPS Abort
63	Entry-Initialization	71	APS Abort
64	Entry-Post 0.05 g	72	CSM CSI Targeting
65	Entry-Up Control	73	CSM CDH Targeting
66	Entry-Ballistic	74	CSM TPI Targeting
67	Entry-Final Phase	75	CSM TPM Targeting
		76	Target ΔV
72	LM Coelliptic Sequence Initiation (CSI)	78	CSM SOR Targeting
73	LM Constant Delta Height (CDH)	79	CSM SOM Targeting
74	LM TPI Targeting		
75	LM TPM Targeting		
76	Target ΔV		
77	LM TPI Search		
78	LM SOR Targeting		
79	LM SOM Targeting		

*CMC is Command Module Computer (CM AGC)

**LGC is Lunar Guidance Computer (LM AGC)

Figure 2.1-4 Programs for a Lunar-Landing Mission

PRO (proceed), and KEY REL (key release). Each of these notations is internally represented by a 5-bit binary code which is transmitted and recognized by the computer. When the operator depresses any one of these buttons on the keyboard, an interrupt program called KEYRUPT enters a request to the Executive for another program that decodes and stores the key code in an input register of the AGC.

The numeric sections of the DSKY panel form three data-display registers, R1, R2 and R3, which can contain up to five numerals each, and three control display registers, VERB, NOUN, and PROG (program), of two numerals each. Each of the three data display registers has a sign section which displays a plus sign, a minus sign or nothing at all (blank). The PROG register indicates the mission program currently operating; the VERB and NOUN registers indicate the display and load activity initiated by the operator or by the computer. All information necessary to operate the display panel on the DSKY is transmitted from the computer through an output register which activates two display characters at a time. The basic language used for communication between the operator and PINBALL is a pair of two-character numbers that represents a verb/noun combination. The verb code indicates the operation to be performed, while the noun code indicates the operand to which the operation (verb) applies. Typical of the verb codes used are those for displaying and loading data. Noun codes call up groups of erasable registers within computer memory. Figures 2.1-5 and 2.1-6 give a list of the verbs and nouns available in the AGC for the CSM program COLOSSUS. (The LM program, LUMINARY, has a similar list.)

In addition to the numeric buttons and verb/noun control buttons, PINBALL responds to the other control buttons found on the DSKY. The RSET button usually turns off the alarm lights on the panel. Should any of these alarm lights remain on after the RSET button is depressed, the condition causing the alarm persists. The ENTR button has two functions: it causes the AGC to execute the verb/noun combination appearing in the VERB and NOUN registers or to accept a newly-entered data word. The CLR button is used to blank R1, R2, or R3 during a data-loading sequence, thus allowing reloading of a data word. The KEY REL button allows internal programs to use the DSKY if the operator has not previously released the DSKY for such use. The KEY REL light is turned on when an internal program attempts to use the DSKY but finds that the astronaut has not released it for internal use; depressing the KEY REL button performs this release. Thus, the operator

REGULAR VERBS

00 NOT IN USE
01 DISPLAY OCTAL COMP 1 IN R1
02 DISPLAY OCTAL COMP 2 IN R2
03 DISPLAY OCTAL COMP 3 IN R3
04 DISPLAY OCTAL COMP 1,2 IN R1,R2
05 DISPLAY OCTAL COMP 1,2,3 IN R1,R2,R3
06 DISPLAY DECIMAL IN R1 OR R1,R2 OR R1,R2,R3
07 DISPLAY DP DECIMAL IN R1,R2
08 SPARE
09 SPARE
10 SPARE
11 MONITOR OCTAL COMP 1 IN R1
12 MONITOR OCTAL COMP 2 IN R2
13 MONITOR OCTAL COMP 3 IN R3
14 MONITOR OCTAL COMP 1,2 IN R1,R2
15 MONITOR OCTAL COMP 1,2,3 IN R1,R2,R3
16 MONITOR DECIMAL IN R1 OR R1,R2 OR R1,R2,R3
17 MONITOR DP DECIMAL IN R1,R2
18 SPARE
19 SPARE
20 SPARE
21 LOAD COMPONENT 1 INTO R1
22 LOAD COMPONENT 2 INTO R2
23 LOAD COMPONENT 3 INTO R3
24 LOAD COMPONENT 1,2 INTO R1,R2
25 LOAD COMPONENT 1,2,3 INTO R1,R2,R3
26 SPARE
27 DISPLAY FIXED MEMORY
28 SPARE
29 SPARE
30 REQUEST EXECUTIVE
31 REQUEST WAITLIST
32 RECYCLE
33 PROCEED
34 TERMINATE
35 TEST LIGHTS
36 REQUEST FRESH START
37 CHANGE PROGRAM
38 SPARE
39 SPARE

EXTENDED VERBS

40 ZERO CDU (W N20)
41 COARSE ALIGN CDU (W N20,N91)
42 PULSE TORQUE GYRO
43 LOAD FDAI ATT ERROR NEEDLES (TEST ONLY)
44 SET SURFACE FLAG
45 RESET SURFACE FLAG
46 ACTIVATE DAP
47 SET LM STATE VECTOR INTO CSM STATE VECTOR

Figure 2.1-5 Verbs Used in Program COLOSSUS

48 LOAD DAP DATA (R03)
49 START CREW DEFINED MANEUVER (R62)
50 PLEASE PERFORM
51 PLEASE MARK
52 MARKED ON OFFSET LANDING SITE
53 PLEASE MARK ALTERNATE LOS
54 START REND BACK UP SIGHTING MARK (R23)
55 INCREMENT CMC TIME (DECIMAL)
56 TERMINATE TRACKING
57 START REND SIGHTING MARK (R21)
58 RESET STICK FLAG AND SET V50N18 FLAG
59 PLEASE MARK (OPTICS CALIBRATION)
60 SET ATTITUDE ERROR REFERENCE TO PRESENT ATTITUDE.
61 SELECT MODE 1 (DISPLAY DAP ATTITUDE ERROR)
62 SELECT MODE 2 (DISPLAY TOTAL ATTITUDE ERROR (N22-N20))
63 SELECT MODE 3 (DISPLAY TOTAL ASTRONAUT ATTITUDE ERROR (N17-N20))
64 START S-BAND ANT CALC (R05)
65 START OPTICAL VERIFICATION OF PRELAUNCH ALIGNMENT (P03)
66 SET CSM STATE VECTOR INTO LM STATE VECTOR
67 START W-MATRIX RMS ERROR DISPLAY
68 CSM STROKE TEST ON
69 RESTART
70 UPDATE LIFTOFF TIME (P27)
71 UNIVERSAL UPDATE-BLOCK ADR (P27)
72 UNIVERSAL UPDATE-SINGLE ADR (P27)
73 UPDATE CMC TIME (OCTAL) (P27)
74 INITIALIZE ERASABLE DUMP VIA DOWNLINK
75 BACKUP LIFTOFF
76 SET PREFERRED ATTITUDE FLAG
77 RESET PREFERRED ATTITUDE FLAG
78 UPDATE PRELAUNCH AZIMUTH
79 START BARBECUE MODE ROUTINE (R64)
80 UPDATE LM STATE VECTOR
81 UPDATE CSM STATE VECTOR
82 REQUEST ORBIT PARAM DISPLAY (R30)
83 REQUEST REND PARAM DISPLAY #1 (R31)
84 SPARE
85 REQUEST REND PARAM DISPLAY #2 (R34)
86 REJECT REND BACK UP SIGHTING MARK.
87 SET VHF RANGE FLAG
88 RESET VHF RANGE FLAG
89 START REND FINAL ATTITUDE MANEUVER (R63)
90 REQUEST REND OUT OF PLANE DISPLAY (R36)
91 BANKSUM
92 START IMU PERFORMANCE TEST (P07)
93 ENABLE W MATRIX INITIALIZATION
94 ENABLE CISLUNAR TRACKING RECYCLE
95 SPARE
96 TERMINATE INTEGRATION AND GO TO P00
97 THRUST FAIL DISPLAY
98 SPARE
99 ENABLE ENGINE IGNITION

Figure 2.1-5 (cont.) Verbs Used in Program COLOSSUS

Noun	Description	Format
00	NOT IN USE	
01	SPECIFY ADDRESS (FRAC)	.XXXXX FRAC .XXXXX FRAC .XXXXX FRAC
02	SPECIFY ADDRESS (WHOLE)	XXXXX. INTEG XXXXX. INTEG XXXXX. INTEG
03	SPECIFY ADDRESS (DEGREE)	XXX.XX DEG XXX.XX DEG XXX.XX DEG
04	SPARE	
05	ANGULAR ERROR/DIFFERENCE	XXX.XX DEG
06	OPTION CODE	OCT OCT
07	CHANNEL/FLAGWORD/ERASABLE OPERATOR	OCT OCT OCT
08	ALARM DATA	OCT OCT OCT
09	ALARM CODES	OCT OCT OCT
10	CHANNEL TO BE SPECIFIED	OCT
11	TIG OF CSI	00XXX. HRS 000XX. MIN 0XX.XX SEC
12	OPTION CODE	OCT OCT
13	TIG OF CDH	00XXX. HRS 000XX. MIN 0XX.XX SEC
14	SPARE	
15	INCREMENT ADDRESS	OCT
16	TIME OF EVENT (USED BY EXT VERB ONLY)	00XXX. HRS 000XX. MIN 0XX.XX SEC
17	ASTRONAUT TOTAL ATTITUDE (USED IN MODE 3 NEEDLES (V43))	XXX.XX DEG XXX.XX DEG XXX.XX DEG
18	BALL ANGLES AUTO MANEUVER	R XXX.XX DEG P XXX.XX DEG Y XXX.XX DEG
19	SPARE	
20	PRESENT ICDU ANGLES	R XXX.XX DEG P XXX.XX DEG Y XXX.XX DEG
21	PIPAS	X XXXXX. PULSES Y XXXXX. PULSES Z XXXXX. PULSES
22	NEW ICDU ANGLES	R XXX.XX DEG P XXX.XX DEG Y XXX.XX DEG
23	SPARE	
24	DELTA TIME FOR CMC CLOCK	00XXX. HRS 000XX. MIN 0XX.XX SEC
25	CHECKLIST (USED WITH V50)	XXXXX.
26	PRIO/DELAY,ADRES,BBCON	OCT OCT OCT
27	SELF TEST ON/OFF SWITCH	XXXXX.
28	SPARE	
29	CSM LAUNCH AZ	XXX.XX DEG
30	TARGET CODE (GYROCOMPASSING VERIFICATION)	XXXXX. XXXXX. XXXXX.

Figure 2.1-6 Nouns Used in Program COLOSSUS

Noun	Description	Format
31	*SPARE*	
32	TIME FROM PERIGEE	XXX. HRS 00XXX. MIN 0XX.XX SEC
33	GETI	00XXX. HRS 000XX. MIN 0XX.XX SEC
34	TIME OF EVENT	00XXX. HRS 000XX. MIN 0XX.XX SEC
35	TIME FROM EVENT	00XXX. HRS 000XX. MIN 0XX.XX SEC
36	TIME OF CMC CLOCK	00XXX. HRS 000XX. MIN 0XX.XX SEC
37	GETI(TPI)	00XXX. HRS 000XX. MIN 0XX.XX SEC
38	TIME OF STATE VECTOR	00XXX. HRS 000XX. MIN 0XX.XX SEC
39	DELTA TIME FOR TRANSFER	00XXX. HRS 000XX. MIN 0XX.XX SEC
40	TFI/TFC VG DELTA V (ACCUMULATED)	XXBXX M-S XXXX.X FPS XXXX.X FPS
41	TARGET AZIMUTH TARGET ELEVATION TARGET IDENTIFIER (PASTE FROM N30)	XXX.XX DEG XX.XXX DEG 0000X
42	APO ALT PER ALT DELTA V (REQUIRED)	XXXX.X NM XXXX.X NM XXXX.X FPS
43	LATITUDE (+ NORTH) LONGITUDE (+ EAST) ALTITUDE	XXX.XX DEG XXX.XX DEG XXXX.X NM
44	APO ALT PER ALT TFF	XXXX.X NM XXXX.X NM XXBXX M-S
45	MARKS (VHF-OPTICS) TFI (NEXT BURN) MGA	XXBXX XXBXX M-S XXX.XX DEG
46	DAP CONFIG	OCT OCT
47	THIS VEHICLE WEIGHT OTHER VEHICLE WEIGHT	XXXXX. LBS XXXXX. LBS
48	GIMBAL PITCH TRIM GIMBAL YAW TRIM	XXX.XX DEG XXX.XX DEG
49	DELTA R DELTA V SOURCE CODE	XXXX.X NM XXXX.X FPS 0000X.
50	SPLERROR PERIGEE TFF	XXXX.X NM XXXX.X NM XXBXX M-S
51	RHO GAMMA	XXX.XX DEG XXX.XX DEG
52	CENTRAL ANGLE OF ACTIVE VEHICLE	XXX.XX DEG
53	RANGE RANGE RATE PHI	XXX.XX NM XXXX.X FPS XXX.XX DEG
54	RANGE RANGE RATE THETA	XXX.XX NM XXXX.X FPS XXX.XX DEG
55	PER CODE ELEVATION ANGLE (E) CENTRAL ANGLE OF PASSIVE VEHICLE	XXXXX. XXX.XX DEG XXX.XX DEG
56	REENTRY ANGLE DELTA V	XXX.XX DEG XXXXX. FPS
57	DELTA R (SOI) (+ PASS VEH LEADS)	XXXX.X NM

Figure 2.1-6 (cont.) Nouns Used in Program COLOSSUS

58	PER. ALT (POST TPI OR SOR)	XXXX.X NM
	DELTA V (TPI OR SOR)	XXXX.X FPS
	DELTA V (TPF OR SOR FINAL)	XXXX.X FPS
59	DELTA V LOS 1	XXXX.X FPS
	DELTA V LOS 2	XXXX.X FPS
	DELTA V LOS 3	XXXX.X FPS
60	G MAX	XXX.XX G
	V PRED	XXXXX. FPS
	GAMMA EI (+ UP)	XXX.XX DEG
61	IMPACT LATITUDE	XXX.XX DEG
	IMPACT LONGITUDE	XXX.XX DEG
	HEADS UP/DOWN (+ UP)	[illegible]
62	VI ,INERTIAL VEL MAG	XXXXX. FPS
	HDOT ,ALT RATE	XXXXX. FPS
	H ,ALT ABOVE PAD RADIUS OF LANDING SITE	XXXX.X NM
63	RTGO ,RNG FROM E.I. TO SPLASH	XXXX.X NM
	VIO ,PREDICTED INERT VEL	XXXXX. FPS
	TFE ,TIME FROM FROM E.I.	XXBXX M-S
64	G, DRAG ACCELERATION	XXX.XX G
	VI ,INERTIAL VELOCITY	XXXXX. FPS
	R TO GO (+ OVSHT)	XXXX.X NM
65	SAMPLED CMC TIME (FETCHED IN INTERRUPT)	00XXX. HRS
		000XX. MIN
		0XX.XX SEC
66	BETA, CMD BANK ANGLE	XXX.XX DEG
	CROSS RANGE ERROR (+ TGT RT)	XXXX.X NM
	DOWN RANGE ERROR (+ OVSHT)	XXXX.X NM
67	R TO GO (+ OVSHT)	XXXX.X NM
	LAT ,PRESENT POSITION (+ NORTH)	XXX.XX DEG
	LONG ,PRESENT POSITION (+ EAST)	XXX.XX DEG
68	BETA, CMD BANK ANGLE	XXX.XX DEG
	VI ,INERTIAL VELOCITY	XXXXX. FPS
	HDOT ,ALT RATE	XXXXX. FPS
69	BETA	XXX.XX DEG
	DL	XXX.XX G
	VL	XXXXX. FPS
70	CELESTIAL BODY CODE (BEFORE MARK)	OCT
	LANDMARK DATA	OCT
	HORIZON DATA	OCT
71	CELESTIAL BODY CODE (AFTER MARK)	OCT
	LANDMARK DATA	OCT
	HORIZON DATA	OCT
72	DELT ANG (+ ACT VEH LEADS)	XXX.XX DEG
	DELT ALT (+ PASS VEH ABOVE)	XXXX.X NM
	SEARCH OPTION	XXXXX.
73	ALTITUDE	XXXXXB. NM
	VELOCITY	XXXXX. FPS
	FLIGHT PATH ANGLE	XXX.XX DEG
74	BETA, CMD BANK ANGLE	XXX.XX DEG
	VI , INERTIAL VELOCITY	XXXXX. FPS
	G , DRAG ACCELERATION	XXX.XX G
75	DELTA ALTITUDE CDH	XXXX.X NM
	DELTA TIME (CDH-CSI OR TPI-CDH)	XXBXX MIN/SEC
	DELTA TIME (TPI-CDH OR TPI-NOMTPI)	XXBXX MIN/SEC
76	SPARE	
77	SPARE	
78	SPARE	
79	RATE (+ INCREASING CDU)	X.XXXX DEG/SEC
	DEADBAND	XXX.XX DEG
	AXIS CODE	XXXXX.
80	TFI/TFC	XXBXX M-S
	VG	XXXXX. FPS
	DELTA V (ACCUMULATED)	XXXXX. FPS
81	DELTA VX (LV)	XXXX.X FPS
	DELTA VY (LV)	XXXX.X FPS
	DELTA VZ (LV)	XXXX.X FPS
82	DELTA VX (LV)	XXXX.X FPS
	DELTA VY (LV)	XXXX.X FPS
	DELTA VZ (LV)	XXXX.X FPS
83	DELTA VX(CONT)	XXXX.X FPS
	DELTA VY(CONT)	XXXX.X FPS
	DELTA VZ(CONT)	XXXX.X FPS
84	DELTA VX(O VEH)	XXXX.X FPS
	DELTA VY(O VEH)	XXXX.X FPS
	DELTA VZ(O VEH)	XXXX.X FPS

Figure 2.1-6 (cont.) Nouns Used in Program COLOSSUS

85	VGX (CONT)	XXXX.X FPS
	VGY (CONT)	XXXX.X FPS
	VGZ (CONT)	XXXX.X FPS
86	DELTA VX (LV)	XXXXX. FPS
	DELTA VY (LV)	XXXXX. FPS
	DELTA VZ (LV)	XXXXX. FPS
87	MARK DATA: OPTICS SHAFT ANGLE	XXX.XX DEG
	OPTICS TRUNNION ANGLE	XX.XXX DEG
88	PLANET UNIT POSITION VECTOR X	.XXXXX
	Y	.XXXXX
	Z	.XXXXX
89	LANDMARK LATITUDE (+ NORTH)	XX.XXX DEG
	LANDMARK LONGITUDE/2 (+ EAST)	XX.XXX DEG
	LANDMARK ALTITUDE	XXX.XX NM
90	REND OUT OF PLANE PARAMETERS Y	XXX.XX NM
	Y DOT	XXXX.X FPS
	PSI	XXX.XX DEG
91	PRESENT OCDU ANGLES - SHAFT	XXX.XX DEG
	- TRUN	XX.XXX DEG
92	NEW OCDU ANGLES- SHAFT	XXX.XX DEG
	TRUN	XX.XXX DEG
93	DELTA GYRO ANGLES	X XX.XXX DEG
		Y XX.XXX DEG
		Z XX.XXX DEG
94	ALTERNATE LOS - SHAFT	XXX.XX DEG
	TRUN	XX.XXX DEG
95	PREF ATT FDAI ANGLES	R XXX.XX DEG
		P XXX.XX DEG
		Y XXX.XX DEG
96	+X AXIS ATT FDAI ANGLES	R XXX.XX DEG
		P XXX.XX DEG
		Y XXX.XX DEG
97	SYSTEM TEST INPUTS	XXXXX.
		XXXXX.
		XXXXX.
98	SYSTEM TEST RESULTS	XXXXX.
	AND INPUTS	.XXXXX
		XXXXX.
99	RMS VALUE OF POSITION ERROR	XXXXX. FT
	RMS VALUE OF VELOCITY ERROR	XXXX.X FPS
	OPTION CODE	XXXXX.

Figure 2.1-6 (cont.) Nouns Used in Program COLOSSUS

has control over the displays he wishes to observe, without being interrupted by an internal request. As will be shown in a discussion which follows on multi-level displays, the KEY REL button can also be used to reestablish displays which have been temporarily suspended.

While the astronaut communicates with the computer by entering information in the DSKY, the computer communicates with the astronaut by a flashing or nonflashing verb/noun display. The loading of data registers provides an example of two-way communication. To load three registers of data, the astronaut selects VERB 25 NOUN XX ENTR, where NOUN XX describes the data involved. He then depresses the ENTR button and the computer responds by flashing VERB 21, telling him to load register R1, which has been blanked. After the astronaut keys in the initial data, he keys ENTR. The computer responds with a flashing VERB 22, indicating that it is ready to accept data in the second register. The process is then repeated for the third register. Since PINBALL is able to distinguish between two modes of the ENTR button (execute verb/noun or enter data), data are not processed until the final component is loaded and the ENTR button is depressed. At this time, the data entered are scaled for each component and stored in the proper location in memory.

When a sign button is depressed before data are entered into each register, numeric information is treated as decimal; otherwise, PINBALL considers the data to be octal. If the operator depresses the 8 or 9 button on the DSKY while loading octal data, the OPR ERR (operator error) light is illuminated, which he can turn off by depressing the RSET button.

PINBALL was first developed to exercise systems-test and operations programs in an early version of the AGC. At that time only one level of priority was provided. Consequently, two internal jobs requiring displays could not run simultaneously. (This was satisfactory then and even for later unmanned flights during which the Boost Monitor Display—a constantly updated sequence of trajectory parameters—was continuously displayed on the DSKY.) But procedures like rendezvous-radar navigation marktaking could not run in the background behind a targeting computation and communicate updated data through the normal display activity in the foreground. With the advent of manned flights, it became clear that the computer would have to communicate with the astronaut on several levels; consequently, development of display-interface software and a hierarchy of priority interrupts was begun. Boost Monitor Display programs in SUNSPOT were the initial components of the complete G&N astronaut/AGC interface software that was further developed in SUNDISK and ultimately refined for COLOSSUS and LUMINARY.

The initial display-interface routine, GOFLASH, was created to save coding for the four or five calls to PINBALL by the Boost Monitor programs. The subroutine approach saved 12 instructions of the 18 otherwise required each time the AGC initiated an information transfer through PINBALL to the DSKY. In a recent COLOSSUS program, there are 45 calls to GOFLASH, which accomplishes a net saving of 540 instructions.

A second level of displays which was added carried a higher priority than normal program displays. These so-called Extended-Verb displays permitted an information request to be keyed in—even though another normal-priority program might be in progress—and to attract the crew's attention via a flashing display, effectively preempting the normal program's DSKY activity. An Extended Verb usually takes the form of an information request which differs from a regular verb in that it cannot be satisfied by simply displaying already available information stored in an erasable-memory location. An Extended Verb requires some data manipulation and ordinarily involves one or more subroutine calls. While the Extended Verb is running, the normal display is held in abeyance. Since sufficient information has to be saved to restore an interrupted display after the interrupt, display points became natural restart points. And because displays are usually natural breakpoints in an extended computation, they provide excellent demarcation points for program phase changes. A special restart mechanism therefore was created to permit "restarts" to pick up at the most recent display. A more comprehensive description of restarts follows in Section 2.1.4.

At about the time the need for Extended-Verb displays was recognized, a similar requirement was recognized for mark displays. During rendezvous, the astronaut is very busy with three four-part operational cycles (navigation, targeting, maneuver, and burn) in succession to be accomplished during brief spans of time. It therefore became virtually mandatory that the Range Radar (LM) and VHF (CM) navigation marktaking be performed automatically without astronaut supervision, but with provision for astronaut intervention if anomalous mark data were obtained. The same priority-interrupt technique implemented in the Extended Verb feature was also implemented to permit navigation marks to be taken while a targeting routine was in progress, and—when they satisfied certain threshold-acceptance criteria—to be incorporated automatically. Only marks that violate accept/reject criteria need be presented for the astronaut's consideration explicitly via the display-interrupt software interface. Since Extended Verbs and marking-program displays shared the same priority level, a restriction was necessarily imposed that no Extended Verb using displays could be imposed during marktaking.

A second higher level of priority-interrupt displays was required both to display anomalous mark data which exceeded the threshold for acceptance and to permit alarm-type displays to override the first two levels. Since targeting programs or Extended Verbs run during the rendezvous programs, a third priority level was needed for alarm conditions and for marks that exceeded the auto-accept threshold. The three-level display hierarchy thus consists of normal displays, which are the lowest level and can be overriden by Extended Verb or mark displays, and third-level priority displays (alarm conditions, excessive updates) which can interrupt displays in both of the lower priority levels.

In addition to the three internally-generated priority-display levels described, the astronaut can key in two higher levels called external monitor request and non-monitor request. Altogether, five levels of display information are provided. After keying in a non-monitor request over an external-monitor request which in turn has overridden the three levels of internal priority display, an astronaut can return to the fourth external-monitor level from the fifth non-monitor level by keying KEY REL, and from external monitor to the third (priority) level via another KEY REL. He can then respond to the priority display and obtain the second and normal display levels, in turn, by keying appropriate responses to each succeeding display level. Thus, while monitoring a program computation and simultaneously taking navigation marks, the astronaut may be notified of an emergency-alarm condition by a priority display and may then initiate two levels of monitor-interrupt displays to discover the cause of the alarm condition before taking appropriate action.

The most significant effect of the additional display routines was that it became possible to have three levels of programs—with displays—running simultaneously. Response by the astronaut to any of the higher level displays would automatically cause the next lower-level display to reappear. This feature gives the astronaut the flexibility of using five levels of displays at a time.

2.1.3.4 Uplink and Downlink

Uplink is the digital telemetry system which enables ground control to load data or issue instructions to the AGC in the same manner employed by the astronaut using the DSKY keyboard. All information received by the AGC via uplink is in the form of keyboard characters. Each character is assigned an identifying code number called its character code. The AGC picks up the transmitted codes (these codes are the same as key codes) and enters a request to the Executive for the program which decodes and accepts them. The PINBALL program which decodes and accepts the transmitted code makes no distinction between inputs from the keyboard or from uplink, and any ground-command sequence normally transmitted via uplink may be duplicated by the astronaut using the keyboard.

The astronaut can choose to reject uplink from ground control by setting a toggle switch on the cockpit control panel to the blocked position.

A Universal Update Program exists in the AGC which facilitates updating the erasable memory and can be called by a number of extended verbs. To protect against the ingestion of erroneous information, the Update Program temporarily stores all new inputs in a buffer and transmits its contents back to ground control via downlink (see below) for verification. Furthermore, storage of state-vector updates (position and velocity) with their associated sphere-of-influence (earth or lunar) are delayed until current state-vector integration is finished.

The Update Program accepts four types of erasable-memory updates:

1. Contiguous Block Update provides ground-control capability to update up to 18 consecutive erasable-memory registers in the same erasable-memory bank.
2. Scatter Update provides ground-control capability to update from 1 to 9 nonconsecutive erasable-memory registers in the same or different erasable banks.
3. Octal-Clock Increment provides ground-control capability to increment or decrement the AGC clock with a double-precision octal-time value.

4. Liftoff-Time Increment provides ground-control capability to increment or decrement the AGC time, LM and CSM state-vector times and ephemeris time with a double-precision octal-time value.

This Universal Update Program capability has been available since SUNDISK (Apollo 7).

Downlink is the digital telemetry system which automatically selects lists (downlists) of internal AGC data for transmission to the ground downlink. Each downlist contains data pertinent to specific mission phases. COLOSSUS has five standard downlists: Powered, Coast and Align, Rendezvous and Prethrust, Entry and Update, and P22 (Orbital Navigation Program). LUMINARY has six standard downlists: Orbital Maneuvers, Coast and Align, Rendezvous and Prethrust, Descent and Ascent, Lunar Surface Align, and Initialization and Update of the Abort Guidance System (AGS). Whenever a new program is entered, a request for its list is made by placing the appropriate code into a downlink register. The downlink program then transmits the complement of this code as an identifier and uses it to select the appropriate list. The complete list is transmitted even if the program is changed during its transmission.

The standard AGC downlist contains 100 words (200 AGC registers). The AGC digital downlink is transmitted at a high rate of 50 words/sec or at a low rate of 10 words/sec. Thus, transmission of one downlist requires two sec at the high rate and ten sec at the low rate.

Certain data on the standard downlists are meaningful only when considered in multiregister arrays. Since the programs which compute these arrays are not synchronized with the downlink program, a "snapshot" is taken of these words so that changes in their values will not occur while these arrays are being transmitted to the ground. When a "snapshot" is taken, several words are stored at the time the first word is transmitted. The other words in the downlist are read at the time of transmission.

There is a special mode of downlink, called Erasable-Memory Dump, which can preempt the standard downlist being transmitted. The transmission consists of all of the erasable banks being transmitted sequentially. One complete pass through erasable requires 20.8 sec. The computer makes two passes through the complete erasable memory before returning to the standard downlist for the current mission phase. Since normal processing continues during the transmission of the Erasable-Memory Dump, some of the registers transmitted could have different contents on the second pass because they may have been recalculated during the transmission time.

This erasable-dump capability can be initiated using an Extended Verb and was developed to support postflight analysis; it can, however, be used whenever information not on a standard downlist is desired.

2.1.4 Error-Detection and Self-Check Features

Considerable effort has been expended over the years to uncover and correct for a number of hardware- or software-initiated problems. These problems can vary from a hardware power failure to the software getting caught in a loop. Both the hardware and software are designed to catch these problems, and the software procedures used to reinitialize (restart) the computer have become relatively standard.

The function of the hardware- and software-restart logic is to restore the current program with a minimum of disturbance to the mission. Fundamentally, this requires that certain specified tasks be called at the end of the correct time intervals (from a suitable base time), and that the specified jobs be reestablished with the proper priorities. In some cases, the proper "restarting" addresses for the jobs and/or tasks should not be at their beginning, but instead at some intermediate location or even at a special location entered only if a restart is encountered. These locations (restart points) are chosen to fall between computations such that when a restart occurs, the program resumes at a point in the program which precedes the place where the problem arose.

To accomplish the required restart functions, the various activities performed by the program software, in essentially independent computations, are divided into "restart groups"; there is provision in the restart software for six groups. One group, for example, might be concerned with the periodic powered-flight navigation cycling; another with orbital integration (perhaps required with powered flight to generate relative CSM/LM display data); a third with the timing of events leading to engine ignition; a fourth with generation of a time display on the DSKY; a fifth with computation of required velocity information for a rendezvous maneuver; and a sixth with a special computation performed shortly before engine ignition (to estimate the length of the burn). All six of these functions could be part of the complete program's computational load (as jobs or tasks) at one time and be in various stages of completion; and, consequently, they could be associated with separate restart groups. Not all computational activity in the program is restart-protected in this fashion; for example, should a restart occur while data are being loaded via the DSKY, the loading sequence must be reinitiated.

A restart group, therefore, can generally be considered to be associated with a particular functional software activity. Each group, in turn, is conventionally divided into a number of "phases" indicating just where the computations should be reinitiated in the event of a restart. The phase information for a given group is retained in both true and complemented form in the erasable memory, giving a total of 12 cells for the six pairs of cells associated with the six restart groups. When the restart software is entered, a check is made to ensure that all six pairs of cells have the proper internal complement relationship. If not, it is concluded that suspect information prevents the satisfactory resumption of computations, and the attempt to perform the restart is abandoned in favor of a FRESH START. FRESH START, which reinitializes the complete guidance system and essentially leaves it in an "idling" configuration, is discussed in Section 2.1.4.2.) The complement relationship could be destroyed if the erasable memory were modified by whatever caused the restart action, such as a power transient, or should the restart occur during certain portions of the programs that change restart-phase information.

Should the restart software conclude that adequate phase information is available (on the evidence of a proper complement relationship for the six pairs of phase data), the RESTART routine can be entered for each restart group that is "active" (a group is made "inactive" by setting the phase of that group to +0, indicating that none of its computations are restart-protected). The RESTART routine, depending on the value of the phase associated with that group, can cause jobs to be established and/or Waitlist tasks to be called at appropriate times via LONGCALL or the normal waitlist routines. The value of the phase information also determines whether one or two such jobs and/or tasks are to be reinitiated, and, additionally, whether the parameters associated with the reinitiation are to be obtained from fixed or erasable memory.

The value of the phase for a particular restart group, properly interpreted, is used to select an appropriate table entry in fixed and/or erasable memory. The table entries, separated by groups, are stored so that memory capacity is not wasted should there be more fixed-memory tables of one type than the other. The polarity with which information is stored in the tables is used to determine whether the table information pertains to a job, a Waitlist task, or a LONGCALL task, and, additionally, to determine which of several available options for defining the reinitiation parameters is to be employed.

During the course of the computations, it is necessary to update the phase value associated with the appropriate group. This can be done directly by loading new phase information into the appropriate group's phase cells or through use of one of several available phase-changing subroutines. The three most commonly used phase-changing subroutines are NEWPHASE, PHASCHNG and 2 PHSCHNG, all of which have a variety of options, depending upon the details of the calling sequence. Each one of these subroutines identifies the nature of the restart desired—fixed-memory table only, fixed and erasable tables, or erasable-memory table only.

The AGC restart mechanism provides great flexibility for restarting with optimal configuration of important computations, at almost no cost in erasable memory and little cost in execution time.

The significance of this restart protection can be appreciated more fully if one considers the consequences of the accidental knockdown of an unprotected engine-on bit during a burn. The following two sections describe remedies for hardware- and software-discovered difficulties and illustrate how self-check procedures contribute to the integrity of the mission program.

2.1.4.1 Hardware Restarts

One kind of program interrupt—a hardware restart—differs markedly from those described in Section 2.1.2.1. This special kind of program interrupt does not result in "normal" resumption of the program; it takes absolute priority over other program interrupts; it cannot be inhibited; and it can even interrupt an interrupt. As part of its generation, a special involuntary-interrupt instruction is produced, causing the hardware to generate a master-clear signal which knocks down all of the outbits.

A hardware restart can be triggered by such hardware problems as power failure, computer-oscillator failure, or parity failure. If the failure is transitory, the restart logic will resume the program flow.

A parity failure indicates possible malfunctions in a fixed or erasable register, in a sense line or in an amplifier. The AGC-stored word length consists of a sign bit, 14 magnitude bits of information and a parity bit. Whenever a register is addressed, odd parity must be observed or a hardware restart will occur. Should the parity error be detected in an erasable-memory register, it will be reinitialized and thus reset by the software-recovery logic. However, should a parity failure occur in a fixed-memory register, either a more serious physical problem exists or the astronaut has accidentally addressed an empty (unused) register.

Hardware restarts can occur upon the software-detection of a program-interrupt failure (RUPT LOCK) revealed if a program interrupt is continuously in effect for a specified period of time or if no program interrupt takes place within an equally long interval. Similarly, a transfer-control failure (TC TRAP) can be discovered. In addition, a special procedure called NIGHT WATCHMAN reveals the failure to address one specific memory location with a certain frequency, thus detecting the inadvertent entrapment in a large program loop.

Several lesser problems are indicated by warning lights and do not cause a restart: Counter Fail, which arises if counter increments occur too frequently or fail to occur following an increment request; PIPA Fail, which arises if no pulses arrive from the Pulsed Integrating Pendulous Accelerometers during a specified period, or if both positive and negative pulses occur simultaneously, or if too long a time were to elapse without at least one positive pulse and at least one negative pulse arriving; and Uplink Too Fast and Downlink Too Fast.

2.1.4.2 Software Restarts

Software restarts are programmed branches into the software-recovery logic. They use much of the same coding as the hardware restarts and, in fact, execute the actual restart in an identical fashion.

Software-restart logic is frequently useful to perform nonproblem functions such as stopping certain computations while allowing others to continue. It is also used when a new mission program is selected via V37. In this case, current processing is stopped; all scheduled jobs, tasks and interrupts are cleared out; all restart groups except the one used by the background-tracking program (if in progress) become inactive; the new program is set up in a restart group; and then the restart is executed to initiate the new program. Restart logic is used similarly in an abort from lunar descent, but in this case, the new program selected would be the abort program. Software restart procedures can also be initiated by such software-detected difficulties as too many tasks in the Waitlist system or a negative input to the square-root subroutine.

Two of the more important alarms which cause software restarts are BAILOUT and POODOO. A BAILOUT initiates a software restart for a problem from which recovery is expected, such as the overflow of job-register sets. A POODOO initiates a software restart for a problem from which a simple recovery is not expected, such as an attempt to take the square root of a negative number. Such a problem can happen if erroneous parameters have been loaded; consequently, a reinitialization of these same parameters will continuously yield the same alarm. In this case, normal computation flow is terminated and a flashing V37 (Change Program) comes up on the DSKY.

A FRESH START reinitializes the complete guidance system and essentially leaves it in an "idling" configuration with all of the output channels (outbits) and pending interrupts knocked down; at this point the program checks to see if the engine-on bit should be restored and if the IMU is in gimbal lock, and it takes whatever protective measures are necessary. FRESH START is the most radical reinitialization available for recovery.

A software program called BANKSUM Check, initiated by an Extended Verb to check all fixed and erasable memory for parity failures, is used principally for systems-test purposes. This routine sums the contents of the addresses within each fixed bank—halting temporarily when the last memory cell is reached. At this point a memory-cell summing routine included in the self-check portion of the fixed memory checks to ensure that the magnitude-of-the-sum is equal to the bank number and provides a DSKY display of the sum for operator review. The feat of having the magnitude-of-the-sum equal to the bank number is accomplished in the assembly process simply by adding an appropriate constant stored at the end of each bank to the correct value of each BANKSUM's magnitude.

As mentioned in Section 2.1.2.2, when no mission functions are being performed, an idling job (DUMMYJOB) is run to check for new jobs while checking fixed and/or erasable memory, depending on the option last selected by the astronaut.

Lastly, should the astronaut want to check the DSKY lamps, they can all be illuminated.

2.2 Major Mission Tasks Accomplished with the Computer Software

2.2.1 Early Approach to Navigation, Targeting, Guidance and Control

The navigation, targeting, guidance and control software specifies and manages the various spacecraft motions required to accomplish each mission phase. Functions of concern include the onboard measurement of rotational and translational motion, the processing of these measurements for display to the crew and ground control, the acceptance from the crew or ground control of desired spacecraft-maneuver instructions, and the execution of the defined maneuvers to change the spacecraft motion by modulating the firing of the various rocket-propulsion systems. In this context, navigation, targeting, guidance and control are defined as follows:

Navigation is the measurement and computation necessary to determine the present spacecraft position and velocity.

Targeting is the computation of the maneuver required to continue on to the next step in the mission.

Guidance is the continuous measurement and computation during accelerated flight to generate steering signals necessary to assure that the position and velocity changes of the maneuver will be those required by navigation measurements and targeting computations.

Control is the management of spacecraft-attitude motion—the rotation to and the stable maintenance of the desired spacecraft attitude during free-fall coasting flight and powered accelerated flight.

The appendices to this report present a functional description of these major program capabilities. Their design and development represent a significant portion of the Apollo software effort. The integration of these guidance, navigation and control programs with mission-oriented programs into a flight rope requires the comprehensive testing and verification effort described in Section III.

The early studies of the major program capabilities began, in most cases, well before the Apollo mission plan was finalized, since most of their concepts were fundamental to the overall task to be performed. For example, rendezvous procedures would be essential to both the earth-orbit and lunar-orbit rendezvous plans.

As a first step in MIT's software efforts, the basic organization of AGC computation and control had to be decided upon and implemented. PINBALL was developed to enable communication between the astronaut and the AGC. Guidance, navigation and control techniques had to be developed for every phase of the Apollo mission—from earth-orbit insertion to soft landing on the lunar surface to reentry into the earth's atmosphere. Similarly, abort procedures had to be developed for every phase. Studies determined the effect of the earth's luminous exponential atmosphere upon space navigation. Star- and horizon-sighting techniques had to be developed. Lunar-orbit determination using star-occultation measurements and the NASA Manned Space Flight Network were investigated. The effects of retrorocket exhaust velocity on visibility were ascertained. Development proceeded on a universal powered-flight guidance program tailored specifically to exploit the powers of an onboard digital computer. In addition, powered-flight steering of a spacecraft using a time-shared digital computer was studied, considering, of course, such factors as performance, response time and fuel conservation. And operating procedures had to be defined for the entire Apollo mission.

These are but a fraction of the many tasks which were studied and implemented before a mission-oriented rope could be integrated. These tasks continue. Flight experience frequently indicates the desirability of improvements or refinements. An example of such ongoing design work is the automation of the rendezvous sequence. Another is the restoration of the GN&C System Saturn-Take-over program as a backup system. With these exceptions and at this advanced date in the program, most changes are of a relatively minor nature.

The lunar-landing objective of the Apollo mission was finally achieved after many preliminary flights, each of which evolved from its predecessor (see Section 1.2). Each flight rope contained not only the programs necessary for the completion of its stated mission, but also many programs which were not of immediate application. In this fashion, existing flight ropes also served to bench-test programs which would be utilized in future flights. For example, the lunar-operations sequence was present in its entirety in SUNDANCE, the rope developed for a manned earth-orbital flight. But SUNDANCE provided the unique opportunity to exercise the lunar sequences in the comparatively safe earth-orbital environment. To prepare the actual lunar-landing sequence, however, those programs still had to be adapted to the conditions expected to prevail at the time of the lunar landing.

2.2.2 The G&N Mission Phases

For tractability the Apollo mission was divided into a number of discrete phases. Although each phase will be discussed somewhat independently, it is essential to note that all phases lead logically and efficiently from one to another in a stepwise fashion.

The lunar-landing mission, Apollo 11, contained all of the completed software programs. While many detailed variations can exist in future missions, the guidance, navigation and control functions remain essentially the same. A synopsis of a typical Apollo lunar-landing mission follows to aid in understanding the comprehensive task which the G&N software performs.

As stated above, the overall Apollo mission trajectory can be divided into several linked phases. Figure 2.2-1 illustrates thirteen such phases. The following paragraphs discuss each of these phases, along with lunar-surface operations.

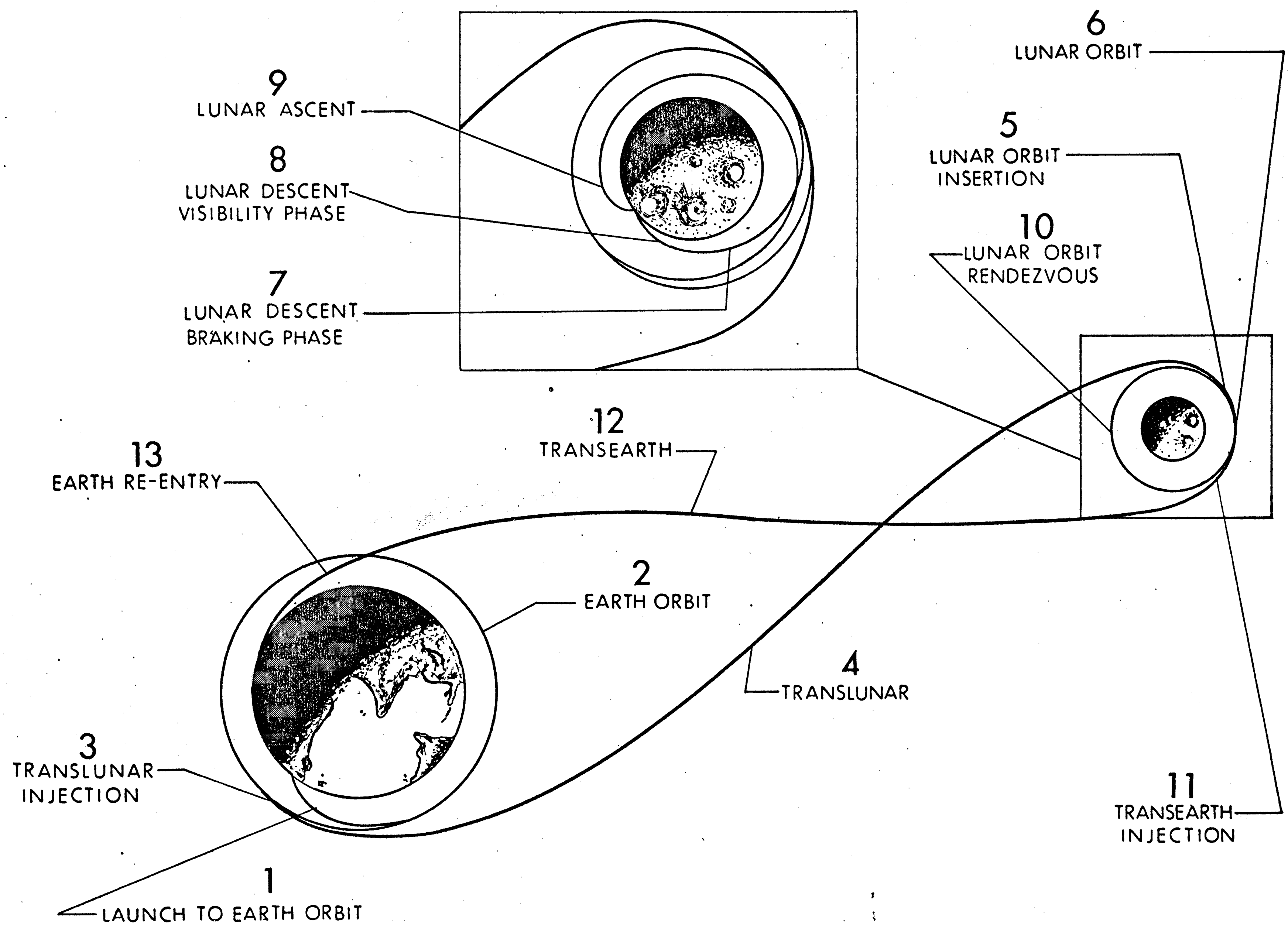


Figure 2.2-1 G&N Mission-Phase Summary

2.2.2.1 Launch to Earth Orbit

Prior to launch there is an intensive and intricate schedule of activity. Automatic programmed checkout equipment performs exhaustive tests of the major subassemblies in two major sequences: countdown demonstration and the actual countdown. Two operating sets of guidance equipment are prepared for the launch. The Saturn guidance equipment in the Saturn Instrument Unit controls the launch vehicle, while the Apollo guidance equipment in the Command Module provides a monitor of Saturn guidance during launch. The Lunar Module GN&C System, after prelaunch testing, is normally powered down for the launch phase of the mission.

Both sets of inertial guidance sensors, Saturn and Command Module, are aligned to a common vertical and launch-azimuth reference. During countdown, both systems are gyro-compassed to an earth-frame reference. Near liftoff, both systems respond to discrete signals to switch over from the earth reference to the nonrotating inertial reference used during boost.

During first-stage flight, the Saturn guidance system controls the vehicle by swiveling the outer four rocket engines. During the initial vertical flight, the vehicle is rolled from its launch azimuth to the flight-path azimuth. The Saturn guidance then controls the vehicle in an open-loop preprogrammed pitch maneuver designed to pass safely thorugh the period of high aerodynamic loading.

Both the Saturn and Command Module guidance systems continuously measure vehicle motion and compute position and velocity. In addition, the GN&C System compares the actual motion history with that expected from the Saturn control equation to generate an error display for the crew.

Shortly after the initial fuel-settling ullage and the second-stage thrust, the aerodynamic pressure approaches zero, the launch escape tower is jettisoned, and the vehicle passes out of the atmosphere. Any required abort, now, would normally be accomplished using the Service Module propulsion to accelerate the module away from the rest of the vehicle.

Since the problems of aerodynamic structure loading are no longer important, the Saturn guidance system now steers the vehicle toward the desired orbital-insertion conditions using propellant-optimizing guidance equations. Thrust-vector control is achieved by swiveling the outer four engines of the second stage.

During second-stage flight, the GN&C System continues to compute vehicle position and velocity, as well as several other flight parameters which can be displayed to the crew. The free-fall time to atmospheric entry, the apocenter altitude and pericenter altitude are the primary displays at this time.

The third Saturn stage (SIVB) has a single main propulsion engine gimballed for thrust-vector control. Roll control is achieved using the small SIVB roll attitude-control thrusters. The Saturn guidance system continues to steer the vehicle to orbital altitude and speed. When orbit is achieved, the main SIVB propulsion is shut down; this usually occurs at about 12 minutes after liftoff on a 100-mile circular orbit.

During the second- and third-stage boost flight, the Command Module is configured to allow the crew to take over the SIVB steering function manually, should the Saturn guidance system indicate failure. Should this switchover occur, presumably the mission could be continued. More drastic failures would require an abort using the Service Module propulsion system.

2.2.2.2 Earth Orbit

The Apollo spacecraft remains attached to the Saturn SIVB in earth orbit. The Saturn system controls attitude by commands to the small SIVB reaction-control thrusters for pitch, yaw and roll.

Ground-tracking navigation data telemetered from the Manned Space Flight Network (MSFN) stations are available to correct the position and velocity of the Saturn navigation system and to provide navigation data for the GN&C System via uplink telemetry. The inertial-subsystem alignment in the Command Module may also be updated by star sightings with the optical subsystem. For these measurements, the crew exercises manual control of vehicle attitude through the Saturn attitude-control system.

Typically, the earth-orbital phase lasts less than three hours for systems checkout before the MSFN-computed signals are transmitted to the Saturn system to initiate the translunar-injection maneuver.

2.2.2.3 Translunar Injection

Translunar injection is performed using a second burn of the Saturn SIVB propulsion system. Saturn guidance and control systems again provide the necessary steering and thrust-vector control to the near-parabolic velocity that puts the vehicle on a so-called "free return" trajectory to the moon. This trajectory is constrained ideally to pass in back of the moon and to return to earth-entry conditions without additional propulsion.

As before, the GN&C System independently generates appropriate parameters for display to the crew for monitoring purposes. Should the Saturn guidance system indicate failure, steering takeover by the crew is possible. The typical translunar-injection thrusting maneuver continues for slightly over five minutes' duration before the SIVB is commanded its final shutdown.

2.2.2.4 Translunar

The spacecraft configuration injected into the translunar free-fall must be reassembled for the remaining operations. An adapter in front of the SIVB houses the LM until this phase of the flight. The astronauts separate the Command and Service Modules from the SIVB and then turn the CSM around for docking with the Lunar Module. To accomplish this, the pilot has a three-axis left-hand translation controller and a three-axis right-hand rotational controller. Output signals from these controllers are processed in the Command Module computer to modulate the firing of the 16 low-thrust reaction-control jets for the maneuver. The normal response from the translation controller is proportional vehicle acceleration in the indicated direction. The normal response from the rotational controller is proportional vehicle angular velocity about the indicated axis.

During the separation and turnaround maneuver, the SIVB control system holds the Lunar Module attitude stationary; this allows for a simple docking maneuver of the Command Module to the Lunar Module docking hatch. The SIVB, Saturn Instrument

Unit, and Lunar Module adapter are staged to leave the Apollo spacecraft in the translunar flight configuration. A further short maneuver puts the SIVB on a separate trajectory which will not interface with the Apollo spacecraft.

Very soon after injection into the translunar free-fall coast phase, MSFN-computed navigation measurements are examined to determine the acceptability of the trajectory. These data indicate whether there is a need for an early midcourse maneuver to correct any error in the flight path which might propagate with time to a larger value, thus avoiding a needless waste of correction-maneuver fuel. This first correction is made—perhaps a few hours from injection—only if it is needed. Ground-tracking data can be telemetered to the spacecraft anytime they are available. Using these ground data or horizon-to-star angle measurements obtained from the onboard sextant, the onboard computer can correct the knowledge of the spacecraft state vector—position and velocity.

Mission control on the ground periodically examines the ground-based radar data for uncertainty in position and velocity and the estimate of indicated velocity correction required to improve the present trajectory. If the indicated position and velocity uncertainties are suitably small and the indicated correction is large enough to be worth the effort, the crew may execute the telemetered midcourse correction. Each midcourse velocity correction requires, first, an initial spacecraft orientation which aligns the estimated direction of the thrust axis along the desired acceleration direction. Once the thrust direction is aligned, the rocket is ignited and controlled by the GN&C System.

Typical midcourse corrections are of the order of 30 ft/sec or less. If a required correction happens to be very small, it is made with the small reaction-control thrusters. Larger corrections require a short burn of the service-propulsion rocket. The direction and magnitude of each burn adjust the trajectory so that the moon is finally approached near the plane and pericynthion altitude that provide for satisfactory conditions for the lunar-orbit insertion and lunar landing.

During the translunar phase, mission control periodically transmits blocks of data via voicelink to the crew to permit safe return in the event of loss of communications. These data include state-vector updates to be loaded by the crew at the appropriate time into the AGC. The data are sent as a precaution against

Unit, and Lunar Module adapter are staged to leave the Apollo spacecraft in the translunar flight configuration. A further short maneuver puts the SIVB on a separate trajectory which will not interface with the Apollo spacecraft.

Very soon after injection into the translunar free-fall coast phase, MSFN-computed navigation measurements are examined to determine the acceptability of the trajectory. These data indicate whether there is a need for an early midcourse maneuver to correct any error in the flight path which might propagate with time to a larger value, thus avoiding a needless waste of correction-maneuver fuel. This first correction is made—perhaps a few hours from injection—only if it is needed. Ground-tracking data can be telemetered to the spacecraft anytime they are available. Using these ground data or horizon-to-star angle measurements obtained from the onboard sextant, the onboard computer can correct the knowledge of the spacecraft state vector—position and velocity.

Mission control on the ground periodically examines the ground-based radar data for uncertainty in position and velocity and the estimate of indicated velocity correction required to improve the present trajectory. If the indicated position and velocity uncertainties are suitably small and the indicated correction is large enough to be worth the effort, the crew may execute the telemetered midcourse correction. Each midcourse velocity correction requires, first, an initial spacecraft orientation which aligns the estimated direction of the thrust axis along the desired acceleration direction. Once the thrust direction is aligned, the rocket is ignited and controlled by the GN&C System.

Typical midcourse corrections are of the order of 30 ft/sec or less. If a required correction happens to be very small, it is made with the small reaction-control thrusters. Larger corrections require a short burn of the service-propulsion rocket. The direction and magnitude of each burn adjust the trajectory so that the moon is finally approached near the plane and pericynthion altitude that provide for satisfactory conditions for the lunar-orbit insertion and lunar landing.

During the translunar phase, mission control periodically transmits blocks of data via voicelink to the crew to permit safe return in the event of loss of communications. These data include state-vector updates to be loaded by the crew at the appropriate time into the AGC. The data are sent as a precaution against the contingency that telemetry and/or voice communication fail prior to the next scheduled update. These updates occur at about ten-hour intervals.

2.2.2.5 Lunar-Orbit Insertion

Prior to lunar-orbit insertion maneuvers, as with all normal thrusting with the Service Propulsion System, the inertial subsystem is realigned using star sightings. Then the GN&C System generates initial conditions and steering parameters based upon targeting parameters telemetered from the ground. The guidance programs initiate engine turn-on, control the direction of the acceleration appropriately, and shut the engine down when the maneuver is complete. Lunar-orbit insertion maneuvers are the two burns typically intended to put the spacecraft in an orbit of approximately 60 nmi altitude. The first thrusting maneuver, behind the moon, slows the spacecraft so that it will be "captured" by lunar gravity into a highly elliptical orbit and not pass on free-return to earth. Then, the second burn, at perilune behind the moon, circularizes the orbit. The plane of the orbit is selected to pass over the preplanned landing region.

2.2.2.6 Lunar Orbit

In lunar orbit, navigation measurements may be made to update the knowledge of the actual orbital motions. A particularly important sighting—that to the intended landing target—provides data for the site's precise location in the lunar navigation coordinate frame. Sufficient measurements must be made and combined with ground-tracking data to provide accurate initial conditions to the Lunar Module guidance system for the LM's controlled descent to the lunar surface.

2.2.2.7 Lunar Descent

During lunar orbits, before separation, the Lunar Module GN&C System is turned on and receives a checkout and its initial conditions, and the rendezvous radar (RR) is self-tested. Before initiation of the Lunar Module descent-injection maneuver, the vehicles are separated; the Lunar Module inertial subsystem receives final realignment from star sightings; the directional tracking and ranging operation of the RR is checked against the radar transponder on the CM; and the maneuver attitude is assumed. The maneuver is made using Lunar Module descent-stage propulsion under control of the module's GN&C System.

During free-fall phases of the Lunar Module descent, the Command Module can make optical tracking and VHF range-only measurements of the Lunar Module for confirmation of its relative orbit. For that part of the trajectory in front of the moon, earth-based tracking provides an independent check. The RR continues to track the CM transponder throughout free-fall for additional trajectory corroboration. At lower altitudes, the Lunar Module landing radar on the descent stage is self-tested prior to powered descent-insertion. Alignment updating of the Lunar Module inertial subsystem is also performed.

2.2.2.7.1 Braking Phase

Powered-descent braking begins when the descent engine is reignited; the velocity- and altitude-reducing maneuver is controlled via the Lunar Module inertial subsystem and autopilot calculations in the computer.

The descent-stage engine can be throttled over the range necessary to provide initial braking and to provide controlled hover above the lunar surface. Engine-throttle setting is automatically commanded by the guidance system to achieve proper path control, although the crew can override this signal with several alternative control modes, if desired.

Thrust-vector control of the descent stage is achieved by a combination of body-fixed reaction jets and limited gimballing of the engine. The engine gimbal angles follow guidance commands in a slow loop (fixed rate command of approximately 0.2 deg/sec), thus causing the thrust direction to pass through the vehicle center of gravity—and minimizing the need for continuous fuel-wasting torques from the reaction jets.

During all phases of the descent, the operations of the various systems are monitored from onboard and earth-based radar. The landing can be retargeted by uplink telemetry or the mission could be aborted for a number of reasons. If the GN&C System performing the descent control is still operating satisfactorily, it would control the abort back to rendezvous with the Command Module. If the primary guidance system has failed, the independent backup Abort Guidance System could steer the vehicle back to orbital conditions for rendezvous.

2.2.2.7.2 Visibility Phase

One significant feature of this phase is that the controlled trajectory is selected to provide the Lunar Module crew with visibility of the landing surface. The vehicle attitude, descent rate, and direction of flight are all essentially constant, so the landing point being controlled by the guidance appears fixed, relative to the window. A simple reticle pattern in the window indicates this landing point in line with a number denoted by computer display. Should the astronaut observe that the landing point is in an area of unsatisfactory surface features relative to other areas nearby, he can select a new landing site for the computer-controlled landing. Alternately, the astronaut has the option of taking manual control of this landing maneuver at any time.

Automatic guidance control during the braking and visibility phases uses weighted combinations of inertial-sensing and landing-radar data, with the weighting dependent upon expected uncertainties in the measurements. The landing radar includes altitude measurement and a three-beam Doppler measurement of three components of Lunar Module velocity with respect to the lunar surface.

At any point in the landing, the astronaut can elect to assume partial or complete control of the vehicle. For instance, one logical mixed mode of operation would have the rate-of-descent controlled automatically by modulation of the thrust magnitude and astronaut manual control of attitude for horizontal maneuvering.

Near the lunar surface, the spacecraft enters a hover phase which may have a variety of conditions, depending upon mission ground rules, crew option and computer program. Descent-stage fuel allowance provides for hovering before touchdown. If hovering is not accomplished, an abort is initiated on the ascent stage. The crew makes final selection of the landing point and maneuvers to it either by tilting the vehicle or by operating the reaction jets for translation acceleration. The inertial-subsystem altitude and velocity computation is updated by the landing radar so that, as touchdown is approached, good data are available from the inertial sensors, since the flying dust and debris caused by the rocket exhaust degrade radar and visual information. Touchdown is made with the spacecraft near vertical and with a downward velocity of less than 4 ft/sec.

2.2.2.8 Lunar-Surface Operations

The period on the moon includes considerable activity in exploration, equipment deployment, experimentation, and sample gatherings. Also during this time, spacecraft systems are checked and prepared for the return. For example, the ephemeris of the Command Module in orbit is periodically updated, and the information is relayed to the Lunar Module crew and computer. The Lunar Module rendezvous radar can also track the Command Module as it passes overhead to provide further data upon which to base the ascent-guidance maneuvers. The inertial subsystem receives final alignment from optical star or planet sightings prior to the start of ascent or, as a backup, the vertical components of this alignment can be achieved by accelerometer sensing of lunar gravity in a vertical-erection loop. Still another backup mode involves using computer-stored knowledge of the spacecraft's inertial alignment at touchdown. Liftoff must be timed to achieve the desired trajectory for rendezvous with the Command Module.

2.2.2.9 Lunar Ascent

Launches from the lunar surface leave the descent stage of the Lunar Module behind, and can be initiated over a range of time by entering a holding orbit at low altitude until the phasing is proper for transfer to the Command Module. A desirable constraint on all ascent-powered maneuvers, as well as abort maneuvers during the landing, is that the following coasting trajectory have sufficient altitude to avoid intersection with the lunar surface. This is a safety consideration which allows for the possibility of failure of the engine to reignite. If the Lunar Module engine thus fails, the spacecraft could then safely coast until a rescue maneuver by the Command Module is accomplished. That is, the Command Module could execute "mirror images" of those thrusting maneuvers that the Lunar Module would have normally performed. Thus, the Lunar Module can be the passive vehicle in the rendezvous exercise.

The initial part of the ascent trajectory is a vertical rise followed by pitchover, as commanded by the guidance equations. The ascent-engine maneuvers are under the control of the GN&C System. The ascent engine is fixed-mounted and nonthrottleable; consequently, thrust-vector control is achieved by complementing the engine thrust with that of the 16 reaction-control jets mounted on the ascent stage. Required commands from guidance terminate thrusting when a suitable rendezvous coast trajectory is achieved.

2.2.2.10 Lunar-Orbit Rendezvous

This phase starts from the low holding orbit achieved by the ascent burn of the previous phase. From this orbit, the RR makes direction and range measurements to the Command Module for refinement of the navigation data in the Lunar Module computer. The phasing of motion between the two vehicles eventually reaches a specific point at which a standard transfer burn puts the Lunar Module on an ascending trajectory to intercept the orbiting Command Module. During this period, radar measurements provide data for the Lunar Module computer's small velocity corrections needed to establish a more accurate intercept trajectory. Coasting continues between and during these corrections until the range to the Command Module is reduced to a few miles.

A series of braking maneuvers under control of the Lunar Module GN&C System and the astronaut is required during the terminal rendezvous phase. During this phase, data from the inertial sensors and the rendezvous radar are utilized. The Command Module pilot can monitor progress with the sextant, with VHF ranging, and with the computer-contained rendezvous program. This operation reduces Lunar Module velocity relative to the Command Module to zero at close range, leaving the Lunar Module pilot in a position to initiate a manual docking maneuver with the translation and rotation control of the reaction jets. These maneuvers are normally done with the Lunar Module, although propulsion or control problems could require the Command Module to take the active role. After final docking, the Lunar Module crew transfers into the Command Module. The remaining ascent stage of the Lunar Module is then jettisoned.

2.2.2.11 Transearth Injection

Navigation measurements made while in lunar orbit determine the proper initial conditions for transearth injection. These measurements are performed as before, using available onboard and earth-based tracking data.

The guided transearth injection, which of necessity is performed behind the moon, is normally made under the control of the GN&C System. Targeting for this maneuver is normally provided by uplink telemetry before the spacecraft passes behind the moon. Several backup means are available to cover possible failures in the primary system. The injection maneuver is controlled to put the spacecraft on a free-fall coast which will attain satisfactory entry conditions near earth.

2.2.2.12 Transearth

The transearth phase is very similar to the translunar phase. During the long coasting phases going to and from the moon, the systems and crew must control the spacecraft orientation. Typical midcourse orientation constraints include ensuring that the high-gain communication antenna can point to earth while remaining within its gimbal limits; that the proper omnidirectional antenna is selected by the crew; and that the spacecraft attitude is not held fixed relative to the sun for too long a period, thus minimizing the effect of local heating. Consequently, a passive thermal-control mode (barbecue) is normally used via the GN&C System to change spacecraft attitude slowly, relative to the sun line-of-sight.

Onboard and ground-based navigation measurements nominally lead to a series of three midcourse correction maneuvers during the transearth flight. Very accurate transearth injection has made it probable that one or more of these maneuvers may be deleted. The aimpoint of these corrections is the center of the safe earth-entry corridor suitable for the desired landing area. This safe corridor is expressed as a variation in flight-path angle of -6.5 deg ±0.05 deg, measured with respect to the local horizontal. A too-high entry could lead to a skipout from the atmosphere; a too-low entry could lead to atmospheric drag decelerations exceeding the crew tolerance.

After safe entry conditions are confirmed by navigation, the inertial platform is aligned or realigned, the Service Module is jettisoned, and the initial entry attitude of the Command Module is achieved.

2.2.2.13 Reentry

Initial control of entry attitude is achieved by GN&C System commands to the 12 reaction jets on the Command Module. As the atmosphere is entered, aerodynamic forces create torques determined by the shape and center of mass. These torques are in a direction toward a stable trim orientation, with the heat shield forward and the flight path nearly parallel to one edge of the Command Module's conical surface. The entry digital autopilot in the GN&C System now operates the reaction jets to damp out any oscillation about this trim orientation. The resulting angle of attack of the entry shape causes an aerodynamic lift; this force is used for entry path control by rolling the vehicle about its wind axis under control of the GN&C System. Range control is achieved by rolling, so that an appropriate component of the lift vector is either up or down, as required. Cross-range control involves rolling the spacecraft so that the lift vector points right or left of the flight path, as required.

Safe reduction of high velocity to suborbital conditions through the energy-dissipation effect of the atmospheric drag forces is the first concern of the entry guidance. At lower velocity, controlling to the earth-recovery landing area is included in the automatic guidance; manual entry maneuvers can also be used as a backup mode. Velocity continues to decrease until deployment of the drogue parachutes. Final letdown is normally by three parachutes to a water landing.

2.2.3 Rope Design Philosophy and Problems Encountered

The principal flight software efforts which, when integrated together, allow such a complicated mission to succeed are coasting flight navigation, targeting, powered-flight guidance and navigation, and digital autopilots. The philosophy which guided the design, development and integration of each of these tasks is presented in this section, and a functional description of each is presented in the appendices.

Early in MIT's Apollo software effort, the engineer who designed a mission program was also responsible for the coding and testing of that program. Because early programs were to fly in unmanned, fixed-sequence flights, mission programs were arranged in a fixed, predefined sequence. AGC memory capacity seemed ample, and programming and verification were relatively simple and straightforward.

With each successive rope, the software task became decidedly more complicated. With the arrival of manned flights, provision for astronaut interaction brought about a requirement for nonfixed program sequences with interfacing routines. The necessity arose for several programs to run simultaneously. Memory requirements began to grow at a staggering rate*. Finally, the mission programs themselves became so complex that it became virtually impossible for an individual design engineer to accomplish all the design, programming and verification tasks by himself. Clearly, the need for a formal design philosophy was at hand.

Mission programs were apportioned into standardized computational, service and interfacing routines. Furthermore, nearly every program was modularized so that there were no assumptions concerning program sequence, except where mandatory. Consequently, the program became tractable, allowing the allocation of analysis, programming and verification to expert programming individuals—each of whom was to become a specialist in his own area.

With this modularization of the programs, it became apparent that many could run in parallel. (The CM AGC Executive allows up to seven to run in parallel, and the LM AGC Executive allows up to eight.) Parallel operation would create DSKY display conflicts, however, because PINBALL originally restricted to one the number of programs which might have access to the DSKY at any one time. But these conflicts had been anticipated, since the multiple-level, DSKY-display capability was being developed concurrently. Furthermore, the DSKY-display capability provided a standard display interface for all programs and established a useful mechanism for restarting programs (see Sections 2.1.3.3. and 2.1.4). The modularization of the programs, together with the multiple-level DSKY displays, allowed the flexibility and program manageability needed to accomplish the Apollo mission.

Problems were attendant throughout the development, however.

Great care had to be exercised in the allocation of erasable memory, since the demand exceeded the available registers; the sharing of erasables wherever possible became standard. With the enlarged staff of programmers, careful control was more critical than earlier. Each individual programmer concentrated on a particular aspect of the program, and frequently was unfamiliar with areas other

---

*Even the relatively simple Apollo 4 program had required no less than 87 percent of the Block I computer memory.

than his own. Considerable effort* was expended in the allocation of erasable storage and in the prevention or correction of erasable-memory conflicts.

Difficulty in "shoehorning" erasable storage** and the ever-attendant problems of erasable-memory conflict were not the only vexations imposed by the meager AGC erasable memory (2048 words): erasable sharing brought on external restraints, causing programs to become less flexible—they had to be programmed to conserve erasable memory even at the cost of simplicity and execution time; and many basic subroutines could not be made reentrant.

From the beginning, restart protection has been provided for all the ropes—at a cost in fixed memory, execution time, and complexity (complexity because a restart could occur anywhere in the program). One school of thought felt such protection was unnecessary; it was unlikely, this viewpoint held, that such a restart would occur in flight at all, and any that did occur would probably be during an unimportant part of the program. However, a more conservative philosophy prevailed, providing safe error recovery—a sobering factor, since little or no redundancy was provided for fault tolerance in the hardware. Simpler, more obvious programming techniques, which might have averted some of the problems encountered, were not used if it were felt that they might restrict the scope and usefulness of the program.

Gradually, provision has been included in the software to check against astronaut procedural errors and to back up hardware failures with alternative software processing; several software procedures have been implemented to ensure that failures of critical switches and indicators can be overcome by special provisions within the program.

---

*At one point an Erasable Committee, consisting of the Assembly Supervisor and representative experts from each of the major areas, would adjudicate every request for an erasable word or bit.

**COLOSSUS 237 (Apollo 8) flew with only 15 unused erasable words, and LUMINARY 69 (Apollo 10) with only 5.

SECTION III

TESTING, VERIFICATION, AND MISSION SUPPORT

For each flight a new assembly of onboard computer programs is integrated and tested. Improvements over the previous flight are included and parameters improvements over the previous flight are included, and parameters are changed to meet specific flight objectives. As mentioned in Section I, this completed assembly of hard-wired and erasable memory is known as a "rope", a name taken from the weaving process by which the fixed memory is manufactured. The present section of this report describes MIT's continuing effort in the qualification and support of each new rope. The support effort is varied in nature. Before release for manufacture, the rope undergoes a vigorous testing and verification program. Specification change procedures provide NASA with control over the software system. Documentation is generated for training and information purposes, as well as for specification control. MIT also supports the Apollo missions by training crews, flight controllers and others, by providing support personnel to NASA, and by actively monitoring each flight.

3.1 Testing and Verification

3.1.1 Testing Philosophy

Because the lives of astronauts are at stake, all components of the Apollo system must undergo exceptionally stringent testing. Schedules have been tight and launches frequent; thus, timely, well-managed testing programs have been necessary. The testing program for Apollo software was designed under additional constraints, because the software is subject to constant change. Improvements are continually suggested by the astronauts, NASA and MIT—even up to the time of launch. The fixed memory, however, must be tested and released for manufacture three to four months prior to flight, to allow for manufacturing time and for integrated testing of the complete vehicle. Thus, an obvious conflict arises between the desire for improvements and the need for testing. As a result, MIT must perform a large amount of work in a short period of time—and with very high accuracy.

In general, the MIT testing program encompasses two major areas—computation and logic. The mathematical portions of the program are tested for computational accuracy, both to discover programming errors and to identify degradation in accuracy resulting from such factors as truncation and roundoff. It is also necessary to test the entire program sequentially to ensure that the proper logical sequences occur.

The first step in the testing program is the preparation of comprehensive test plans. A test plan specifies the objective of the test, the broad initial conditions, and the sequence of program operation, and it identifies the criteria (test points) upon which the results are to be judged. Preparation of test plans requires the cooperation of the designers and programmers who are intimately acquainted with the particular coding being tested, as well as coordination by those familiar with the overall program structure. Test plans thus serve to organize, control and evaluate the testing program.

After preparation of the test plan, the second step in the testing procedure is to generate specific initial-condition data and a detailed operating sequence, including astronaut operations when applicable. The third step is to perform the test on the All-Digital or Hybrid Simulators and to collect the test-point data from on-line printouts and post-run edits. Comparison data are collected from other simulations. The fourth step is to compare the test-point data from the various sources and to make a judgment concerning the future course of the test. It is not unusual for a test to go through the second, third and fourth steps repeatedly before being judged successful. The final step is the documentation of the test.

Testing procedures developed along with the programs. In the early conceptual and engineering stages, MAC* programs were written by the designers to test their ideas before AGC coding was started. When small pieces of AGC coding were completed, they were individually tested to see that all logical branches were correct and that they yielded the desired arithmetic outputs. As these pieces of coding were integrated to form larger blocks, interfaces were tested to verify that the

*As explained in Section I, MAC is a high-level programming language for general-purpose computers, developed at MIT for scientific applications. It is not to be confused with MIT's Project MAC. The latter was named independently, some years later, and is unrelated to the MAC language.

pieces of coding which were tested independently would also work together. Much insight and planning were necessary to ensure that sufficient representative tests were run on combinations of programs, since the length and complexity of the integrated system software made it impossible to test every conceivable sequence of events. However, all of those sequences which could reasonably occur for a particular mission were vigorously tested.

As work progressed from subroutines to the major-program level, testing emphasis shifted from the individual bits and branches to the overall performance, computational accuracy, scaling problems, and major logic flow. It was important to determine whether the design was adequate to perform the required functions.

As it reached completion, the integrated flight rope required performance and stress testing. Typical mission sequences, such as navigation, targeting and powered flight, were simulated. Testing was also designed to ensure that the computer could accomplish all the required tasks in real time. (If the AGC is asked to do too many things at once, a restart will occur, and valuable time will be lost.) It was also important to test the effects of off-nominal procedures and data upon computer functioning.

After the early missions were flown and the testing program became well defined, it became unnecessary to duplicate the above testing for each new mission. The program worked—only the changes and additions needed exhaustive testing. As a flight approached, the testing emphasis shifted towards those program sequences and combinations which were anticipated for the mission.

3.1.2 Levels of Testing

Formally, the testing effort has been subdivided into six levels:

Level 1 testing was part of the early design effort. As a particular set of specifications was created, design engineers coded the equations in MAC and performed various test cases to identify possible computational and logical difficulties, such as loss of acceptable accuracy and range of variables.

Level 2 testing began when a block of AGC coding was completed for the above specifications. The programmer would test the coding on the All-Digital Simulator. Only those factors directly influencing the block of coding were included in the simulation. Results of Levels 1 and 2 testing were compared and distributed among MIT personnel.

Eventually, Levels 1 and 2 were combined by building edit programs in the All-Digital Simulator which processed the data through both MAC and AGC equations, and printed comparisons. As the overall programs became well developed, new design changes would thus undergo "unit testing", which took the place of Levels 1 and 2.

Level 3 testing was done by the programmers to verify the operation of complete programs or routines. Digital and hybrid simulations were used to ensure that the smaller blocks of coding fit together logically. As each logical path of the coding was tested, it was traced on a master copy of GSOP, Section 4, including test number and date. (Section 4 is the NASA-approved specification document for software-logic flow, as discussed in Section 3.2.1 of this report.)

Level 4 testing required the cooperation of designers and programmers, using both digital and hybrid simulations. Sequences of several programs were tested, corresponding to possible mission usage. These tests verified the proper communication from program to program and investigated conflicts in such areas as erasable-memory usage and time sharing, between the major programs. Test points were compared with the edit programs in the All-Digital Simulator, or, if edit programs were unavailable, with an engineering simulation. Completion of Level 4 testing corresponded to release of a program for manufacture.

The programs underwent continual change during Levels 3 and 4 testing due to new specifications, as well as problems uncovered by the testing program. Level 5 testing repeated all the Levels 3 and 4 tests on the final rope which was released for manufacture, and thus verified the continual validity of these earlier tests.

Level 6 testing, which took place after the rope was released for manufacture, made use of the All-Digital Simulator to establish performance specifications, and the Hybrid Simulator to reveal program anomalies. Level 6 testing on the All-Digital

Simulator was oriented toward the particular flight; these tests used the expected timeline, operational trajectories, procedures and erasable data. Expected one-sigma and three-sigma errors in equipment and in state vectors were employed to give a broad range of performance data. Results of the tests were analyzed by the designers and programmers, and presented to NASA as predictions of the Guidance, Navigation and Control System's performance.

3.1.3 Testing Tools

Software designers and programmers used various simulations in the development and testing of the flight programs. The All-Digital Simulator bore the largest brunt of the testing effort. It afforded the most precise and repeatable simulation of the AGC and its environment. The Hybrid Simulator permitted the tester to interface directly with a program by means of a DSKY and to make on-the-spot changes if necessary. The Engineering Simulator provided quick turnaround, thus permitting multiple runs with changes in many parameters. The following sections briefly describe each of these simulations, with emphasis on those aspects pertinent to their use in the testing and verification program.

3.1.3.1 All-Digital Simulator

The All-Digital Simulator has been the most powerful tool in the verification program. It exists entirely as coding on a general-purpose digital computer, and is composed of two logically independent sections, linked by an interface routine. The AGC Instruction Simulator simulates the operation of the Apollo Guidance Computer, both in storage layout and in detailed arithmetic and logical operation. The Environment, made up of a number of MAC-coded subroutines, simulates all relevant aspects of the hardware and flight environment within which the AGC operates. This environment includes effects of the engine, spacecraft dynamics, optics, IMU, radar, astronaut interactions, atmospheric and gravity effects, and celestial-body motion. Almost every aspect of the environment which can conceivably interact with the flight program is included.

During a simulated sequence, the Instruction Simulator advances through the AGC program, instruction by instruction, simulating the detailed operations performed by the AGC in executing each instruction. After each instruction cycle, the state of the simulated computer, including such factors as instruction sequencing, contents of erasable storage, interrupt activity and clock incrementation, is identical to the state of an actual AGC executing the same program; in addition, truncation, round-off, overflow and timing exhibit the same behavior on the simulated AGC as they do in the real one.

In the course of advancing through the AGC program, the Instruction Simulator encounters instructions which refer to input or output operations, such as the reading of an input counter or the setting of an output discrete. A program known as the Communicator examines all such input/output references and determines whether immediate interaction with the Environment simulation is required by the specific action of the AGC. When input data are required by the Instruction Simulator, the Communicator tries to provide this information by extrapolation from the previous Environment state. If this can be done, control returns immediately to the Instruction Simulator. Should the Communicator not have a valid extrapolation formula, there will be a full Environment update. In general, the Communicator updates the Environment over the longest possible time interval consistent with maintaining simulation accuracy.

By maintaining a high degree of similarity between the simulated and the real AGC-Environment interface, the simulated AGC can be subjected to computational loads and dynamic situations which closely approximate the conditions of a real mission. Precision in the simulated AGC performance is degraded primarily by inaccuracies in AGC or Environment models. These inaccuracies may be deliberate, representing a compromise between fidelity and computational speed, or may be of unknown cause and difficult to evaluate; however, the inaccuracies are all within the precision needed to test the programs vigorously.

In addition to providing the Instruction Simulator with all the necessary inputs for the simulation to run, the Environment serves as a standard against which flight software performance can be judged. This is because many of the tasks required of the AGC involve measurement and computation of factors in the external surroundings, such as spacecraft attitude and trajectory, the effects of gravity, and sensor errors. Inaccuracies can arise in these AGC computations for a number of reasons: information from the sensors may be imperfect; the measurements available may have to be processed before the information required can be obtained; space and time limitations in the AGC, with its short, 15-bit single-precision accuracy, also introduce errors; finally, programming errors can lead to subtle or gross miscalculations. All of these error sources are represented in the All-Digital Simulator. However, the Environment portion of the simulator has available or can generate the "true" value of the quantity being measured and the "true" value of the quantity being computed. Although the "true" quantities in the Environment simulation are obtained from finite precision mathematical models, the 64-bit accuracy of the MAC-coded environment is far greater than the AGC provides, and the models are more comprehensive than those used in the flight programs. For example, the Environment can compute the "true" altitude of the Lunar Module above the simulated lunar surface. This altitude can serve as a standard by which to judge the AGC-computed altitude. Post-run edits permit the user to make this type of comparison on any pertinent section of the software.

The Digital Simulator provides the user with numerous output options, traces, dumps and edits, which permit detailed analysis of AGC performance. Before processing each instruction, the Instruction Simulator checks whether there is a user-interrupt attached to that instruction. These interrupts can be initiated by accessing a memory location, or can be made conditional upon various parameters of the computer state or upon the number of accesses to a location. Thus, the user can interrupt the program to dump onto magnetic tape any portion of the AGC memory or the Environment. He can flag the time an instruction occurs, change any register, or even terminate the run. The user may periodically dump a "snapshot" of the entire simulator from which a subsequent simulation can be initiated. This feature, commonly called "rollback", is extremely valuable when many hours have been invested in a simulation run that has terminated for one reason or another. The results may be examined, changes made to the AGC program or the Environment, and the run continued in a deterministic manner. Since the simulation is entirely digital, it has bit-by-bit repeatability, and any changes between runs can be attributed to modifications by the user. As previously mentioned, the editing capability causes information to be stored and then analyzed at the end of the simulation by a MAC program.

Generally, debugging of AGC programs proceeds by testing individual elements of programs on the Digital Simulator separately, and then gradually merging the elements into a working rope. The AGC programmer uses the simulation in early stages of development to debug preliminary coding. The program under test is executed in a simulation, and, by using the various diagnostic tools, the programmer can determine where errors exist.

In later stages of rope development, the Digital Simulator can be used to verify the adequacy of the various guidance, navigation and control programs to perform required tasks in a flight environment. The implementation of specific guidance, navigation or control laws on the AGC often leads to problems with scaling, job sequencing, or timing. These problems may be uncovered in simulation and result in redesign of some of the control algorithms. The closed-loop simulation of the AGC interacting with the vehicle is able to test the adequacy of the steering and autopilot design in many ways that are not possible through analysis alone. In the final stages of program development, the simulator may be used to generate long verification runs which demonstrate the full mission capability of the rope.

The All-Digital Simulator plays the largest part in the testing and verification program. Among its advantages are the exact reproducibility of tests, and the availability of many user options. One disadvantage is that the user cannot interface directly with the program. All required environment and astronaut actions must be decided upon before the test, and changes cannot be made until computer printout is returned to the user. Another disadvantage is that, in a few circumstances, the simulation may be forced to run much slower than real time, as when high-frequency bending is being simulated, and the Instruction Simulator has to wait while digital approximations are being calculated in the Environment. For these cases the Hybrid Simulator is the more appropriate testing tool, and complements the capabilities of the Digital Simulator.

3.1.3.2 Hybrid Simulator

The Hybrid Simulator combines analog and digital computers with various pieces of G&N hardware to provide a real-time simulation of the flight programs. By interfacing with the simulation through a DSKY and various hand controllers and switches, the user can control the flow of the program in process and can make on-line modifications if necessary. This capability is especially pertinent, since the Apollo system involves such a high degree of man/machine interaction. The user may be a designer testing a new design, a programmer verifying his coding, a human-factors engineer evaluating crew procedures, or an astronaut familiarizing himself with the system. Two complete simulators exist, one for the Command Module and one for the Lunar Module. Mockups of the CM and LM cockpits are interfaced with each of the hybrid computers to provide an environment for realistic replication of crew functions associated with the G&N system.

Analog and digital computers are both necessary to provide real-time simulations. Such high-frequency effects in the environment as bending and actuator dynamics are simulated by analog computers, since a digital computer cannot respond in real time with the accuracy needed. Repetitive mathematical and data-processing functions, however, are best performed by the digital computer.

In the Hybrid Simulator, actual Apollo LM and CM computers are used; however, Core Rope Simulators replace all of the AGC memory with erasable memories, thus facilitating conversion from one rope assembly to another. Core Rope Simulators also provide many useful features to aid in program analysis, such as the ability to monitor and change memory locations, and to stop and single-step either computer. Actual Coupling Data Units interface with the AGCs, but the remaining G&N hardware, as well as spacecraft dynamics and the external environment, are simulated. The cockpits feature planetarium displays and television for use by the optics equipment and for simulated lunar landing.

Operation of the Hybrid Simulator requires the participation of an AGC user and a computer operator. An XDS 9300 computer controls the simulation. It initializes, checks and modes the analog computers. It loads the Core Rope Simulator with an AGC program, sets up the values of variables, uplinks erasable-load values to the AGC, and turns the entire simulation on. At this point, the AGC user will call up on the DSKY the AGC program to be verified. This can be done either from the DSKY in the hybrid laboratory or from the one in the cockpit mockup.

During operation, data are taken from the AGC every two seconds in two ways: the cockpit displays, the DSKY and the Core Rope Simulator provide visual data displays; and the telemetry simulator transfers the AGC downlists directly to an XDS 9300 program which records each downlist, together with a selected "snapshot" of pertinent simulation parameters, onto magnetic tape. Following a simulation, the downlink tape is run through an Edit program to produce an arrayed, scaled and labeled line-printer output in a format convenient for comparing AGC and XDS 9300 quantities. Strip-chart recorders are used for recording simulated variables from the analog computers and also for some digital-computer variables after digital-to-analog conversion.

A disadvantage of the Hybrid Simulator is that the results are not exactly repeatable. The output of the analog computers can vary slightly with time, thus preventing a microscopic quantitative analysis and comparison of results. However, qualitative analysis and the checking of logical branches are facilitated by the fast turnaround time, and the ease with which AGC assemblies can be loaded into the Core Rope Simulator and changed as necessary.

3.1.3.3 Engineering Simulator

The Engineering Simulator was designed to aid in early analysis and Level 1 testing. The software logic specified in the GSOP was coded directly in the MAC language and run with a greatly simplified environment. The engineering simulation was also used to help evaluate AGC-coded performance on the All-Digital Simulator. As the Edit capabilities of the All-Digital Simulator were developed and improved, the Engineering Simulator became less important. However, the high operating speed and simple environments of the engineering simulations made them especially suited to statistical analysis of various techniques, such as rendezvous. The user could run many trials with changes in parameters, thus forming a large data base for statistical judgments. It would have been extremely costly and time-consuming to perform such runs on the All-Digital Simulator.

3.1.3.4 Systems Test Laboratory

The Systems Test Laboratory contains two complete G&N hardware systems —one each for the LM and CM. Although used principally to check out the hardware and hardware/software interfaces, the systems provide a software test and verification capability not present in any of the other simulators, since they include actual radars, optics and Inertial Measurement Units. This hardware complement allows the meticulous checking of radar and optics programs and further provides real hardware/software interfaces, with all of their inherent random characteristics. It is these characteristics that can never be duplicated on any simulator.

In the course of checking out the hardware/software, the operators have ofttimes uncovered bugs which otherwise would not have been discovered, since on a simulator all of the possible vagaries of an actual hardware/software union might not have been simulated.

Most problems which occur during flight can be readily explained, but it remains to be proven in the Systems Test Laboratory if that explanation is indeed correct. For example, during the Apollo 11 lunar descent several alarms came up on the DSKY indicating that the computer was saturating without apparent reason. A suspicion that the rendezvous-radar power switch was in the wrong position was confirmed via voicelink to the crew, thus erasing initial doubts about equipment failure. This explanation for the troubles encountered during lunar descent was later verified in the Systems Test Laboratory when the lunar-descent programs were run with the hardware in the incorrect switch configuration.

From a software-testing point of view, one disadvantage the Systems Test Laboratory has is that it makes no provision for spacecraft dynamics, but this is of little consequence since the Hybrid Simulator does. The Hybrid Simulator serves as the tool for the great bulk of those tests which require an astronaut/software/hardware interface. However, those programs which utilize interfaces with the optics and radar are tested in the Systems Test Laboratory. In a real sense, therefore, these facilities complement one another.

3.2 Software Specification Control

The Guidance System Operations Plan (GSOP) is the NASA-approved specification document for each new rope. Before release for manufacture, the coding should fulfill all of the performance requirements and logic specified in the GSOP. Changes in this specification from one flight to the next must be approved by the NASA Software Control Board (SCB) in the form of a Program Change Request or Program Change Notice. There are, however, many points in the coding which are "below" the GSOP level of specification. Changes to coding not covered by the GSOP may be made without NASA approval, but require internal MIT review in the form of MIT Assembly Control Board (ACB) approval. After the rope is released for manufacture, an Anomaly form is used to report detected deviations from the specification.

The GSOP is by definition an incomplete specification, in that it does not accurately reflect such program factors as timing, flag setting, restarting, display of data, jobs and tasks, or erasable structure. Changes in coding below the specification level of the GSOP do not require NASA approval. Thus, there is a certain amount of freedom of implementation available to MIT. However, MIT performs internal change control by requiring Assembly Control Board approval of all changes not specifically covered by other documentation. ACB requests are used primarily to conserve coding and improve program efficiency.

Various meetings with NASA serve to define and control software implementation. The Software Development Plan Meeting is held regularly at MIT to review the status of the software effort and plan future development. Three meetings have been used to mark official NASA acceptance of a rope. (See Section 3.2.3.) The First Article Configuration Inspection (FACI) followed Level 4 testing, and was the preliminary approval to release a rope for manufacture. Upon completion of Level 5 testing, the Customer Acceptance Readiness Review (CARR) marked approval of the complete functioning of the rope. About one month before flight, the Flight Software Readiness Review (FSRR) approved the rope for the particular flight details and uses. Since Apollo 8, the FACI and CARR have not been used.

3.2.1 The Guidance System Operations Plan (GSOP)

As discussed briefly in Section 1.3.2, the Guidance System Operations Plan is the specification document for the software effort. It is published separately for the Lunar Module and the Command Module. The GSOP is updated with each new program release, thus providing NASA with ready and accurate control over the software and system operations. In addition to its role as a specification document, it has served as a working document within MIT to coordinate the inputs of the various groups, and as a testing foundation for simulator personnel. It has also served MSC personnel and contractors as a G&N description and as a crew-training aid.

The GSOP is published in six sections, each a separate volume. Section 1, Prelaunch, contains prelaunch calibration and test operations. Section 2, Data Links, describes programs and data for digital uplink and downlink between the onboard computer and the ground. Section 3, Digital Autopilots, describes the autopilot design and function. Section 4, Operational Modes, specifies the logic flow of the software coding for most programs and routines. (Since Section 4 does not specify the coding itself, programmers are relatively free to use the most convenient method of coding for a particular situation.) Section 5, Guidance Equations, is an engineering-oriented view of the guidance and navigation computations as used by the logic described in Section 4. Section 6, Control Data, is a summary of the data used in the All-Digital and Hybrid Simulators to verify the flight programs.

3.2.2 Change Control Procedures

All changes to program specifications must be submitted for NASA approval as either a Program Change Request or Notice. The Program Change Request (PCR) is a request for a change, originating either at NASA or MIT. It is given a preliminary review for technical content by the MIT program engineer and by the NASA Flight Software Branch, then held for Software Control Board action. Composed of representatives of various branches of NASA, the SCB may disapprove a change, order a more detailed evaluation from MIT, or order MIT to implement the change. This decision involves overall mission considerations and scheduling, as well as the particular software considerations.

Although a Program Change Notice (PCN) follows the same approval procedure as a PCR, it is a notification by MIT that a change is being made, rather than a request for a change. The PCN is used for clerical corrections to the GSOP, or for changes which must be made for program development to continue. The use of PCNs to authorize changes has some risk, in that formal SCB action may disapprove the PCN, requiring the undoing of the change.

An Anomaly is a failure of the program to perform to the specification. Anomaly reports result from testing and inspection after rope release. They may be originated by NASA, MIT, or the other contractors to report program irregularities or deviations in expected performance. The Anomaly form submitted to the Flight Software Branch contains a detailed description of the Anomaly, including its cause, how the Anomaly is recognized, its effect on the mission, avoidance procedures, recovery procedures, and suggested program corrections. Since Anomalies occur late in the preparation for a mission, after a rope has been manufactured, the disposition is usually to write a "program note" for the present mission, and correct the problem in a future release. Sometimes, however, as of result Anomalies, new PCRs, or problems discovered in testing, it is necessary to re-release a rope. When the decision is made to fix an Anomaly, authorization may be given in one of two ways. If the Anomaly has no effect on the GSOP, a routing slip is attached to the Anomaly with direction to fix the problem. If the Anomaly has GSOP impact, a PCR or PCN is prepared and processed in the normal manner. Approval by MSC of the PCR is the authorization to fix the GSOP and the program.

Program and Operational Notes are prepared by the NASA Flight Software Branch and reviewed by MIT personnel with the crews in attendance before each flight. The purpose of Program Notes is to advertise to the crew and flight controllers known subtleties and Anomalies in a rope, and to provide workaround procedures.

3.2.3 Software Control Meetings

Various meetings among NASA, MIT, and the other contractors serve to disseminate information about software status, to control changes in specification, and to mark formal acceptance of the released flight rope by NASA.

The Software Development Plan Meeting is held biweekly at MIT, with NASA represented by the Flight Software Branch. Reports are presented by MIT on the programs in development, and problems are discussed at the programming level. These are working meetings, long and detailed, where many policy decisions are made, and misunderstandings ironed out.

Periodically the Software Development Plan Meeting is expanded to include the Chairman of the Software Control Board, thus forming the Joint Development Plan Meeting. More formal presentations are included, and crucial decisions made.

Following each meeting, the Software Development Plan group issues a plan to organize and control schedules, personnel assignments, and other internal requirements. The plan presents the status of PCRs and Anomalies, and includes detailed milestones of program development, testing, verification and documentation.

In accepting a rope for a specific flight, NASA's original concept was to conduct three milestone meetings:

1. First Article Configuration Inspection (FACI)
2. Customer Acceptance Readiness Review (CARR)
3. Flight Software Readiness Review (FSRR).

The FACI was to culminate MIT's testing program through Level 4 and to provide a "B" release which could be used for training purposes. In a joint MIT/MSC meeting, working groups would review the results obtained from Levels 3 and 4 testing to ascertain whether the rope was ready to undergo Configuration Control. The review at the FACI was directed towards assuring that the program reflected the GSOP specifications, and that the testing program was sufficient and proper. The FACI would approve manufacture of a "B" rope, and authorize MIT to conduct formal Level 5 tests on the rope, using a NASA-approved "Qualification Test Plan".

The CARR was conducted to review the results of Level 5 testing and authorize the manufacture of an "A" release to be used on the mission. All aspects of the program were to be approved, not only those expected for the forthcoming mission.

Following the CARR, MIT conducted Level 6 testing, oriented toward the particular flight. NASA and other contractors also tested the rope, using the expected data and trajectories. Anomalies were reported and documented. The FSRR was then conducted four to six weeks prior to launch, to review program performance under actual mission requirements. This was done to determine whether additional testing or workaround procedures were necessary, and to formally accept the rope for use on the flight.

In actuality, no rope has been accepted according to this plan. The "B" release which follows Level 4 testing has been flown in every mission since Apollo 8, and has often been manufactured prior to FACI. The FACI and CARR Meetings have fallen into disuse, since Software Development Plan Meetings and telephone conferences have provided NASA with a more efficient working format for information and control of rope development. The FSRR is left as the only official meeting for the analysis and acceptance of a rope.

3.3 Documentation Generation and Review

MIT software documentation is necessary for specification control as well as mission support, general communication and training purposes.

The GSOP (described above in Section 3.2.1) is the NASA-approved specification document for each rope, but also provides general information about the software system. Sections 3 and 5 of the GSOP include much of the engineering analysis underlying the control systems and guidance equations. The logic flow of the programs given in Section 4 provides a convenient format for individuals to develop an operational understanding of the guidance and navigation functions on the spacecraft, without having to delve into the actual computer coding. Early versions of Section 4 included the crew-abbreviated and expanded G&N checklists, linking the operational details with the software logic. The checklist format was a DSKY display/crew response sequence. It included pertinent options, and those systems operations which interfaced with the G&N. Later, to expedite document reproduction, the checklist was separated from the GSOP and included in the Functional Description document. It was integrated by MSC into the complete onboard checklist, with format and content essentially unchanged.

The Functional Description document was created to provide an operationally-oriented description of interfaces between the G&N hardware and software, and between the G&N and the backup systems. This document also served in the training and familiarization of crew and crew-support personnel. It was the first MIT document to include a detailed description of all G&N hardware, as well as telemetry outputs and complete backup and malfunction-detection procedures. It detailed those steps the crew would perform to determine where a failure had occurred if one or more symptoms of subsystem malfunction appeared. These procedures were presented in flowchart format, and have been incorporated into the contingency checklist section of the onboard flight-crew data file under the direction of the Astronaut Office and Flight Crew Support Division of MSC. The Functional Description document was updated with each mission. Outside critique by other subcontractors through MSC helped MIT to maintain a high degree of accuracy. The document was last updated for Apollo 12 and has been discontinued, since hardware design and operations have stabilized. Many of the software aspects of the Functional Description document will be fulfilled by a Users' Guide, described below.

The computer listing of the AGC rope also serves a documentation role. This listing is a printout, line by line, of each instruction and location in the rope. However, "remarks" have been liberally added to the listing to aid the user in following and understanding the various programs. A program may include a general description, a list of calling programs, explanations of various branches, and other aids to understanding the logic flow, depending on the individual programmer. There is also a general section of remarks, including lists of verbs, nouns, alarm codes and flagwords. A symbol table provides a cross-reference for symbols used in the various programs and gives their definitions and uses.

A document flowcharting the computer programs has evolved from a series of blue-line charts to the present, bound, mission-specific volumes. These flowcharts are distinct from those of GSOP Section 4, in that they follow in detail how the AGC coding has been implemented. The flowcharts are produced by a documentation group separate from the programmers, which not only makes for standardization, but can serve as an independent check on the validity of the coding. Whenever possible, the flowchart is keyed to the equations of GSOP Section 5. Comments are freely used to clarify a program's function and to define for the benefit of the reader such terms as variables, units and scale factors. Thus, the flowcharts can replace the computer listing as a reference source for many purposes, and can provide a commentary and guide for those cases where the listing must be consulted as the primary source.

The above documents, as well as the Apollo Operations Handbook (published by NASA), have been used for crew training purposes. However, they have generally appeared to be too detailed and inclusive for easy assimilation of information by flight crews. For this reason, the Users' Guide to Apollo GN&CS Major Modes and Routines is being written. The Users' Guide presents the basic operation of the onboard system for use by crew members and flight controllers who have no prior G&N experience. The objective is to comprise all programs, routines, and extended verbs defined by the GSOP, describing their operation, theory and interrelationships in sufficient detail for a crew member to gain the prerequisite understanding on which to base a more rigorous study of specific, flight-particular details and procedures. The Users' Guide is not mission oriented, but is updated periodically to reflect major software changes.

In addition to maintaining the above documents, MIT reviews various NASA publications. The Apollo Operations Handbook (AOH) Volume 2, for the CM and for the LM, is reviewed for accuracy and conformity with each successive onboard program. All current operational Anomalies and program notes are incorporated. Changes to the AOH are submitted on a Proposed Operational Procedures Change (POPC) form to the Apollo Program Control Office and indicate the recommended wording for each desired change. MIT also reviews POPCs submitted by other contractors. The AOH is kept up to date, and the final version is released one month before launch. This document, however, is not designed to be mission oriented; its primary function is to specify the physical characteristics of a given spacecraft —the spacecraft's role during a given mission is treated only peripherally.

Flight Plans and Mission Rules documents are reviewed upon receipt by the current MIT Mission Program Engineer. In addition, the following documents (in preliminary and final editions), issued by the MSC Data Priority Coordination group, have been reviewed by a large number of MIT design and flight-support personnel.

a. Abort Summary Document
b. CSM Rendezvous Procedures Document
c. LM Rendezvous Procedures Document
d. Reentry Procedures Document
e. LM Descent/Phasing Summary Document
f. Lunar Surface Operations Document

These volumes have been reviewed and commented upon within a three-week response period. Communication is mainly in the form of informal comments submitted to MSC Data Priority personnel through the MIT Mission Program Engineer responsible for that flight and vehicle covered in the particular document.

3.4 Mission Support

MIT's tasks in mission support are varied. Crews, flight controllers, and others are given formal and informal briefings, as well as simulator training. MIT personnel are assigned to NASA for flight support; and, via telephone link from Cambridge, MIT plays an important role in real-time support during missions.

3.4.1 Crew Support

A series of flight-crew classroom briefings on the G&N system were developed by MIT personnel to meet several objectives. These briefings sought to define fundamental problems in guidance and navigation, and to show the solutions as mechanized in Apollo. They described the G&N system capabilities and limitations, with emphasis on the reasons for particular programming, and they introduced the Apollo flight crews to detailed G&N procedures, operational control, software moding, and onboard program logic. Flight crews of every mission from AS-204 through Apollo 14, as well as NASA ground flight controllers and representatives of other NASA subcontractors, have been briefed by MIT in these classroom sessions. The training sessions have evolved into a format emphasizing system mechanization, rather than fundamental problems behind the techniques. This change in emphasis was a natural result of greater crew sophistication in understanding the nature of the G&N system. Time limitations also forced strict adherence to matters of immediate mission success.

The great majority of presentations were prepared and presented by the engineers who designed, built and analyzed the G&N system. Other presentations were prepared by simulation verification personnel and the respective Mission Program Engineers. Direct contact between the flight crews and MIT personnel benefited both parties and added a depth of appreciation for each other's problems and goals.

Each training session related to a particular mission and onboard program. Although particularly beneficial from the crew's point of view, this policy placed a sizable burden on MIT engineers at those times when crew briefing conflicted with program release. A possible alternative would have been to have two or three persons devoting their energies to understanding the entire G&N system, solely for crew-training purposes. However, the extremely rapid change and development of onboard programs made this a virtually impossible task.

In addition to the formal classroom briefings, there were periodic special crew briefings with MSC personnel to resolve issues of primary importance. MIT also monitored crew training on the Command Module and Lunar Module Simulators at Kennedy Space Center, and helped in troubleshooting possible system or simulation Anomalies. MIT personnel were thus available to explain G&N operations to flight crews and simulation personnel. These less formal approaches are considered to have been as essential to efficient crew use of the G&N system as the more formal classroom sessions.

To expedite crew procedures, MIT investigated short sequences of crew interactions with parts of the G&N system. These "part-task" evaluations sought to determine the limits of a human's ability to perform various G&N tasks, identifying those environmental factors most significant to performance.

A prime tool for these studies was the MIT Sextant Simulator. This device duplicated full optical motion of the sextant and provided optical images of landmarks, horizons and stars. It was used to verify marking accuracy during navigation-sighting tasks performed under a variety of environmental constraints. These tasks included star/landmark, star/horizon, star/reticle, flashing LM beacon, and simulated Apollo Optical Telescope star sightings. Tests were also performed on KC-135 zero-g flights to verify marking accuracies, and to determine the necessity for tethering the crew members during task performance. The CSM and LM cockpit mockups of the Hybrid Simulator also were used to evaluate crew tasks, such as attitude maneuvers, landing-point redesignations, and IMU-alignment sightings, as well as end-to-end flight sequences.

3.4.2 Flight Support

MIT's role in flight support has undergone considerable change over the course of the program. Early, during the development of the GN&C System, MIT was asked to support the Flight Control Division by sending personnel to be trained as flight controllers. In response to this request MIT assigned several people to the Manned Spacecraft Center in Houston. As it was, those MIT personnel who supported the Flight Control Division at MSC soon lost touch with the new developments at the Laboratory during those early days of rapid change of both hardware and software.

Before any missions were flown, MIT's real-time support underwent a change. As a result, the first four Apollo flights, all of which were unmanned and of less than one day's duration, were supported by the software specialist (dubbed "rope mother") responsible for the development of the onboard computer program and by a representative of the hardware division.

With Apollo 7, the first manned flight, two significant developments occurred. First, the program had become too large and complex for one, seemingly omniscent rope mother to oversee, thus requiring the overall responsibilities to be delegated to a large number of persons (see Section 2.2.3), one of whom was designated Mission Program Engineer. His responsibility included monitoring of the flight programs from the end of Level 6 testing through the real-time mission support. He represented MIT at Mission Control Center, Houston, and participated in any real-time decision making. Second, MIT was asked to provide software specialists to support the Flight Software Branch. These individuals were assigned to Houston on a 6- to 12-month basis and reported directly to the Flight Software Branch.

From Apollo 7 on, MIT has had the availability of a console in the Flight Dynamics Staff Support Room at MSC; and since Apollo 10, this has become a permanemt assignment. The MIT console is concerned not only with software aspects, but with the operation of the GN&C System as a whole; thus, it complements the adjacent Flight Software Branch console.

Since mission support is generally accomplished on a person-to-person basis, it has been advantageous to use a constant small group of people to represent MIT during the missions. Thus, the flight controllers develop confidence in the capability of particular individuals to respond to any mission-critical situation.

During each flight since AS-202, MIT in Cambridge has maintained direct contact with Mission Control Center, Houston through a Scheduling, Conferencing and Monitoring Arrangement (SCAMA). This consists of three dedicated telephone lines, one for two way phone conversations, one for "listen only" air-to-ground communications between the spacecraft crew and mission control, and the last for "receive only" teletype transmissions of Guidance, Navigation and Control parameters stripped from raw telemetry data.

Beginning with Apollo 7, SCAMA facilities were moved into a large room, a digital clock was added to keep track of ground elapsed time, and an input to the XDS 9300 computer was added in parallel with the teletype. This last addition allows a computer-editing process to take place on the telemetry information. Teletype messages and edited data are stored on magnetic tape for recall if required. The edited data are also printed for immediate verification by G&N specialists.

The minimum manpower required for flight support at MIT, Cambridge is three persons per shift, three shifts per day for round-the-clock coverage throughout the mission. These three people are a communicator, a software specialist and a hardware specialist. It is the communicator's responsibility to coordinate SCAMA phone conversations, to maintain a chronological events log and an action-item file, and to call in appropriate experts as required. The software specialist is cognizant of the entire program code and is expert in a particular section of coding current in the flight program. The hardware specialist is cognizant of all operational aspects of the G&N hardware and investigates any variances observed in the telemetry data.

Available for use as troubleshooting tools, procedural verifiers or mission phase predictors are two operational G&N Systems in the Systems Test Laboratory (one for each vehicle) and two Hybrid Simulators (one for each vehicle). These are all loaded with the appropriate flight programs prior to lift-off and maintained in operational readiness throughout the mission.

APPENDIX A

MAJOR PROGRAM CAPABILITIES— Coasting-Flight Navigation

The navigation function of the Apollo spacecraft GN&CS is conducted during all phases of the Apollo lunar mission. As mentioned in Section 2.2, the mission phases for the spacecraft GN&CS are:

1. Launch to earth-orbit monitor
2. Earth-orbit navigation monitor
3. Translunar-injection maneuver monitor
4. Earth-moon (translunar) midcourse navigation and guidance
5. Lunar-orbit insertion maneuver
6. Lunar-orbit landing-site sightings
7. Descent-orbit injection maneuver
8. Lunar-landing maneuver
9. Lunar-ascent maneuver
10. Lunar-orbit rendezvous navigation and control
11. Transearth-injection maneuver
12. Moon-earth (transearth) midcourse navigation and guidance
13. Earth-reentry and landing

The navigation function during many of these mission phases is pure inertial navigation using the IMU and the computer. Typical maneuver phases of this type are the translunar injection, lunar-orbit insertion, lunar ascent, transearth injection, and earth reentry. These mission phases are characterized by large acceleration forces due to the spacecraft engines or atmospheric entry.

During all free-fall or coasting phases of the Apollo mission—cislunar, orbital and rendezvous—the onboard system employs the same navigation concept, a recursive formulation of the optimum linear estimator originally devised by R.E. Kalman. This concept incorporates measurement data sequentially without recourse to the batch-processing techniques common to other methods. Matrix inversion is avoided by regarding all measurement data as single-dimensional or scalar, with the measurement characterized by a geometry vector. These features allow a navigation formulation compatible with the complexity and computational limitations of the onboard computer. A further important feature of this concept is that, within the framework of a single computational algorithm, estimates of quantities such as rendezvous-radar biases (in addition to position and velocity) can be included by the simple expedient of increasing the dimension of the state vector. This appendix has been restricted to these three mission phases which utilize recursive navigation techniques.

A.1 Cislunar Navigation

The cislunar phases of the Apollo mission are the translunar trajectory between earth orbit and the moon, and the transearth trajectory from lunar orbit to the reentry into the earth's atmosphere. These two cislunar trajectories are illustrated in Fig. A.1-1, along with typical navigation sighting periods for the Command Module GN&CS. The primary mode of navigation for the Apollo cislunar phases is the Manned Space Flight Network (MSFN), a system of earth-based tracking stations. Within this system, ground-based radar-tracking data are processed in the Real Time Computation Center of the NASA Manned Spacecraft Center to determine the spacecraft state vector, and to compute required midcourse correction maneuvers. These are telemetered to the spacecraft for targeting the translunar and transearth trajectories to their desired terminal conditions. The onboard spacecraft GN&C System acts in a backup navigation capacity during these two phases. During the cislunar phases the GN&CS provides the self-contained capability to determine the spacecraft's state vector, using onboard measurements, so that the spacecraft can establish and target a safe-return trajectory to the earth if communications from the earth were lost. The sighting schedule illustrated in Fig. A.1-1 on the outbound translunar trajectory is the schedule used to checkout and calibrate the spacecraft navigation-sighting system under nominal conditions when the trajectory is being determined by the ground-tracking stations. Shown on the return transearth trajectory is a schedule which would be typical in the case where communication to the spacecraft were lost in the vicinity of the moon, necessitating that the system onboard the spacecraft navigate and control the return trajectory to earth. In this abort case, the objective of the spacecraft GN&CS is to determine and control the transearth trajectory such that the required earth-entry corridor conditions are achieved for

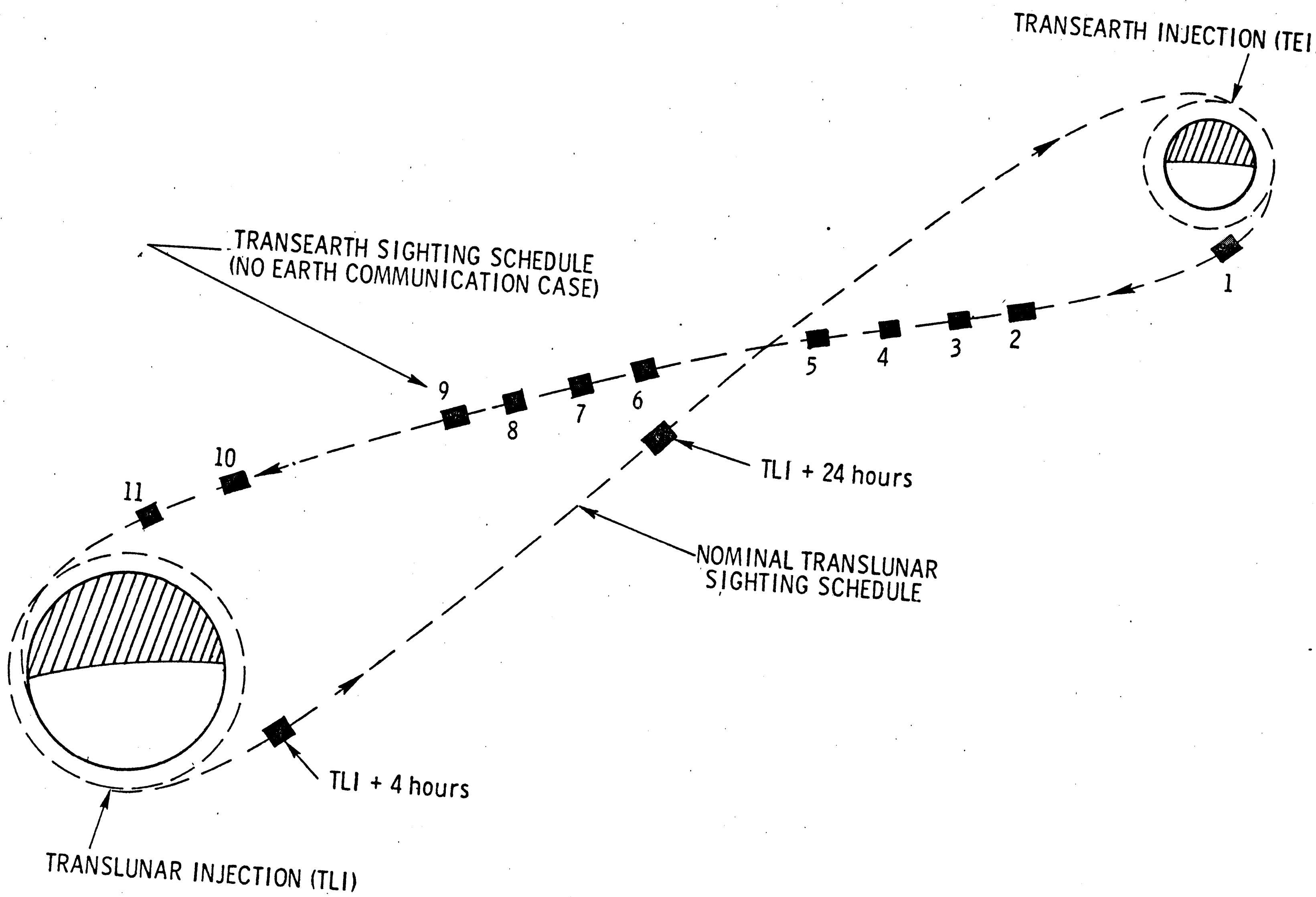


Figure A.1-1. Apollo Cislunar Navigation Phases a safe return. The navigation measurement used to achieve this objective is an optical star/horizon or star/landmark measurement made with the CM sextant.

For a cislunar navigation sighting, either the astronaut or the midcourse navigation program in the AGC points the sextant's two lines-of-sight at a specified reference star and landmark or horizon. It is then the navigator's task to center the superimposed star-image onto the landmark, if landmarks are being used, or onto the substellar point of the horizon, if the horizon target is being used. A sextant view of a typical star/horizon measurement is illustrated in Fig. A.1-2 at the moment when the navigator signals the computer to record the sextant trunnion angle and time of mark. This illustration is typical of the star/horizon view in the sextant during the first sighting interval shown on the translunar phase in Fig. A.1-1 when the spacecraft is approximately 30,000 nmi from the earth. In the Apollo lunar missions to date, the sunlit horizon of the earth has provided a more consistent and useful target for cislunar navigation sightings than landmarks, due to cloud cover and limited sunlit surfaces over major portions of cislunar trajectories. For navigation sightings using the moon, landmarks are preferred over horizons for their greater accuracy. Either the near or far substellar point of a horizon can be used in a star/horizon measurement, as shown in Fig. A.1-3. In this type of a measurement, it is important to superimpose the star-image on the sunlit horizon as close to the substellar point as possible and minimize the measurement plane misalignment error illustrated in Fig. A.1-3.

As previously stated, a single navigation concept is used in the Apollo spacecraft G&N systems for all coasting phases of the mission. A simplified functional diagram of the cislunar-navigation concept is shown in Fig. A.1-4. In this case, free-fall equations of motion extrapolate a six-dimensional state vector (position and velocity), along with the error-transition matrix, to the time at which a navigation measurement is to be made. After the reference star and planet landmark or horizon have been selected by the navigator, an estimate of the angle (A_{EST}) between this star and target is computed, based upon the extrapolated state vector, the reference star and the planet target. A measurement geometry vector ($\underline{b}$) is also determined, based upon the estimated vehicle state vector, reference star, planet target, and the type of measurement being made. For cislunar-navigation measurements, this geometry vector ($\underline{b}$) lies in the reference star/target planet plane and is normal to the planet line-of-sight, as illustrated in Fig. A.1-5. When the navigator superimposes

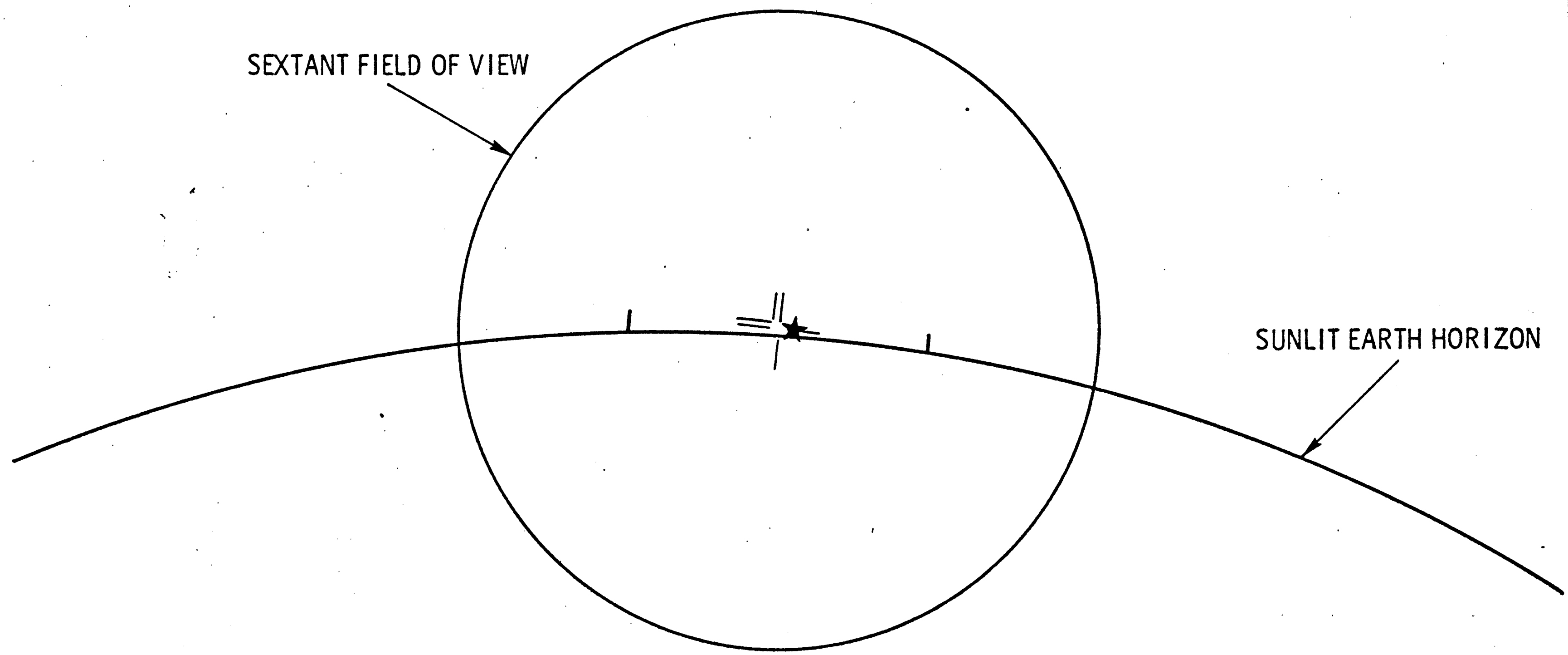


Figure A.1-2 Sextant View of a Typical Star/Horizon Measurement for First Cislunar Sighting Interval

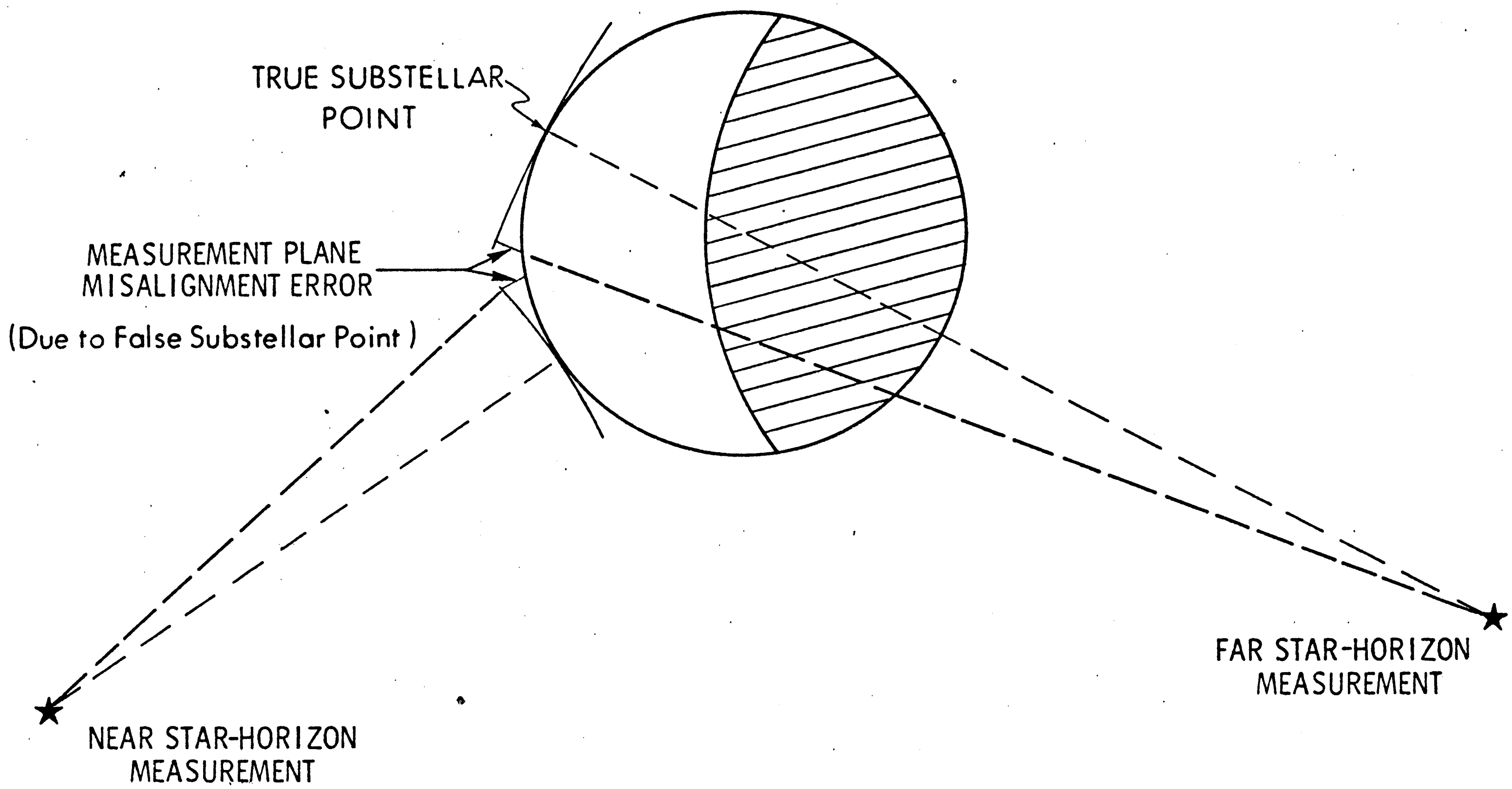


Figure A.1-3 Sextant Star/Horizon Measurements

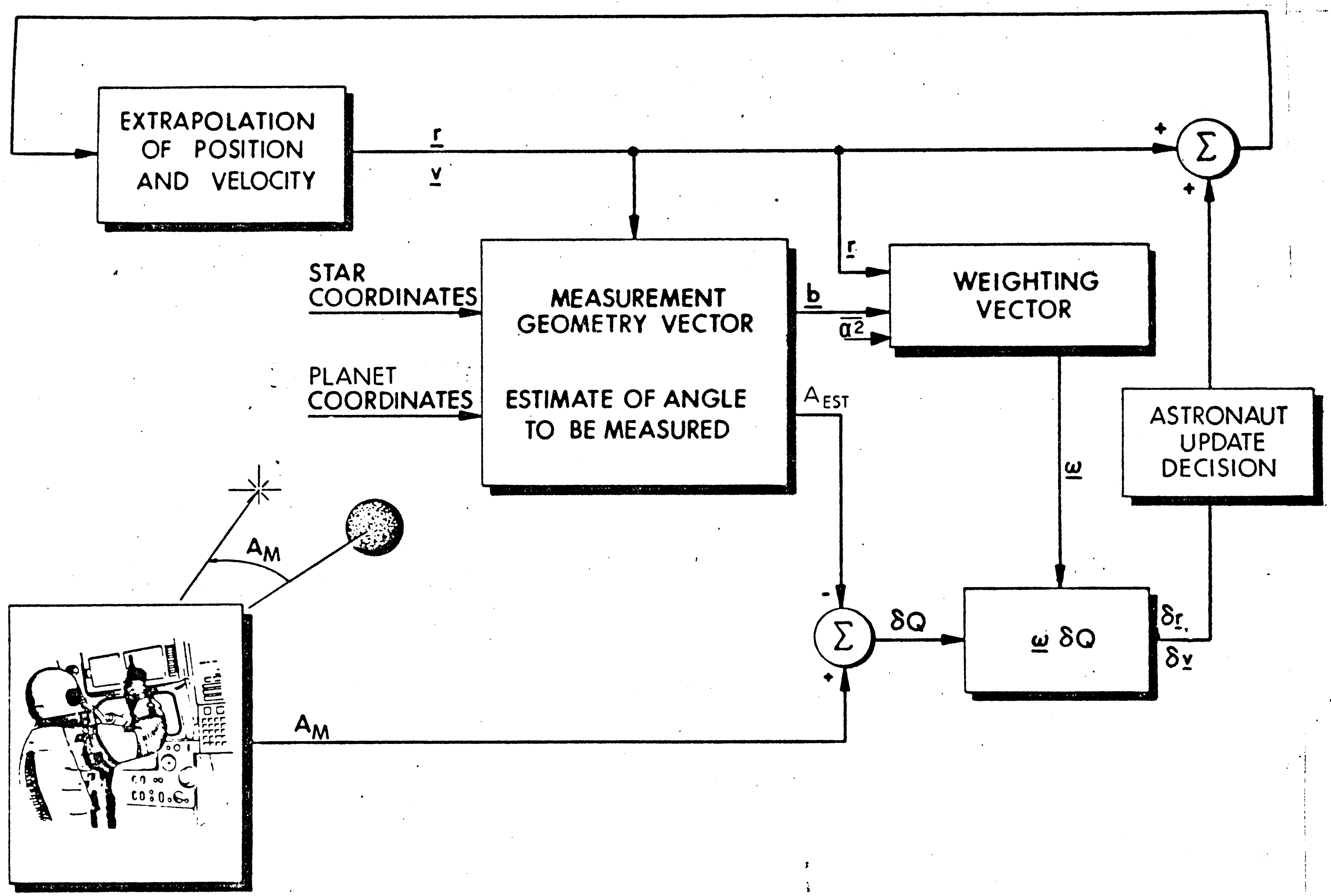


Figure A.1-4 Simplified Functional Diagram of Cislunar Navigation the images of the reference star on the planet target in the sextant field-of-view, the computer compares the measured angle, A_M, with the estimated angle, A_{EST}. With reference to Fig. A.1-4, the following are algebraically combined to form a weighting vector, $\underline{\omega}$: the measurement geometry vector, $\underline{b}$, the error-transition matrix, W, which is extrapolated to the measurement time (and updated, to take into account incorporation of a measurement for use with subsequent measurements), and the *a priori* mean-squared measurement error, $\overline{\alpha^2}$. A statistically optimum state-vector update, ($\delta\underline{r}$, $\delta\underline{v}$), is then computed from the difference in the estimated and measurement angles, δQ, and the weighting vector, $\underline{\omega}$. The geometry vector, $\underline{b}$, used to determine the weighting vector, $\underline{\omega}$, represents to a first-order approximation the variation in the measured quantity (A_M in this case) resulting in variations in the components of the state vector. This concept is illustrated in simplified form in Fig. A.1-5, depicting a single cislunar star/horizon navigation measurement and position update. The estimated position of the spacecraft is updated in this simplified example along the measurement geometry vector, $\underline{b}$, by an amount $\delta\underline{r}$, such that A_{EST} equals A_M. This example is simplified in two major respects: first, the magnitude of the update, δr, would be a function of the statistics of the sextant-measured errors and the extrapolated error-transition matrix, W, and would seldom make the measured and estimated angles exactly agree; second, the update shown in Fig. A.1-5 is entirely along the measurement geometry vector, $\underline{b}$, which might be valid for the first navigation measurement taken, but on subsequent measurements the weighting vector, $\underline{\omega}$, will rotate $\underline{b}$ by the correlation represented in W, such that the update $\delta\underline{r}$ will not be along the $\underline{b}$ vector. This correlation feature is central to the navigation concept. It should be recognized, however, that even though the cislunar star/horizon measurement directly updates the vehicle state vector in only one direction, $\underline{b}$; the other position and velocity components are also updated to a lesser, but still significant extent, through correlation. To achieve the greatest accuracy in cislunar navigation, sequential star/horizon measurements are ideally chosen so that the measurement planes of sequential sightings are separated by about 90 deg.

An important point to be noted in the cislunar-navigation functional diagram of Fig. A.1-4 is that, after the state-vector update has been computed by the AGC, this update is displayed to the navigator for his review, and he personally decides whether to accept or reject the update and navigation measurement. If the state-vector update computed from the first navigation sighting taken after several hours without navigation sightings exceeds a predetermined threshold, or if the update is fairly

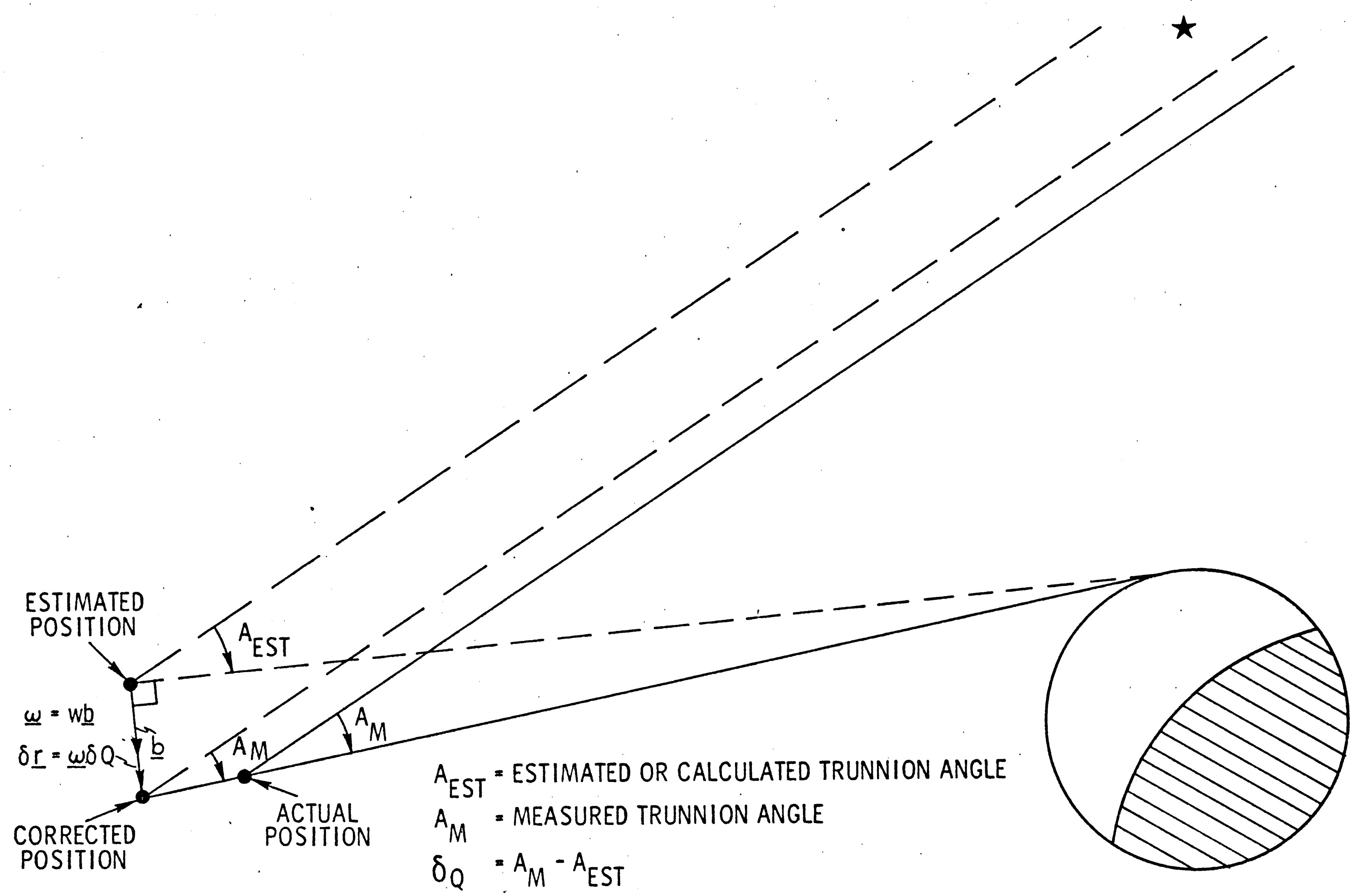


Figure A.1-5 Simplified Star/Horizon Navigation Updating close to the threshold and the navigator is uncertain as to the sighting accuracy or identification of the target, he would reject the update. Upon rejection he repeats the navigation sighting. If the state-vector update is essentially identical to the previous (unincorporated) update, the navigator is logically obliged to accept the update and incorporate it into the state vector. Normally, after the first few navigation sightings and updates in a sighting period, all subsequent updates will fall below the preselected threshold and are routinely accepted.

Spacecraft cislunar-navigation accuracy is primarily limited by (a) unmeasured or unaccounted-for perturbing forces on the vehicle, (b) computational precision and computer-word length, and (c) optical measurement errors. In the Apollo GN&CS the optical measurement errors are the most serious. These measurement errors arise from:

a. Planet-lighting limitations
b. Sextant optical-design limitations
c. Horizon-phenomena uncertainties
d. Astronaut-sighting inability to determine the substellar point on the horizon, and to superimpose the star/horizon images during the presence of spacecraft attitude motion.

In the initial prototype Apollo spacecraft-sextant design a blue-sensitive photometer was included for horizon detection, but this was subsequently removed from the production systems since it had been decided that earth-based radar tracking would be the primary source of cislunar navigation. Without the photometer the navigator must select an altitude point (horizon locator) that can be consistently repeated from one navigation sighting to the next. It is believed, based upon simulation and flight experience, that the higher altitudes of the sunlit horizon provide the most consistent reference for navigation sightings where atmospheric phenomena are less likely to cause perceptual uncertainties. This reference altitude is approximately 32 km above the earth. Figure A.1-6 is a further illustration of how atmospheric weather conditions, such as clouds, can change the apparent horizon altitude in the lower atmosphere, and why a higher altitude reference was chosen. Each Apollo navigator must choose his own particular horizon altitude and try to maintain this reference throughout the cislunar phases. From post-flight analysis data of five Apollo lunar missions, this reference altitude has varied between 17 km

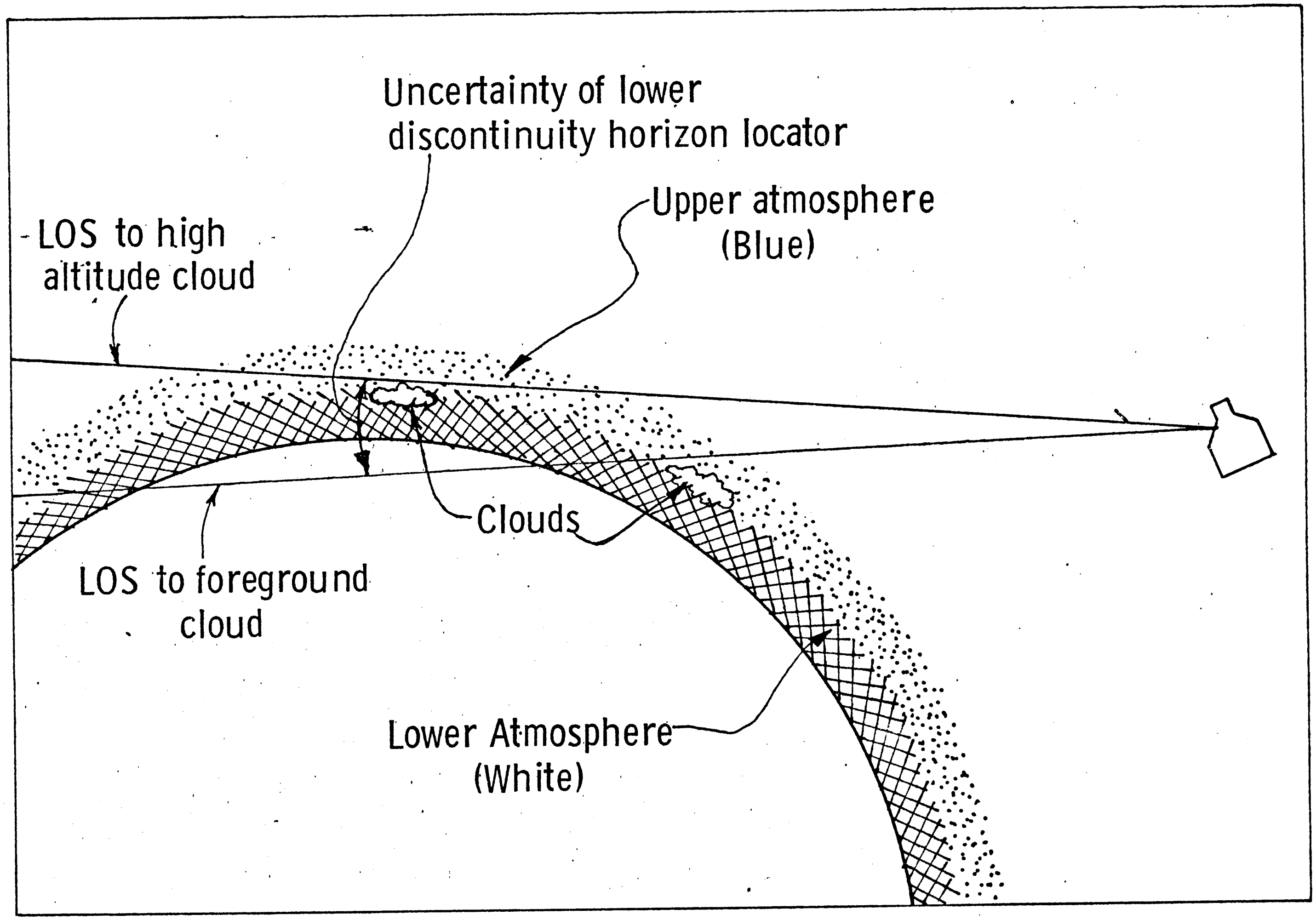
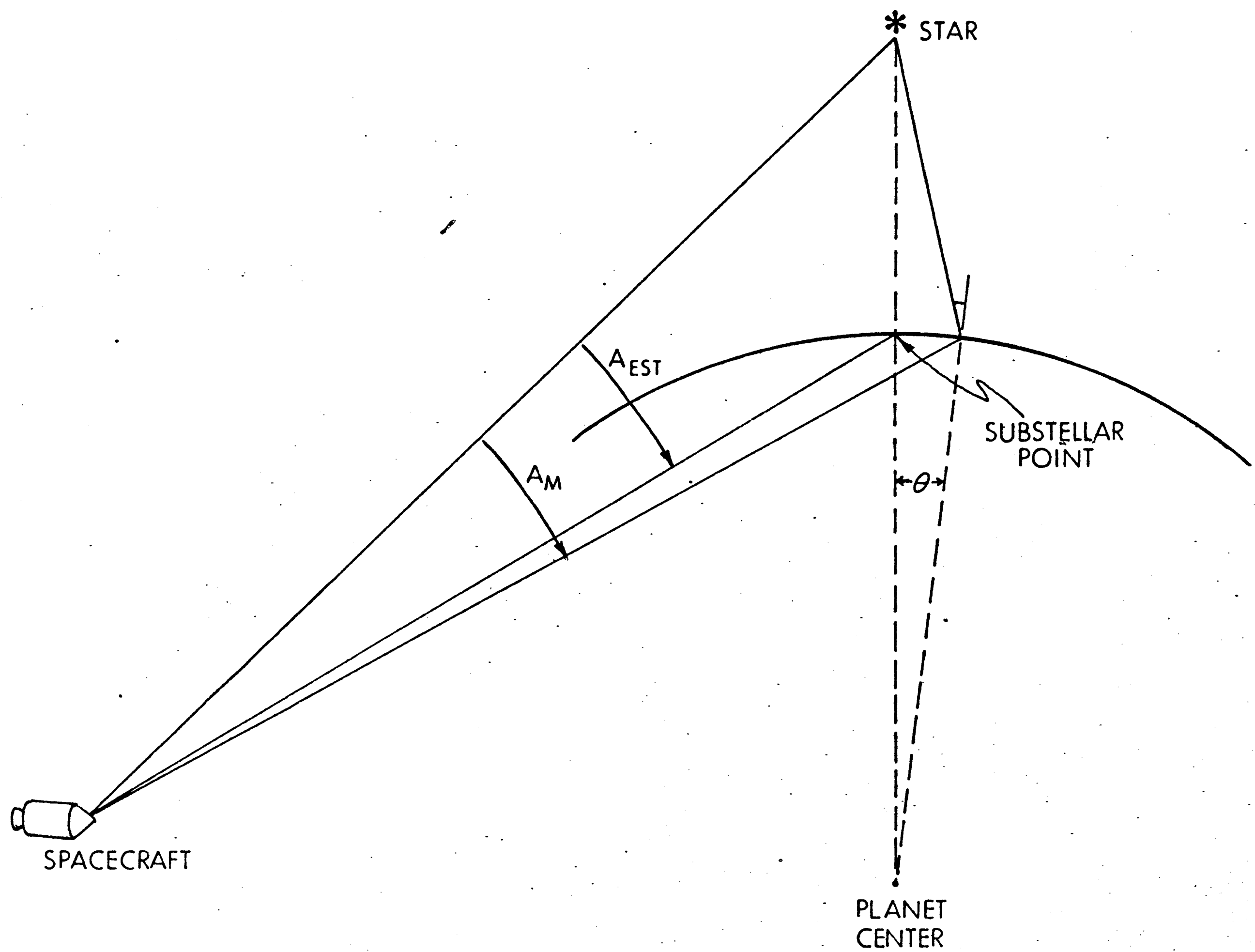


Figure A.1-6 Illustration of Cloud-Top Problem with the Use of Apparent Horizon as a Locator



θ *MEASUREMENT PLANE MISALIGNMENT ANGLE*

A_M *MEASURED TRUNNION ANGLE*

A_C *ESTIMATED TRUNNION ANGLE*

Figure A.1-7 Measurement-Plane Misalignment to 44 km, but the consistency of an individual navigator once he has chosen this altitude reference is the most important feature with respect to navigation accuracy rather than the absolute value of this reference altitude. Since it is difficult to determine the absolute-reference altitude a given navigator will use before a mission, the first translunar-sighting period shown in Fig. A.1-1 at TLI+4 hours is used to calibrate the sextant and determine the reference altitude of the sunlit horizon the navigator will use for that flight by letting him sight on the horizon and then check this initial sighting against the predicted sighting angle using telemetered angle data. These initial horizon sightings along with other translunar sightings, are used to update the stored reference altitude in the guidance computer. Navigation sightings using the lunar horizons are naturally not complicated by the atmospheric effects encountered for the earth horizon. As a result these are more straightforward —the biggest problem being the roughness of the lunar terrain itself.

The third factor previously listed that affects navigation accuracy is the astronaut's ability to correctly superimpose the reference star on the horizon at the substellar point. This point is contained in the measurement plane defined by the spacecraft, star, and center of the planet at the point of tangency of the line-of-sight from the spacecraft to the horizon. Measurement plane misalignment is illustrated in Fig. A.1-7. In general, the star is not placed at the substellar point, but slightly to one side or the other, due to the dynamic nature of the measurement, since small attitude changes continually take place, or due to insufficient range and resulting curvature of the horizon resulting in a perceptual limitation to the accurate determination of the substellar point. This type of measurement error causes the sextant trunnion angle to be too large for a near-horizon (Fig. A.1-3) and too small for a far-horizon measurement.

The Command Module G&N system for cislunar navigation is basically a computer-aided manual operation. The navigator must initiate the navigation program, calibrate the optics, select the desired star and planet horizon, make the sightings, and finally accept or reject the resulting state-vector update computed by the AGC. In essence, the navigator is system manager, mission-sequence controller, subsystem-interface coordinator, and performer of specialized tasks too difficult or costly to automate. The AGC performs the basic navigation computations using the manually-controlled optical sighting data. Manual control was deemed desirable for the Apollo cislunar-navigation sighting operation to minimize the number of active GN&C units, thereby conserving power (since only the computer and sextant are needed). Recall that the GN&C System acts in a backup navigation capacity for safe earth return during those cislunar phases when the navigator has ample time to conduct navigation sightings. Operational experience during the lunar missions indicates that attitude maneuvering of the vehicle to a landmark or horizon while maintaining star acquisition manually for accurate sightings is a difficult task. For this and other reasons, such as the desire for passive thermal control of the vehicle in an automatic mode, it was decided to keep the GN&C System IMU powered during cislunar flight. The IMU availability affords additional assistance to the navigator by providing automatic control of the optics to the selected reference star and automatic control of the attitude maneuver to the computed substellar point. Furthermore, automatic vehicle-attitude hold during star acquisition and star-acquisition maintenance during the maneuver to the horizon also are particularly helpful under light-vehicle conditions during the transearth phase. With these aids, the astronaut's navigation-sighting task is effectively eased, and his major task becomes one of fine correction of the vehicle attitude—performing the delicate task of superimposing the star/horizon images and marking when superposition is achieved.

In the analysis of the cislunar-navigation data, it is felt that, even though the computer-aided manual-sighting performance is adequate for the Apollo missions, further accuracy can be achieved by making the sighting operation more automatic with the implementation of a horizon photometer designed to utilize two spectral regions of the sunlit horizon. The role of the navigator would still be important in handling other unforeseen problems that might arise during the missions, such as scattered-light conditions in the optics requiring alternate stars to be chosen for navigation, and reflections from debris and particles making star recognition difficult, if not impossible, under some conditions. The human navigator is well suited to handle problems of this type, while the GN&C System can be designed to relieve the navigator from the more routine but overburdening detail of the navigation operation.

A.2 Rendezvous Navigation

The nominal lunar-orbit rendezvous-trajectory profile for Apollo missions is illustrated in Fig. A.2-1. This profile is referred to as the concentric flight

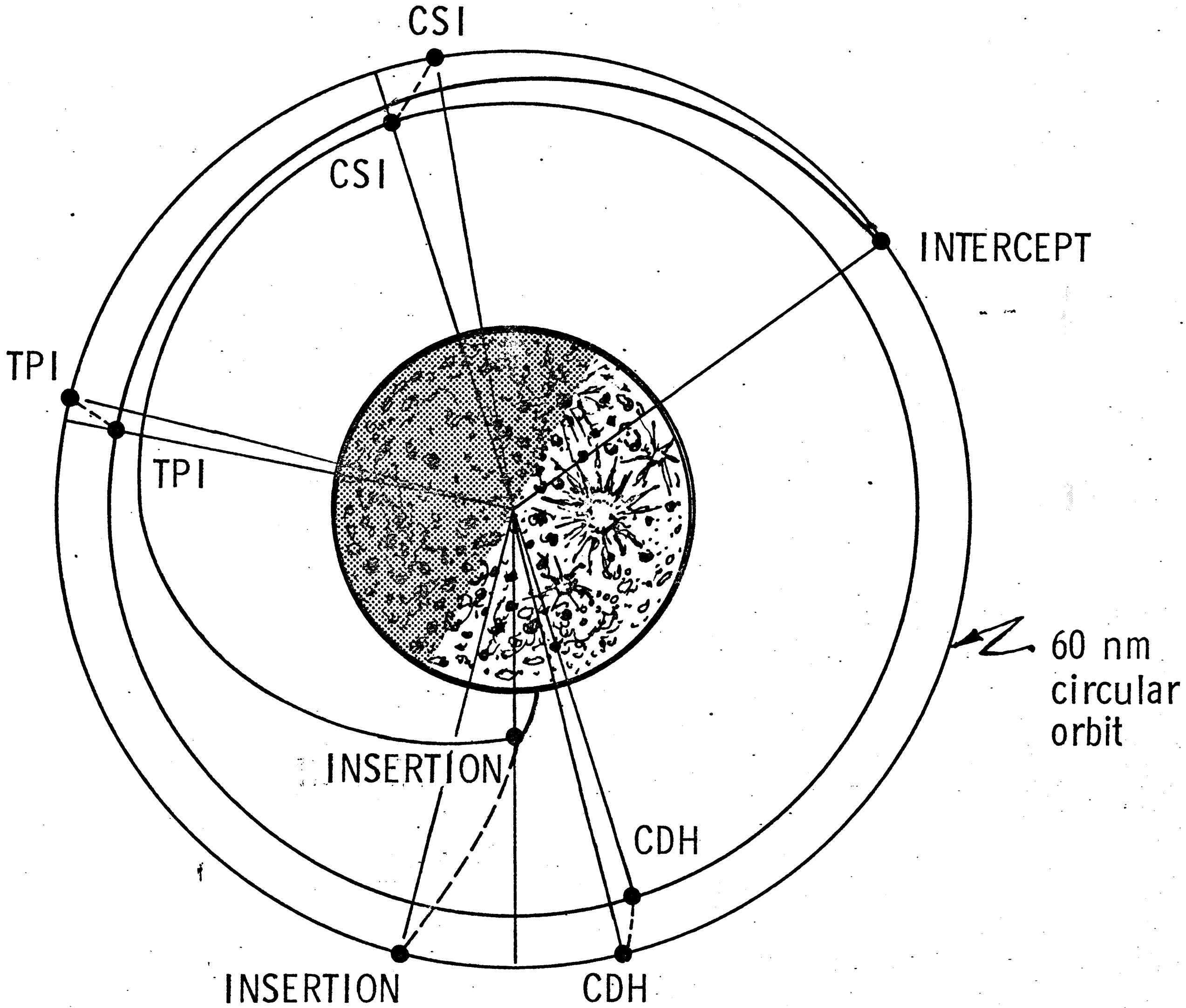


Figure A. 2-1 Nominal Apollo Rendezvous Profile plan and consists of two phasing-type maneuvers after lunar-ascent insertion (CSI-coelliptic sequence initiation; CDH-constant differential height maneuver) to place the active vehicle (LM) in a coelliptic orbit at an essentially constant altitude difference below the passive vehicle (CSM). After these conditions are established, the transfer-phase-initiation maneuver (TPI) places the LM on an intercept trajectory with the CSM. A series of midcourse correction maneuvers (MCCs) are normally made to improve or maintain this intercept trajectory so that the astronaut can manually perform the terminal-braking maneuvers before the intercept point. The objectives of the spacecraft rendezvous-navigation system are to maintain and update the estimated vehicle position and velocity vectors with relative tracking data so that the three major maneuvers and the midcourse corrections of the rendezvous profile can be correctly computed and executed, thereby minimizing propellant usage and achieving an accurate intercept trajectory with the CSM such that the manual terminal-rendezvous maneuvers can be efficiently performed.

In the rendezvous profile of Fig. A.2-1, both the active and passive vehicles conduct simultaneous rendezvous navigation; thus the CSM can provide maneuver information to the LM for backup purposes or execute retrieval maneuvers, if required. The LM rendezvous-tracking sensor is an amplitude-comparison, monopulse tracking radar which tracks a transponder on the Command Module. This radar provides range, range rate, and the two antenna-tracking angles (specified shaft and trunnion) as measurement data to the onboard rendezvous-navigation program. This operation is automatic and requires only general monitoring by the astronauts in the LM. The Command Module astronaut uses the optical sextant to manually track a flashing beacon, located below the LM rendezvous-radar antenna, or reflected sunlight from the LM to provide tracking data to the CM rendezvous program. This operation is similar to that used in cislunar navigation except that only the sextant-articulated star line-of-sight is used for rendezvous tracking, and the sextant tracking angles are referenced to a stable coordinate frame to which the inertial measurement unit is aligned. On the Apollo 7 and 9 missions, CM rendezvous navigation employed only optical-tracking data. On following missions a modification to the vehicle very-high-frequency (VHF) communication system provided relative-range information to the CM rendezvous-navigation program. After initially starting the VHF range system, these range data are processed automatically by the onboard Apollo CM guidance computer, while sextant optical tracking is still a manual task. Figure A.2-2 summarizes the tracking measurement

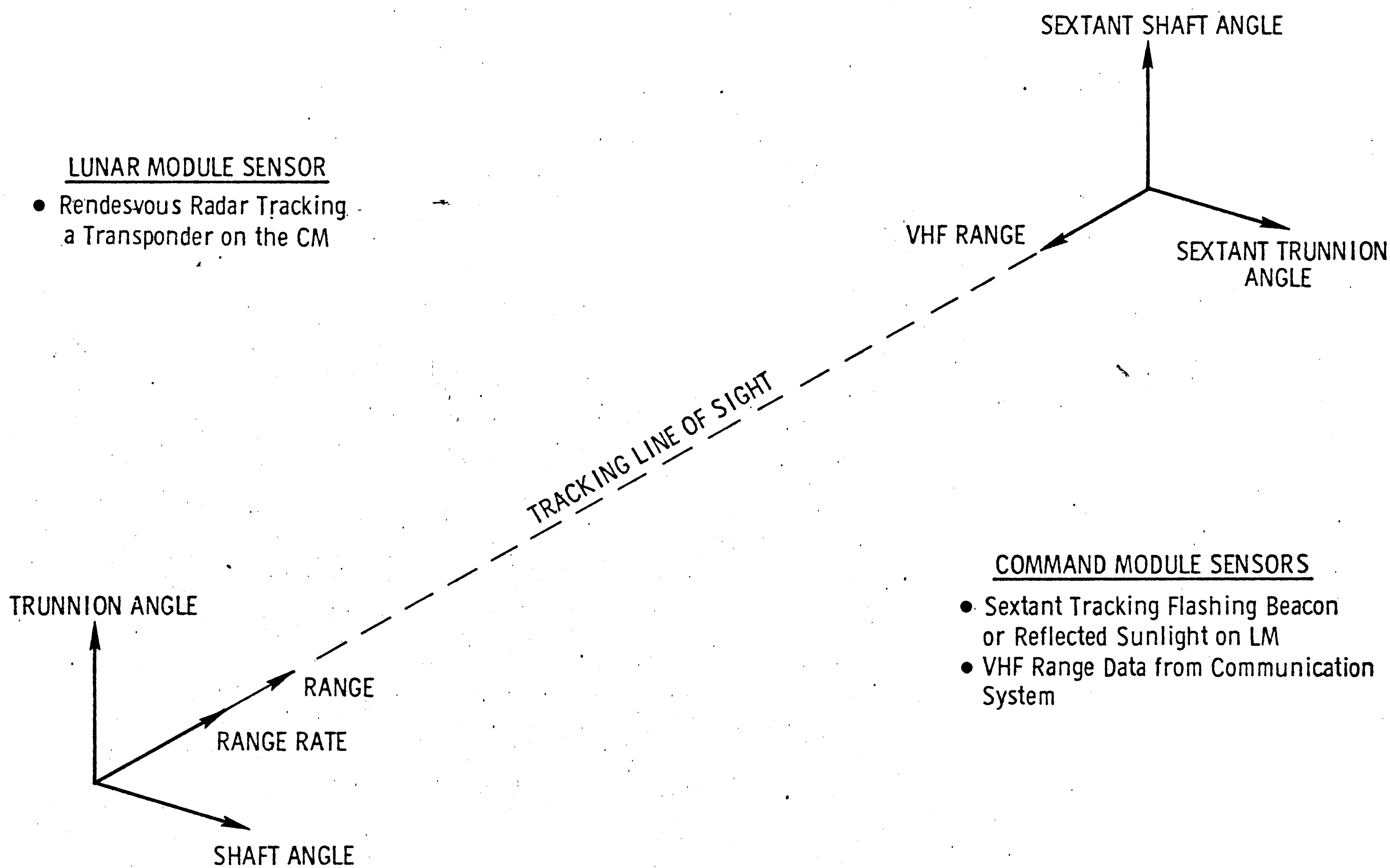


Figure A. 2-2 Command and Lunar Module Rendezvous Measurement data used in both the LM and CM for rendezvous navigation. Even though the tracking sensors on the two vehicles are quite different, the identical navigation concept is used in each vehicle for rendezvous navigation.

As mentioned before, a single navigation concept is used in the Apollo spacecraft GN&C Systems for all coasting phases of the mission. A simplified rendezvous navigation functional diagram is shown in Fig. A.2-3 and is similar to that for the cislunar navigation of Fig. A.1-4, except for differences required by the tracking sensor and target vehicle. It might be noted that the cislunar navigation and trajectory control is essentially a rendezvous problem between the spacecraft and the moon, so it is not surprising that the same navigation concept can be applied for the cislunar and rendezvous phases. With reference to Fig. A.2-3, the active and passive vehicle state vectors are extrapolated to the time a navigation measurement is to be taken by the coasting-integration program. An estimate of the rendezvous measurement, A_{EST}, is computed from the two extrapolated state vectors and subsequently compared with the measured tracking data, A_M. The difference, δQ, is then combined with the appropriate weighting vector to compute an update, ($\delta \underline{r}$, $\delta \underline{v}$), to the spacecraft estimated state vector. Several important points should be noted in this operation. First, the operation just described is done sequentially for each of the four tracking data (range, range rate, antenna shaft angle, antenna trunnion angle) that constitute a navigation measurement in the Lunar Module GN&C System (Fig. A.2-2) and, likewise, sequentially for the two sextant angles and VHF range data in the Command Module. Second, the computed state-vector update, ($\delta \underline{r}$, $\delta \underline{v}$), for each tracking measurement is automatically checked in the computer against a preselected threshold. If the magnitudes of the computed $\delta \underline{r}$ and $\delta \underline{v}$ are both less than their respective threshold levels in the state-vector alarm test, the update is automatically incorporated in the state-vector estimate. If they exceed the threshold levels, the astronaut is informed and must decide whether to incorporate or reject the update. Typically, the navigator would reject the update until he were sure that the proper target was being correctly tracked and then accept later updates. The state-vector update monitoring in the rendezvous-navigation program is, therefore, a semiautomatic operation, whereas it is a completely manual operation in the cislunar navigation program. As shown in Fig. A.2-3, either the active or passive vehicle state vector can be updated by the rendezvous navigation program. This decision is made early in the mission and is not normally changed thereafter. The velocity changes resulting from rendezvous maneuvers are automatically incorporated into the active vehicle

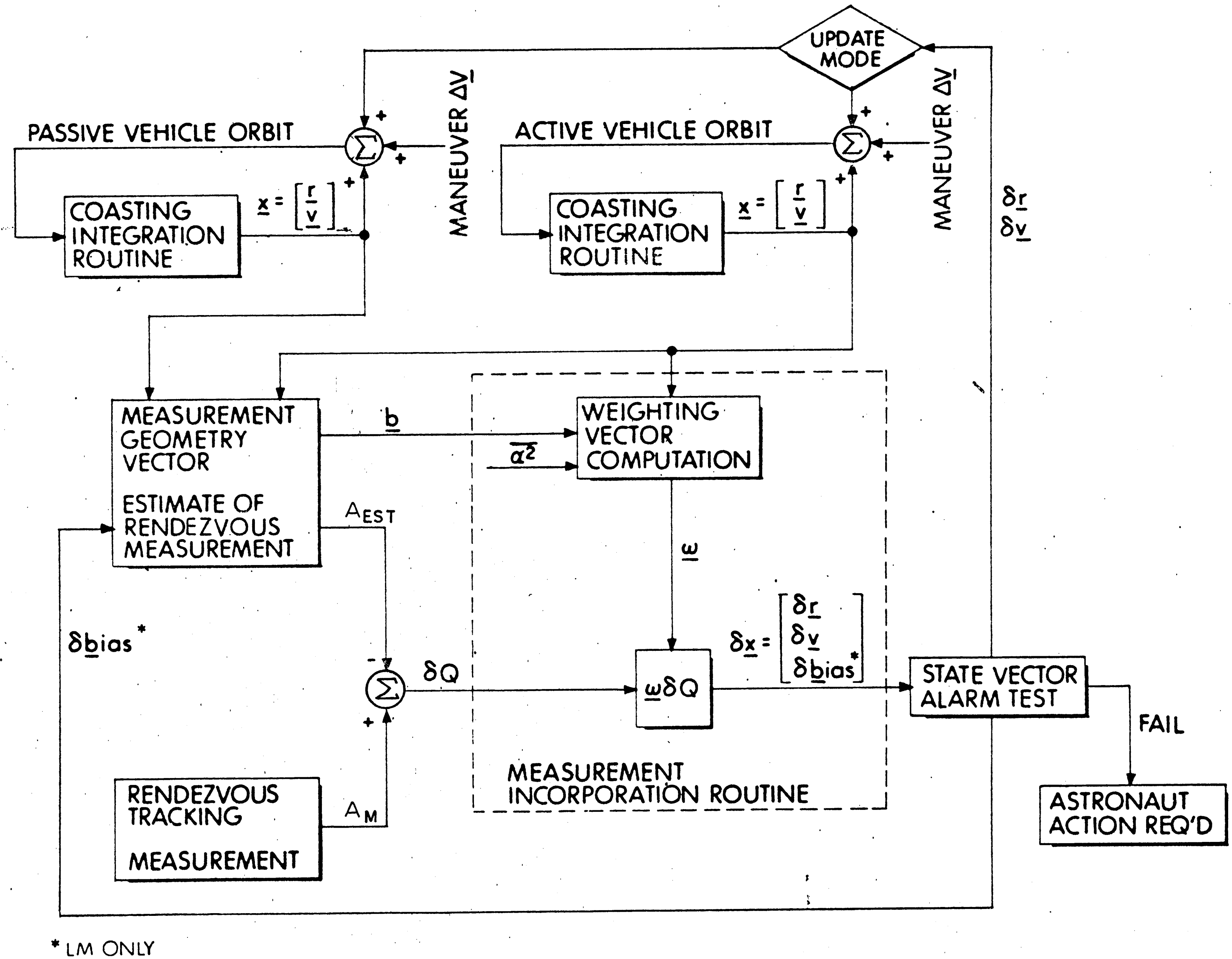


Figure A. 2-3 Simplified Functional Diagram of Rendezvous Navigation system through the IMU and are then communicated to the other vehicle for incorporation by the astronaut in the passive GN&CS. Finally, in the case of LM rendezvous navigation, the state vector and error-transition matrix used in the navigation-measurement incorporation routine of Fig. A.2-2 are increased from six to nine dimensions to estimate the angle biases in the rendezvous-radar tracking data. The rendezvous radar is not rigidly mounted to the inertial-measurement and navigation base (as are the CM optics), and the structural bias between the radar antenna and inertial unit is, therefore, estimated along with the six dimensions of the position and velocity of the state vector. In practice, only two of the additional dimensions are used for antenna-bias estimation, with the ninth element set to zero. As shown in Fig. A.2-3, the estimated antenna-bias angles are automatically incorporated in the estimated rendezvous measurement calculation.

Figure A.2-4 illustrates a simplified rendezvous-navigation angle-measurement incorporation similar to that shown in Fig. A.1-5 for the cislunar-navigation case. In the example of Fig. A.2-4, the position correction $\delta\underline{x}$ does not lie completely along the measurement geometry vector, $\underline{b}$, because of the correlation represented in the weighting vector between the error in the measured direction (represented by $\underline{b}$ in this case) and the errors in the unmeasured position and velocity directions. As mentioned in the discussion of cislunar navigation, this correlation is zero for the first angle measurement, but then builds up over subsequent measurements taken along the trajectory. In the rendezvous-navigation case, direct measurements of range and range rate are made in the LM GN&CS (Fig. A.2-2), so the velocity components normal to the line-of-sight are the only dimensions of the state vector dependent upon correlation for updating. In the CM GN&CS case, there are no direct navigation measurements of velocity in any direction, and this update information is completely dependent upon correlation. The navigation concept was most dramatically demonstrated in the first manned Apollo mission (Apollo 7), in which optical-sextant tracking was used to control a successful rendezvous intercept (TPI) and the following midcourse-correction maneuvers. During Apollo 9, the CM acted as a backup and monitor to the active LM during rendezvous, again using only sextant tracking data for the onboard navigation measurement.

The rendezvous recursive-navigation program employs various approximations and linearizations to make the implementation of the navigation computation,

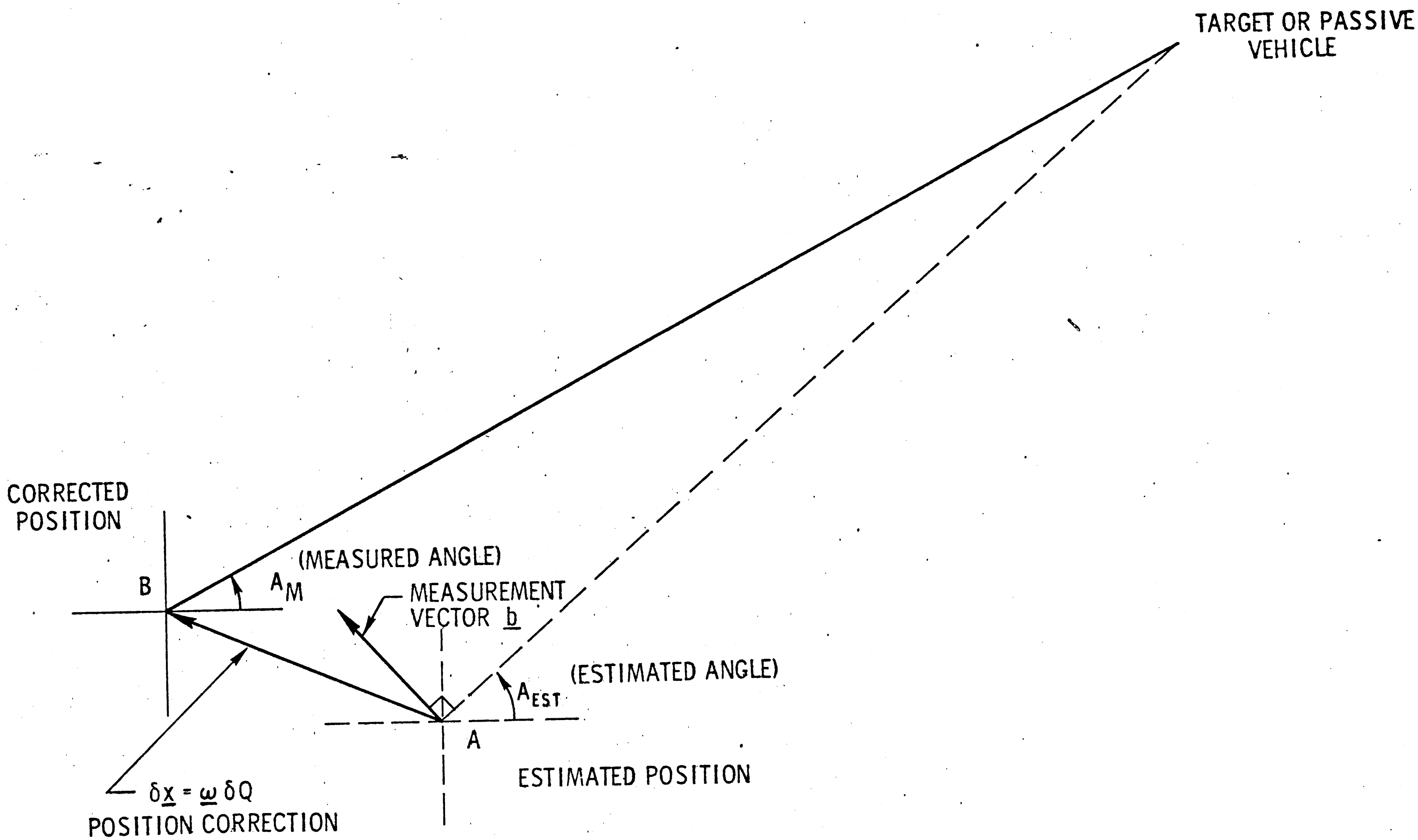
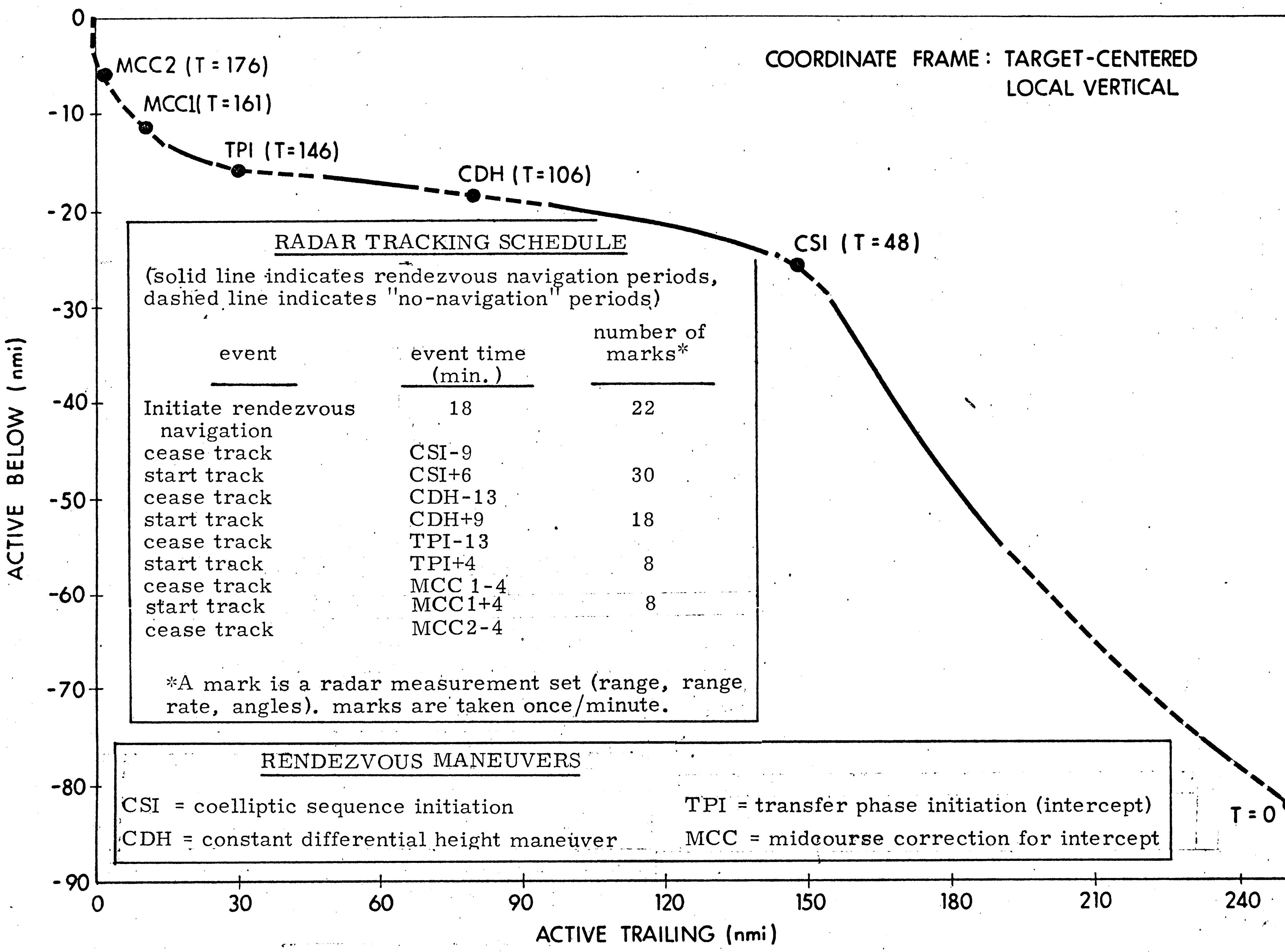


Figure A. 2-4 Simplified Rendezvous-Navigation Angle Measurement Incorporation practical. As a result, the accuracy of the error-transition matrix used in the weighting vector computation of Fig. A.2-3 degrades, resulting in erroneous correlation information after extended tracking periods. The filter matrix is, therefore, periodically reinitialized during the rendezvous-navigation tracking phases. It has been determined in cislunar-navigation simulations, however, that a single W-matrix initialization at the start of the sighting schedule provides sufficient accuracy.

Figure A.2-5 represents a relative-trajectory profile for the nominal lunar-landing-mission rendezvous phase. This trajectory is the same as that shown in Fig. A.2-1, except that the coordinates (0,0) are centered on the passive CM vehicle. The solid-line portions of the trajectory in Fig. A.2-5 represent the rendezvous-navigation phases where tracking data are taken at one-minute intervals in the LM GN&CS. The CM takes similar, if not slightly extended, tracking intervals, but both vehicles suspend navigation prior to major trajectory-correction maneuvers in order to prepare for targeting and execution of these maneuvers. The CM GN&CS normally computes a mirror-image maneuver of that computed by the LM so that it can execute a retrieval, should the LM fail to complete the maneuver.

During the rendezvous phases of an Apollo mission, three active navigation systems normally operate during the entire rendezvous profile. These are the LM, CM, and earth-tracking navigation systems. During the later phases of the rendezvous profile (TPI maneuver preparation to intercept), two additional navigation monitors are active: the LM Abort Guidance System, and crew observations checked against precomputed "chart" solutions for the major rendezvous maneuvers. A measure of the consistency of the three major navigation systems during rendezvous can be gauged by comparing the rendezvous maneuvers computed by each of these systems, since their computations are based upon the navigated state vectors established independently by each system using different tracking sensors. Post-flight analysis of the lunar-rendezvous phases of the Apollo 10 and 11 missions show that there was very close agreement in all cases where comparisons can be made, indicating a high degree of accuracy for all three rendezvous-navigation systems and concepts. The flight experience provided by the five Apollo rendezvous missions to date indicates that the rendezvous navigation concept used in the spacecraft GN&C Systems is highly accurate and versatile in its capability to use a variety of types of navigation measurements.



RADAR TRACKING SCHEDULE

(solid line indicates rendezvous navigation periods, dashed line indicates "no-navigation" periods)

event	event time (min.)	number of marks*
Initiate rendezvous navigation	18	22
cease track	CSI-9	
start track	CSI+6	30
cease track	CDH-13	
start track	CDH+9	18
cease track	TPI-13	
start track	TPI+4	8
cease track	MCC 1-4	
start track	MCC1+4	8
cease track	MCC2-4	

*A mark is a radar measurement set (range, range rate, angles). marks are taken once/minute.

RENDEZVOUS MANEUVERS

CSI = coelliptic sequence initiation

CDH = constant differential height maneuver

TPI = transfer phase initiation (intercept)

MCC = midcourse correction for intercept

Figure A. 2-5 Typical Relative Motion Plot

A.3 Orbital Navigation

As mentioned earlier, the basic objective of all the coasting-flight navigation routines is to maintain estimates of the position and velocity vectors of both the CSM and LM. Coasting-flight navigation achieves this goal by extrapolating the six-dimensional state vector, using a Coasting Integration Routine; and by updating or modifying this estimate with tracking data gathered by the recursive method of navigation (see Section A.1).

As with cislunar and rendezvous navigation, the basic input to the orbital-navigation routine is scanning telescope or sextant tracking-angle data indicated to the computer when the astronaut depresses the MARK button signifying that he has centered the optical reticle on the tracking target—which in orbital navigation is a landmark. The primary output of the orbital navigation routine is the estimated CSM state vector and estimated landmark coordinates. A simplified orbital-navigation functional diagram is shown in Fig. A.3-1 as similar to Figs. A.1-4 and A.2-3 for cislunar and rendezvous navigation. The navigation procedure involves computing an estimated tracking measurement, A_{EST} based on the current state-vector estimates. This estimated measurement is then compared with the actual tracking measurement, A_M, to form a measured deviation δQ. A statistical weighting vector, $\underline{\omega}$, is computed from statistical knowledge of state-vector uncertainties and tracking performance, $\overline{\alpha^2}$, plus a geometry vector, $\underline{b}$, determined by the type of measurement being made. The weighting vector, $\underline{\omega}$, is defined such that a statistically-optimum linear estimate of the deviation, $\delta\underline{x}$, from the estimated state-vector is obtained when the weighting vector is multiplied by the measured deviation δQ. The vectors $\underline{\omega}$, $\underline{b}$ and $\delta\underline{x}$ are of nine dimensions for orbital navigation.

To prevent unacceptably large incorrect state-vector changes, certain validity tests are included in the navigation procedure; the astronaut tracks a landmark and acquires a number of sets of optical angle data before the state-vector updating process begins. During the data-processing procedure the landmark is out of sight, and it is not possible to repeat the tracking. Before the first set of data is used to update the estimated state vector, the magnitudes of the proposed changes in the estimated CSM position and velocity vectors, δr and δv, respectively, are displayed for astronaut approval. In general, successive accepted values of δr and δv decrease during the processing of the tracking data associated with one landmark. Thus, if

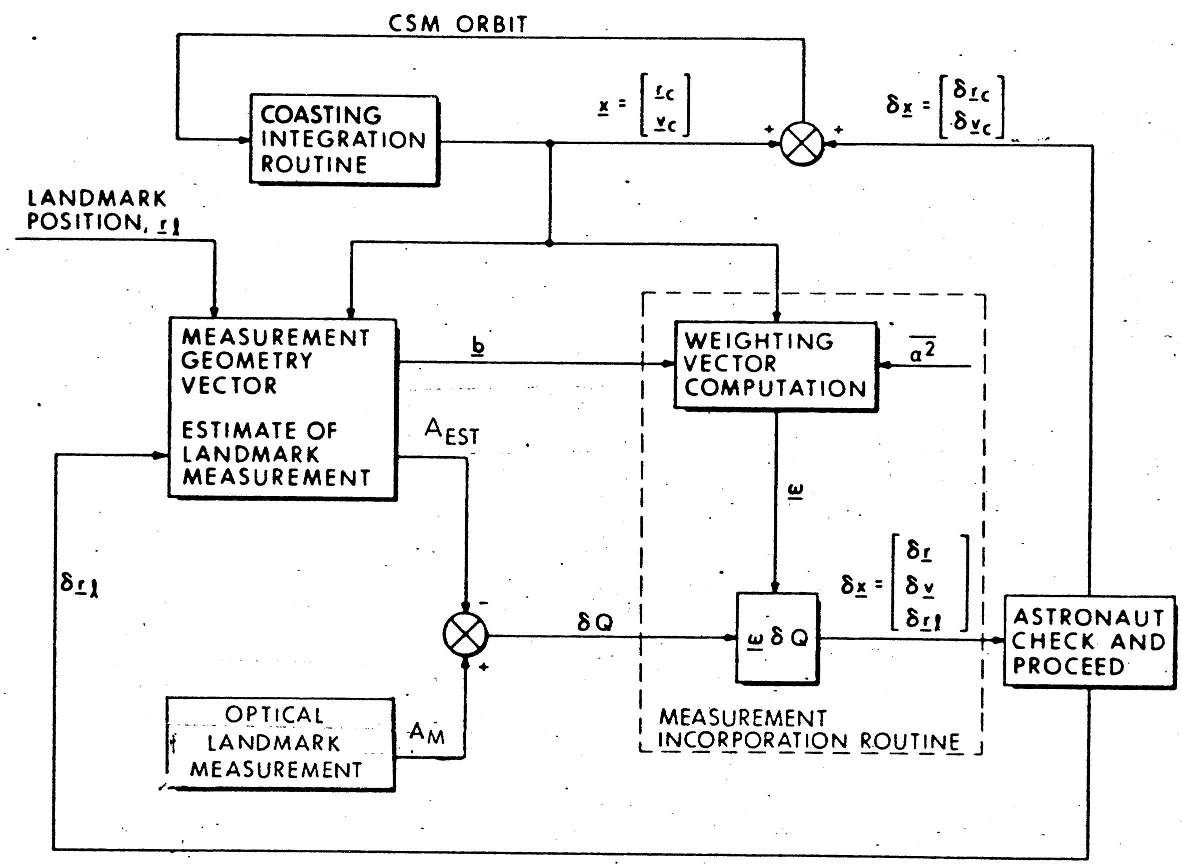


Figure A. 3-1 Simplified Functional Diagram of Orbital Navigation the MARK REJECT button has been used to erase all inaccurate marks, all state-vector updates should be either accepted or rejected. If the first displayed values of δr and δv are judged to be valid, all data associated with that landmark are accepted.

The orbit navigation routine can be used in lunar orbit in lunar-landing missions and in earth orbit during abort situations or alternate missions. This routine has further extensive onboard-navigation and landmark-mapping capabilities, but these are not yet being used in the fashion originally intended, since optical marks are not processed onboard. Consequently, as in cislunar navigation, the primary mode of navigation for the Apollo orbital phases is the Manned Space Flight Network and its Real Time Computation Center in Houston.

Procedures that ensure proper landmark acquisition and marktaking are a precondition to successful landmark navigation. To initially acquire and maintain optical tracking, the CSM must be oriented such that the CSM-to-landmark line-of-sight falls within the scanning telescope's field of view. In the CSM GN&CS there is no automatic vehicle-attitude control during the landmark-tracking procedure. Consequently, any desired attitude control must be accomplished manually by the astronaut using the Rotational Hand Controller or Minimum Impulse Controller or by use of the Barbecue Mode Routine.

Should the astronaut wish, he may use the Automatic Optics Positioning Routine to aid in the acquisition of the landmark. This routine has two modes which are relevant to orbit navigation. In the landmark mode (which is useful for acquisition of a specified landmark), the routine drives the optics to the estimated direction of the specified landmark. The computations and positioning commands in this routine are repeated periodically provided the optics mode switch is properly set. In the advanced ground-track mode (which is useful in lunar orbit for surveillance, selection, and tracking of possible landing sites), the routine drives the CSM optics to the direction of the point on the ground track of the spacecraft at a time slightly more than a specified number of orbital revolutions ahead of current time. Thus, in the advanced ground-track mode, the astronaut is shown continuously the ground track of the CSM for a future revolution. The basis for this mode is that it is desirable to select a landing site near the CSM orbital plane at the LM lunar-landing time.

After the astronaut has acquired the desired landmark (not necessarily the one specified to the Automatic Optics Positioning Routine), he switches the optics mode to MANUAL and centers the scanning telescope or sextant reticle on the landmark. When accurate tracking is achieved, he presses the optics MARK button, causing the time of the measurement and all optics and IMU gimbal angles to be stored in the AGC. Up to five unrejected navigation sightings of the same landmark may be made during the tracking interval, and all sets of navigation data must be acquired before processing of the data begins.

After the astronaut has completed the tracking of a landmark, he is asked by the computer whether he wishes to identify the tracked landmark. If he does, he then enters into the AGC through the keyboard, the coordinates of the landmark.

Should the astronaut not identify the landmark, the Landing Site Designation procedure is then used for the navigation-data processing. In this process the landmark is considered to be unknown, and the first set of navigation data is used to compute an initial estimate of the landmark location. The remaining sets of data are then processed to update the estimated nine-dimensional CSM-landmark state vector.

Whether the landmark is identified or not, one further option is available to the astronaut. He may specify that one of the navigation sightings is to be considered the designator for an offset landing site near the tracked landmark. In this case, the designated navigation data set is saved, the remaining sets of data are processed as described above, and then the estimated offset landing-site location is determined from the saved data. This procedure offers the possibility of designating a landing site in a flat area of the moon near a landmark suitable for optical navigation tracking, but not for landing.

Each set of navigation data used for state-vector updating and not for landing-site designation or offset produces two updates. For the first navigation-data set, the magnitudes of the first proposed changes in the estimated CSM position and velocity vectors, δr and δv, respectively, are displayed for astronaut approval. If the astronaut accepts these proposed changes, then all state-vector updates will be performed, and all the information obtained during the tracking of this landmark will be incorporated into the state-vector estimates.

After all of the sets of navigation data have been processed, the final estimated landmark-position vector is converted to latitude, longitude, altitude coordinates displayed on the DSKY. Should the astronaut designate the tracked landmark to be the landing site, then the landing-site coordinates and landing-site vector are saved in erasable memory. In this manner, the original coordinates of the landing site can be revised or a new landing site selected.

Figure A.3-2 shows the geometry for tracking a landmark in a 60-nmi circular lunar orbit. Recommended marktaking technique requires that five marks be taken equally spaced over the plus-55-to-minus-55-deg marktaking window. The advantage of oblique lines-of-sight on the first and last marks diminishes rapidly beyond ± 45 deg. Consequently, marks taken symmetrically and at equally spaced intervals are preferred to marks taken asymmetrically at the extremes of the marktaking window. The interval between marks for the 76-deg (100-sec) minimum marktaking window is 19 deg (25 sec); for the 110-deg (180-sec) maximum window, the interval between marks is 27.5 deg (45 sec).

The final operation is to convert the nine-dimensional error-transition matrix to a six-dimensional matrix with the same CSM position and velocity estimation error variances and covariances. The reason for this procedure is that the nine-dimensional matrix, when it is initialized for processing the data associated with the next landmark, must reflect the fact that the initial landmark-location errors are not correlated with the errors in the estimated CSM position and velocity vectors. Of course, after processing measurement data, these cross correlations become non-zero, and it is for this reason that the nine-dimensional procedure works, and that it is necessary to convert it finally to a six-dimensional form.

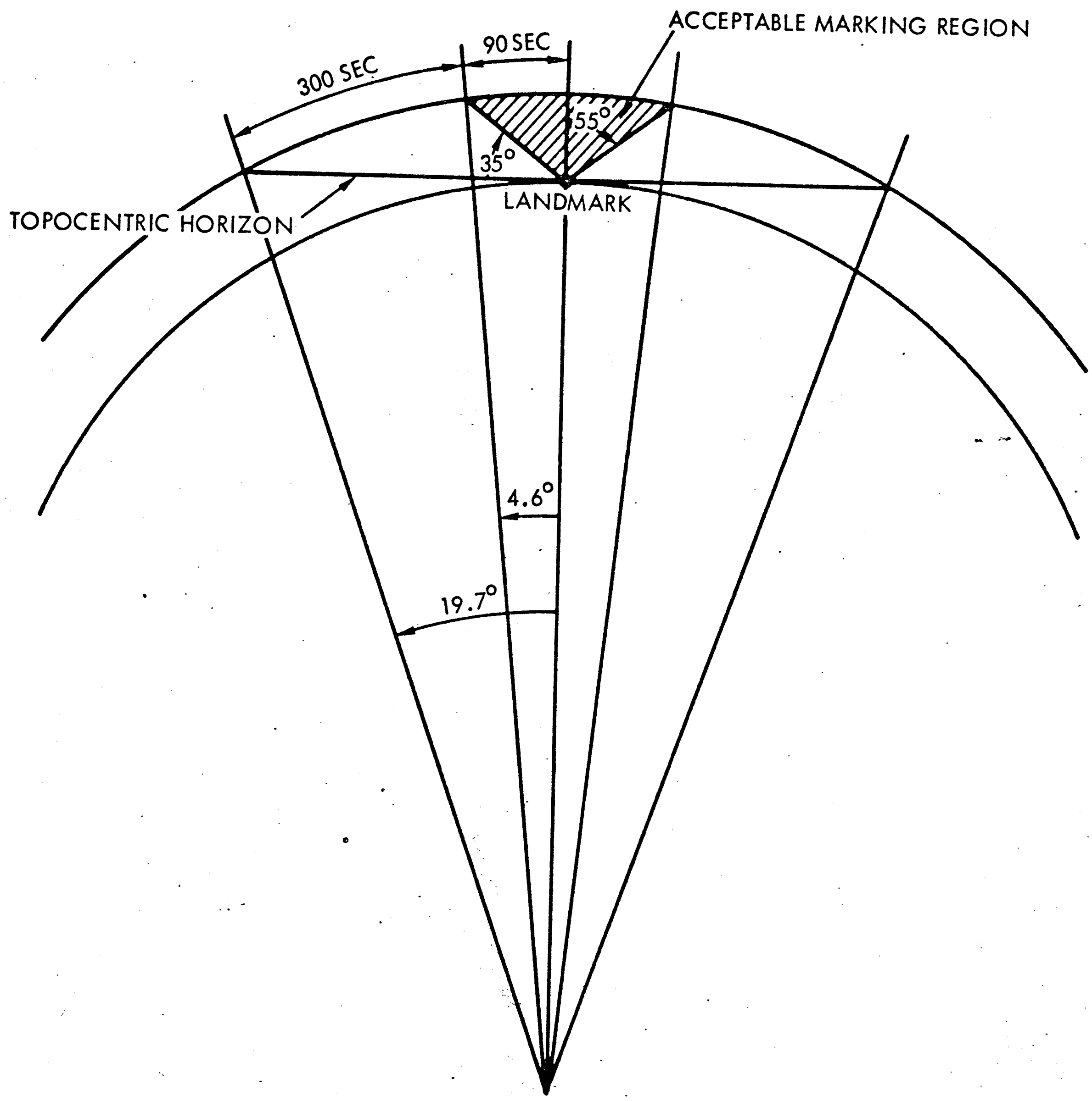


Figure A. 3-2 Landmark-Tracking Geometry for a 60-Nautical-Mile Circular Lunar Orbit

APPENDIX B

MAJOR PROGRAM CAPABILITIES—
Targeting

As stated in Section 2.2.1, targeting is the computation of the maneuver required to continue on to the next step in the mission. Specifically, targeting computes the velocity change required of the spacecraft to obtain a certain objective, such as a change of orbit or a certain reentry corridor or an aimpoint. An aimpoint can itself be quite variable; it can be a point on the surface of the moon or a point in space where another vehicle will be at a specified time in the future or a location below and behind that vehicle at that same projected time.

The AGC does not possess a targeting capability for every phase of the lunar landing mission; consequently, the Real Time Computation Center (RTCC) in Houston provides the targeting for many of the nominal and abort phases of the lunar mission. There are, in fact, five classes of maneuvers, four of which involve targeting:

1. Pretargeted maneuvers comprise earth-orbit insertion, translunar injection, lunar landing, lunar ascent, and reentry.
2. Ground-targeted maneuvers comprise lunar-orbit insertion, transearth injection, descent-orbit insertion, various orbital changes around the moon, translunar and transearth midcourse corrections and return-to-earth aborts.
3. Rendezvous maneuvers comprise coelliptic-sequence initiation (CSI), constant differential height (CDH), transfer-phase initiation (TPI), transfer-phase midcourse (TPM), and out-of-plane maneuvers.
4. Return-to-earth (RTE) maneuvers comprise cislunar aborts which might occur after loss of communication with the ground.
5. Untargeted maneuvers comprise docking, passive thermal control, crew-originated attitude maneuvers, etc.

Clearly, untargeted maneuvers need not be discussed here. Pretargeted maneuvers have unchanging objectives which are included within the actual program computations. Ground-targeted, rendezvous and return-to-earth maneuvers may have varying objectives which may not be anticipated beforehand; consequently, onboard targeting programs permit considerable flexibility in the prevailing conditions and objectives when they are implemented. It is the latter targeting programs which will be discussed in this appendix, as well as the targeting computations upon which these programs are based.

B.1 Targeting Computations

Targeting programs are classified by the type of maneuver targeted (either External ΔV or Lambert) or by the method of computation (iterative or noniterative). External ΔV is an open-loop, constant-attitude maneuver which permits easy out-of-the-window monitoring. To date, all ground-targeted maneuvers have used External ΔV. The principal disadvantage of External ΔV is that it is open loop with respect to the targeted conditions (required velocity). Any variations from the RTCC-assumed models for thrust and mass flow during the burn can result in a trajectory which could require a further trimming maneuver—and hence cause a propellant penalty. Computation of required velocity for a generalized External-ΔV maneuver (External-ΔV targeting, as opposed to External-ΔV guidance) is extremely complicated, effectively precluding an onboard AGC External-ΔV targeting capability.

Lambert maneuvers, however, are closed-loop with respect to the targeted conditions, in that they periodically update required velocity, a function of present and targeted state. Thus the effects of non-nominal thrust and flow rate are minimized. A disadvantage of Lambert targeting is that it lends itself to intercept problems, as opposed to trajectory-shaping problems. As a result, Lambert targeting accommodates only a small proportion of the maneuvers required in an Apollo mission.

Several targeting techniques are used in Apollo—some of which are External-ΔV or Lambert and all of which employ External-ΔV or Lambert guidance. The onboard return-to-earth targeting program (P37) produces a conic solution which is utilized by Lambert guidance. CSI and CDH targeting prepare inputs for External-ΔV guidance. TPI and TPM are Lambert problems and utilize Lambert targeting to generate Lambert-guidance inputs directly. Ground-targeted maneuvers use whichever targeting techniques will accomplish the current goal and generate inputs to External-ΔV guidance.

The choice between an iterative or noniterative method of computation depends, generally, on the extent to which perturbations affect the solution. Since no analytic expression completely describes the forces acting upon a vehicle traveling between the earth and the moon, targeting of such a trajectory involves first an analytic approximation, then orbital integration to determine the error, a second approximation to compensate for the error, and so on, bracketing the solution until either an imposed iteration limit is reached or the approximation converges on the desired solution.

The accuracy of any rendezvous computation depends upon a good knowledge of the state vectors of the two vehicles with respect to each other. Since coelliptic-sequence initiation is performed after injection or abort, the initial estimate of the LM state vector could be quite poor. Normally, ample time is available for repeated rendezvous navigation to improve the probability of good state-vector estimates before the CSI maneuver. Even in the off-nominal case, there would be sufficient time to take a certain minimum number of marks to ensure a good rendezvous.

Average G (see Section C.1.1.1), which improves knowledge of the state vector during powered flight, tends also to slightly degrade the estimate of that vector due to accelerometer uncertainties; thus rendezvous navigation is needed repeatedly to ensure the high quality of the state vector.

B.2 Ground-Targeted Maneuvers

All ground-targeted maneuvers are transmitted to the AGC via voice or telemetry uplink. Sufficient data could be transmitted to permit immediate execution of a powered-flight program but, instead, an onboard pseudo-targeting buffer program (P30) is executed prior to the maneuver. This pseudo-targeting approach has several advantages over direct maneuver execution: it provides meaningful (perhaps critical) displays to the astronaut; it can itself generate many of the inputs required by the guidance program, permitting a significant reduction in the required number of uplink variables (especially important for voice uplinks which must be entered via the DSKY); and it is designed to accept conceptually simple inputs for a crew-originated maneuver in an emergency situation when ground communication is unavailable. Furthermore, this approach serves as a backup for the onboard rendezvous-targeting programs in the highly unlikely event that the onboard primary systems in both the CM and LM fail.

RTCC ground targeting considers and accounts for such predictable effects as nonimpulsive burns, mass and thrust variations, guidance/steering rotation and Lambert aimpoint. Such unpredictable effects as thrust dispersions from nominal, or cg or V_G misalignments are ignored.

The buffer program provides inputs only to External-ΔV guidance. A similar program was devised for Lambert guidance, but it was never used and has since been deleted.

B.3 Rendezvous Maneuvers

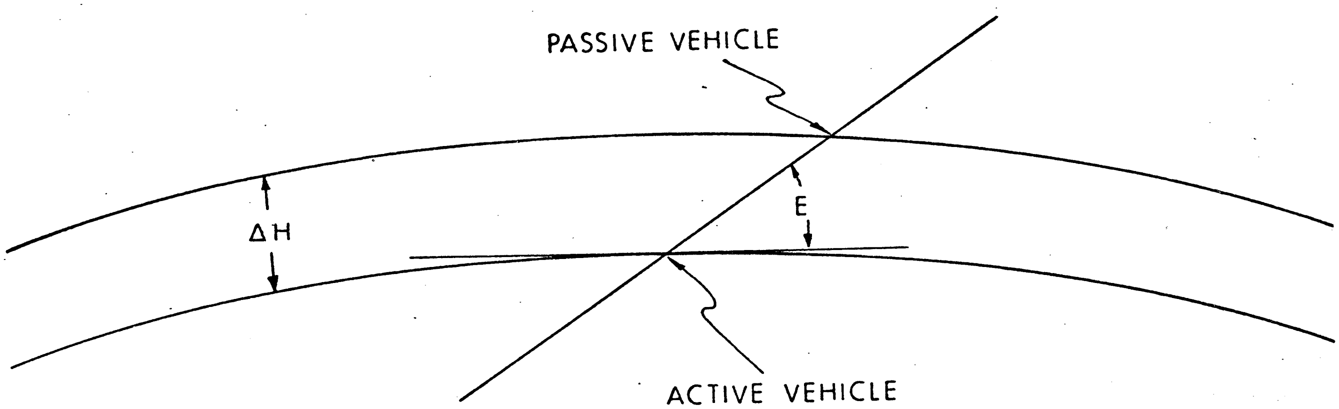
There are two basic profiles which achieve rendezvous: direct transfer to intercept, and rendezvous using intermediate parking orbits. The AGC has the means for targeting both of these maneuvers. Each method has its own significant advantages and disadvantages. The direct transfer is fast; however, maneuver magnitudes can be quite large, imposing a possible fuel penalty, high closing rates and nonstandard lighting and intercept conditions. On the other hand, parking-orbit rendezvous allows the final phase to be standardized (thus simplifying crew training), and permits smaller maneuver magnitudes (minimizing the effects of a poor maneuver); it is, however, long and drawn out—taking several hours.

Gemini used the parking-orbit rendezvous because its computer did not have a navigation filter, and the final approach was planned to allow for easy crew monitoring. Early NASA incredulity concerning the new Apollo system favored the smaller maneuver magnitudes of the parking-orbit rendezvous to minimize the effects of a single bad burn—should one occur. Later, after confidence in the Apollo system was established, the parking-orbit rendezvous retained its precedence principally because of the ease in training the crew for the standardized final approach.

Nominally, the LM is the active vehicle throughout the entire rendezvous sequence. During this time the CSM is computing the maneuvers it would perform if it were the active vehicle—these being mirror images of the LM maneuvers. Should the LM experience a major failure (e.g., in its propulsion system), the CSM is instantly prepared to begin retrieval.

Any parking-orbit rendezvous must eventually target a direct transfer to intercept; the difference is that a parking-orbit rendezvous establishes conditions for a standardized transfer. The onboard capability for parking-orbit rendezvous is known as the concentric flight plan (CFP) and usually involves two parking orbits. Figure B.3-1 illustrates the concentric flight plan.

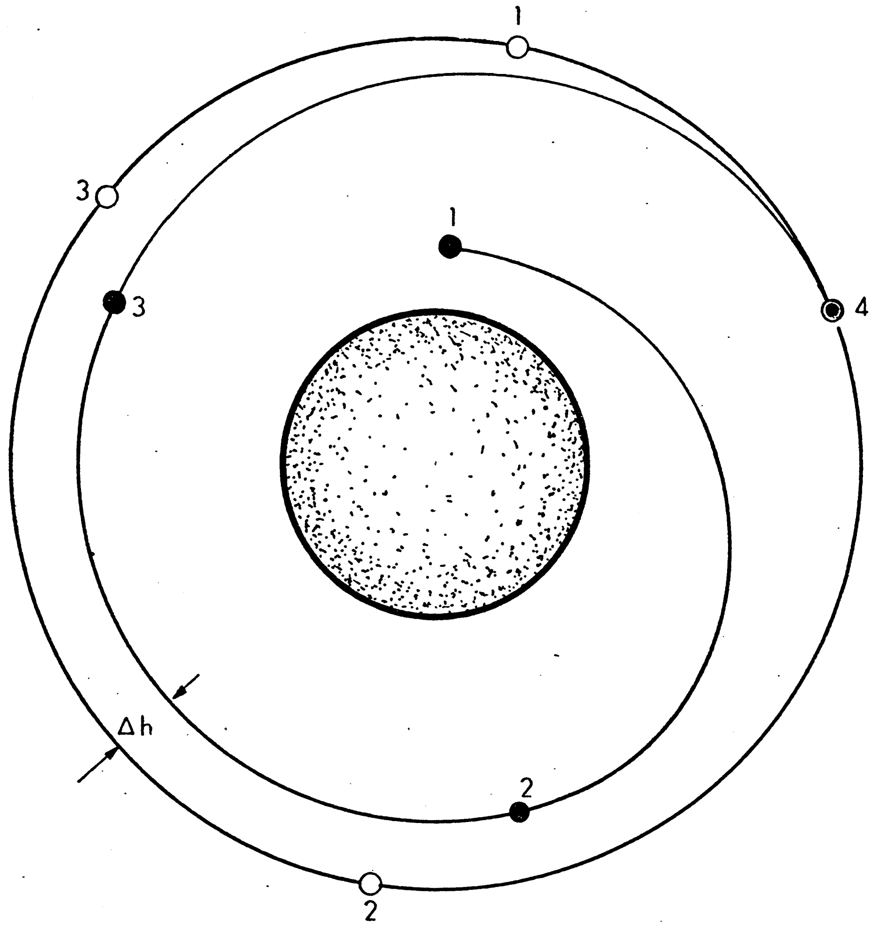
The first two maneuvers of the CFP are designed to place the active vehicle in an orbit that is coelliptic or concentric with the orbit of the passive vehicle (there is a constant altitude differential) and has a certain phase-angle/altitude-differential relationship defined by the elevation angle (E) at a specified time. These maneuvers are called the coelliptic-sequence initiation (P32) and constant differential height (P33). Coelliptic orbits, together with the proper elevation angle, produce



such desirable conditions as slow closing rates, easy astronaut takeover in the event of computer malfunction, and easy discovery of errors by monitoring elevation-angle changes.

Transfer to intercept, whether or not done in the context of parking-orbit rendezvous, involves simply planning the time and place of intercept and aiming to hit the spot at the proper time.

The third maneuver of the CFP is called transfer-phase initiation (P34), which targets intercept trajectories from the CDH parking orbit. Transfer-phase midcourse (P35) targets midcourse corrections (MCCs) to such trajectories.



● — ACTIVE VEHICLE

○ — PASSIVE VEHICLE

Concentric Flight Plan Events

1. CSI Maneuver
2. CDH Maneuver
3. TPI Maneuver
4. Intercept

(Δh = differential altitude following CDH)

Figure B. 3-1 Concentric Flight Plan

Prior to actual intercept, a sequence of manual braking maneuvers is initiated to prevent collision and provide a proper attitude and rate of closure for docking.

All of these rendezvous programs can select either the LM or the CSM as the active vehicle and can be flown in either earth or lunar orbit.

B.3.1 Coelliptic Sequence Initiation (CSI) and Constant Differential Height (CDH)

The CSI computation iteratively solves a piecewise-continuous boundary-value problem which can be described as follows: given the current state vectors of both vehicles at CSI-ignition time $\left[t_{IG}(\text{CSI})\right]$, what maneuver is required of the LM (assuming it is the active vehicle) to arrive at a point, TPI, at a specified time in the future, $t_{IG}(\text{TPI})$*, and with an elevation angle of 27 deg** from the CSM?

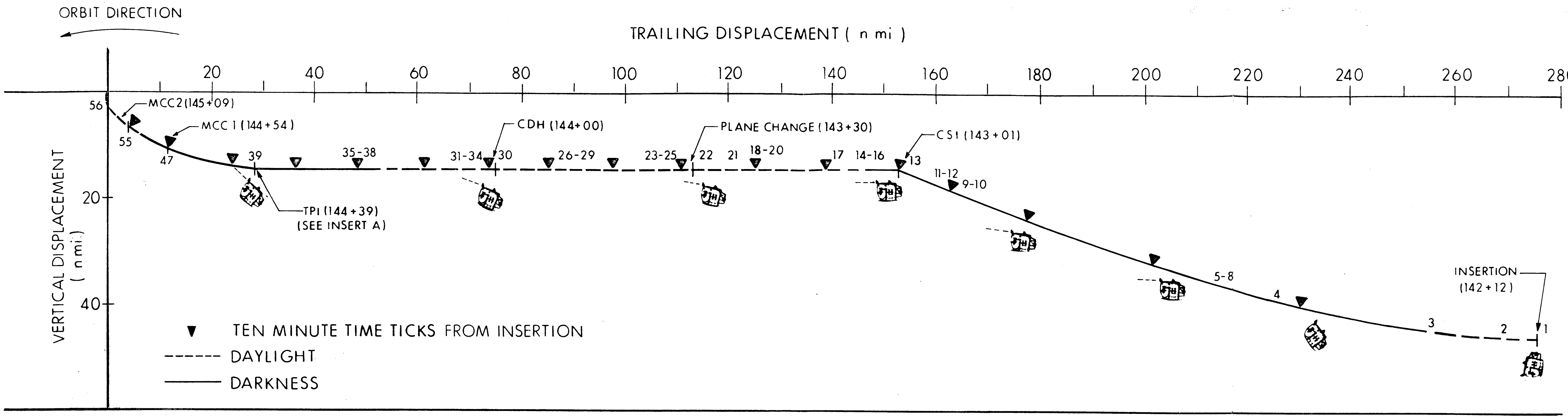
Nominally, the ascent maneuver from the lunar surface targets the Lunar Module into a 9-by-45-mile parking orbit and the CSM is in a 60-mile circular orbit; consequently, the CSI maneuver done near apogee essentially coelliptisizes the parking orbit at about a 15-mile differential altitude and the CDH maneuver will be very small. Figure B.3-2 graphically illustrates a typical LM vertical displacement from the CSM as a function of its trailing distance. Since, during rendezvous, the LM is at a lower altitude than the CSM, it will be moving at a higher orbital velocity and will be constantly catching up with the CSM. On this graph the CSI maneuver occurs at about apogee; thus, as predicted, the CDH maneuver is very small. On this figure the acronym RR refers to rendezvous radar; AGS refers to the Abort Guidance System, which serves as a backup to the GN&CS.

The CSI computation yields two solutions—one each for CSI and CDH. In actual practice the CSI maneuver is burned from this solution, but usually the CDH maneuver ΔV is merely displayed.

---

*From $t_{IG}(\text{TPI})$ to intercept, the vehicles will travel about 130 deg relative to the center of the moon (central angle of travel, also known as angle of transfer); consequently, $t_{IG}(\text{TPI})$ is planned to occur at the midpoint of darkness to ensure good lighting conditions for docking.

**The 27-deg elevation angle is a backup requirement. In the event of a computer failure at TPI, this elevation angle permits a relatively simple manual maneuver to replace the automatic TPI. This manual maneuver is performed by aligning the LM'S thrust vector along the line-of-sight to the CSM.



LM MAJOR EVENTS

	GROUND ELAPSED TIME	EVENT
1.	142+12	INSERTION
2.	142+13	AGS TARGET CSI
3.	142+16	IMU REFSMMAT ALIGN (P52)
4.	142+25	INITIALIZE W-MATRIX (10000, 10, 15)
5.	142+26	INITIATE RR NAVIGATION (P20)
6.	142+26	GN&CS TARGET CSI (P32)
7.	142+27	AGS UPDATE AND ALIGN
8.	142+28	INITIATE AGS RADAR UPDATES
9.	142+49	TERMINATE RR NAVIGATION (23 MARKS)
10.	142+49	GN&CS FINAL CSI COMP
11.	142+52	TERMINATE AGS RADAR UPDATES (9 UPDATES)
12.	142+53	AGS FINAL CSI COMP
13.	143+01	CSI THRUSTING MANEUVER (P41)
14.	143+03	INITIALIZE W-MATRIX (2000, 2, 5)
15.	143+04	RESUME RR NAVIGATION
16.	143+04	TARGET CDH (GN&CS P33 AND AGS)
17.	143+07	RESUME AGS RADAR UPDATES
18.	143+21	TERMINATE RR NAVIGATION (17 MARKS)
19.	143+22	OBTAIN CSM RDOT FOR PLANE CHANGE
20.	143+22	EXTERNAL ΔV FOR PLANE CHANGE (P30)
21.	143+24	TERMINATE AGS RADAR UPDATES (7 UPDATES)
22.	143+30	RCS PLANE CHANGE MANEUVER (P41)
23.	143+31	REINITIALIZE W-MATRIX (V93)
24.	143+32	RESUME RR NAVIGATION
25.	143+32	TARGET CDH (GN&CS P33 AND AGS)
26.	143+48	TERMINATE RR NAVIGATION (15 MARKS)
27.	143+48	GN&CS FINAL CDH COMP
28.	143+51	TERMINATE AGS RADAR UPDATES (7 UPDATES)
29.	143+52	AGS FINAL CDH COMP
30.	144+00	CDH THRUSTING MANEUVER (P41)
31.	144+02	REINITIALIZE W-MATRIX (V93)
32.	144+03	RESUME RR NAVIGATION
33.	144+03	TARGET TPI (GN&CS P34 AND AGS)
34.	144+05	RESUME AGS RADAR UPDATES
35.	144+27	TERMINATE RR NAVIGATION (24 MARKS)
36.	144+27	GN&CS FINAL TPI COMP
37.	144+30	TERMINATE AGS RADAR UPDATES (9 UPDATES)
38.	144+31	AGS FINAL TPI SEARCH COMP
39.	144+39	TPI THRUSTING MANEUVER (P41)
40.	144+41	REINITIALIZE W-MATRIX (V93)
41.	144+41	RESUME RR NAVIGATION
42.	144+41	TARGET MCC1 (GN&CS P35 AND AGS)
43.	144+41	RESUME AGS RADAR UPDATES
44.	144+51	TERMINATE RR NAVIGATION (9 MARKS)
45.	144+51	TERMINATE AGS RADAR UPDATES (6 UPDATES)
46.	144+51	GN&CS AND AGS FINAL MCC1 COMP
47.	144+54	MCC1 THRUSTING MANEUVER (P41)
48.	144+56	REINITIALIZE W-MATRIX (V93)
49.	144+56	RESUME RR NAVIGATION
50.	144+56	TARGET MCC2 (GN&CS P35 AND AGS)
51.	144+56	RESUME AGS RADAR UPDATES
52.	145+06	TERMINATE RR NAVIGATION (9 MARKS)
53.	145+06	TERMINATE AGS RADAR UPDATES (6 UPDATES)
54.	145+06	GN&CS AND AGS FINAL MCC2 COMP
55.	145+09	MCC2 THRUSTING MANEUVER (P41)
56.	145+12	BRAKING (P47)

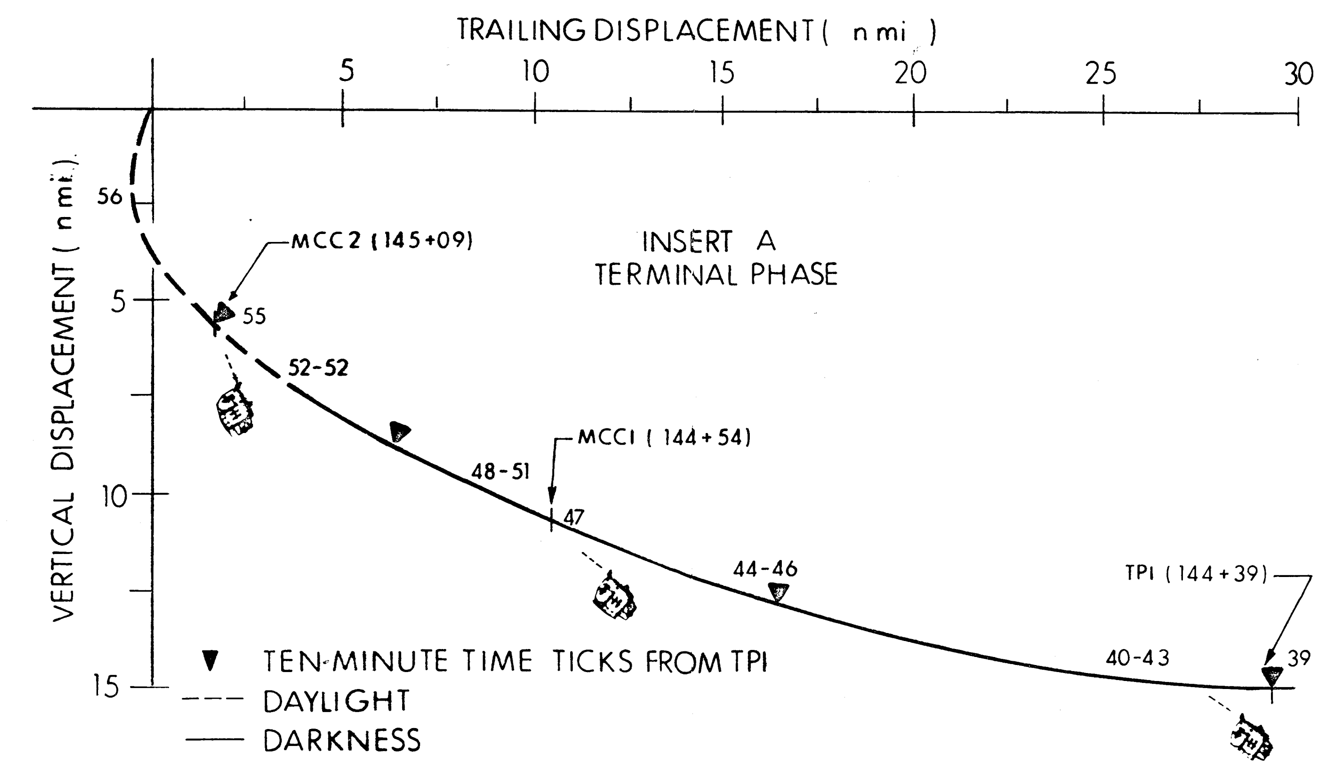


Figure B. 3-2 Typical Mission Rendezvous: CSM-Centered Motion

State-vector errors tend to propagate during the nontracking period of an orbital maneuver. To ensure the best possible state vectors for inputs to the CDH computation, further tracking is performed after the CSI burn; then the CDH maneuver is recomputed. However, if for some reason (such as loss of visibility) tracking cannot be performed between the two maneuvers, the astronaut will still call for the CDH maneuver—just as he would in the nominal case—but with degraded accuracy.

Other than not achieving the desired boundary conditions, there are a few other situations which could render unacceptable a CSI solution. One is when the various burns occur too closely together to permit tracking in between; another is when the LM is phased improperly with the CSM; still another is a too-large ΔV or a too-small ΔH. DSKY alarms notify the astronaut should one of these situations occur; he can then adjust his inputs to the CSI program until he achieves a satisfactory solution. Typically, the CSI and CDH burns are 180 deg apart, with a plane-change maneuver halfway in between to make the orbits coplanar.

The inputs the astronaut can adjust are t_{IG}(TPI), CDH apsidal crossing and the standard TPI line-of-sight elevation angle, E. He would not adjust the vehicles' state vectors, which are also inputs. The astronaut carries onboard charts which give him nominal as well as non-nominal inputs to the CSI computation. Although the onboard rendezvous-targeting computations are prime, the astronaut can always check his inputs with the CSM or the ground if he chooses.

When the CDH burn is completed, the elevation angle is small (e.g., 9 deg), the LM is catching up with the CSM and tracking is resumed. About 40 minutes later the elevation angle is 27 deg, indicating TPI time.

B.3.2 Transfer Phase Initiation (TPI) and Transfer Phase Midcourse (TPM)

If the CSI and CDH maneuvers successfully produce the desired boundary conditions at TPI time, the transfer phase will possess several desirable properties, such as a slow closing rate; predictable changes in the elevation angle, E; nearly-coincident TPI-thrust direction with the line-of-sight; and a $|\Delta V|$, measured in feet per second, which is nearly twice the $|\Delta H|$ measured in miles. These properties allow easy monitoring and easy astronaut takeover should the computer malfunction during or after the TPI maneuver.

The TPI computation solves the Lambert (intercept) problem directly. It solves for the maneuver required of the LM to intercept the Command Module at a given angle of transfer* from t_{IG}(TPI). In earth orbit, this Lambert solution uses offset aimpoints to compensate for the effects of earth oblateness.

Typically, the Lambert problem is computed two more times after the TPI burn to make transfer-phase midcourse corrections. Of course, time is allowed between all of these burns for further tracking.**

After the second midcourse correction (about 4 miles from the CSM), the astronaut begins a manual braking schedule during which he maintains a collision course and brakes his vehicle. Finally, the LM comes up in front of the CSM while pitching over, so that each docking hatch faces the other.

The GN&CS rendezvous function ends with the initiation of the manual braking schedule.

---

*Typically, 130 deg of CSM travel, or a transfer time of about 43 min.

**Historically, two other rendezvous routines were planned and subsequently discarded—TPI Search, and stable orbit rendezvous (SOR).

The TPI Search was intended to provide a fast return in an abort situation. It could be very costly in fuel consumption, but that compromise was deemed necessary, for example, in the case of a partial failure of the life-support systems. The program allowed the astronaut to manually iterate central angles of travel to find an early-rendezvous trajectory which would minimize fuel consumption. After the initial burn, the problem would be solved again to provide MCCs, as in the procedure used for the standard TPI.

TPI Search doesn't provide the easy monitoring and standard final transfer that the concentric flight plan does; it could be fuel costly but it didn't have to be, because there were fewer burns; and it could have large closing rates necessitating a busy braking schedule. As NASA confidence in the entire system grew, this type of contingency maneuver was discarded.

The SOR targeted an intercept with the CSM's orbit at a fixed distance behind or ahead of it. This aimpoint was used in a Lambert calculation to determine the necessary maneuver. After the SOR burn, navigation and midcourse corrections could further improve the accuracy of the maneuver. At the point where the two orbits cross, another maneuver would have been performed to match the two vehicles' orbital velocities. Needless to say, none of the burns mentioned here are impulsive. Final closure and docking would be accomplished manually.

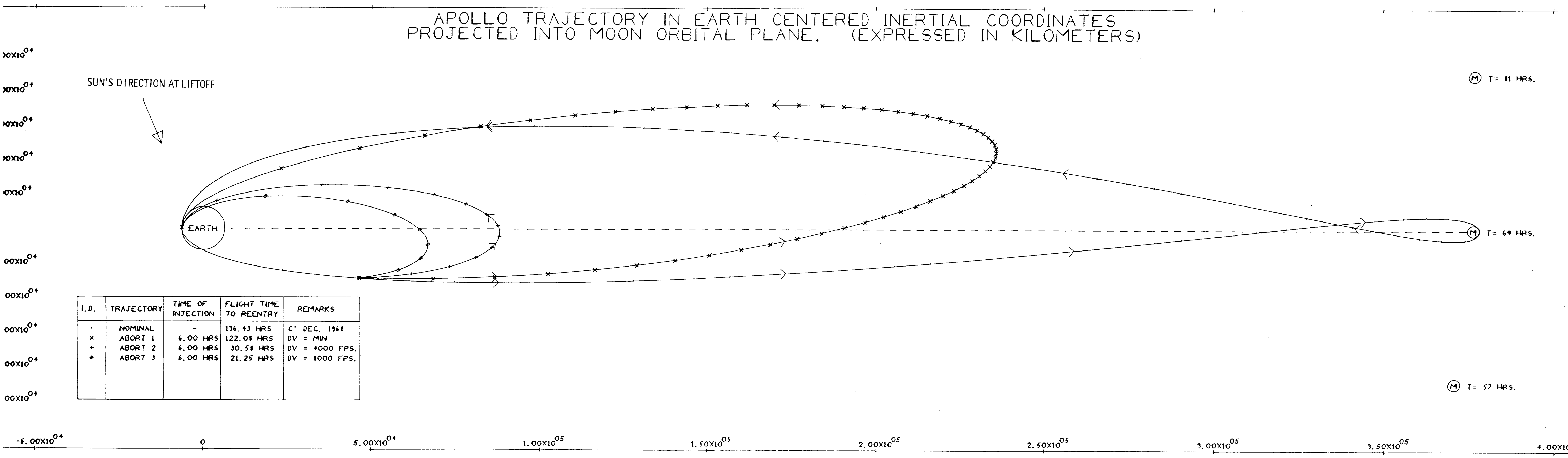
B.4 Return to Earth

The principal task of the return-to-earth program (P37) is to provide an onboard targeting capability to return the spacecraft to a proper earth-reentry corridor in the event an abort occurs while there is a loss of communications with the ground. With this program, safe returns can be achieved from earth orbit, from trajectories resulting from translunar-injection SIVB-powered maneuver failure, from translunar coast (outside the lunar sphere of influence), and from transearth coast, including midcourse corrections (also outside the lunar sphere of influence). Figures B.4-1, B.4-2, and B.4-3 indicate typical RTE trajectories at TLI+6 hours, TLI+20 hours and TLI+50 hours, respectively.

B.4.1 Options

P37 originally provided the crew with three basic options—minimum-fuel return, minimum-time return and landing-site designation. Limited computer capacity has since dictated the elimination of the landing-site designation option; nevertheless, a limited amount of designation is still possible, since ΔV directly affects the return time which, due to earth rotation, influences the landing site. Three of the inputs for either remaining option are desired ignition time (t_{IG}), desired velocity change (ΔV_D), and desired reentry angle. The choice between minimum-fuel and minimum-time return is determined by the input ΔV_D (zero for the minimum-fuel option). The program provides a display of the landing-site latitude and longitude, required velocity change, the spacecraft's velocity magnitude at 400,000-ft entry altitude measured above the Fischer ellipsoid, the flight-path angle at 400,000-ft entry altitude, and the transit time to that point from the time of ignition. After the astronaut selects which propulsion system to use, SPS or RCS, the following quantities are displayed: middle-gimbal angle at ignition, time-of-ignition and time-from-ignition.

Two principal steps are used to compute the return trajectory. First, a two-body conic solution is generated which satisfies the constraints dictated by the inputs, thus yielding a fast two-body approximation which is displayed to the astronaut. Should he so elect, he can continue on to a precision solution recomputed to consider gravitational perturbations. The conic characteristics are then used as a basis for the development of a precision trajectory within the original constraints. The



I.D.	TRAJECTORY	TIME OF INJECTION	FLIGHT TIME TO REENTRY	REMARKS
·	NOMINAL	-	136.43 HRS	C' DEC. 1968
×	ABORT 1	6.00 HRS	122.08 HRS	DV = MIN
+	ABORT 2	6.00 HRS	30.58 HRS	DV = 4000 FPS.
◆	ABORT 3	6.00 HRS	21.25 HRS	DV = 8000 FPS.

Figure B. 4-1 Typical Abort Trajectories for TLI+6 Hours

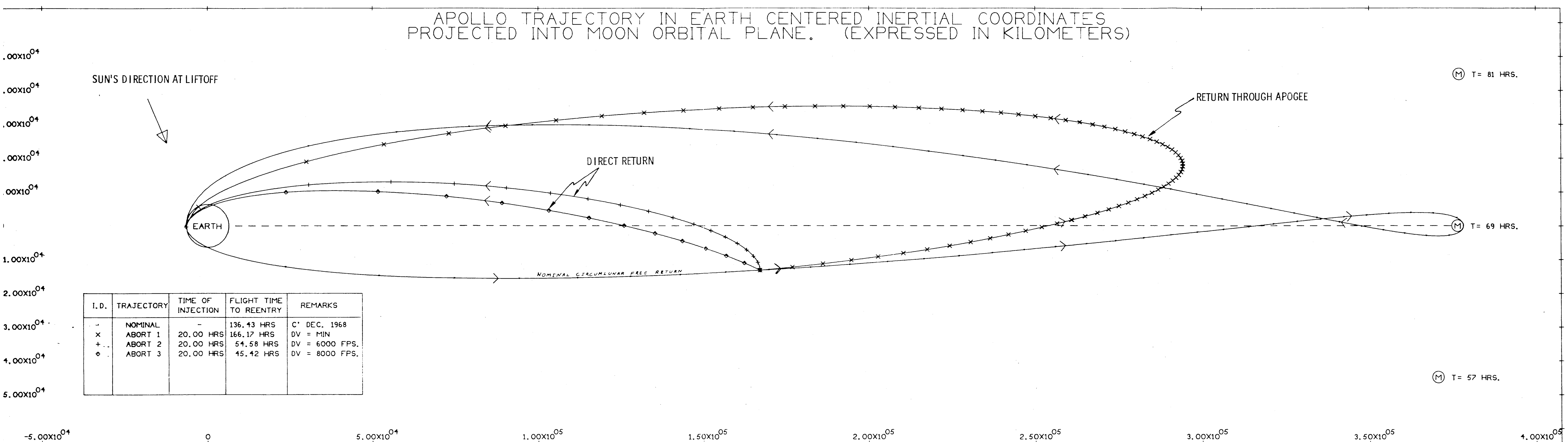


Figure B. 4-2 Typical Abort Trajectories for TLI+20 Hours

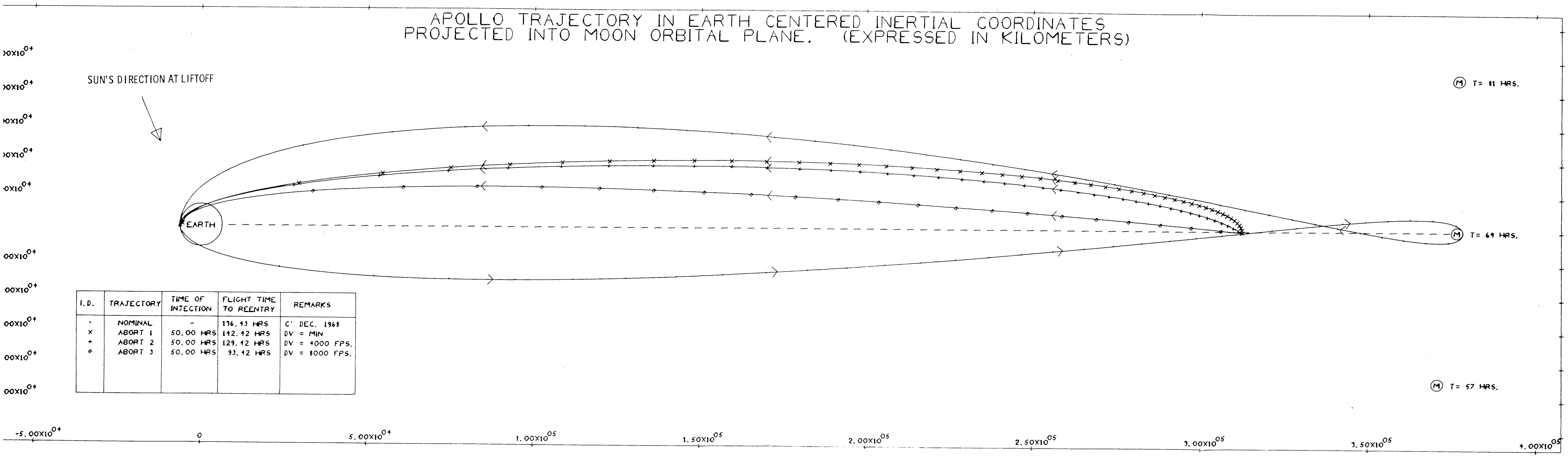


Figure B.4-3 Typical Abort Trajectories for TLI+50 Hours precision-trajectory displays can differ significantly from the conic displays, especially in the case of long transit-time returns. Thus the astronaut is obliged to check the new displays before accepting the precise solution.

B.4.2 Two-Body Problem

The fuel-critical two-body problem consists of generating a conic trajectory which returns the spacecraft to a specified reentry radius and flight-path angle under the constraint of minimizing the impulsive ΔV required to achieve the trajectory. The time-critical two-body problem is similar, except that the impulsive ΔV is prescribed rather than minimized. In either case, the premaneuver state vector is assumed to be known. Since reentry conditions do not constrain the trajectory plane, an in-plane maneuver is most efficient, thus reducing the problem to two dimensions.

During the program design, consideration was given to a closed-form solution to the minimum-fuel problem. This would have required solving for the roots of a general fourth-order polynomial. The logic necessary to eliminate imaginary and physically unrealizable roots made this approach unattractive. It was concluded that an iterative search, using the cotangent of the post-maneuver flight-path angle as an independent parameter, would be a better approach to the problem. The latter approach always converges on a physically realizable solution and greatly simplifies the equations. More important, the same equations can be used in the solution of the time-critical problem, and additional trajectory constraints can be readily included.

Computation of the two-body solutions to both the fuel-critical or time-critical problems is complicated by initial uncertainty in the reentry radius and flight-path angle. The reentry radius is defined to be 400,000 ft above the Fischer ellipsoid and therefore varies with reentry latitude. The flight-path angle is a function of the reentry speed and is represented by a polynomial curve fit. An initial rough estimate is made of both the radius and flight-path angle. Based on the resulting reentry latitude and speed, a better estimate of the proper radius and flight-path angle can then be made. This iteration is automatically continued until satisfactory terminal conditions are achieved. Then the landing site is computed, based on either a prestored entry range or the range resulting from a half-lift reentry profile.

The two-body solution does not reflect trajectory perturbations resulting from the earth's oblateness, the moon, or the finite duration of the maneuver. In most cases, however, it presents a good approximation to the maneuver size and comes within 10 to 15 deg of the true landing site. After the state vector is extrapolated to the ignition time, about 15 sec are required to compute the conic solution.

B.4.3 Precision Solution

A precision solution to the return-to-earth problem may be initiated after the user has an opportunity to evaluate the conic approximation. Although the precision solution still assumes an impulsive maneuver, it does compensate for trajectory perturbations due to the earth's oblateness and the moon. An attempt is made to satisfy the original time-critical or fuel-critical constraint, in addition to the reentry radius and flight-path angle requirements obtained from the two-body solution. The object of the precision phase is to compute a new post-impulse velocity which compensates for the perturbed gravitational field. The technique used to accomplish this also makes use of the two-body solution to determine the new post-impulse velocity, by rerunning the conic computations with an offset reentry radius.

The precision solution displayed to the crew contains no significant approximations. The accuracy of the solution is largely a function of the accuracy of the onboard state vector used by the program. The answers, therefore, are comparable to those obtained from a large ground computer.

The running time of the precision phase is significant, and is primarily a function of the time of flight from ignition to reentry. It typically ranges between one and two minutes per ten hours of flight time.

B.4.4 General Considerations

The return-to-earth program is included in the onboard computer primarily to provide return-targeting capability in the event that communication with the ground is lost. During translunar coast the availability of the LM for communications backup makes the probability of a communications loss very small. Therefore, the mission phase most likely to require use of the return-to-earth program is transearth coast. In this case, the only requirement is to provide targeting for small midcourse corrections which ensure that the spacecraft reenters in the center of the entry corridor. Small minimum-fuel maneuvers, primarily in the horizontal direction, have negligible effect on the landing site; they are made several times during the return trip, and are preceded by periods of cislunar navigation.

Although the most probable use of the RTE program is in the fuel-critical mode for midcourse corrections, the crews have shown considerable interest during training in using the time-critical mode to achieve some degree of landing-site control. As explained earlier, landing-site longitude variability can be accomplished by varying the input ΔV. Optimum use of both the conic and precision phases of the program can minimize the computer time required for a solution which results in a particular longitude. Since the conic solution provides a fast approximate answer, it should be rerun until a ΔV is determined which results in a longitude close (within approximately 15 deg) to the desired longitude. Within this region the longitude is a fairly linear function of the ΔV. This solution should be continued through the precision phase to accurately establish the longitude error. Following this, the ΔV should be adjusted slightly and another conic solution computed. By comparing this solution to the previous conic solution, the sensitivity of longitude to changes in ΔV can be determined. A simple, manual computation can determine how much change in ΔV is required to bring the precision longitude to the desired longitude. This straightforward linear technique has proven to be quite efficient in producing a solution with a particular landing-site longitude.

APPENDIX C

MAJOR PROGRAM CAPABILITIES—
Powered-Flight Guidance and Navigation

C.1 Fundamentals of Powered-Flight Guidance and Navigation

The Apollo mission profile consists of many discrete segments, each with definable characteristics and objectives. (The segments are discussed in detail in Section 2.2.) The translunar-coast, transearth-coast, earth-orbit, and lunar-orbit phases represent examples of coasting flight—periods when the spacecraft trajectory is subject only to local gravitational forces. Connecting these coasting-flight segments are briefer, powered-flight segments, such as translunar injection (TLI), transearth injection (TEI), and midcourse corrections (MCCs)—periods when the spacecraft trajectory is subject not only to local gravitational forces, but also to purposeful application of thrusting and/or aerodynamic acceleration. In addition to those instances where it serves to accomplish a transfer between two coasting-flight phases, powered flight also occurs whenever either the initial or terminal state is on the earth or lunar surface. Thus, boost from and entry into the earth's atmosphere and descent to and ascent from the lunar surface are also powered-flight phases.

C.1.1 Powered-Flight Navigation

Powered-flight navigational requirements differ fundamentally from the navigational requirements of coasting flight.

During coasting flight, gravitational accelerations are the dominant influence upon the spacecraft's state vector. These accelerations are modeled quite closely, considering both the primary and secondary celestial bodies and the sun, and are integrated accurately to span the long coasting-flight intervals. Thus, precision integration greatly reduces the requirement for frequent state-vector updates either from the ground or computed onboard from optics and VHF ranging measurements.

During powered flight, spacecraft dynamics and scarcity of time preclude the use of optics and VHF ranging measurements; consequently, powered-flight navigation uses accelerometer measurements of aerodynamic and thrusting accelerations and a greatly simplified gravity model*.

C.1.1.1 Gravity Computation

The gravity calculations are performed in a straightforward manner. The equations of motion for a vehicle moving in a gravitational field are

$$\frac{d\underline{r}}{dt} = \underline{V} \quad ,$$

and

$$\frac{d\underline{V}}{dt} = \underline{g} + \underline{a}_T \quad ,$$

where $\underline{r}$ and $\underline{V}$ are the position and velocity vectors with respect to an inertial frame of reference. The measured acceleration vector of the vehicle, $\underline{a}_T$, is defined as the vehicle acceleration resulting from the sum of rocket thrust and aerodynamic forces, if any; $\underline{a}_T$ would be zero if the vehicle moved under the action of gravity alone. The vector sum of $\underline{a}_T$ and $\underline{g}$, the gravitational vector, represents the total vehicle acceleration.

Position and velocity are obtained as a first-order difference-equation calculation through a simple computational algorithm:

$$\Delta \underline{V}_a(t_n) = \underline{V}_a(t_n) - \underline{V}_a(t_{n-1}) \quad ,$$

$$\underline{r}(t_n) = \underline{r}(t_{n-1}) + \left[\underline{V}(t_{n-1}) + \frac{1}{2}\Delta\underline{V}_a\right]\Delta t + \frac{1}{2}\underline{g}_{n-1}(\Delta t)^2 \quad ,$$

$$\underline{V}(t_n) = \underline{V}(t_{n-1}) + \Delta\underline{V}_a(t_n) + \frac{1}{2}(\underline{g}_n + \underline{g}_{n-1})\Delta t \quad .$$

---

*For lunar-centered flight, a simple spherical force field is considered. For earth-centered flight, a first-order oblateness term is added. No "other body" effects are considered.

The vector $\underline{V}_a$ is the time integral of the nongravitational acceleration forces, the components of which are the outputs of three mutually orthogonal integrating accelerometers situated in the spacecraft's Inertial Measurement Unit. The gravitational vector $\underline{g}_n$ is a function of position at time t_n. This method is called "Average G" because velocity is updated by means of the average of the gravity vectors at the extremes of the measurement intervals.

A careful error analysis of a vehicle in earth orbit has demonstrated the above algorithm to yield errors of approximately 100 ft and 0.2 ft/sec after a period of 35 minutes, using a 2-sec time step and rounding all additions to eight decimal digits. (Errors increase for smaller time steps due to the effects of accumulated round-off errors; interestingly, errors also increase for larger time steps as truncation errors assume significance.) Compared to typical accelerometer scale-factor errors, the errors in the Average-G algorithm are smaller by several orders of magnitude.

C.1.2 Powered-Flight Guidance Using Cross-Product Steering

Powered-flight guidance and steering control the direction of applied thrusting acceleration to meet the targeted terminal conditions. The guidance law generates the necessary trajectory-control variables for the steering law, in addition to preparing the requisite displays to enable the astronauts to monitor the progress of the maneuver.

Steering controls the direction of vehicle thrusting accelerations according to the behavior of the guidance trajectory-control variables. Steering is implemented either manually or automatically. Manually, the astronaut implements steering by using his hand controller to apply translational accelerations via the Reaction Control System (RCS) Autopilot. Automatically, steering is implemented by a cross-product steering law which generates spacecraft-attitude commands for the Thrust Vector Control (TVC) Autopilot. (These autopilots are discussed in Appendix D.)

Powered-flight guidance depends upon the generation of an instantaneous required velocity ($\underline{V}_R$), corresponding to the present spacecraft position such that the targeted terminal conditions are achieved. Subtracting the current velocity ($\underline{V}$)

from the required velocity results in the instantaneous velocity to be gained ($\underline{V}_G$):

$$\underline{V}_G = \underline{V}_R - \underline{V} \quad .$$

Hence powered-flight guidance is commonly called $\underline{V}_G$ guidance. The steering law attempts to null this $\underline{V}_G$ (i.e., to cause the orbital velocity to approach the desired velocity) by controlling the direction of spacecraft thrusting acceleration.

For manual nulling of $\underline{V}_G$, the current $\underline{V}_G$ is an adequate control variable when expressed and displayed in a convenient coordinate system, e.g., along the orthogonal RCS jet axes. The astronaut selects X/Y/Z translation to null the corresponding displayed $\underline{V}_G$ components.

C.1.2.1 Cross-Product Steering

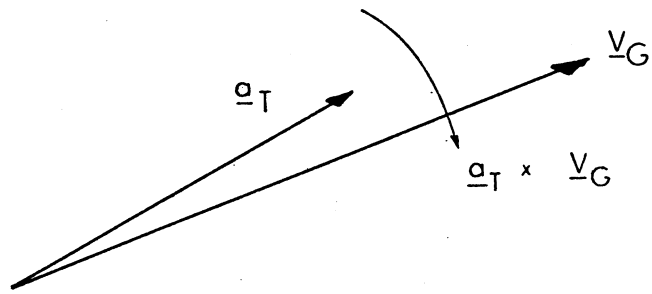
For automatic $\underline{V}_G$ nulling, the cross-product steering law generates vehicle-attitude commands for the TVC autopilot, which stabilizes vehicle attitude; in steady state the thrust vector will go through the spacecraft's center of gravity. Vehicle-attitude commands thus are essentially thrust-vector pointing commands. Cross-product steering therefore controls the direction of thrusting acceleration, $\underline{a}_T$, ensuring that the vector combination of properly oriented thrusting acceleration and inherent gravitational acceleration eventually nulls the guidance $\underline{V}_G$.

A sensible thrust-vector control system points the thrust vector in the general direction of $\underline{V}_G$. An autopilot rate-command signal, $\underline{\omega}_c$, proportional to $\underline{a}_T \times \underline{V}_G$,

$$\underline{\omega}_c = K(\underline{a}_T \times \underline{V}_G) \quad ,$$

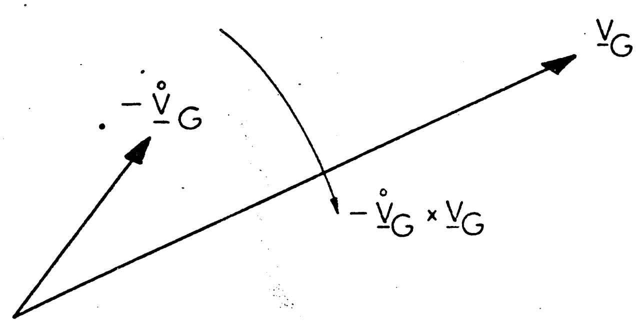
causes the vehicle to rotate in a direction to align $\underline{a}_T$ and $\underline{V}_G$ as desired, and it vanishes (i.e., commands attitude hold) when the commanded alignment is achieved. This maneuver is depicted vectorially in Fig. C.1-1.

For burns that are brief relative to the orbital period, $\underline{a}_T \times \underline{V}_G$ steering proves adequate. For longer burns, where gravitational influences produce significant orbit "bending", however, a better control policy consists of aligning $-\dot{\underline{V}}_G$ with $\underline{V}_G$ to permit limited—although not quite optimal—fuel economy. For burns of the Service



$$\underline{\omega}_c = K\,(\underline{a}_T \times \underline{V}_G)$$

Figure C.1-1 $\underline{a}_T \times \underline{V}_G$ Steering Commands



$$\underline{\omega}_c = K\,(-\dot{\underline{V}}_G \times \underline{V}_G)$$

Figure C.1-2 $-\dot{\underline{V}}_G \times \underline{V}_G$ Steering Commands

Propulsion System, $\underline{a}_T$ is the dominant component of $-\dot{\underline{V}}_G$ and hence suggests the control law,

$$\underline{\omega}_c = K(-\dot{\underline{V}}_G \times \underline{V}_G) \quad .$$

The new term, $-\dot{\underline{V}}_G$, is generated as part of the guidance policy. This maneuver is depicted vectorially in Fig. C.1-2. When one considers the dominant role of $\underline{a}_T$ in the generation of $\dot{\underline{V}}_G$, the similarity of Fig. C.1-1 and Fig. C.1-2 becomes apparent.

Concern for control of attitude changes during the maneuver suggested that a linear combination of the two steering policies might take advantage both of the simplicity of $\underline{a}_T \times \underline{V}_G$ steering and of the fuel economy of $-\dot{\underline{V}}_G \times \underline{V}_G$ steering:

$$\underline{\omega}_c = K\left[(\underline{a}_T - \gamma\dot{\underline{V}}_G) \times \underline{V}_G\right] \quad .$$

Implementation of this generalized cross-product steering policy is facilitated by the generation of a special $\underline{b}$ vector, closely related to $\dot{\underline{V}}_G$:

$$\dot{\underline{V}}_G = \underline{b} - \underline{a}_T \quad .$$

When the latter two equations are combined, the following generalized cross-product steering law results:

$$\underline{\omega}_c = K\left[(\underline{a}_T - c\underline{b}) \times \underline{V}_G\right] \quad .$$

This maneuver is depicted in Fig. C.1-3. Guidance supplies $\underline{V}_G$ and the $\underline{b}$ vector; Average-G accelerometer readings provide $\underline{a}_T$; c is specified for the particular maneuver; and K is again selected for a fast, stable response.

The result of the generalized cross-product steering policy is a vector rate of change of body attitude, with components lying along the three body axes. The roll component is ignored (since there is a separate roll autopilot which simply maintains zero roll rate), and the pitch and yaw components become commanded attitude rates for the pitch and yaw channels of the TVC autopilot.

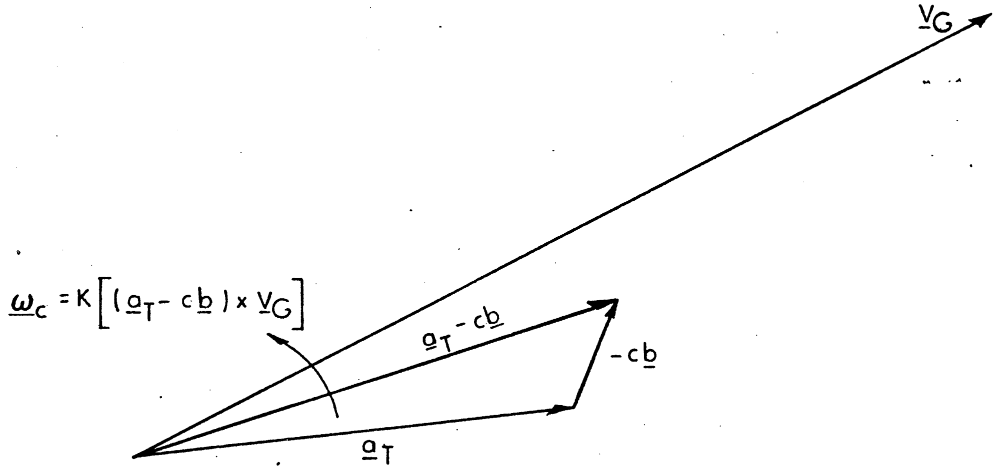


Figure C.1-3   Generalized Cross-Product Steering

C.1.2.2 Comparison of Explicit and Implicit Guidance Policies

Powered-flight guidance strategy employs two basic approaches, explicit and implicit.

Explicit guidance refers to guidance policies in which the output variables ($\underline{V}_R$, $\underline{V}_G$, $\underline{b}$, etc.) are computed from algorithms which explicitly involve present and terminal vehicle states. Implicit guidance refers to guidance policies in which the algorithms only implicitly involve vehicle states. For example, the direct computation of $\underline{V}_G = \underline{V}_R - \underline{V}$ involves $\underline{R}$, $\underline{V}$, and $\underline{R}_T$ explicitly, whereas in the integration of $\dot{\underline{V}}_G = -Q^* \underline{V}_G - \underline{a}_T$ (see below), nowhere do $\underline{R}$ and $\underline{V}$ appear explicitly. Direct $\underline{V}_G$ calculation is thus an explicit mechanization, while $\dot{\underline{V}}_G$ integration is an implicit mechanization of a $\underline{V}_G$ guidance policy.

Explicit calculation of $\underline{V}_R$ has several disadvantages. Computational times are long, the algorithms require large storage capacity, and a different set of algorithms is required for each type of maneuver. For some mission phases (e.g., TLI), no known closed-form solution exists for the steering vector $\underline{b}$, thus requiring simplified solutions involving "close-to-nominal" assumptions. With non-nominal flight (e.g., contingency aborts), such disadvantages become very severe.

The Lambert guidance policy implemented for Apollo consists of a mix of explicit and implicit forms. Between explicit $\underline{V}_G$ solutions, $\underline{V}_G$ is extrapolated by integrating the $\dot{\underline{V}}_G$ implicit-guidance equation. This permits use of a very crude $\underline{b}$ vector, and largely removes the close-to-nominal requirements that might otherwise exist.

Implicit guidance, as mentioned above, refers to guidance policies in which the output variables are computed from algorithms where the terminal state and the present state do not appear explicitly. Generally, the instantaneous required velocity is a function of present position and time,

$$\underline{V}_R = \underline{V}_R(\underline{r}, t)$$

The time derivative of V_R can be expressed by applying the total derivative law:

$$\dot{\underline{V}}_R = \frac{d\underline{V}_R}{dt} = \frac{\partial \underline{V}_R}{\partial t} + \left\| \frac{\partial \underline{V}_R}{\partial \underline{r}} \right\| \underline{V} \quad .$$

Substitution from the basic equation,

$$\underline{V}_G = \underline{V}_R - \underline{V} \, ,$$

and application of certain $\dot{\underline{V}}_R$ relationships along the solution trajectory (in particular, $\dot{\underline{V}}_R \Big|_{\underline{V}=\underline{V}_R} = \underline{G}$) yields, after some manipulation,

$$\dot{\underline{V}}_R = \underline{G} - Q^* \underline{V}_G \, ,$$

where Q^* is the 3x3 matrix of partial derivatives,

$$\left\| \frac{\partial \underline{V}_R}{\partial \underline{r}} \right\| \quad .$$

This equation states that the time rate of change of $\underline{V}_R$ along any path is equal to $\underline{G}$ (as it would be along the correct free-fall path) plus a correction if the actual path is not the free-fall path (as $\underline{V}_G \rightarrow 0$, $\dot{\underline{V}}_R \rightarrow \underline{G}$).

Substitution of this result into the time derivative of $\underline{V}_G = \underline{V}_R - \underline{V}$ yields an expression for $\dot{\underline{V}}_G$:

$$\dot{\underline{V}}_G = -Q^* \underline{V}_G - \underline{a}_T \quad .$$

This simple first-order equation can serve as a guidance policy, involving only the thrusting acceleration inputs (from the Average-G accelerometer readings) and the matrix Q^*. Nowhere do present vehicle position or velocity, present velocity required, or terminal (targeted) state appear explicitly. Terminal-state, present velocity, and present required velocity are all implicit in the current velocity to be gained. Present position is implicit in the Q^* matrix. A single initial explicit calculation of $\underline{V}_R$ to obtain the initial $\underline{V}_G$ is required. From then on the equation for $\dot{\underline{V}}_G$ is self-sufficient, requiring only the external generation of the elements of Q^*. Implicit guidance is often referred to as $\dot{\underline{V}}_G$ guidance, whereas explicit guidance, as mentioned above, is called $\underline{V}_G$ guidance. All necessary inputs to the cross-product routine are readily available: $\underline{V}_G$ from the integration of the $\dot{\underline{V}}_G$ guidance equation, $\underline{a}_T$ from the Average-G accelerometer readings, and $\underline{b} = \dot{\underline{V}}_G + \underline{a}_T$.

In general, the Q^* elements are functions of position and time. Where flight is close to some nominal trajectory, the position dependence can be removed, and the elements can be represented simply as functions of time. Further simplifications can be made, such as using two or three straight-line segments with perhaps one or two discontinuities; using simple straight lines; using simple constants; and even using zero for selected elements.

The advantage of implicit guidance is its computational simplicity. One disadvantage is the requirement for much preflight simplification and approximation to determine adequate representations for the Q^* elements. Another disadvantage is the severe restriction to near-nominal flight—greater here than it was for explicit guidance.

The interrelationships of explicit/implicit guidance and cross-product steering should be noted. Initial $\underline{V}_R$ and $\underline{V}_G = \underline{V}_R - \underline{V}$ calculations are made in both the explicit and implicit approaches. Thereafter, only in the explicit case are periodic resolutions for instantaneous $\underline{V}_R$ and $\underline{V}_G$ required. Implicit guidance simply integrates the $\dot{\underline{V}}_G$ equation. To do so, however, periodic updates of the Q^* matrix (partial derivatives,

$$\left\| \frac{\partial \underline{V}_R}{\partial \underline{r}} \right\| ,$$

functions of time and position) are required. Note in the $\dot{\underline{V}}_G$ expression that definition of a $\underline{b}$ vector,

$$\underline{b} = -Q^* \underline{V}_G ,$$

yields an equation

$$\dot{\underline{V}}_G = \underline{b} - \underline{a}_T ,$$

which was introduced above, during cross-product steering discussions. Both explicit and implicit guidance laws use the cross-product steering law. For implicit guidance, the $\underline{b}$ vector used by cross-product steering is already available, appearing as the $-Q^* \underline{V}_G \triangleq \underline{b}$ in the $\dot{\underline{V}}_G$ equation; no additional computations are required. For explicit guidance, however, the cross-product steering requirement for $\underline{b}$ presents an added computational burden; analytic expressions for $\underline{b}$ must be implemented, solely for inputs to cross-product steering.

There are two ways to circumvent the computational burden, however. First, one can revert to simple $\underline{a}_T \times \underline{V}_G$ steering (i.e., by setting c = 0 in the generalized cross-product steering law). Then, the value of $\underline{b}$ is immaterial and need not be computed. The implementation of Lambert-targeted ASTEER guidance employs such a simplification (see Section C.1.2.4). Fuel penalties for the non-optimal c = 0 are minimal in the situations where ASTEER is used.

Alternatively, the $\underline{b}$ vector can be obtained with adequate accuracy by a simple numerical differentiation using a first-order back difference. If the equation for the $\underline{b}$ vector ($\underline{b} = -Q^* \underline{V}_G$) is substituted into the equation for $\dot{\underline{V}}_R$ ($\dot{\underline{V}}_R = \underline{G} - Q^* \underline{V}_G$), the following expression for $\underline{b}$ results:

$$\underline{b} = \dot{\underline{V}}_R - \underline{G} \quad .$$

A simple first-order back difference of $\underline{V}_R$ is suggested:

$$\dot{\underline{V}}_R(\tau) = \frac{\underline{V}_R(\tau - \Delta\tau) - \underline{V}_R(\tau)}{\Delta\tau} \quad .$$

The $\underline{b}$-vector equation then becomes simply:

$$\underline{b}(\tau) = \frac{\underline{V}_R(\tau) - \underline{V}_R(\tau - \Delta\tau)}{\Delta\tau} - \underline{G}(\tau) \quad .$$

Note that $\underline{G}(\tau) = G[\underline{R}(\tau)]$ now involves present vehicle state explicitly (obtained from Average G). Time- and memory-consuming explicit $\underline{b}$-vector calculations are no longer required. Preprogramming and close-to-nominal trajectory restrictions inherent in the explicit $\underline{b} = -Q^* \underline{V}_G$ implementation are avoided. There are some geometries for which the above equation gives degraded $\underline{b}$ vectors, but generally the results are acceptable.

The final form of powered-flight guidance is a combination of explicit and implicit guidance laws. Periodically, the explicit calculation of $\underline{V}_G = \underline{V}_R - \underline{V}$ is made; this involves direct computation of $\underline{V}_R$ (e.g., via the iterative Lambert subroutine, discussed below). The necessary velocity computations are slow, requiring several seconds (2 to 4 normally, but 10-sec cycles have been observed). During the explicit computational interval, implicit guidance is used to maintain a current estimate of $\underline{V}_G$ for steering. The required $\underline{b}$ vector is obtained from simple back differences of $\underline{V}_R$, with local gravitation taken into consideration.

C.1.2.3 Lambert Powered-Flight Guidance

Lambert powered-flight guidance solves the problem of reaching a specified target position at a specified intercept time. The period of free-fall coasting flight which follows the termination of the active guidance phase may last a matter of hours (one or more orbits in rendezvous) or days (during return to earth following a translunar orbit or a transearth midcourse correction).

Computations for Lambert powered-flight guidance can be done either onboard the spacecraft or telemetered from the ground. Lambert targeting (TPI or TPM) generates the target vector, the intercept time, the time of ignition, and various other parameters and displays. Return-to-earth targeting generates these same Lambert-type inputs for the Lambert guidance. Guidance prepares the necessary trajectory-control variables during the active transfer phase between the two periods of coasting flight. The $\underline{V}_G$ is nulled either manually (via the astronaut's hand-controller activity) or automatically (via the cross-product steering law).

Lambert powered-flight guidance involves both explicit and implicit guidance policies. Briefly, the explicit task involves the direct calculation of required velocity, $\underline{V}_R$, and velocity-to-be-gained:

$$\underline{V}_G = \underline{V}_R - \underline{V} ,$$

where $\underline{V}$ is the current velocity. The implicit task involves the extrapolation of $\underline{V}_G$ during the explicit $\underline{V}_G$ computation interval.

Lambert's theorem states that the time of flight (t_f) along the solution trajectory depends only upon the length of the semimajor axis, a, of the solution conic (valid for general conics); the sum of the distances to the initial (r_1) and final (r_2) points of the arc from the center of force; and the length of the arc (c) joining these points:

$$t_f = t_f(a, r_1 + r_2, c) \quad .$$

Time of flight is the known specified quantity:

$$t_f = t_2 - t_1 \quad .$$

The task of the Lambert subroutine is to determine iteratively the solution conic, in particular the parameter a. Required velocity is then simply the velocity associated with the solution conic at the present position:

$$\underline{V}_R = \underline{V}_R(\underline{R}_1, a) \quad .$$

The present velocity to be gained is simply the difference between present velocity required and present orbital velocity. The $\underline{b}$ vector, required only for the steering task, is computed by back-differencing $\underline{V}_R$ and accounting for local gravitation, as described above.

C.1.2.4 Lambert ASTEER Guidance

Lambert ASTEER guidance is an explicit guidance policy which solves the problem of reaching a specified target vector at approximately a specified time. ASTEER guidance utilizes a single initial solution of Lambert's theorem by the Lambert iterator, e.g., for the semimajor axis, a, of the desired coasting-flight conic:

$$t_f = f(a, \underline{R}_1, \underline{R}_T') \quad .$$

Since the variation of a during the brief rendezvous powered maneuvers directed by the AGC on the LM is negligible, a can be assumed constant (leading only to minor error in the time of actual intercept). As in the nominal Lambert case discussed in the previous section, Lambert targeting provides the offset target vector, $\underline{R}_T'$, and the time of flight, $t_f = t_T - t_1$. The initial $\underline{V}_{R_1}$ is computed explicitly for the initial a:

$$\underline{V}_{R_1} = f(a_1, \underline{R}_1, \underline{R}_T') \quad .$$

Thereafter, explicit $\underline{V}_R(\tau)$ updates are computed as functions only of $\underline{R}(\tau)$, using the initial a:

$$\underline{V}_R(\tau) = f(a_1, \underline{R}_\tau, \underline{R}_T') \quad .$$

Initial and updated explicit $\underline{V}_G$'s are computed via the equation,

$$\underline{V}_G(\tau) = \underline{V}_R(\tau) - \underline{V}(\tau) \quad .$$

By assuming constant $a = a_1$, the Lambert-iterator resolutions can be avoided and time saved, thus permitting $\underline{V}_G$ updates every 2-sec Average-G steering cycle.

ASTEER avoids the steering $\underline{b}$-vector problem simply by setting $\underline{b} = \underline{0}$. This is convenient from the vantage of avoiding unnecessary $\underline{b}$-vector computation time, and empirically justified for those situations where ASTEER is used (e.g., short burns such as LM-active rendezvous maneuvers, where $\underline{a}_T \times \underline{V}_G$ steering might be expected to work, anyway).

C.1.2.5 External ΔV Powered-Flight Guidance

External ΔV powered-flight ($\dot{\underline{V}}_G$) guidance is an implicit guidance policy designed for ground targeting of lunar-orbit insertion (LOI), descent-orbit initiation (DOI), transearth injection (TEI), midcourse correction and abort maneuvers and for onboard targeting of certain rendezvous maneuvers. External ΔV was implemented for several reasons:

a. The explicit $\underline{V}_R$ and $\underline{b}$ computations planned originally for the LOI, DOI, and TEI maneuvers required extensive fixed- and erasable-memory storage, and consumed too much computer time.

b. External ΔV is a simple, fast, reliable guidance/steering mode, compared to the slower—and more complex—Lambert and ASTEER modes. Considerable experience with external ΔV maneuvers had been gained during the Gemini program, whereas Lambert maneuvers were a new development.

c. External ΔV is characterized by an inertially-fixed burn attitude, which simplifies the problems of burn monitoring and manual takeover in the event of partial or complete primary-system failure.

d. Reduced computational burdens permit more time for nominal activity (displays, autopilots, compensation, etc.) and extended verb activity.

e. External ΔV is conceptually simpler, e.g., for crew-originated maneuvers. Fewer inputs are required, with fewer demands made upon the crew.

Disadvantages of External ΔV are a fuel penalty on long burns and the requirement for ground targeting and uplink communication of the targeting parameters—time of ignition and initial $\underline{V}_G$ (in local vertical coordinates); but these disadvantages are offset by the tremendous savings in computer-memory capacity which an onboard targeting program would have required. Voice-linked loads of planned and contingency maneuvers are also transmitted in case of a communications failure. Fuel penalties, even during the long LOI, TEI, and translunar aborts, are minimal.

As previously mentioned, External ΔV is an implicit $\dot{\underline{V}}_G$ policy. There is no reference to a required velocity; targeting supplies the initial $\underline{V}_G$ direction, and $\underline{V}_G$ is extrapolated from

$$\dot{\underline{V}}_G = \underline{b} - \underline{a}_T ,$$

discussed above. The $\underline{b}$ vector for External ΔV maneuvers is <u>identically</u> zero, meaning that $\underline{V}_G$ is affected only by thrusting accelerations.

A unique feature of External ΔV prethrust computations is an approximate compensation for finite burn times, in which the in-plane component of the initial $\underline{V}_G$ is rotated in the direction of the angular-momentum vector, by half the predicted central angle of the burn. Current vehicle mass and a nominal thrust level (2- or 4-jet RCS, or the SPS) are also used in the calculation. No attempt is made to account for mass variation during the burn.

C.1.2.6 Thrust-Cutoff Sequencing

Automatic termination of thrust is provided for powered-flight maneuvers, since the time constants of computation, display and astronaut-reaction would result in an unacceptable ΔV for a manual termination. The guidance/steering calculations generate an estimate of time-to-cutoff, which is then displayed for burn-monitoring purposes. When time-to-cutoff drops below four sec, the steering commands are set to zero, the steering and time-to-cutoff calculations are disabled, and a task is scheduled to issue the engine-cutoff discrete at the proper time. The guidance $\underline{V}_G$ calculations continue (in case they should be needed for manual $\underline{V}_G$ trimming) following engine shutdown.

The time-to-cutoff calculations involve the present $\underline{V}_G$, the just-measured $\Delta\underline{V}_a$ (Average G, Section C.1.1), and the most recent $\underline{b}$ vector (identically zero for ASTEER and External ΔV steering). As depicted in Fig. C.1-4, the desired effect is the minimization of cutoff $\underline{V}_G$ magnitude, assuming that the direction and magnitude of $-\dot{\underline{V}}_G$ are constant.

The equation for time-to-cutoff (TTC), measured from the time of the accelerometer reading associated with $\underline{V}_G$ and $\dot{\underline{V}}_G$,

$$TTC = \frac{\underline{V}_G \cdot \mathrm{UNIT}(-\dot{\underline{V}}_G)}{\dot{V}_G}\left[1 - \frac{\underline{V}_G \cdot \mathrm{UNIT}(-\dot{\underline{V}}_G)}{2V_E}\right] - \Delta T_{TO} ,$$

contains a first-order variable-mass approximation (V_E is the exhaust velocity) and a biasing impulsive ΔV tailoff time, ΔT_{TO}. The vector dot product represents the projection of $\underline{V}_G$ onto the current $-\dot{\underline{V}}_G$ direction.

Cutoff residuals comprise the cross-axis component shown in Fig. C.1-4, plus axial cutoff errors due to the approximations associated with the TTC algorithm. As cutoff approaches, the approximations become quite good $\left(\underline{b} \rightarrow \underline{0} \text{ as } \underline{V} \rightarrow \underline{V}_R,\right.$ and $\left. V_E \gg \underline{V}_G \quad \mathrm{UNIT}(-\dot{\underline{V}}_G)\right)$; axial cutoff errors are therefore on the order of the theoretical maximum of two accelerometer pulses, provided that engine parameters (V_E, constant thrust, ΔT_{TO}, etc.) are modeled accurately.

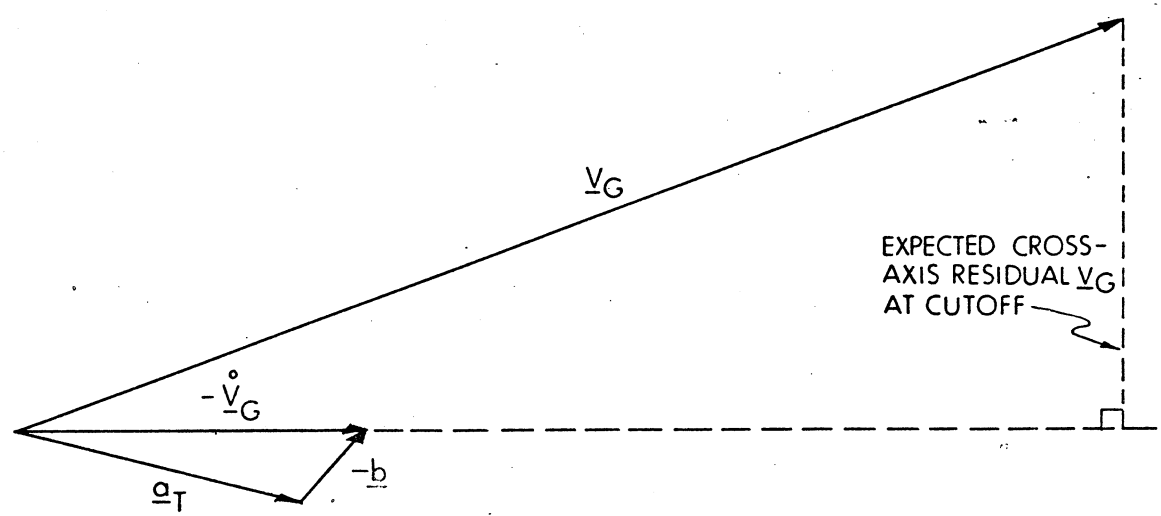


Figure C.1-4 Time-to-Cutoff Geometry

The cross-axis component of residual $\underline{V}_G$ at cutoff, however, can be quite large, especially for short burns during which the steering/autopilot loop dynamics prevent recovery from initial vehicle and thrust-pointing errors, the dynamics associated with a moving center of gravity, or $\underline{b}$-vector dynamics in a severe-burn geometry. However, the post-burn trimming maneuver can accommodate these situations.

C.2 Thrust Monitor Program

The thrust monitor program monitors and displays velocity changes applied to the spacecraft during manual maneuvers and during maneuvers not controlled by the GN&C System. The program initially suspends state-vector updating by the VHF range-link (i.e., resets the update and track flags), and advances the vehicle-state vector to the current time. This operation continues until the state vector is extrapolated ahead to the ignition time. The Average-G routine discussed in the previous section is then initiated, thus allowing the earliest possible ignition. Average G remains in operation until the program is terminated upon completion of the maneuver. The primary output of the thrust-monitor program is the measured maneuver $\Delta\underline{V}$, expressed in vehicle coordinates.

The thrust-monitor program is normally employed during two major maneuvers—the translunar-injection maneuver controlled by the Saturn guidance system and the manually-controlled terminal rendezvous maneuvers required for a CSM retrieval of the LM. During the TLI maneuver, the astronaut can call up the display of inertial velocity altitude and altitude rate. During the active CSM terminal rendezvous maneuvers, relative range, range rate, and the angle of the vehicle X axis relative to the horizontal plane are available for display.

C.3 Earth-Orbit Insertion Monitor Program

The AGC program for earth-orbit insertion monitors the performance of the Saturn launch vehicle from the detection of the liftoff discrete to the accomplishment of earth orbit. In the event of a Saturn inertial-platform failure during boost, however, the Command Module AGC provides a backup capability to guide the Saturn along a prescribed trajectory which is discussed in some detail in Section D.5.

Since a close relationship must exist between the guidance and navigation equipment on both the Saturn launch vehicle and the Command Module, it is important to detail that relationship, beginning prior to launch.

As explained in Section 2.2.2.1, two sets of guidance equipment are prepared for launch: The Saturn guidance equipment in the Saturn Instrument Unit controls the launch vehicle, while the Apollo guidance equipment in the Command Module provides a monitor of Saturn guidance during launch. The Saturn and CM are both inertially aligned to a common vertical and launch azimuth reference. During countdown, both systems are gyro-compassed to an earth-frame reference. Near liftoff, they both respond to discrete signals to switchover from the earth reference to the nonrotating inertial reference used during boost.

Prior to liftoff, the AGC interrogates the liftoff discrete every half-second, and when liftoff is detected, the computer initiates the earth-orbit insertion program. Should the discrete fail or malfunction, the astronaut can initiate the program. At liftoff, the current clock reading updates the universal time previously stored in the AGC, and the AGC clock itself is then zeroed.

During first-stage flight, the Saturn Instrument Unit controls the flight by swiveling the engine's outer four rockets. For inertial vertical flight, the vehicle is rolled from its launch azimuth to the flight-path azimuth. Saturn guidance then controls the vehicle in an open-loop, preprogrammed pitch maneuver designed to pass safely through the critical period of high aerodynamic loading.

Throughout this period, the AGC's powered-flight navigation routine (Average G, Section C.1.1) calculates the position and velocity of the Saturn vehicle. This routine is initialized with the position and velocity at liftoff, computed using knowledge of the earth-rotation vector, observed time of launch, and preloaded values of launch-site longitude and geodetic latitude. (To relate the stable-member (SM) orientation at liftoff to the basic reference-coordinate system, the direction of Z_{SM} is calculated first. The Z_{SM} axis, which is aligned along the true direction of gravity at the launch pad, defines the astronomical latitude. The astronomical latitude is the derived angle between the zenith and the equator, measured by the observed angle between the horizon and the polar axis. The geodetic latitude, which is also defined as the angle between the zenith and the equator, is calculated on the basis of the earth's reference ellipsoid. Since the astronomical latitude is approximately equal to the geodetic latitude at the launch pad, the geodetic latitude is substituted in the numerical calculation of the direction of gravity.)

The earth's polar axis is described in reference coordinates for each space mission by storing in the AGC the quantities necessary to determine precession and nutation. On this basis, the Z_{SM} may also be computed in the reference system.

The CMC assumes that X_{SM} and Y_{SM} are in the plane of the horizon at the launch site. During prelaunch, the X_{SM} is kept aligned with the launch azimuth, an angle in the horizon plane measured positive from North in an easterly direction to X_{SM}. Since North and South are determined by the line of intersection between the horizon plane and the plane determined by the earth's polar axis and the zenith, X_{SM} can be computed in reference coordinates. X_{SM} and Z_{SM} in reference coordinates determine Y_{SM} in reference coordinates, thus allowing the transformation (REFSMMAT) from earth-reference to stable-member coordinates.

Both the Saturn and CM guidance systems continuously measure vehicle motion and compute position and velocity. In addition, the GN&C System compares the actual Saturn trajectory with that to be expected from the AGC, using a sixth-order polynomial approximation—thus generating an attitude-error display for the crew. During boost (first-stage only), these attitude errors are available to the Saturn Instrument Unit for nulling, should a takeover be required. Should a takeover be required during second or third-stage boost, the only commands the Instrument Unit can receive are from the Rotational Hand Controller which the astronaut manipulates as he compares DSKY displays of velocity, altitude and altitude rate with a nominal trajectory profile available on printed card. In this fashion the spacecraft can be flown safely into earth orbit with relatively minor errors in apogee and perigee.

C.4 Entry Guidance and Mission Control Programs

The phase of the mission beginning with the jettison of the Service Module by the Command Module and ending with safe arrival at the designated landing site comprises three major control functions—entry guidance, mission control programs, and Entry Digital Autopilot. (For a summary of the entry phase, see Section 2.2.2.14.)

The entry guidance directs the CM to a safe return at the designated landing site. The mission control programs (P61 through P67) inform the crew of their location along the entry trajectory, as determined by the Entry Guidance. The Entry DAP uses the output of the entry guidance (i.e., roll command) to perform automatically all maneuvers necessary for reentry.

This section of Appendix C discusses entry guidance and mission control programs. The Entry DAP is discussed in Appendix D, which describes all of the Digital Autopilots.

C.4.1 Entry Guidance

A spacecraft returning from a lunar mission reenters the earth's atmosphere at velocities exceeding escape velocity. To cope with the sensitive dynamics involved, automatic entry guidance and control are employed. Although escape velocity is only 40 percent greater than orbital velocity, the sensitivity of the reentry range capability to variations in the critical entry flight-path angle* is several orders of magnitude greater for entry from lunar missions than from earth-orbital missions. The automatic guidance system is expected to provide quick response and to minimize the effects of variation of the actual environmental characteristics (such as CM aerodynamics and the atmosphere) from the design standards.

The automatic entry-guidance system was designed with two main objectives, having the following priority: (1) a safe return to the earth's surface, and (2) landing-point control. For the CM to splashdown successfully at the designated landing site, these objectives must be met. Safe return means that the deceleration during reentry should never exceed a prescribed limit, nor should the spacecraft skip back out of the atmosphere at greater than orbital velocity. The reentry trajectory may include a free-fall, ballistic lob portion out of the atmosphere in order to reach a distant landing point, but this must be done at subcircular velocity. The midcourse guidance phase has the initial responsibility for a safe return in that the spacecraft must be steered into an acceptable entry corridor from which a safe return is possible.

---

*The entry flight-path angle is defined as the angle between the inertial velocity vector and local horizontal at the entry interface altitude of 400,000 ft above the reference geoid.

The entry guidance must achieve the objective of range-control without interfering with the objective of a safe return. Acceptable reentry-angle values (about the nominal) define a region called the reentry corridor. A nominal entry-angle value for a lunar mission is approximately -6.5 deg. (A nominal entry-angle value for an earth-orbital mission is approximately -1.5 deg.)

To fulfill its main objectives, the entry guidance must cope with errors. The distinction is made between navigation errors and steering errors.

Navigation errors are inaccuracies in the determination of the spacecraft's own position and velocity. These errors cannot be removed by guidance design. It is sometimes convenient to think of a navigation error as the error in where the spacecraft thinks the target is. Causes of navigation errors include:

(1) Errors in the indicated initial position and velocity at the start of reentry.
(2) Initial misalignment of the IMU.
(3) IMU gyro and accelerometer errors.

Steering errors, on the other hand, depend on guidance design and represent the spacecraft's inability to reach the position where it thinks the target is. In a well-designed system, the expected miss-distance should be approximately the same as the expected navigation error near the end of the reentry trajectory. A subtle type of steering error involves control actions which are taken early in entry, while the velocity is still supercircular. The danger is that during this phase, while the sensitivities are high, control actions based on navigation errors may be improper and result in a large enough trajectory deviation such that later control actions are incapable of compensating sufficiently for the early mistakes. Thus, steering errors are a function of navigation errors. Specifically, the chief troublemaker is the error in indicated rate-of-climb during the supercircular phase. The effect of this error is minimized by the use of a reference trajectory during the pull-up portion, Upcontrol. On the other hand, navigation errors are not a function of steering errors or the steering scheme, except indirectly.

A controlled-entry flight to the designated landing point is achieved by taking advantage of the aerodynamic lift capabilities of the CM. The CM is a wingless, axially symmetric, reentry vehicle constructed so that its center of gravity is displaced from its axis of symmetry. When flying hypersonically in the atmosphere, it trims with a low, constant ratio of lift to drag. The only means for perturbing the trajectory in a controlled manner is to roll about the wind axis (velocity vector) with reaction-control jets, permitting the lift vector to be pointed anywhere in the plane perpendicular to the wind axis. The roll angle defines the orientation of the lift vector relative to the trajectory plane—i.e., the plane containing the wind axis and the position vectors. The component of lift in the trajectory plane is the means of trajectory control for the down-range flight, while the component of lift out of the trajectory plane is for control of cross-range flight.

The entry-guidance equations regularly compute the desired lift direction currently necessary if the trajectory is to reach the designated landing site. In actual flight, the Entry DAP causes the CM to roll about the wind axis so that the actual lift direction is forced into agreement with the desired lift direction. This results in achieving the desired in-plane component for down-range control. The rolling maneuver yields an out-of-plane component of lift that is used for lateral steering. After the cross-range error is removed, the lateral lift component is an unwanted by-product of the steering, and its effect is constrained to an acceptably small value by the guidance, which causes the CM to roll periodically so as to reverse the sign of the lateral lift and null out lateral drift. Since the in-plane component is the fundamental controlled quantity, in that it controls down-range flight, its sign normally remains unchanged during the nulling process (lateral switching). In effect, the restriction on the sign of the in-plane component during lateral switching demands that the CM roll through the smaller of the two possible angles—through the so-called shortest path—when lateral switching takes place; the Entry DAP normally causes this type of maneuver. But in certain instances where such a maneuver would cause the vehicle to fall short of the target, the entry guidance insists on rolling over the top without regard for the angle size, and a program switch is set to inform the roll DAP that the CM is being commanded to roll through the larger angle. In summary, the entry guidance continuously provides a single steering quantity—the commanded roll angle—that is to be achieved by the CM.

Entry guidance is further described by using the illustrative deceleration profile of Fig. C.4-1. At point 1, the CM lift is initially directed upward for nominal-to-steep entry angles; for shallow angles, lift is directed downward. A function of the guidance is to establish the initial lift by roll-angle command, which

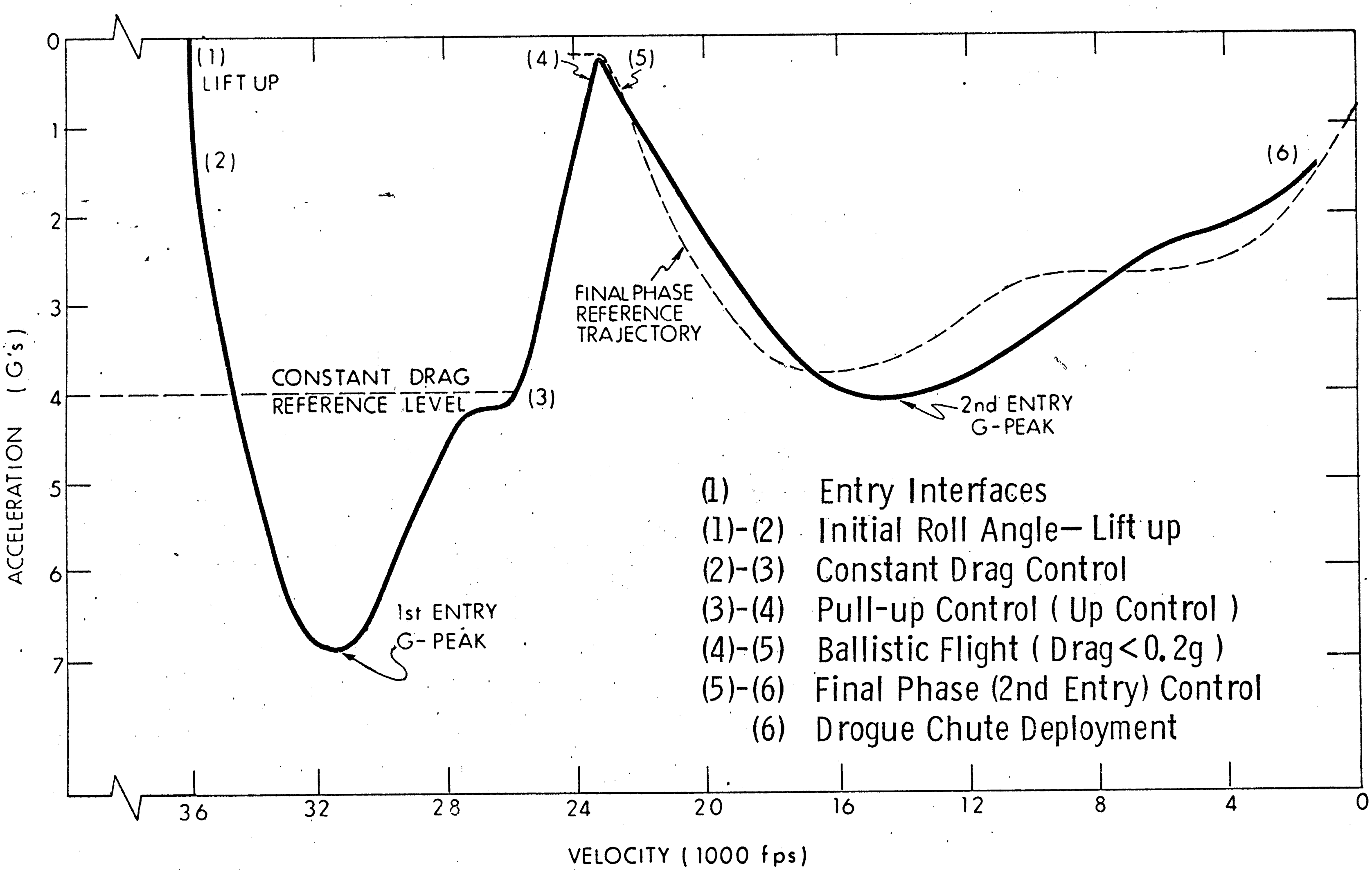


Figure C. 4-1 Typical Deceleration Profile is held constant until the CM is in the atmosphere and the deceleration level reaches 0.05g. Prior to 0.05g, the Entry DAP holds the CM in the attitude of hypersonic trim at the specified roll angle. After 0.05g, the atmosphere holds the CM in trim, with rate-damping assistance in pitch and yaw by the DAP. At this point, the guidance begins the lateral-range calculations which will permit a small (15 deg) roll-angle deviation. The g-level continues to increase until about 1.3g, when constant drag control begins—attempting to maintain a constant deceleration (point 2). Velocity and range-to-go decrease. When the rate of descent reaches 700 ft/sec, a trajectory-search, in addition to constant drag, begins to determine if the predicted range for a constant lift-to-drag ratio (L/D) flight from the spacecraft's current location would yield the desired range. The predicted range is made up of analytic expressions based on a candidate reference trajectory consisting of segments for pull-up, ballistic lob and final (second-entry) phase. If the predicted range is not within 25 nmi of the desired range, the constant drag control continues to be flown. When the predicted range is within 25 nmi of the desired range, the guidance begins to steer the CM along the pull-up reference (point 3). When the terminal conditions of the Upcontrol reference are met, a ballistic lob is flown if the drag becomes less than 0.2g (point 4). The guidance maintains a roll command constant at the most recent value. Should the drag become less than 0.05g, the roll command is set to zero and, in addition, the inadequate aerodynamic stability in this low-dynamic pressure region requires that pitch and yaw hypersonic trim again be maintained by the three-axis DAP. On reaching the peak altitude, the CM falls back toward the earth and reenters the atmosphere at point 4. At 0.05g, the DAP relinquishes. The final phase of guidance is entered when the acceleration builds up and exceeds a 0.2g threshold. The final-phase guidance (point 5) steers the CM to the landing site, using a prestored reference trajectory. Short-range trajectories omit the ballistic and Upcontrol phases; very long-range trajectories omit the constant-drag phase.

Two remaining functions of entry guidance are the g-limiter logic, which modifies the roll commands during the final phase to avoid exceeding the maximum 8g limit, and the lateral logic, which periodically switches the desired lift direction from one side of the trajectory plane to the other to allow the roll angle to control both down-range and lateral range.

Three entry-guidance functions are used throughout the trajectory—Navigation, Targeting and Mode Selector. The first and second of these functions begin in Program P61; the third begins in P63. Navigation updates the position and velocity vectors using acceleration data. Targeting computes the current landing site vector based on earth rotation during an estimated flight time and calculates the present range and lateral range for the landing site. Mode Selector selects each entry-guidance phase on the basis of the current position along the entry trajectory.

C.4.2. Entry Mission Control Programs

The entry trajectory is broken into major segments, each identified with a mission program number, such as P61, shown in Fig. C.4-2 for both orbital and lunar entry. After the crew selects P61, the programs run in numerical sequence. Selection of the next major mode is automatically determined by the mission control program prior to beginning P63 and by entry-guidance thereafter. While each phase of the entry guidance is operating, the DSKY displays the corresponding mission control program, P61 through P67, to inform the crew of the spacecraft's location along the entry trajectory. The mission control programs form the framework within which the crew is able to monitor the phases of flight beginning with the maneuver to CM/SM separation attitude and continuing until droge-chute deployment.

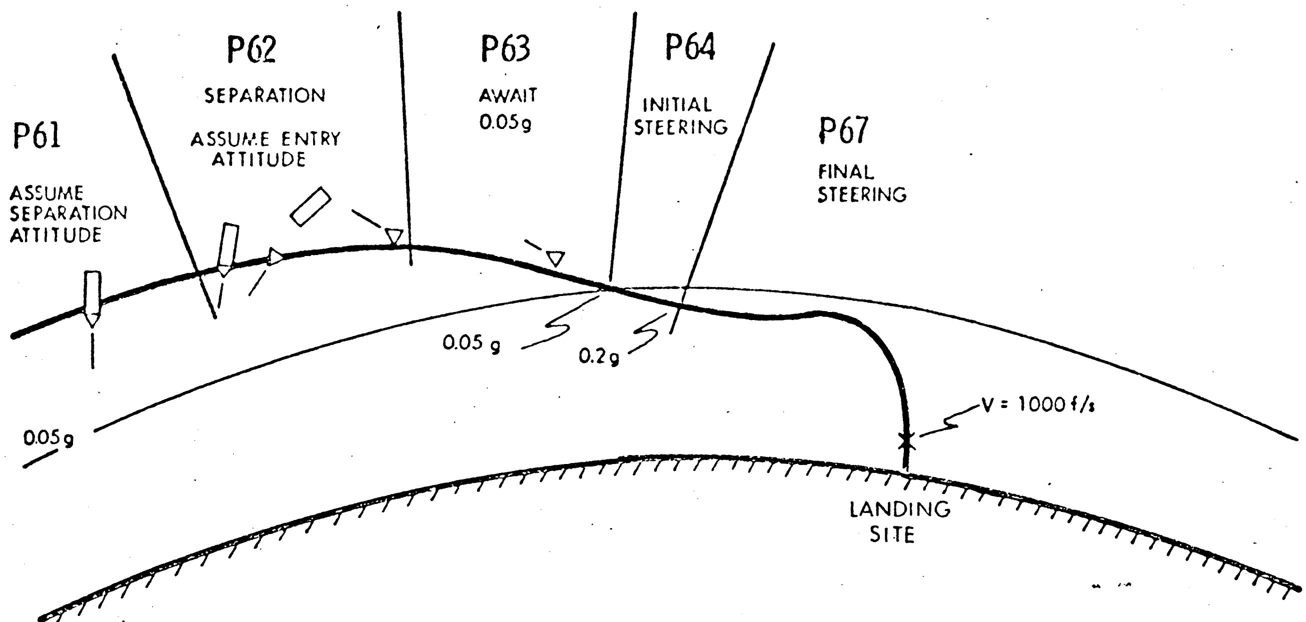
The mission control programs for entry are summarized in the following sections:

C.4.2.1 Entry Preparation, P61

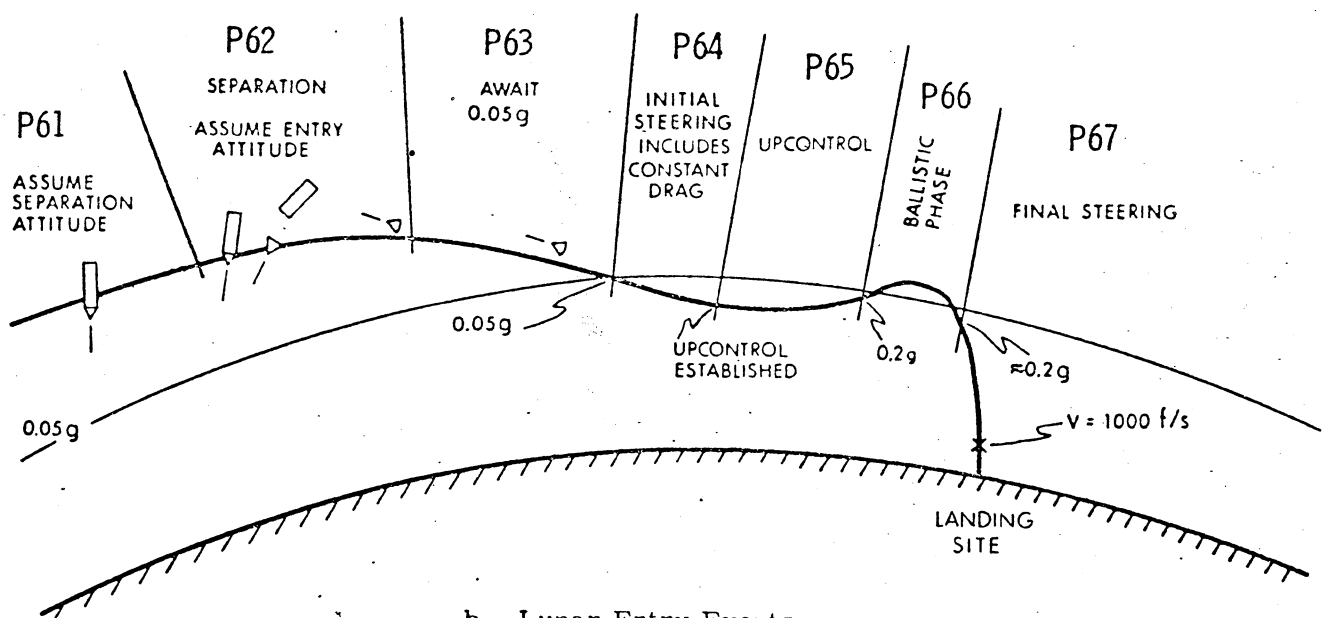
The Entry Preparation Program starts the entry sequence for the CM. P61 initiates onboard navigation and checks the IMU alignment. Next, it calculates data so that the crew can initialize the Entry Monitor System (EMS)* and also monitor certain future trajectory check points. The data from the P61 displays are also compared with the ground-supplied EMS data and serve as a check on the operation of the AGC. On completion of the displays, P62 is begun.

---

*The EMS is a backup system used to monitor entry performance; it provides a roll-attitude indication, a range-to-go counter and a velocity and acceleration plot to compare with acceptable velocity/acceleration profiles.



a. Orbital Entry Events



b. Lunar Entry Events

Figure C. 4-2 Typical Entry Events along Trajectory

C.4.2.2 CM/SM Separation and Pre-entry Maneuver, P62

The CM/SM Separation and Pre-entry Maneuver Program establishes the trajectory-monitoring displays for the crew, accepts crew notification that CM/SM separation has occurred, activates the Entry DAP, and orients the CM to the correct attitude of hypersonic trim. When the DAP first becomes active, it establishes attitude hold, following separation, until the crew verifies the correctness of the display of initial roll-attitude specification and target location. After verification, the DAP initiates and performs the maneuver to bring the CM into entry hypersonic-trim attitude at the specified roll attitude.

When the ENTRY DAP determines that the maneuver to entry attitude is essentially completed, P63 is begun.

C.4.2.3 Entry Initialization, P63

The Entry Initialization Program activates the entry guidance and establishes the pre-0.05g guidance-monitoring display. During P63, the DAP continues to hold the CM to the hypersonic trim with respect to the computed relative wind axis, as the CM approaches the atmosphere. When the guidance senses that the atmospheric drag level has exceeded 0.05g, P64 is begun.

P63 is the last P60 mission control program (until P67) to actually perform a program-sequence function. By comparison, P64, P65, P66, and the initial portion of P67 are merely milestones established by entry guidance along the trajectory.

C.4.2.4 Post-0.05g, P64

The Entry Post-0.05g Program establishes the guidance-monitoring display for the crew, initiates the lateral guidance and performs the trajectory planning function by finding a candidate reference trajectory that is compatible with the range to be flown. The guidance decides that the current entry is from an earth orbit if the velocity is less than 27,000 ft/sec at 0.05g (the start of P64); lateral steering is activated and, when the drag exceeds 0.2g, P67 is begun. The guidance decides that the entry is from lunar orbit if the velocity is greater than 27,000 ft/sec at 0.05g; lateral steering is activated, the trajectory search is begin, and the guidance directs the CM to fly to a constant drag level, about 4g. The guidance looks for a reference trajectory that satisfies the range requirement, using some or all of the trajectory segments: Upcontrol phase, ballistic phase, and final phase. Throughout P64, the DAP follows roll commands issued by entry guidance, and performs coordinated rolling with rate-damping pitch and yaw. Except as mentioned in P66, the DAP continues this behavior throughout the flight. When constant-drag flight has reduced the predicted range to within 25 nmi of the desired range, P65 is begun. If an Upcontrol reference trajectory is not found, for which the terminal velocity exceeds 18,000 ft/sec, then P67 is begun.

C.4.2.5 Upcontrol, P65

The Entry Upcontrol Program establishes the guidance-monitoring displays for the crew. P65 is flown for all lunar entries except for those where the landing-site range is short. During P65, the guidance steers the CM along the pull-up reference trajectory toward a possible controlled ballistic lob. When the termination conditions of the Upcontrol reference are met, the phase is over:

a. When the drag acceleration becomes less than the terminal drag condition, P66 is begun.

b. If the rate of descent becomes negative while the velocity has nearly reached the terminal velocity conditions, then pull-up flight is ended and P67 is begun.

(For intermediate-range trajectories, the terminal drag condition may be greater than 0.2g, and the net effect of satisfying the terminal drag condition is a direct transfer to P67 by way of P66.)

C.4.2.6 Ballistic, P66

The Entry Ballistic Program establishes the guidance-monitoring displays for the crew. P66 signifies the ballistic lob portion of the trajectory and lasts as long as the atmospheric drag level is less than 0.2g. Because of the rarified atmosphere, the guidance ceases to produce new values for the desired lift-vector direction. While the drag level remains greater than 0.05g, the CM desired roll attitude is the most recent roll command (from P65); pitch and yaw hypersonic trim are still maintained by the atmosphere. If the drag level becomes less than 0.05g, pitch and yaw attitude control by the DAP is activated to maintain the CM in hypersonic trim (as during P62 and P63); in addition, a wings-level, lift-up roll attitude is established for the CM by providing a zero roll-angle command. When the drag again exceeds 0.05g, the DAP reverts to pitch and yaw rate dampers. Whenever the atmospheric drag exceeds 0.2g, P67 is begun.

C.4.2.7 Final Phase, P67

The Entry Final Phase Program establishes the guidance-monitoring displays for the crew. P67 is the terminal portion of all entry trajectories. The guidance steers the CM to the landing site by generating the desired lift direction based on perturbations away from an onboard stored reference trajectory. The guidance also prevents the load factor from exceeding 8g. When the relative velocity decreases to 1000 ft/sec, the guidance ceases to generate new steering commands, but maintains the most recent value for the CM terminal roll attitude. P67 is terminated at crew option following drogue-chute deployment, although for telemetering coverage, termination is delayed as long as possible.

C.5 Lunar-Landing Guidance and Navigation

One of the most important phases of the Apollo mission is the guidance and navigation of the Lunar Module during the deceleration maneuvers prior to touchdown on the lunar surface. This section discusses the guidance and navigation capabilities onboard the LM for this powered-landing phase, which is presented in the context of the complete mission in Section 2.2.2.7.

The basic function of the LM guidance and navigation systems during powered landing is to take the spacecraft from a nominal initial altitude of about 50,000 ft and a velocity of approximately 5600 ft/sec and safely land it at an assigned site on the moon with virtually zero touchdown velocity. Several conditions and constraints govern the means by which the powered landing is accomplished:

a. The Descent Propulsion System (DPS) propellant must be utilized in an efficient manner, i.e., the required velocity increment should be as small as possible.

b. The selected landing site must be visible to the astronaut through the window of the LM for a time interval of at least 75 sec immediately prior to touchdown.

c. New state-vector estimates and steering commands for the LM cannot be obtained more frequently than once every 2 sec.

d. The DPS must be operated either at a fixed high-throttle setting (close to maximum thrust) or as a continuously-throttleable engine over a limited range of lower throttle settings—with the direction of applied thrust essentially parallel to the longitudinal axis of the LM.

e. The astronaut must have the capability of manually redesignating the landing site during the interval when the site is visible. (Conditions prevailing at 500-ft altitude must be "comfortable" such that manual takeover can be accomplished with ease.)

The navigation and guidance systems each perform different functions during the powered landing maneuver, as indicated in Fig. C.5-1. The navigation system basically determines (estimates) the state of the vehicle, i.e., its position and velocity. The guidance system uses the navigation information to compute specific-force commands for use in steering the vehicle. The following two parts of this section describe the guidance and navigation systems which enable the LM to accomplish the stated objectives during the powered-landing maneuver.

C.5.1 Guidance System Description

The mission requirements of efficient fuel utilization and a 75-sec landing-site visibility interval during the powered-landing maneuver are in direct conflict. For best fuel management during the powered maneuver, the Descent Propulsion System should be operated at the largest permissible thrust level, with the thrust direction slightly above the local horizontal. For the astronaut to have visibility of the landing site through the LM window, however, the longitudinal axis of the vehicle (and hence, the thrust direction) must be in a nearly vertical orientation. Continuous throttle control is required during the final part of the landing maneuver, moreover, to properly shape the trajectory (to meet visibility requirements) and to achieve the desired terminal conditions (position and velocity) at touchdown.

To accomplish these objectives in an efficient manner, the landing maneuver is divided into three major phases, as indicated in Fig. C.5-2. For convenience

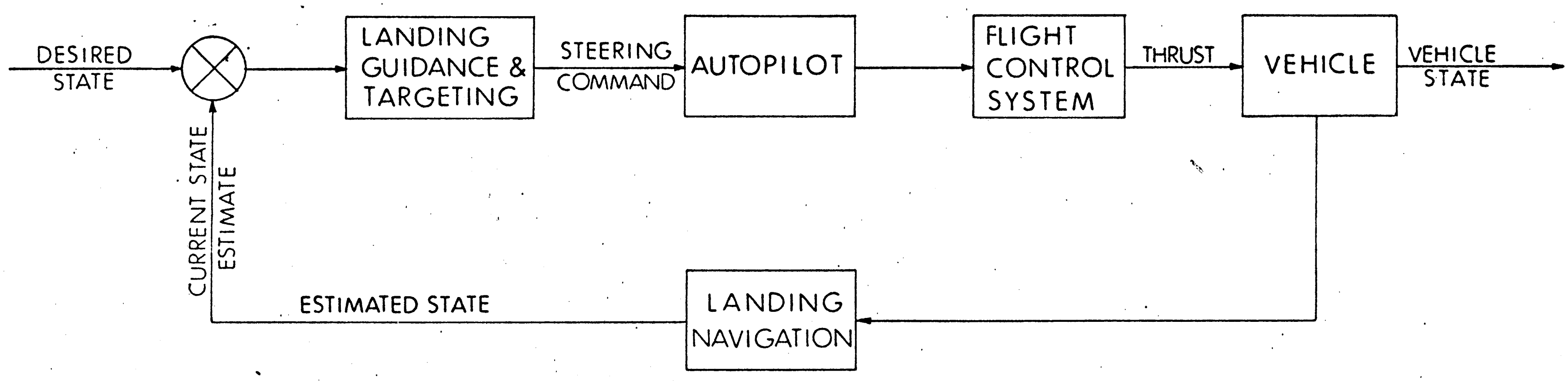


Figure C. 5-1 Functional Diagram Showing Relationships between LM Guidance and Navigation Systems

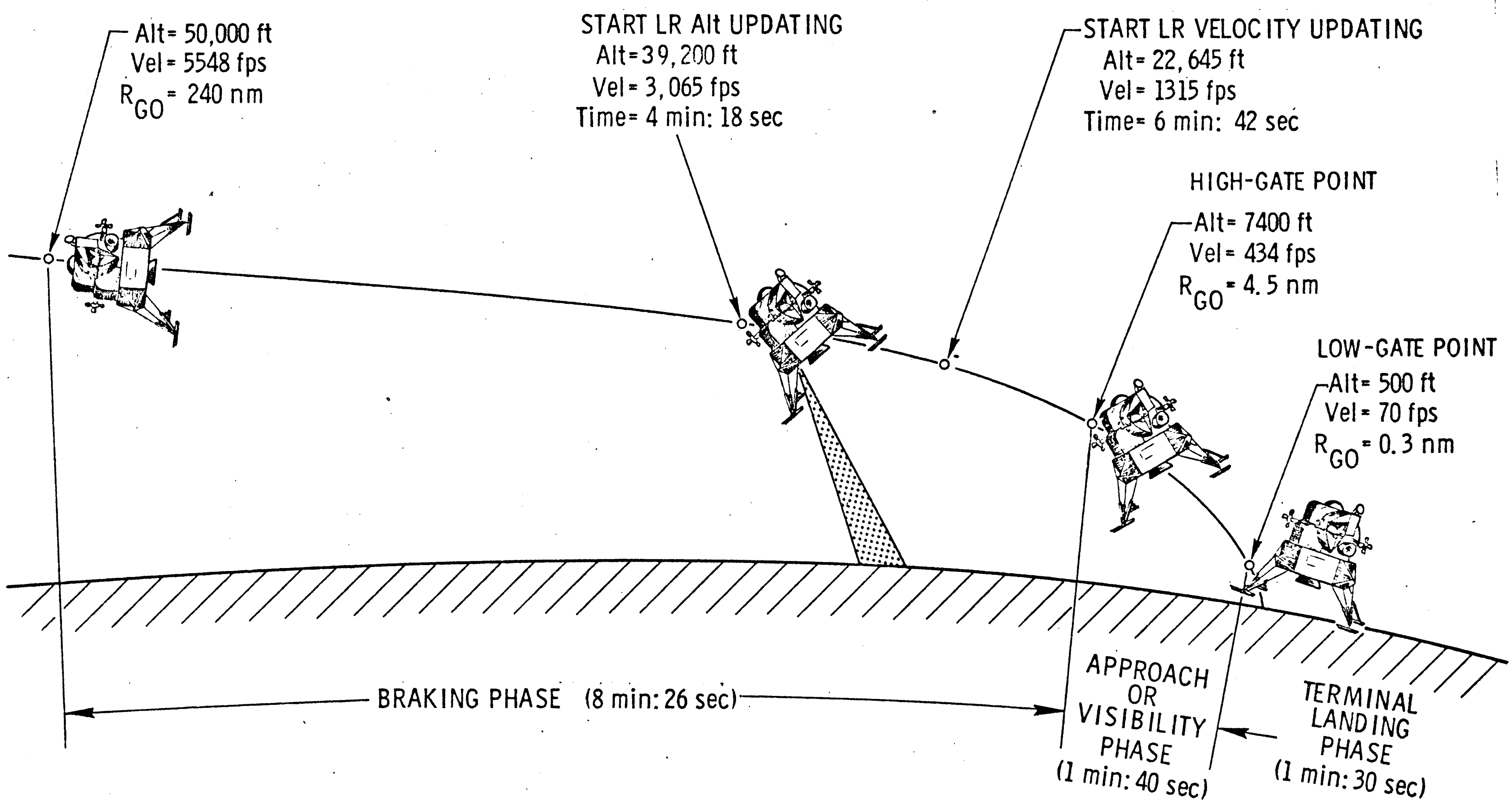


Figure C. 5-2 Powered Lunar-Landing Trajectory Phases these three phases are referred to as the "braking" and "visibility" phases, and the terminal-descent (hover) maneuver. The major deceleration of the vehicle is accomplished during the braking phase, which is typically about 470 sec in duration. The braking phase is the longest with respect to time and range covered. The major objective of this phase is to establish the desired initial altitude and velocity conditions for the following visibility phase within efficient propellant usage limits; the bulk of the spacecraft's orbital velocity is removed during the braking phase. The visibility (approach) phase is designed such that the commanded vehicle attitude at reduced engine-throttle setting allows the astronaut to view the landing area for the first time. During this second phase the astronaut can—if he so desires—redesignate the landing site toward which the GN&C System is controlling the Lunar Module. A furthur objective of the visibility phase is to establish and maintain a trajectory from which the astronaut can easily take over control when he desires. During the third and final terminal-descent phase, the astronaut manually controls the LM through the final vertical descent to lunar touchdown. For convenience, the braking-phase terminal point is referred to as the "High-Gate"* point, and the visibility-phase terminal point is called the "Low-Gate" point. The terminal point for the hover is, of course, the landing site.

The guidance system solves the powered-landing guidance problem essentially as successive two-point boundary-value guidance problems. First of all, the guidance system executes an algorithm which determines the exact time and spacecraft attitude required at ignition. Ignition nominally occurs at the periapsis of the prior orbit. The spacecraft is maneuvered to that desired attitude and 35 sec prior to lighting the engine, the DSKY is blanked for five sec to notify the astronaut that everything is proceeding smoothly. Ullage is commanded for 75 sec prior to ignition—the beginning of the braking phase. For 26 sec the engine operates at minimum thrust, about 12 percent of the engine's rated thrust of 10,500 pounds.

---

*The "target" for the braking phase lies very near the lunar surface, projected forward about 62 sec past High Gate, but the phase actually ends at High Gate. The reason for avoiding the designated target (other than the obvious intersection with the lunar terrain) is that as time approaches zero with the guidance equations, the gain of the guidance equations approaches infinity. To avoid that gain variation, the targets of the braking phase are projected about a minute downstream from the desired braking-phase terminus. For the same reason, the target for the visibility phase lies some 10 sec beyond Low Gate.

With sufficient propellant, the guidance can permit substantial landing-site redesignation; as much as 7000 ft forward or to either side, or about 4500 ft backward. The guidance system requires no fixed landing site and indeed provides a relatively gentle standard approach to Low Gate whether the landing site has been redesignated or not.

Nominally, the descent trajectory is planar; however, redesignation results in a nonplanar trajectory. The descent trajectory during both the braking and visibility phases is provided by a vector polynominal which determines a three-dimensional line in space.

During almost the entire visibility phase, the guidance system maintains spacecraft attitude so that the landing site nearly coincides with the reticle (Landing Point Designator) etched on the Lunar Module window; the computer displays on the DSKY a number which informs the astronaut where along this reticle to look to see where the computer is taking him.

When the Low-Gate aim conditions have been met, the Lunar Module begins the so-called terminal-descent maneuver, which is nearly vertical. The point where this maneuver is started is dependent upon mission ground rules, crew option, and the erasable load provided for the specific mission.

The terminal-descent program in Apollo 11 and 12 automatically nulled the horizontal components of velocity and provided a 3 ft/sec rate of descent. In Apollo 13 and subsequent flights, the astronaut must specify the rate of descent by means of his rate-of-descent (ROD) controller; there is also an attitude-hold mode in which spacecraft attitude (and hence horizontal velocity) can be manually controlled—not necessarily nulled—to provide the desired translation across the lunar terrain.

The fuel allotment of the DPS provides for hovering immediately prior to touchdown. If the spacecraft does not enter this terminal descent phase, an abort is initiated on the ascent stage (see the next section of this Appendix).

Thrust magnitude and orientation time histories are shown in Fig. C.5-3 for a typical simulated landing trajectory. As can be seen, the throttle is operated at maximum thrust for most of the braking phase. During this period the DPS is

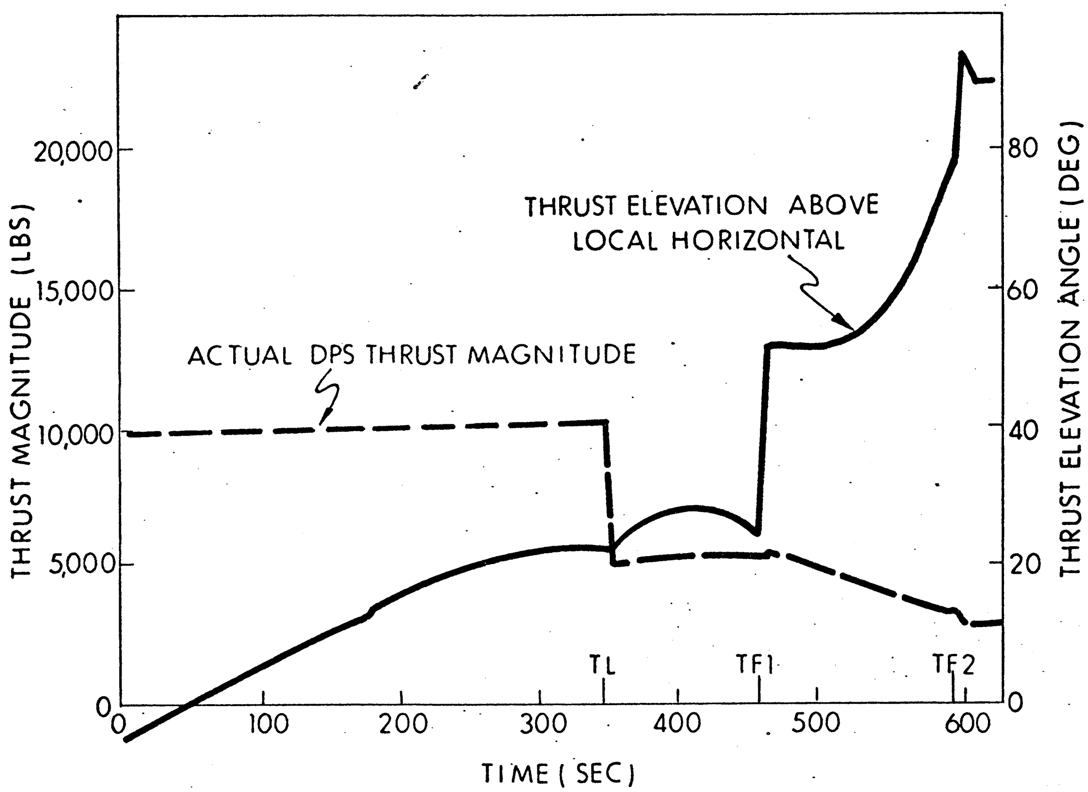


Figure C.5-3 Thrust Profile for Reference Trajectory oriented along the direction of the command specific force even though, due to limitations*, the throttle cannot follow the thrust command. The DPS is operated thereafter in the continuously-throttleable range between 12 and 58 percent of maximum thrust.

C.5.2 Navigation System Description

The navigation concept used by the LM GNC&S controls the braking and visibility phases of the lunar-landing maneuver and then continues to provide altitude and velocity data to the astronaut in the final hover phase. This approach employs an inertial navigation system updated by a doppler landing radar. The first three to four minutes of the braking phase is completely controlled by the inertial navigation system. When the altitude has decreased to between 40,000 and 30,000 ft, as illustrated in Fig. C.5-2 for the Apollo 11 mission, the landing radar is activated and initial altitude updates are accepted by the navigation system. When the velocity has decreased to less than 2000 ft/sec, landing-radar velocity corrections are next incorporated. It is important that the landing radar altitude and velocity updates be achieved during the braking phase for two reasons: first, to guarantee a safe altitude condition at the start of the approach phase, and second, to make the major trajectory changes before the visibility phase, so that the commanded vehicle altitude will be relatively free of control transients during this second phase, thus allowing the astronaut to visually determine and evaluate the landing area and have time to redesignate the landing site, if necessary. The visibility phase lasts only 100 sec or less, so major navigation-update changes resulting in large commanded vehicle attitude changes are undesirable.

The landing-radar sensor used to update the inertial navigation system during the landing maneuver is a four-beam doppler radar with the beam configuration shown in Fig. C.5-4. Three beams are used for velocity determination and one for range. The landing radar antenna is mounted at the base of the LM descent stage and can be oriented in one of two fixed positions. The first antenna position is used during the braking phase when the vehicle attitude is essentially horizontal,

---

*Due to a hard mechanical stop, the engine is incapable of delivering more than 94 percent of the rated thrust.

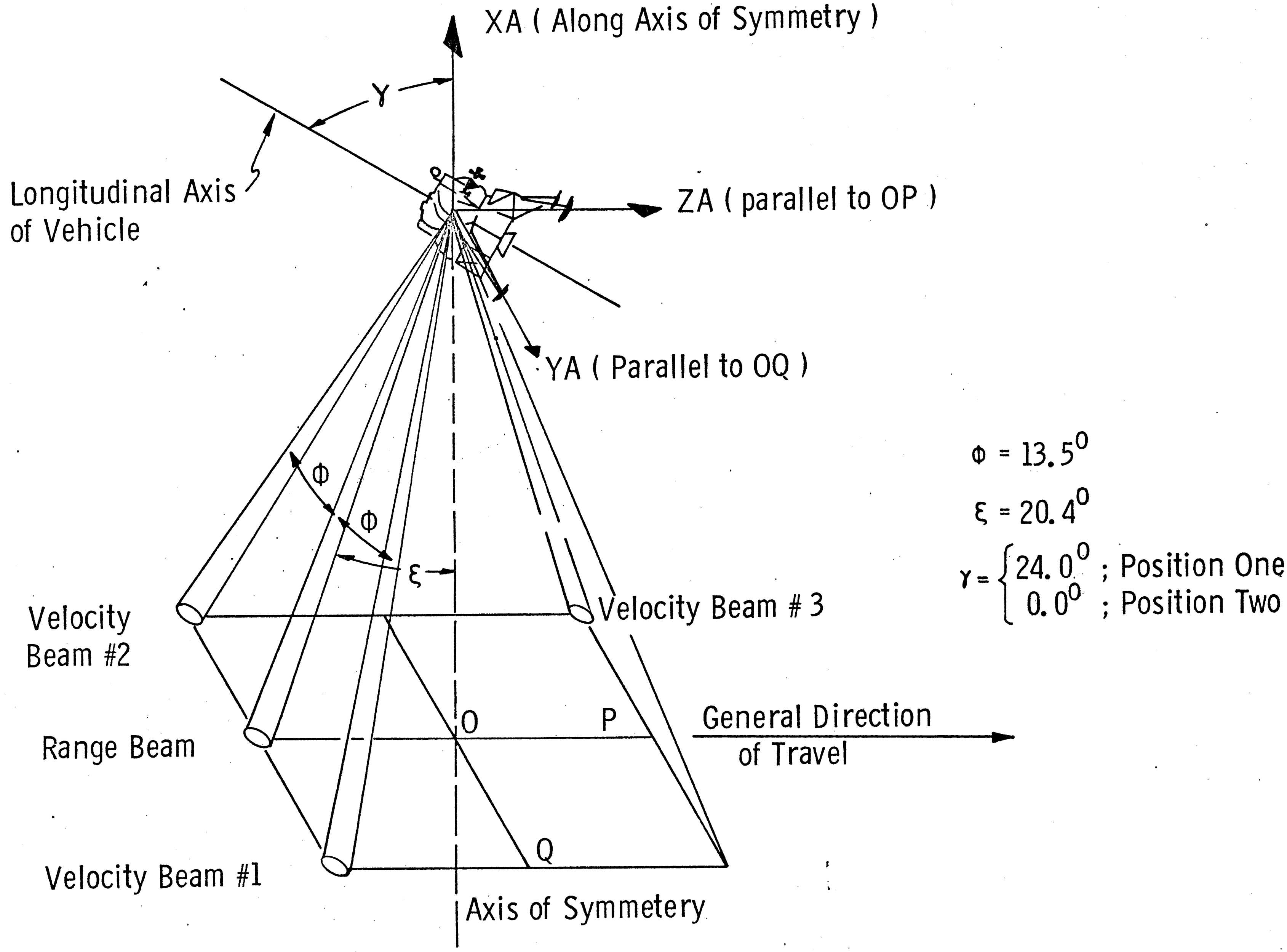


Figure C. 5-4 Landing-Radar Beam Geometry and the second is used in the approach and landing phases as the LM orientation nears a vertical attitude.

The navigation problem during the powered landing is significantly different from that during the coasting cislunar and rendezvous navigation phases discussed in Appendix A. In addition to gravitational forces, a large thrust acceleration acts on the vehicle at all times during the landing maneuver. Another significant difference is that the guidance system controlling the vehicle thrust and attitude continuously uses navigation data throughout the maneuver, as opposed to the intermittent updates maneuvers typical for cislunar and rendezvous phases. Despite these major differences, the general navigation concept used during landing is very similar to that used during coasting phases.

A simplified landing navigation and control functional diagram is illustrated in Fig. C.5-5. During the landing maneuver, the inertial measurement unit is active at all times and provides the specific force data necessary for extrapolation of the state-vector estimates. During the initial part of the braking phase, the IMU is the only navigation sensor employed, and the estimated vehicle-state vector is used to command required vehicle attitudes and and engine throttle levels so that the desired terminal conditions for the mission phase are achieved. When the landing radar is activated later in the braking phase, the vehicle altitude above the lunar surface is estimated in the computer and then compared with the landing-radar measured range data converted to altitude. The difference between these two parameters is automatically checked in a landing radar (LR) data test to verify that the LR is operating normally. If the data check satisfactorily, a correction term is computed by applying a precomputed weighting factor to the altitude difference*, as shown in Fig. C.5-5. This correction is then used to update the state-vector estimate and to correct the landing trajectory in the next guidance-equation computation cycle. After velocity lock-on is achieved by the LR at a lower altitude and velocity, a similar operation is conducted for velocity updates to the state vector. When both altitude and velocity updating are being done, the navigation-measurement data from the LR are controlled by the onboard computer. Altitude is updated every two sec, and each component of velocity every six sec.

*If the LR data test is failed a given consecutive number of times, the DSKY alerts the astronaut to this fact; he can force acceptance of the rejected data should he desire, but this option would not normally be used.

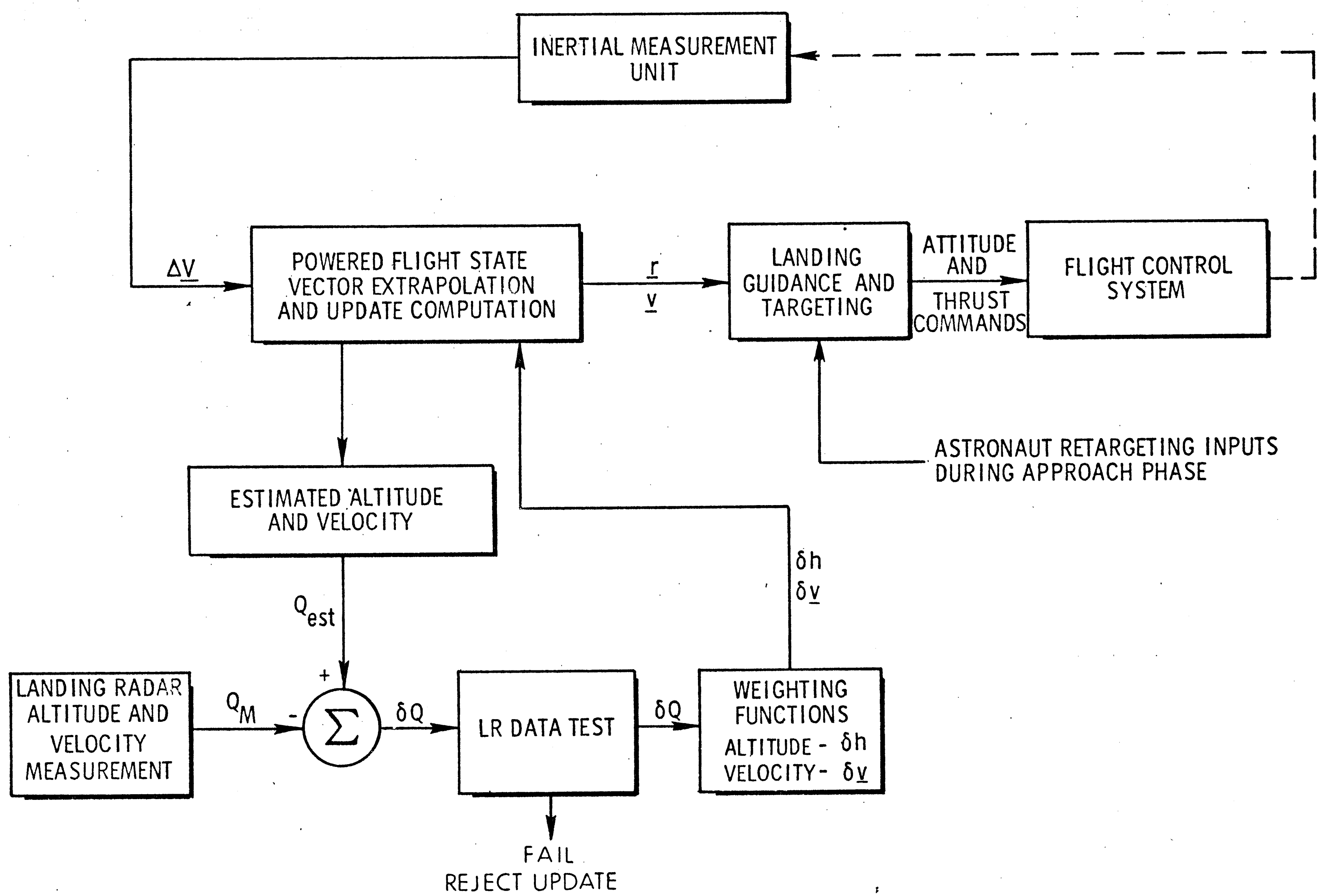


Figure C. 5-5 Simplified Functional Diagram of Lunar-Landing Navigation and Control

With reference to Fig. C.5-5, it should be noted that LR measurement data are used to update only four of the six components of the estimated state vector, i.e., altitude and the three velocity components. The down-range and cross-track position estimation errors are not updated by LR measurements. The astronaut can correct these two horizontal-plane position errors during the visibility phase by incrementally retargeting the guidance equations. In this operation, the landing site to which the GN&CS is controlling the vehicle is displayed to the astronaut by a DSKY number referenced to a grid pattern on the LM window. As previously mentioned, it is important that the major altitude and velocity corrections to the state vector be completed before the visibility phase so that the astronaut can effectively assess the landing area and correct the cross-track and down-range deviations, if required.

The weighting functions used to compute the state-vector updates in Fig. C.5-5 are significantly different from the time-varying statistical weighting functions used in the cislunar and rendezvous navigation phases. These powered-landing weighting functions are linear approximations to the statistical optimum navigation-filter weighting functions based upon inertial and LR sensor accuracies, lunar terrain uncertainties, and measurement bias errors. Since computation time is a critical parameter during the landing maneuver, the LR weighting functions are precomputed and stored. The altitude weighting function is a linear function relative to estimated altitude. Since altitude updates are typically started some 90 nmi from the landing site, lunar terrain altitude variations relative to the landing site were important factor in shaping this function. The velocity weighting function is linearized with respect to estimated velocity; it is truncated to fixed small values at the low-velocity conditions to minimize commanded attitude changes during the more critical terminal phases of the landing maneuver, and to avoid incorporating large LR velocity updates in the velocity region where LR dropouts can occur due to near-zero velocity conditions along various radar beams. The LR measurement weighting functions are uncoupled or noncorrelated with any measurement direction other than that along which the LR measurement is being made. Whereas correlation in the navigation-update weighting functions is very important in the cislunar and rendezvous navigation concepts, it is intentionally avoided in the landing maneuver navigation since implementation of such a correlation function was not considered practical due to modeling uncertainties and G&N system computation time limitations in the landing maneuver.

Several important differences exist between the operations of power-landing navigation and those of cislunar and rendezvous navigation. First, the state-vector updating procedure and monitoring in powered landing is completely automatic, since the navigation is time-critical. Next, the navigation-measurement weighting function used for the landing navigation is a precomputed linearized approximation to a statistically optimum weighting function, and is uncorrelated between state-vector components. Finally, the landing-navigation function updates only four of the six state-vector components, and the astronaut is required to manually correct the remaining horizontal position deviations during the final two phases of the landing maneuver.

C.6 Lunar Ascent and Abort Guidance

Once the Lunar Module has separated from the Command Module for the descent to the lunar surface, a means must exist for the LM to ascend back into a parking orbit preparatory to the rendezvous and docking of the two spacecraft. In the context of the Apollo mission, the ascent can occur in the planned circumstance, after a safe touchdown on the lunar surface; or in the unplanned circumstance, when an emergency situation exists. Whether the ascent be nominal or abort, the guidance equations which determine the ascent maneuvers are identical; only the initial conditions fed into the guidance equations are different.

By definition, nominal ascent can occur only after the crew of the LM has signalled to the computer an "acceptable" landing on the surface. By extension, nominal ascent guidance is the program used for any ascent from the lunar surface after an acceptable landing—even if an emergency ascent is deemed necessary prior to the planned time of liftoff.

Also by definition, an abort guidance program is used for ascent in two general situations: (1) when an emergency develops during descent, prior to touchdown on the lunar surface; and (2) when the crew does not acknowledge the touchdown as acceptable, choosing instead to ascend as quickly as possible.

The ascent and abort guidance programs are essentially open loop, in that they receive no positional updates from navigation. The programs perform their guidance function exclusively from signals received from the Lunar Module's Inertial

Measurement Unit. Initialization for ascent to the required parking orbit is dependent upon prevailing conditions.

On the basis of the initial conditions, the computer determines which of the LM's propulsion systems will be employed during ascent. Two systems are available, and either or both may be employed for a particular ascent—depending, of course, on the fuel available in each and the estimated fuel which will be required to reach the planned parking orbit. The Ascent Propulsion System (APS) is used alone for an ascent from the lunar surface or for an abort occurring very close to touchdown. The Descent Propulsion System (DPS) is used alone only when its remaining fuel volume is sufficient to propel the LM into the parking orbit—up to about 300 sec into the braking phase of the lunar descent. When an abort is called at a later time, the DPS will be flown to depletion; parking-orbit injection will then be accomplished using either the APS or the RCS. The foregoing procedures must, of course, be modified should the abort be caused by a propulsion-system failure.

A flag (LETABORT) internal to the lunar descent programs permits aborts to be called once the lunar programs have been activated. Throughout the descent phase, the crew can call up abort guidance; this remains true until after touchdown. Once the crew decides that a safe landing has occurred and that surface operations can commence, they confirm it to the AGC by calling the lunar-surface program, which cancels the LETABORT—and leaves the nominal ascent program as the sole means of achieving ascent.

In a nominal ascent, the target of the ascent is planned well ahead of time, based on the state vector of the orbiting Command Module. The LM must lift off the lunar surface at such a time and with such velocity that it will insert into a parking orbit with a favorable position and velocity relative to the CM. Knowing the state vector of the Command Module, the ascent program selects the proper liftoff time.

Initialization for an ascent from the lunar surface is straightforward because it is known in advance. But initialization for an abort during descent is quite involved because neither the necessity nor the time of an abort can be known in advance. In addition, the phase angle—the angular relationship between the LM and the CM—at the time of the abort is equally unknown and must be estimated at initialization.

Consequently a variable targeting system must be employed for an abort, based upon the past information of the CM state vector and an estimate of the LM's current position.

Initialization establishes the initial conditions, sets up initial displays, decide on which targets to use, and brings up the guidance equations into the servicer loop. In all three ascent programs, the initialization routine does this by putting an address into an erasable location toward which the end of the servicer routine goes and which it recognizes as an address; the routine then continues on with the guidance.

Abort initialization also establishes the parking-orbit injection in the radial and cross-range directions. (With a fixed-thrust engine, if these two positional components are controlled, the third component—down-range position—is indirectly controlled.) Depending on the propulsion system which is used to begin an abort, two abort guidance programs are available. The descent abort program begins with the DPS, and the abort-stage program uses only the APS.

No matter which ascent program is called, once the initialization of the program is completed, the initialization routine simply slips out of the loop.

The ascent trajectory for either a nominal launch or an abort is preplanned: the LM thrusts to a vertical ascent for about 0.8 sec, after which it begins a pitch-over maneuver and accepts new data from the IMU. The importance of the 0.8-sec vertival ascent becomes apparent in the case where, upon touchdown on a boulder-strewn surface, the LM begins to topple or if, during descent, the LM begins to tumble.

Once the guidance equations take over control of the program, they produce a commanded thrust vector based on the inertial platform's accelerometer readings and the newly-inputted CM state vector. The thrust-vector components are converted into changes in the CDU angles by a subroutine called FINDCDUW; these angles are then transmitted to the Digital Autopilot. Thus, the LM DAP is the control system during either nominal or abort ascent.

Ascent into the parking orbit required for rendezvous ends when the target velocity commanded (in radial and cross-range positions) is achieved parallel to the plane of the Command Module orbit. At that point, the rendezvous porgrams take over.

C.7 FINDCDUW — A LM Powered-Flight Guidance/Autopilot Interface Routine

As explained in the previous sections of this Appendix, the LM navigation and guidance systems perform complementary functions during the powered-flight maneuvers. Basically, navigation estimates the position and velocity of the Lunar Module, and guidance uses this information to compute the commands needed to steer the vehicle. The guidance-system commands are computed in terms of thrust vectors and must be translated into angular commands to be accepted by the LM DAP. The routine devised to accomplish this conversion is called FINDCDUW. (The "W" in FINDCDUW represents "window control".

Specifically, FINDCDUW aligns the LM thrust vector with the commanded direction during powered maneuvers and aligns the reticle in the window in the direction commanded by the guidance elevations. During descent, the reticle is aligned with the landing site; during ascent, it is aligned in the forward direction of motion.

FINDCDUW receives from the guidance program thrust and window-pointing commands which determine a required spacecraft attitude. Gimbal-angle changes corresponding to this attitude are computed and limited in magnitude to the maximum which may be commanded in one guidance-program computation period (20 deg in 2 sec). The limited gimbal-angle commands are then divided into 20 increments, each one of which is equal to the change which can be expected in one DAP cycle (0.1 sec). Finally, the gimbal-angle increments are sent to the autopilot, along with the corresponding attitude-rate commands and a bias angle whose magnitude is computed from the commanded rates and the control authority. The bias angle is utilized to smooth transient behavior in the autopilot. All of these constraints upon the autopilot commands result in a continuous, rate-limited attitude profile extending over whatever duration is required to complete the maneuver.

There is further constraint imposed upon the angle of the middle gimbal to avoid gimbal lock*. Provided the middle gimbal angle is not initially in gimbal lock, and provided the combination of input commands does not yield a terminal attitude in gimbal lock, FINDCDUW renders it impossible to pass through gimbal lock by constraining the middle gimbal angle to the range between its initial and terminal values. Furthermore, by avoiding gimbal lock, FINDCDUW makes the GN&CS abort guidance not dependent upon manual intervention. Other schemes for gimbal-lock avoidance produce similar outcomes, but are computationally less direct.

---

*A three-degree-of freedom gimbal system such as the one used on the Apollo IMU can cause problems due to a phenomenon called "gimbal lock". Gimbal lock occurs when the IMU's outer axis becomes parallel to its inner axis. In this position, all three axes of gimbal freedom lie in a plane, and no axis is in a direction to absorb instantaneously rotation about an axis perpendicular to this plane. Thus, at gimbal lock the inner stable member can be pulled away from its inertial alignment. Even though a three-degree-of-freedom gimbal system geometrically allows any relative orientation, the required outer-gimbal angular acceleration needed at gimbal lock to maintain stabilization exceeds servo capability. One direct solution to gimbal-lock problems is to add a fourth gimbal and axis of freedom which can be driven so as to keep the other three axes from getting near a common plane. However, the cost in complexity and weight for a fourth gimbal is considerable. Fortunately, Apollo IMU operations are such that gimbal lock can generally be avoided by constraining spacecraft motion relative to the gimbals. If just one of these gimbal-angle changes is constrained, the problem is entirely averted.

APPENDIX D

MAJOR PROGRAM CAPABILITIES— Digital Autopilots

D.1 Developmental History of the Digital Autopilots

Early in 1964, MIT was asked by NASA to determine the feasibility of implementing digital autopilots (DAPs) in the guidance computers. This request was occasioned by rather widespread and growing dissatisfaction with the degree of flexibility offered by the analog autopilot then being employed. (By chance, a switchover was impending from the Block I G&N system to the Block II system, with its enlarged computer memory and improved electronics CDUs.) Furthermore, it was felt that a digital autopilot as the primary control system, backed up by the preexisting analog autopilot, could provide far greater mission reliability than the analog autopilot alone, and that a loss of the nonredundant analog autopilot might have meant mission failure—or worse, loss of the crew and spacecraft.

But of the advantages to digital control, perhaps the greatest was the above-mentioned flexibility: the coding effort required to modify or replace autopilot functions, such as manual operational modes, could be bought more cheaply than corresponding changes to an analog system. In addition, the ease of implementing nonlinear functions, such as deadzones, parabolic curves and counters, reinforced the arguments for a digital autopilot. Although a digital autopilot had not yet been flown in any manned vehicle, digital controllers had been discussed in the control literature and had been flight-tested in at least one unmanned guided missile.

During the summer of 1964, MIT was given the go-ahead on DAP design and implementation. This decision followed MIT studies showing that digital controllers were not only feasible, but offered improvements in control performance as well. This conclusion led to the choice of the digital autopilot as the primary control system.

D.1.1 CSM DAPs

Control of the CSM and CSM/LM vehicles, via the CM Apollo Guidance Computer, involves three separate DAPs, one each for coasting flight, powered flight, and atmospheric entry. In addition, the AGC onboard the Command Module provides for takeover of Saturn steering during boost.

The coasting-flight DAP, which fires the Reaction Control System (RCS) jets for attitude control, provided significant improvements in both performance and flexibility over the Block I system. This was achieved by a number of new features, among them an improvement in the automatic-maneuver routine and the addition of several manual-control modes. By early 1965, the basic RCS autopilot functions were laid out, including phase-plane and jet-select logic, a new maneuver routine, and interfaces for the various manual modes. Along with the development of these functions, some additional features were implemented, such as the sharing of rotational and translational jets during some maneuvers (e.g., ullage), and a rate estimator employing Kalman filtering. Though the RCS DAP functions were not optimized in a rigorous sense, a primary concern was to use as little RCS fuel as possible for attitude maneuvers, since the fuel supply was a limited fixed quantity. Throughout the design effort and even into the flights, the RCS DAP design remained flexible to accommodate many modifications incorporated to improve the capability and performance of the system. RCS DAP functions are discussed in detail in Section D.2 below.

For powered flight, the problem was to maneuver the vehicle according to commands received periodically from the guidance loop. Changes are effected in pitch and yaw by thrust-vector control (TVC), i.e., by deflecting the thrust vector relative to the spacecraft by gimballing the Service Propulsion System engine mounted on the Service Module. Roll control, which could not be achieved with the SPS engine, requires the use of RCS jets with attendant control logic. Because the SPS engine-positioning servos are linear devices*, the TVC loops could be designed around linear constant-coefficient compensation filters. This design allowed analytical determination of the stability of the bending and sloshing modes, which

*The position and rate limits (4.5 deg and 9.0 deg/sec, respectively) on engine motion were large enough to ensure they would not normally be encountered.

proved to be a great asset during the course of the development. The two vehicle configurations, CSM alone and CSM/LM, were sufficiently different to warrant separate TVC autopilot designs. Naturally, each of these DAPs went through several iterations as better bending and slosh information became available. In fact, it was not until the Apollo 10 mission that the final TVC DAP design was flown. Section D.3 discusses the TVC DAP used for pitch and yaw changes and RCS roll control during powered flight.

The Entry DAP controls vehicle attitude from CM/SM separation, occurring about 20 minutes before atmospheric entry, through to drogue-chute deployment. This DAP entails three automatic control modes: (1) three-axis attitude maneuver and stabilization prior to encountering the atmosphere; (2) coordinated roll maneuvers in the atmosphere, which control lateral and longitudinal range by rotating the lift vector; and (3) pitch and yaw rate damping about the aerodynamically-stable trim attitude. The first Entry DAP, which was designed off-line during the Block I flights, was developed in an attempt to reduce the RCS fuel used in the atmospheric roll maneuvers. This consideration was important because roll maneuvers normally used more RCS fuel than the extra-atmospheric and rate-damping modes combined. The design was successful, and by the time the decision had been made to use digital autopilots, this first Entry DAP had proved it could significantly reduce RCS fuel usage significantly. Further improvements, especially in the phase-plane logic, allowed even greater savings, until by the first Block II flight (Apollo 7), the atmospheric-roll maneuvers used only about one-eighth of the RCS fuel that the Block I system would have used. The combined performance of the guidance/autopilot system during entry has been excellent, as measured by the small target miss-distances (averaging roughly one nautical mile). The Entry autopilot is discussed in Section D.4.

The Apollo Guidance Computer onboard the CM also includes provision for takeover of Saturn steering during boost, should inertial reference fail in the Saturn Instrument Unit. Until Apollo 13, takeover was provided as a manual function; more recently an automatic capability has been added. Section D.5 discusses AGC takeover of Saturn steering.

D.1.2 LM DAP

The design of the LM DAP differed considerably from that of the CSM DAPs, due mainly to differences in the configuration of the two vehicles. Aside from the fact that the LM would not experience atmospheric flight, the major difference was that all modes of LM coasting and powered flight, both for the descent and ascent stages, used essentially the same basic control logic; i.e., the LM DAP had to be far more integrated than the CSM autopilots. One of the main factors necessitating this integrated design was that neither the descent engine nor the ascent engine was intended to control vehicle attitude. The ascent engine was rigidly mounted, while the descent engine could be gimballed only at the very slow fixed rate of 0.2 deg/sec. This meant that RCS jet firings would be required for attitude control in LM powered flight as well as coasting flight.

Given this level of integration, the individual control modes had their own developmental histories. The state estimator, used to provide the DAP with information on angular rates and accelerations, was first implemented as a Kalman filter with time-varying gains. These gains were stored in tabular fashion in fixed memory. However, early testing revealed estimation errors resulting from perturbations such as propellant slosh and CDU quantization. To reduce these errors, the estimator was changed by replacing the table of time-varying gains with several constant gains, and introducing an additional nonlinear filter to reduce the perturbation on the attitude measurements.

The descent-engine control system was originally designed merely to keep the engine pointing through the vehicle center of gravity. This acceleration-nulling mode required only a knowledge of the vehicle's angular acceleration and the control effectiveness of the engine. The RCS jets would then be used for attitude control. It was soon noticed, however, that given a suitable control law, the engine's Gimbal Trim System was capable of providing full attitude control when vehicle rates and accelerations were low, thus saving RCS fuel. The control law chosen was a modified time-optional law.

As in the CSM, the LM RCS control laws for powered or coasting flight were based on phase-plane logic. This logic, which varies with vehicle configurations and flight conditions, includes such features as variable switch curves, biasing of the deadbands, and separate firing logic for large and small phase-plane errors. Section D.6 provides descriptions of the various functions of the LM autopilot.

D.2 CSM Reaction Control System (RCS) Autopilot

The Reaction Control System Digital Autopilot (RCS DAP), an integral part of the CSM GN&C System, provides automatic and manual attitude control and stabilization and manual-translation control of the Apollo spacecraft. The autopilot is designed to control four spacecraft configurations during the so-called coasting phases of flight—CSM/SIVB, CSM/LM, CSM/LM ascent stage, and CSM alone. For the latter three configurations, control forces and moments are provided by the Service Module Reaction Control System, which employs 16 rocket thrusters mounted in groups of four, known as quads; for the CSM/SIVB configuration, control forces and moments are provided by the SIVB RCS system. CSM control is achieved by jet-on and jet-off command signals supplied to the solenoids of the thrusters.

The RCS DAP receives and processes data from various internal and external sources. Measurements of spacecraft attitude are provided by the Inertial Measurement Unit (IMU) through Coupling Data Units (CDUs), which serve as gimbal-angle encoders. Attitude rate information is derived from these measurements. Manual attitude commands are generated by the Rotational Hand Controller (RHC) and interface with the computer through an input channel. The inputs are discrete* in nature, specifying the direction of rotation required about each of the spacecraft axes. The Translational Hand Controller (THC) interfaces in a similar fashion and provides translational-acceleration commands along each of the spacecraft axes. The minimum-impulse controller (MIC) interfaces with another input channel; each deflection of the MIC produces a single short rotational impulse (14 msec) of the SM RCS jets about the appropriate axes.

Selection of the various autopilot control modes is governed by a panel switch. Using the DSKY, the crew may also specify several autopilot control parameters, such as angular deadbands, maneuver rates, thruster-quad status and spacecraft-

---

*The AGC has 15 input and output channels whose bits are individually distinct (i.e., discrete). Each bit either causes or indicates a change of state, e.g., liftoff, zero optics, SPS-engine on, RCS-jet on, etc.

mass properties. Automatic attitude maneuvers are performed using internal steering commands provided by the guidance and navigation programs and initiated by keyboard request. In response to these steering commands, the RCS DAP issues jet-on and jet-off commands to the Reaction Control System and generates attitude error signals for display on the flight-director attitude-indicator (FDAI) error meters.

D.2.1 Modes of Operations

The RCS DAP may operate in one of three modes selected by the crew via the Command Module AGC MODE switch*. These three modes can be summarized as follows:

D.2.1.1 Free Mode

RHC commands are treated as minimum-impulse commands. Each time the RHC is moved out of detent, a single 14-msec firing of the control jets results on each of the axes commanded. If there are no RHC commands, MIC commands are processed. If neither RHC nor MIC commands are present, the spacecraft drifts freely.

D.2.1.2 Hold Mode

If there are no RHC commands, the vehicle is held about the attitude reached upon switching to hold or upon termination of a manual rotation. RHC commands override attitude hold and result in rotations at a predetermined rate on each of the appropriate control axes for as long as the RHC remains out of detent. When the RHC returns to detent in all three axes, all angular rates are driven to within a deadband; attitude hold is then established about the new spacecraft attitude. In the Hold mode, all MIC commands are ignored.

---

*THC commands are accepted in any mode and are combined with the rotation commands whenever possible. In the event of a quad failure, however, rotation control will assume priority over translation; i.e., translations are ignored if they would induce rotations that could not be compensated by RHC (or automatic) commands or if they were to cancel a desired rotation. The crew is responsible for performing ullage with the THC and by the selection and management of ±X-translation quads, as described below.

D.2.1.3 Auto Mode

If there are no RHC commands, the DAP accepts rate and attitude commands from the guidance programs to bring the vehicle to the desired attitude at the specified rate. RHC commands override automatic maneuvers and are interpreted as rate commands, as in the Hold mode. When RHC commands cease, automatic maneuvers are not immediately resumed, but rather attitude hold is established about the new orientation. The automatic maneuver can be resumed only by astronaut action via the DSKY. In the absence of automatic-maneuver commands, the DAP functions exactly as in the attitude-hold mode. In the Auto mode, as well, MIC commands are ignored.

D.2.2 Crew Control of the RCS DAP Configuration

D.2.2.1 DSKY Operation

D.2.2.1.1 Data Loading

Most of the autopilot variables over which the crew has control are loaded by means of Verb 48, which is normally executed before the DAP is turned on for the first time and anytime thereafter when the crew wishes to change or update the data. Verb 48 displays, successively, three nouns (only two are needed for the RCS autopilot) for loading and verification.

Noun 46 permits the loading of data relating to current spacecraft configuration; the choice of quads to be fired for X-axis translations; the size of the angular deadband for maneuvers in the Hold and Auto modes; and the specified rate for RHC activity in the Hold and Auto modes and for automatic maneuvers supervised by a special routine for coasting-flight attitude maneuvers, KALCMANU. In addition, Noun 46 includes information on jet selection for roll maneuvers and on the operational status of the quads.

Noun 47 allows the loading of the current weight (in pounds) of the CSM and of the LM. The spacecraft moments of inertia and other pertinent parameters, such as propellant loading and cg offset, are stored in the CM AGC as a function of these keyed-in weights.

Noun 48 allows the loading of data pertinent only to the TVC DAP.

D.2.2.1.2 Other DSKY Operations

After Verb 48 has been completed, Verb 46 may be used to establish autopilot control of the spacecraft. If the specified configuration is CSM alone, CSM/LM, or CSM/LM (ascent-stage only), and if the Thrust Vector Control DAP is not running, the RCS DAP begins initialization. If the specified configuration is SIVB/CSM, RHC commands are sent to the SIVB control system for manual rate control.

Verb 49 calls up R62, a routine that permits the crew to specify a final vehicle attitude. This attitude can be achieved by means of an attitude maneuver supervised by the special routine, KALCMANU.

Routine 61, the tracking-attitude routine, provides the RCS DAP with the information required to automatically track the LM during rendezvous navigation. Whenever R61 requires an attitude maneuver resulting in any gimbal-angle change of 10 deg or more, it will, after an appropriate interval, check a flagbit and, if appropriate, call the attitude-maneuver routine, R60.

Verb 79 calls R64, the barbecue-mode routine which is closely related to utilization of the RCS autopilot. R64 enables the crew to perform (1) passive thermal control, a roll about the X-axis of the stable member; (2) an orbital rate-drive procedure about the Y stable-member axis; and (3) deadband changing without the requirement for direct erasable loading. With Noun 16, the astronaut informs the computer of the time (ground-elapsed time) at which he wishes the X- or Y-axis maneuvers to begin. For that maneuver Noun 79 displays the commanded rate, the RCS DAP deadband, and the stable-member axis about which the maneuver will occur.

D.2.2.2 Attitude-Error Displays

The RCS autopilot generates three types of attitude errors for display on the FDAI error meters, all of which are updated every 200 msec.

Mode 1 displays autopilot phase-plane errors as a monitor of the RCS DAP and its ability to track automatic-steering commands. In this mode, the display is zeroed when the MODE switch is placed in the Free position.

Mode 2 displays total attitude errors to assist the crew in manually maneuvering the spacecraft to the spacecraft attitude (gimbal angles) specified in Noun 22. The attitude errors with respect to these angles and the current CDU angles are resolved into RCS control axes.

Mode 3 displays total astronaut attitude errors with respect to the spacecraft attitude (gimbal angles) in Noun 17 to assist the crew in manually maneuvering the spacecraft. The attitude errors with respect to these angles and the current CDU angles are resolved into RCS control axes.

D.2.3 RCS DAP Implementation

The RCS autopilot was designed to perform a number of functions during a mission:

a. Attitude hold and stabilization
b. Automatic maneuvering, including
   (1) large-angle spacecraft reorientations
   (2) automatic tracking of the LM
   (3) passive thermal control
   (4) orbital-rate drive
c. Manual attitude-rate control
d. Manual rotational minimum-impulse control

D.2.3.1 Attitude Hold and Stabilization

The RCS autopilot is formulated as a sample data system which, in the attitude-hold mode, attempts to null the computed set of body-attitude errors and the spacecraft angular velocity. Figure D.2-1 depicts the logic of this RCS mode. The input to the attitude-hold logic is a set of reference angles corresponding to the desired outer-, inner-, and middle-IMU gimbal angles. In addition to computing estimated body rates, during each 0.1-sec sampling interval, the DAP compares these desired angles with current CDU angles. These rates are derived from IMU-gimbal-angle differences, which are transformed into corresponding body-angle differences and smoothed by a second-order filter. For greater accuracy, the commanded angular acceleration of the RCS jets is included in the computation.

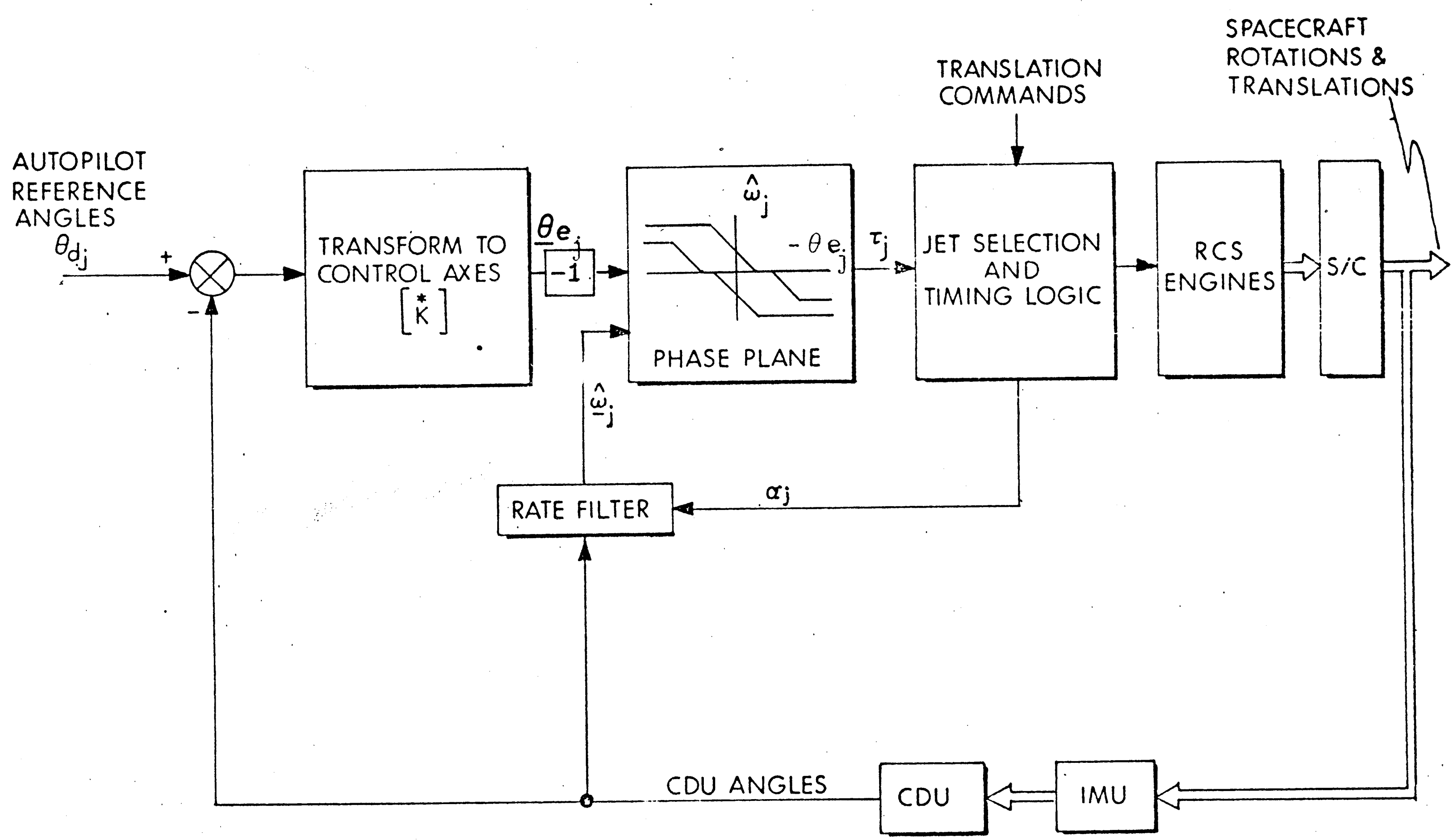


Figure D.2-1 Functional Diagram of CSM RCS Attitude-Hold Logic

As a function of the attitude error and the attitude rate, nonlinear switching logic (phase-plane logic) is used to generate RCS jet-on times, indicating the required polarity and duration of the thruster torque for each control axis. A jet-selection logic combines the rotation commands with the translation commands from the THC and selects the individual jets to be fired.

D.2.3.2 Automatic Maneuvering

The RCS autopilot can perform several different types of automatic rotations. Figure D.2-2 is a functional diagram of the automatic-control logic. The first of the automatic rotations is the ability to perform large-angle reorientations of the spacecraft, as required for main-engine (SPS) thrust-axis alignments prior to powered-flight thrusting maneuvers. A special steering routine performs these maneuvers in a fuel-efficient manner. The RCS DAP can also orient the CSM such that the LM is continuously within the optics field-of-view during rendezvous navigation. In addition, the autopilot can be used to establish the thermal-balancing roll rotation required to maintain uniform solar heating of the spacecraft during extended periods of drifting flight. The autopilot can be used to produce a rotation at an astronaut-specified rate about the Y stable-member axis. In earth and lunar orbit, if the Y stable-member axis is aligned normal to the orbital plane and if the specified rate corresponds to orbital rate, this capability provides a pseudo-local-vertical tracking mode.

D.2.3.3 Manual Attitude-Rate Control

The Rotational Hand Controller interfaces with the computer by means of discrete inputs which indicate a positive, negative, or zero rotation command for each control axis, in accordance with the duration of RHC deflection and some phase-plane switching logic. With rate-command control, the RCS autopilot causes a vehicle attitude rate to be generated in response to these commands. The magnitude of the command rates is specified by the astronaut in the autopilot data-load procedures (see Section D.2.2.1.1). Four rates are available for selection: 2.0, 0.5, 0.2 and 0.05 deg/sec. Figure D.2-3 provides the logic for RCS manual-rate control.

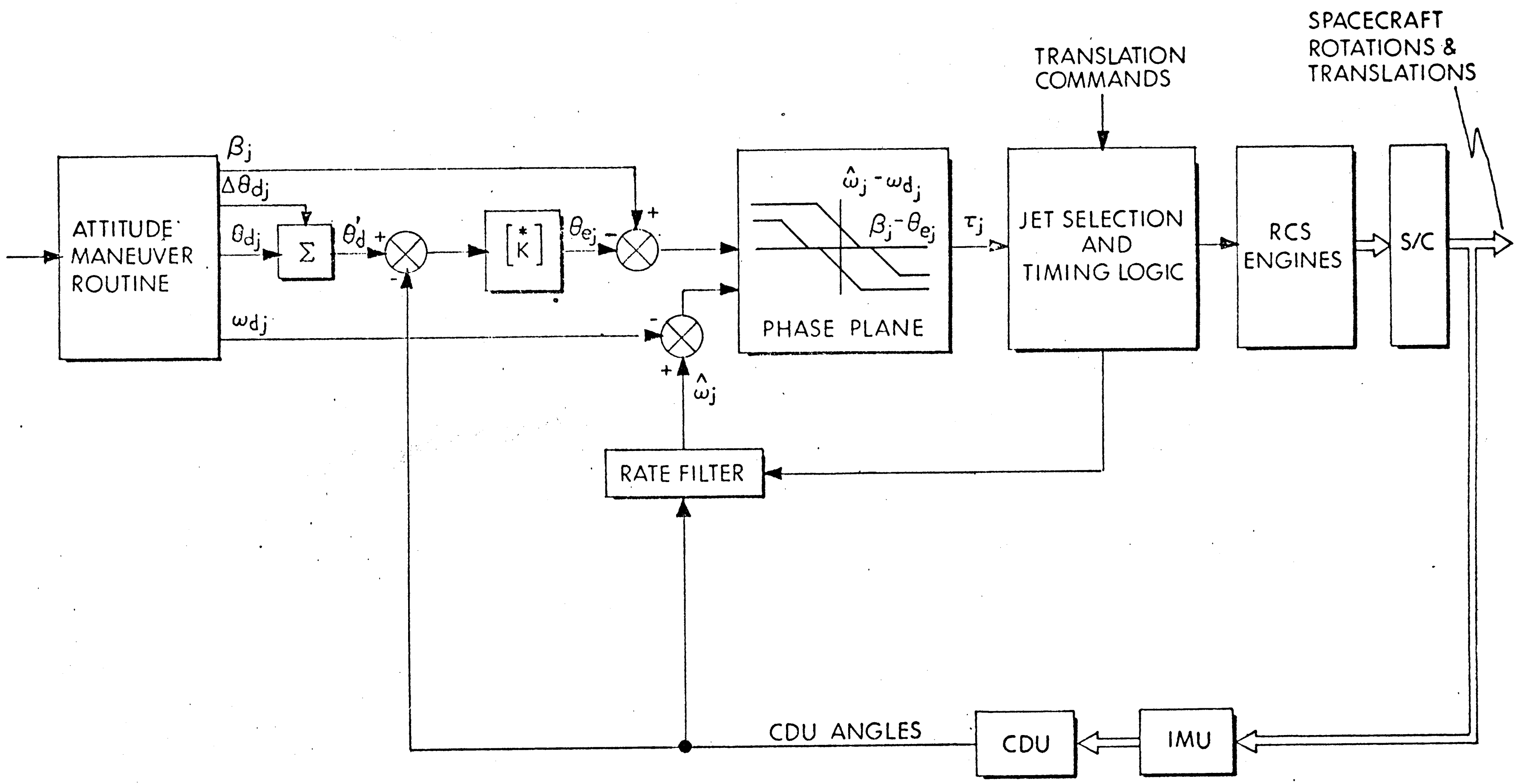


Figure D. 2-2 Functional Diagram of CSM RCS Automatic Control Logic

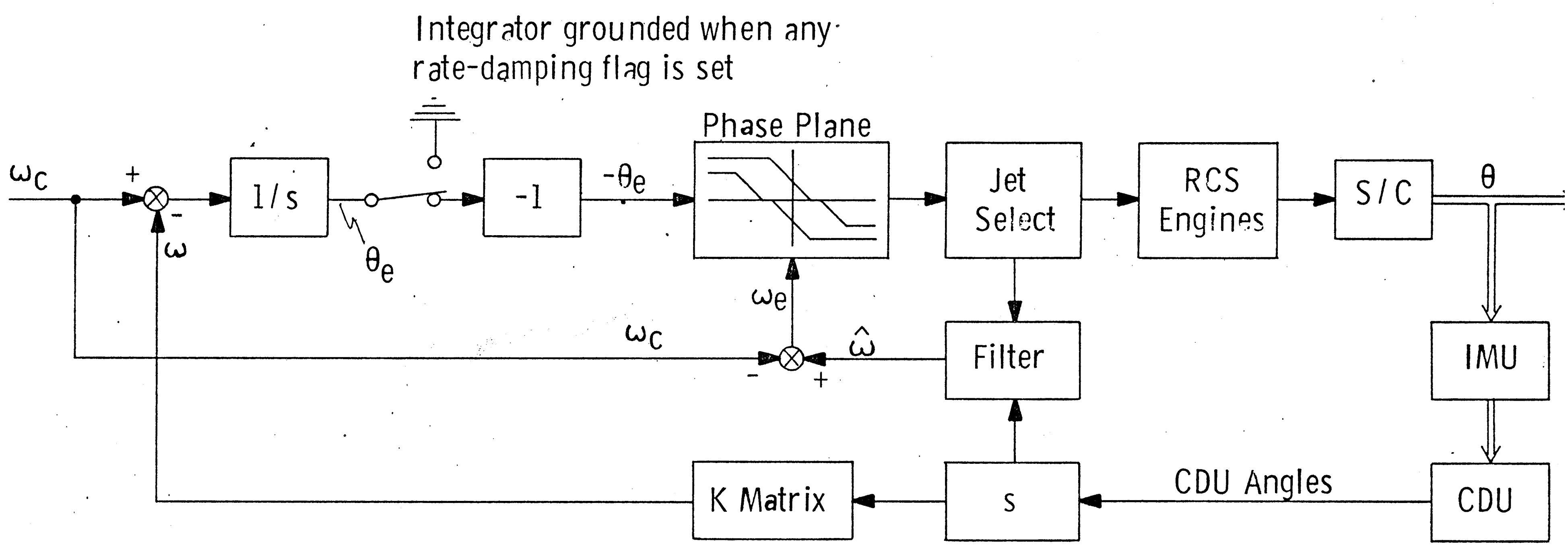


Figure D. 2-3 Functional Diagram of CSM RCS Manual-Rate Control

D.2.3.4 Manual Rotational Minimum Impulse Control

With manual rotational minimum-impulse control, a deflection of the MIC or the RHC at the navigation station produces small rotational commands to the RCS jets about each of the commanded axes. Each time the controller is deflected, the autopilot generates a jet-on command which lasts for 14 msec, thereby producing a small change in the spacecraft angular velocity about the appropriate axis. No other control action is taken until the controller is again deflected to produce another minimum impulse. This control is particularly useful for navigation sightings using the onboard sextant.

D.2.4 Restart Behavior of the RCS DAP

Should a restart occur during RCS DAP operation, any jets that happen to be on will be turned off, and reinitialization of the RCS DAP will be scheduled. This reinitialization is the same as the initialization caused by RCS-DAP turn-on using Verb 46, with the exception that the attitude reference angles are not changed. Automatic maneuvers governed by R60 that were in progress at the time of a restart will not automatically be resumed; rather, attitude hold will be established following reinitialization; automatic maneuver can be resumed by appropriate DSKY action.

D.3 CSM Thrust Vector Control (TVC) Autopilot

D.3.1 Summary Description

During powered-flight, the pitch and yaw control of the spacecraft is achieved through the deflection of a gimballed engine of the CSM Service Propulsion System. Attitude control about the roll axis is provided by the Reaction Control System jets. The computation of gimbal-servo commands in response to computed errors between commanded and measured attitudes is the function of the Thrust Vector Control Digital Autopilot, implemented in the Apollo Guidance Computer onboard the CM.

D.3.1.1 TVC Pitch and Yaw Control

The following is a summary outline of TVC pitch and yaw control: (Figure D. 3-1 is a functional block diagram of the TVC pitch and yaw control.)

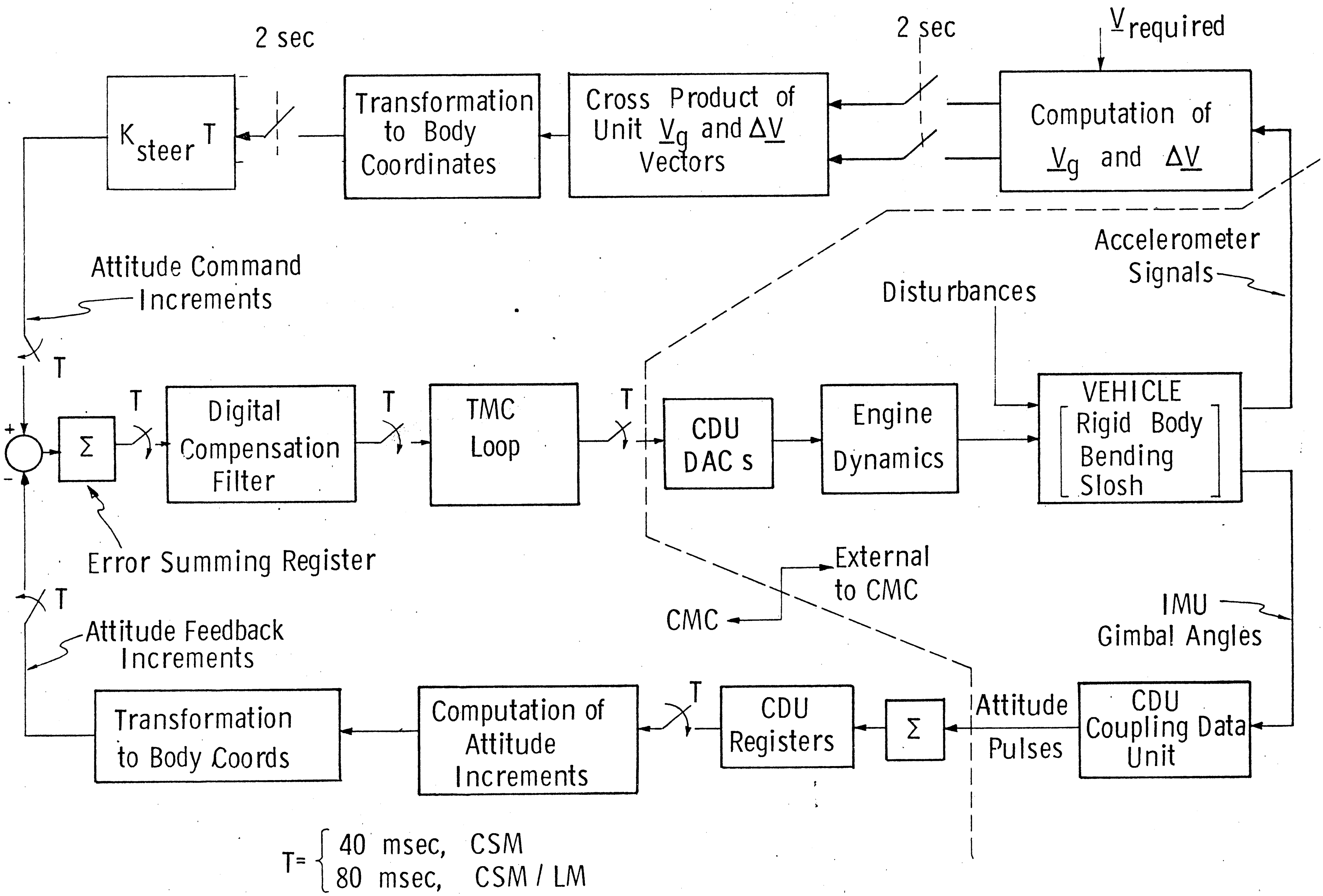


Figure D. 3-1 Functional Diagram of Thrust Vector Control System (Pitch or Yaw)

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

A CDU measures and digitizes the gimbal angles of the IMU and transmits this information to the AGC, where they are stored in a CDU register. The CDU register is sampled regularly by the DAP program which back-differences the CDU angles to obtain the incremental changes over each sampling interval. The CDU increments over each sampling interval are then transformed into body coordinates and subtracted from the commanded increments generated by the steering program. The resultant differences are summed and represent attitude errors, expressed in body coordinates. The respective attitude errors are fed to the pitch and yaw compensation filters, whose outputs, together with estimated trim signals, are the commands to the engine-gimbal servos in pitch and yaw. The compensated signals control rapid transients and determine system bandwidth and the stability of the vehicle slosh and bending.

The CSM/LM DAP has two compensation filters. First is a high-bandwidth filter which stabilizes transients at engine ignition and slosh for the nominal bending effects and propellant loading expected during a lunar mission; this high bandwidth is achieved, however, at a very slight expense to the slosh-stability margin. Second is a low-bandwidth filter which provides poorer transient response, but stabilizes slosh for all propellant loadings. The autopilot begins a TVC burn in the high-bandwidth mode, and remains in that mode unless the astronaut initiates a switchover to the low-bandwidth mode. To retain the maximum advantage of the high-bandwidth filter, this switchover is performed <u>only</u> in the highly improbable case when an observed slosh instability leads to excessive engine oscillations. In the undocked CSM, where bending and slosh are less problematic, the DAP can utilize an even higher-bandwidth filter with considerable success.

As shown in Fig. D.3-2, the command signal to each engine-gimbal servo is comprised of the compensation-filter output and a bias (or trim) from a Thrust Misalignment Correction (TMC) loop. This loop trims the compensated command such that a zero output from the compensation filter will still cause the thrust vector to pass exactly through the center of gravity (cg), when there is no cg movement and no motion of the thrust vector relative to the commanded angle.

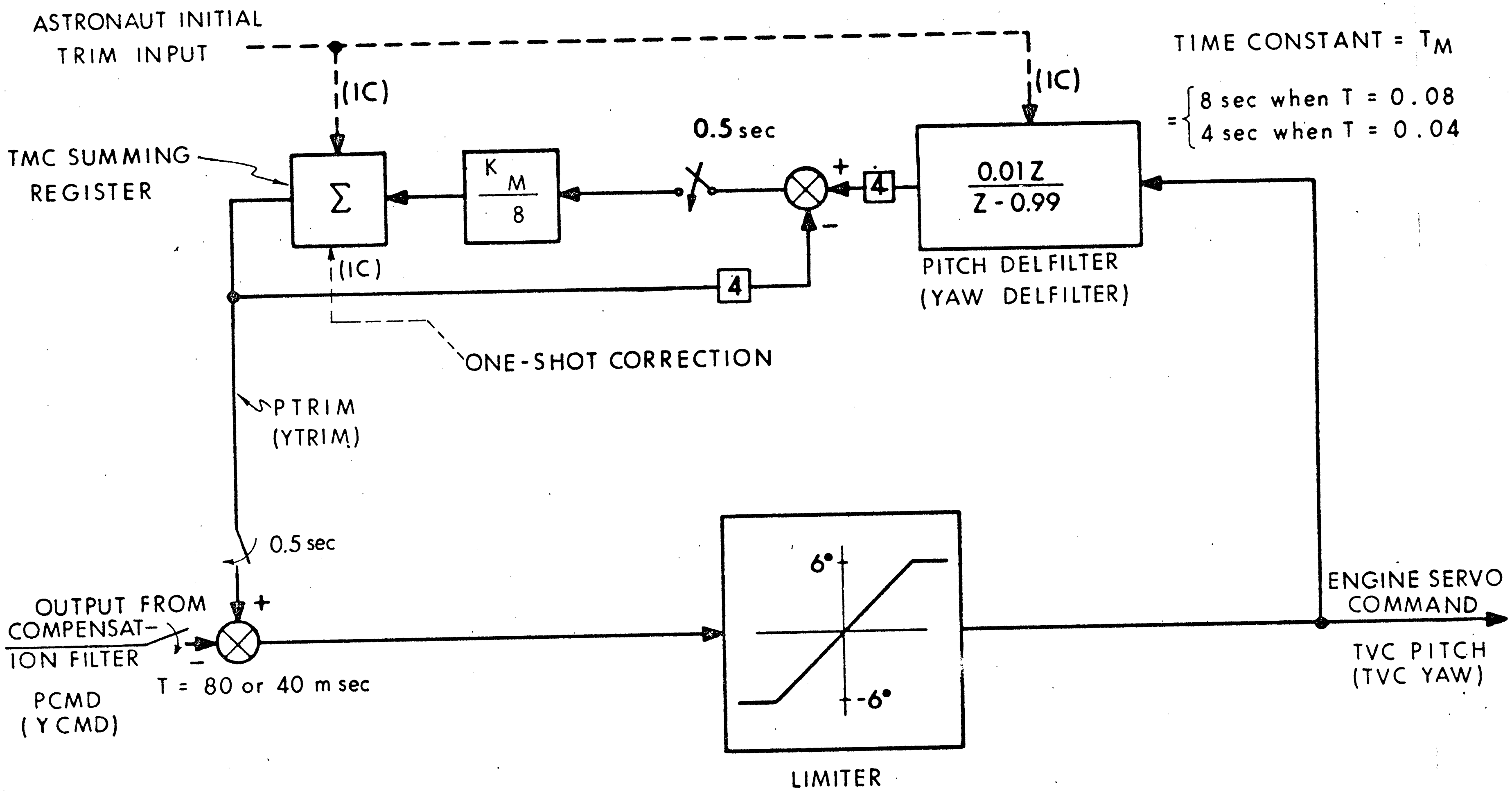


Figure D. 3-2 Functional Diagram of Thrust Misalignment Correction (TMC) Loop

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 slowly integrated to correct for thrust-to-cg misalignment. This 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.

The TMC summing register and DELFILTER are initialized at the beginning of a TVC burn by the astronaut. Later the summing register is reinitialized to implement a so-called one-shot correction.

In the case of the CSM DAP, the TMC summing register is not incremented until a one-shot correction is made 3 sec after engine ignition; this one-shot correction adds to the contents of the register twice the change which has occurred in the DELFILTER output (the factor of two being required to compensate for the transient lag of DELFILTER). Following this correction, the TMC summing register is incremented every 0.5 sec, as shown in Fig. D.3-2.

The CSM/LM DAP increments the TMC summing register from the beginning of the burn. However, the summing register is reinitialized in the event of a low-bandwidth switchover to the current value of DELFILTER (based on the assumption that the switchover occurs beyond the initial transient of DELFILTER). At switchover, the low-bandwidth filter is zeroed, so that the entire burden of supplying the servo command is shifted to the TMC summing register. Following switchover, the TMC loop continues to operate with the same gain, sampling frequencies and DELFILTER time constant.

The operation of the TVC DAP cannot be considered separately from that of the steering loop which interacts with the autopilot. The steering-loop operations are shown in Fig. D.3-1 for the case of External ΔV guidance (see Section C.1.2.5). This form of guidance—which has been used for the SPS burns in all the lunar missions to date—is based on achieving a commanded velocity change, ΔV_c, which is specified prior to each burn. This desired velocity change is inserted into the steering program as an initial value of the computed velocity-to-be-gained vector,

V_G, which is computed in inertially-fixed basic reference coordinates. The V_G vector is reduced during the course of the burn by subtracting the accelerometer-measured increments, ΔV. The ΔV increments are computed at 2-sec intervals by summing the accelerometer pulses accumulated over these intervals, and by transforming the resulting vector components from IMU coordinates to basic reference coordinates. The purpose of the steering loop is to align the vehicle thrust vector with the current velocity-to-be-gained vector. This is achieved by commanding a vehicle-turning rate which is proportional to the normalized cross-product of these two vectors. This cross product is computed in basic reference coordinates and transformed into body coordinates every 2 sec. (Section C.1.2.1 discusses cross-product steering for powered-flight guidance.)

The pitch- and yaw-axis components of the vector cross-product are multiplied by a proportionality constant, K_{steer}, to obtain the commanded rates about these axes—and are multiplied by the autopilot sampling frequency, T, to obtain attitude-command increments which are supplied to the autopilot every T seconds. (K_{steer} has three different values, employed respectively with the CSM DAP and the two modes of the CSM/LM DAP.)

D.3.1.2 TVC Roll Control

The TVC Roll DAP is designed to provide attitude and rate control about the roll axis using the RCS jets. Its function is strictly attitude hold. The orientation of the CSM about the roll axis is held within a specified deadband throughout the burn. The outer-gimbal angle of the IMU, which is parallel to the vehicle roll axis, is read and processed to yield approximate roll-attitude and roll-rate measurements. A switching logic in the phase plane is then used to generate commands to the RCS jets. (The Roll DAP will not be described further in this report so that adequate space can be given to the pitch and yaw autopilots, which have the major role in thrust vector and velocity control.)

D.3.2 Design Requirements of the TVC DAP

D.3.2.1 General Design Considerations

The primary requirement which the TVC pitch and yaw DAP programs must fulfill is to provide, in conjunction with the external-guidance loop, satisfactorily small velocity-pointing errors at thrust cutoff. The DAPs must also limit excursions in vehicle attitude and in thrust-vector orientation to minimize propellant usage and gimbal-servo clutch wear and to facilitate pilot monitoring. The DAP programs must function satisfactorily with uncertain initial conditions and vehicle characteristics that vary during the burn. These factors are discussed below:

D.3.2.2 Initial Conditions and Time-Varying Thrust Misalignment

The TVC DAPs will experience several initial perturbations at SPS engine-ignition time:

a. Non-zero initial attitude errors in pitch and yaw—To avoid bending excitation in the CSM/LM configuration, the TVC DAP neglects initial attitude errors which could result from an ullage maneuver. In the undocked CSM, bending is less problematic and these errors are correctly initialized.

b. Non-zero initial attitude rates in pitch and yaw—Off-nominal RCS-jet performance during ullage may lead to attitude rates at SPS ignition time of up to 1 deg/sec.

c. Initial lateral slosh-mass displacement—Flight results have shown initial slosh-produced oscillations of up to 0.1 deg in the vehicle attitude.

d. Initial longitudinal propellant displacements (in the event of a no-ullage ignition.)

e. Thrust-vector misalignment—Before thrust initiation, the AGC supplies trim signals to the engine-gimbal servos to orient the anticipated thrust vector through the estimated cg position. Very likely, however, some error will occur in the alignment. There are two sources of thrust-vector misalignment—uncertainties in the thrust-vector orientation and uncertainties in the estimation of the cg position. These sources can yield a 3σ misalignment angle ranging from 1.4 deg (full) to 0.98 deg (empty) for the CSM, and 1.25 deg (full) to 0.71 deg (empty) for the CSM/LM.

In addition, due to propellant consumption, the cg position will vary with time. There is also a possibility of angular deflections of the thrust vector within the nozzle, as a result of uneven erosion.

The maximum predicted rates of change in the thrust-to-cg angle for the CSM/LM are about 0.003 deg/sec, in the pitch and yaw planes. For the CSM, the figures are 0.0083 deg/sec in the pitch plane, and 0.014 deg/sec in the yaw plane. Thrust-vector deflection due to nozzle erosion is estimated to be less than ±0.2 deg in any 20-sec interval. The total erosion deflection over a long burn will be within ±0.3 deg.

D.3.2.3 Vehicle Characteristics

The dynamic characteristics of the CSM/LM are sufficiently different from those of the CSM-alone to require separate autopilot programs tailored to the characteristics of each configuration. The three principal differences resulting from the two spacecraft configurations are as follows:

a. Bending-mode frequencies of the CSM/LM are as low as approximately 2 Hz; bending-mode frequencies of the CSM-alone begin at approximately 5 Hz.

b. Both in the moment arm from the gimballed engine to the center of gravity and in the vehicle moment of inertia, the two configurations differ substantially. Thus, a given deflection of the gimballed engine of the CSM-alone can produce as much as four times the angular acceleration as the same deflection in the CSM/LM vehicle.

c. The fuel and oxidizer slosh behavior in the CSM/LM vehicle differs from that in the CSM-alone because of the additional slosh masses in the LM tanks, the effects of the increased mass and moment of inertia of the overall vehicle, and the differences in cg location.

The stabilization of slosh and the avoidance of its excitation are of equal importance for both vehicles. The stabilization of the bending modes and the avoidance of their excitation is primarily a problem in the docked CSM/LM configuration.

D.3.2.4 Design Approach

The discrete, quantized nature of the digital computations have not presented any major impediments to the design of the TVC autopilots. The effects of quantization at the D/A and A/D interfaces have been found to be negligible in the cases of the CSM and CSM/LM TVC DAPs. The effects of finite word length and fixed decimal-point arithmetic in the Command Module AGC have been reduced to negligible levels in these DAPs by (1) careful attention to the manner in which the control equations are solved; (2) proper scaling of the computer variables; (3) use of double-precision variables where required; and (4) proper attention to the manner in which (1), (2), and (3) are combined with the selection of the sampling rates to maximize both the linear operating range and the precision of each digital-filtering operation. Conventional design techniques employing Z and W transforms, root loci and frequency response characteristics were found to be adequate for the design of the TVC autopilots.

D.3.3 TVC DAP Implementation

D.3.3.1 Compensation Filters

A basic TVC autopilot program provides a generalized sixth-order filter which consists of three cascaded second-order sections. The CSM/LM configuration uses all three sections; the CSM only two. The second-order factors for the cascade sections have been selected in such a way as to minimize the transient excursions of the signals between sections, thus allowing these signals to be scaled to take advantage of the available digital word lengths—and thereby minimizing the effects of round-off errors. Generally, it has been found best to group zeros and poles having similar frequencies together in the same cascade section. Where this has not been possible (as in the CSM/LM low-bandwidth third cascade), an attempt has been made to keep the steady-state gains of the numerator and denominator from becoming too dissimilar. (The steady-state gain of either the numerator or the denominator is found simply by setting $z = 1$ and adding the coefficient values.)

D.3.3.1.1 Switchover from High Bandwidth to Low Bandwidth

In the CSM/LM mode, provision is made for manual switchover from the normal high-bandwidth filter to a slower low-bandwidth filter. Once this switchover is commanded, the computer calls a section of coding which is designed to (1) zero the filter storage locations; (2) update the TMC summing registers with new values of PTRIM and YTRIM based on the DELFILTER outputs; (3) load the low-bandwidth coefficients from fixed memory into erasable memory; and (4) load new values for the DAP gain, the TMC loop gain, and the steering gain.

No provision is made during a burn for returning to the high-bandwidth mode once the switchover has taken place. On the next burn, however, the TVC initialization logic reloads the high-bandwidth coefficients from erasable memory.

D.3.3.2 TVC DAP Variable Gains

The DAP gains are established initially and updated periodically using a small routine which is called every 10 sec and computes a piecewise-linear approximation to the curve for $I_{AVG}/T\ell_x$ versus SPS propellant weight. This value is then multiplied by the gain constant KTLX/I to obtain the TVC DAP gain, K_z. Consequently, the gain relationship for the TVC DAP is given by

$$K_z = (KTLX/I)\ (I_{AVG}/T\ell_x)$$

D.3.3.3 Trim Estimation

Three sources of trim information are provided to the TVC DAPs: (1) the initial values of PTRIM and YTRIM from a DSKY entry prior to the first burn; (2) a single-shot correction shortly after ignition (for CSM only), plus repetitive corrections during the burn (both for CSM and CSM/LM); and (3) an end-of-burn update at the engine-off command. In addition, a one-shot trim update is made in the CSM/LM mode if the high-bandwidth to low-bandwidth switchover is executed.

At the time of the engine-off command, a current update of the trim estimates is made. (This consists of picking off the pitch and yaw DELFILTER values and loading them into the trim registers.) The basis for this final update is that, for CSM/LM burns of less than about 25 sec, DELFILTER tracks the actual engine position faster than the full TMC loop—and therefore provides a better trim estimate for the next burn. Before each burn the astronaut reviews the computer-stored trim values for acceptability. Ordinarily he will not alter these trims unless the vehicle configuration has changed since the last burn.

D.3.3.4 Restart Protection

Much of the computer logic, including that of the TVC DAP, must be protected against restarts. (As discussed in Section 2.1.4.2, restarts are caused by such events as power transients or a parity fail on a memory-read instruction.)

Briefly, restart protection involves (1) storing the results of certain computations in temporary locations; (2) setting a flagword to indicate that the computation is completed; and then (3) performing a copy cycle to copy the computed results from the temporary registers into their normal registers. In this way, restarts occurring during a computation operation cause that operation to be repeated, while restarts occurring during a copy cycle require only that the copy cycle be repeated. In the TVC DAP there are copy cycles for the pitch and yaw channels, as well as for the DAP-related routines.

D.3.3.5 Computer Storage and Time Requirements

The total AGC memory used by the TVC DAP, including the Roll DAP, is about 1500 words. This breaks down to 1320 words of fixed memory, 26 words of nonsharable erasable memory (which must be preserved throughout the mission), and 154 sharable erasable words used for scratch-pad computation and temporary storage during TVC only. The pitch and yaw channels together require only about 500 words of fixed memory; the remaining 820 words are used by the Roll DAP and by DAP-related logic, such as the TVC initialization and monitoring routines, the mass-properties routine, and the restart routine.

The computer time used by the TVC DAPs is as follows: for the CSM, about 7 msec per channel per 40-msec sample, or about 35 percent of the available computer time; and for the CSM/LM, about 8 msec per channel per 80-msec sample, or about 20 percent of the available computer time. In addition to this, the time required for the combined Roll DAP and monitoring operations is about 10 msec every 0.5 sec.

D.3.3.6 Selection of Sampling Frequencies

The sampling frequencies of the TVC DAPs and their associated steering loops were selected as follows:

a. The sampling frequencies employed in the compensation filters and feedback loops of the autopilots were made high enough to ensure that none of the major bending modes of these vehicles would be subject to the so-called "folding effect". The CSM sampling frequency of 25 Hz (T = 40 msec) is roughly five times the minimum bending frequency of that vehicle. The CSM/LM sampling frequency of 12.5 Hz (T = 80 msec) is about five times the minimum bending frequency of the CSM/LM.

b. The TMC loop's low-pass filter, DELFILTER, requires a fairly high sampling frequency to attenuate the high-frequency components of the engine-servo command. It is convenient to operate this filter at the sampling frequencies employed by the autopilot compensation filters. The same DELFILTER coefficients are used for both autopilots. These coefficients produce a 4-sec time constant at the CSM sampling frequency and an 8-sec time constant at the sampling frequency of the CSM/LM. These time constants represent a compromise between the conflicting requirements of (a) attenuation of slosh oscillations, and (b) accurate tracking of slow variations in the servo command.

c. The TMC summing register is incremented at a low sampling frequency of 2 Hz, or approximately 12 rad/sec. This sampling frequency is adequate for the TMC loop, whose active frequency range is below 2 rad/sec.

d. The steering-loop computations generate the attitude rate command ($\dot{\theta}_c$) once every 2 sec—i.e., at a sampling frequency of 0.5 Hz or about 3 rad/sec. This sampling frequency is well beyond the requirements of the CSM and CSM/LM steering loops, whose open-loop crossover frequencies are 0.15 rad/sec and less.

e. The staircase waveform of $\dot{\theta}_c$ is smoothed by the process of generating command increments, $\dot{\theta}_c T$, to be summed at each sampling period, T. This smoothing process has been found essential for preventing an adverse interaction between steering-loop sampling and the slosh-mode oscillations—both of which occur in the same frequency range.

D.3.3.7 Effects of Computational Time Delays

All computational time delays have been neglected in deriving the transfer functions representing the autopilot and steering loops. The effects of these delays are examined below.

In the case of the autopilot loop, there is a computational delay of 3.3 msec between the time the IMU gimbal angles are read and the time the engine commands are released. This delay has a negligible effect on autopilot stability. For example, the maximum frequency at which the delay could have been of any importance is the 7.5 rad/sec maximum lead frequency of the CSM autopilot. At this frequency, the computational lag produces a phase shift of only 1.4 deg.

The TMC loop has a computational delay equal to one autopilot sampling period, T. This delay produces a negligible effect in the low-frequency range where the TMC loop is effective. It can be shown that the effects of this delay are so small as to be imperceptible on the plots of the open-loop characteristics of the autopilots.

A larger computational delay of about 0.4 sec occurs in the steering loop. However, the effects of this delay are negligible in the low-frequency range, where the steering loop interacts with the autopilot. For example, the delay has the largest effect in the case of the CSM/LM high-bandwidth mode, where it results in the open-loop characteristics being altered by less than 0.8 dB and 3 deg at any frequency between 0.1 rad/sec and 0.5 rad/sec.

D.3.4 TVC DAP Operation

D.3.4.1 Pre-burn Initialization

During an SPS burn, the functioning of the TVC DAPs is automatic, but there are several interfaces that must be properly established prior to ignition.

First, a small routine called the DAP Data Load may be called by the astronaut several minutes or more before the burn. This routine displays such information as (1) the masses of the CSM and LM which are used by the AGC to compute the autopilot gains; (2) the current engine trim angles to place the thrust vector through the center of gravity; and (3) two flagwords which tell the computer whether the LM is docked or not, and which of the RCS jets to use for translational and roll thrusting. These displayed quantities may be accepted as is, or changed by keyboard entry if so desired. (For example, if the LM had just undocked from the CSM, the Command Module pilot would call up this routine and key in a new value for the flagword that indicates the docked/undocked configuration.)

The second important pre-burn function is the initialization of the digital-to-analog converters that transmit the pitch and yaw commands to test the SPS engine servos. This initialization procedure involves zeroing the D/A converters and energizing a relay to complete the electrical paths to the engine servos. The test entails commanding a sequence of ± 2 deg deflections, both in the pitch and the yaw gimbal servos, which the astronaut can monitor on the SPS gimbal-angle indicator dial. Upon completion of the test, the trim values are commanded in preparation for ignition. Should the astronaut bypass the test, the trim angles will be commanded directly. Thus, at the end of this gimbal-trim routine, the SPS engine will have been aligned for ignition, and the servos will be energized and ready for the TVC commands during the burn.

A third pre-burn activity is the ullage maneuver in which the astronaut fires the +X translational RCS jets for about 20 sec preceding ignition to settle the liquid propellant in the tanks. (This maneuver is not required if the tanks are nearly full.)

D.3.4.2 Start-up Sequence

Following the pre-burn initialization, the TVC DAPs are started by a call from the thrusting program to the TVC initialization sequence about 0.4 sec after the ignition command. This delay is provided to accommodate for the delay between the ignition command from the AGC and the achievement of full engine thrust.

The DAP initialization provides for all remaining TVC preparation: (1) it zeros the erasables used for storing past values of the filter variables; (2) it loads the DAP coefficients and gain, the TMC loop gain, and the steering gain; (3) it initializes the TMC loop, the attitude-error integrators, and the DAP commands; and (4) it prepares the attitude-error needle display on the FDAI with a special initialization call.

With the initialization completed, the DAPs are ready to operate. The TVC program provides for time-separating the pitch and yaw computations, so that the Pitch DAP calls the Yaw DAP one-half sample period later, and vice-versa. This separation of the two channels ensures that other computer functions (e.g., steering computations and telemetry) can function at their assigned rates throughout the burn.

In conjunction with the regular cyclical operations of the Pitch and Yaw DAPs, the DAP-monitoring routine TVCEXECUTIVE comes up at 0.5-sec intervals to: (1) call the Roll DAP; (2) call the attitude-error needle display and provide it with an update; (3) update the DAP variable gains at 10-sec intervals; and (4) perform the one-shot and repetitive corrections for the TMC loop.

D.3.4.3 Shutdown Sequence

The engine-shutdown sequence originates in the thrusting program after the steering computations predict that the time-to-go to reach the desired cut-off velocity is less than 4 sec. At that time the engine cut-off time is computed, allowing for the expected thrust tail-off characteristics. Following the cut-off command from the AGC, the TVC DAP continues to function for about 2.5 sec, after which the RCS DAP is called in its attitude-hold mode.

D.4 CM Entry DAP

As explained in Section C.4, the aerodynamic-lift capabilities of the Command Module permit a controlled-entry flight to a designated landing point. The CM is a wingless, axially symmetric, reentry vehicle constructed with its center of gravity displaced from the axis of symmetry. When flying hypersonically in the atmosphere, the CM trims with a constant low ratio of lift to drag. The sole means for perturbing the trajectory in a controlled fashion is to roll about the wind axis (velocity vector) with the reaction-control jets, permitting the lift vector to be pointed anywhere in the plane perpendicular to the wind axis. The roll angle defines the orientation of the lift vector relative to the trajectory plane—i.e., the plane containing the wind axis and the position vectors. Down-range control is achieved via the component of lift in the trajectory plane; and cross-range control via the component of lift out of the trajectory plane.

In actual flight, an Entry DAP causes the CM to roll about the wind axis so that the actual lift direction is forced into agreement with the desired lift direction commanded by the entry-guidance equations. This results in achieving the desired in-plane component for down-range control. The rolling maneuver also yields an out-of-plane component of lift used for lateral-range control. When the cross-range error is minimized, the lateral-lift component becomes an unwanted by-product of the steering, and its effect is constrained to an acceptably small value by the guidance, which causes the CM to roll periodically so as to reverse the sign of and null the lateral drift. Since the in-plane component is the fundamental controlled quantity—in that it controls down-range flight—its sign normally remains unchanged during this nulling (lateral switching) process. The restriction on the sign of the in-plane component during lateral switching in effect requires that the CM roll through the smaller of the two possible angles (i.e., through the so-called shortest path) during lateral switching. The Entry DAP normally commands this type of maneuver. (In certain instances, where such a maneuver would cause the spacecraft to fall short of its target, the entry guidance demands a roll through the larger of the two angles, and accordingly informs the roll DAP.)

Entry DAP design is simplified by the fact that, within the atmosphere, the CM is aerodynamically stable. Since stability is no problem, only rate dampers are used in pitch and yaw. Furthermore, aerodynamic forces are utilized to do most of the work during a coordinated roll.

D.4.1 Exoatmospheric and Atmospheric Entry DAPs

The Entry autopilots are designed to perform automatically all maneuvers for all phases of entry flight starting with positioning the CM in the entry attitude prior to entry interface and continuing until drogue-chute deployment. Such capability requires several modes of operation, as illustrated functionally in Fig. D.4-1. The two basic modes are exoatmospheric and atmospheric. The atmosphere is defined to begin when the G forces acting upon the spacecraft exceed 0.05g.

In the exoatmospheric mode, the CM has three-axis attitude control based on the Euler set* R, β, α. The Euler set defines the angular attitude of the CM body axes (identical to standard aircraft axes, from the pilot's viewpoint) with respect

---

* The Euler set is trajectory-related and independent of inertial reference.

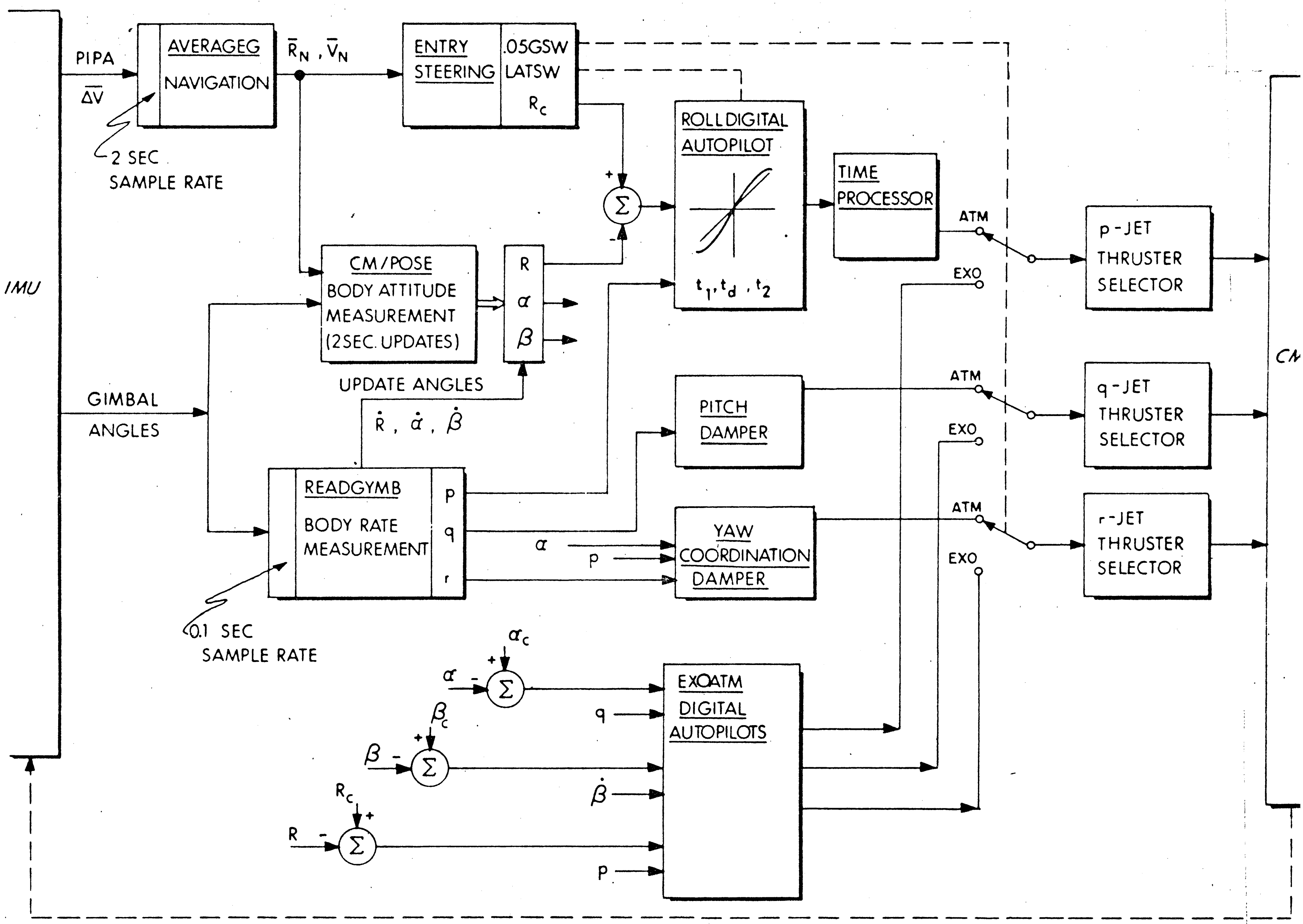


Figure D. 4-1 Functional Diagram of Entry Digital Autopilot to a right-handed vector triad defined as having its X axis along the negative of the computed wind axis, and its Y axis normal to the trajectory plane, i.e., the cross-product of wind axis and position vector. The Euler sequence is roll R about the X axis, yaw β about the Z axis, and pitch α about the Y axis.

The exoatmospheric DAP maneuvers the CM from an arbitrary attitude into a local trim attitude relative to the local wind axis (computed from the current state vector). The maneuver is basically a pitch-over until α is about -20 deg—the hypersonic trim angle of attack.

At the same time, the CM yaws until β goes to 0 deg. The roll angle is held constant until $|\alpha|$ becomes less than 45 deg, at which time the CM does a coordinated roll maneuver about the wind axis until R goes to either 0 deg or 180 deg, as specified by the crew.

A policy based on the Euler attitude-rate equations is used to drive the Euler errors to zero. In the pitch DAP, $\dot{\alpha}$ is considered to be equal to pitch rate, q; $\alpha_c - \alpha$ and $\dot{\alpha}$ are used to command the pitch (q-axis) jets. In the roll and yaw DAPs, the attitude rates, $\dot{R}$ and $\dot{\beta}$, are considered to be orthogonal axes rotated through the angle α with respect to the orthogonal axes, p and r. To decouple the roll and yaw axes, the attitude rates, $\dot{R}$ and $\dot{\beta}$, are assigned to the nearest jet axis, p or r. Thus, for $|\alpha| > 45$ deg, the roll DAP, using $R_c - R$ and $\dot{R}$, fires its yaw (r-axis) jets, and the yaw DAP, using $\beta_c - \beta$ and $\dot{\beta}$, fires the roll (p-axis) jets. When $|\alpha| < 45$ deg, the roll DAP, using $R_c - R$ and $\dot{R}$, fires the roll (p-axis) jets, and the yaw DAP, using $\beta_c - \beta$ and $\dot{\beta}$, fires the yaw (r-axis) jets.

Each navigation cycle, the Euler angles are computed from state-vector data, and are compensated for computation time delay. Between the navigation cycles, the angles are updated at each 0.1-sec DAP cycle by integrating the Euler attitude rates obtained by resolution from the CM body rates. The atmospheric mode is selected whenever the atmospheric drag exceeds 0.05g. The CM measurement continues as before.

The exoatmospheric DAP drives α and β to the commanded hypersonic-trim values. In the atmosphere, these angles are essentially angle of attack and angle of side slip, and the CM is controlled by orienting the lift vector. The amount of lift available for control purposes (i.e., the ratio of lift to drag), is determined by the cg location. Preflight ballast procedures ensure that the Z_{cg} offset will give the desired lift-to-drag ratio, L/D.

Within the atmosphere, aerodynamic forces tend to keep β essentially zero, and α essentially at trim α_t, so that attitude control is required only in roll. As the CM rolls, its X axis is constrained to roll about the wind axis at the angle α_t. This coordinated maneuver requires that yaw rate r be equal to p tan α_t. For both the pitch and yaw axes, rate damping maintains pitch and yaw angular rates within prescribed limits.

D.4.2 Phase-Plane Logic

The Entry DAP is, in reality, six separate digital attitude controllers—three for the exoatmospheric mode and three for the atmospheric mode. There is some consolidation, in that several controllers use common phase-plane logic.

Three axes of the exoatmospheric DAP operate each 0.1 sec and use the phase-plane logic of Fig. D.4-2. The X axis is attitude error, such as β_c-β; the Y axis is attitude rate, such as $\dot{\beta}$. The biased deadzone is utilized to obtain a minimum limit-cycle frequency; however, since the logic is used on a sampled basis, the deadzone is effectively enlarged by one sample time and proportionally as shown in the lower portion of Fig. D.4-2. The logic is constructed so that errors are reduced at a rate between 2 deg/sec and 4 deg/sec.

The behavior of the logic is illustrated by the upper portion of Fig. D.4-2. If, at the DAP sample time, the error and error rate correspond to a point in the shaded area, the indicated jet is turned on; should the point lie in the clear area, the jet is turned off. A typical trajectory is illustrated, and the dots represent updates. Between DAP updates, the jets remain on, if already on, and off, if already off.

The foregoing DAP logic is valid for both single-ring* and dual-ring thruster operation. However, dual-ring operation uses about twice the propellant as single-ring.

---

* The CM has two independent sets (rings) of thrusters to provide the redundancy necessary to assure safe operation of the jets.

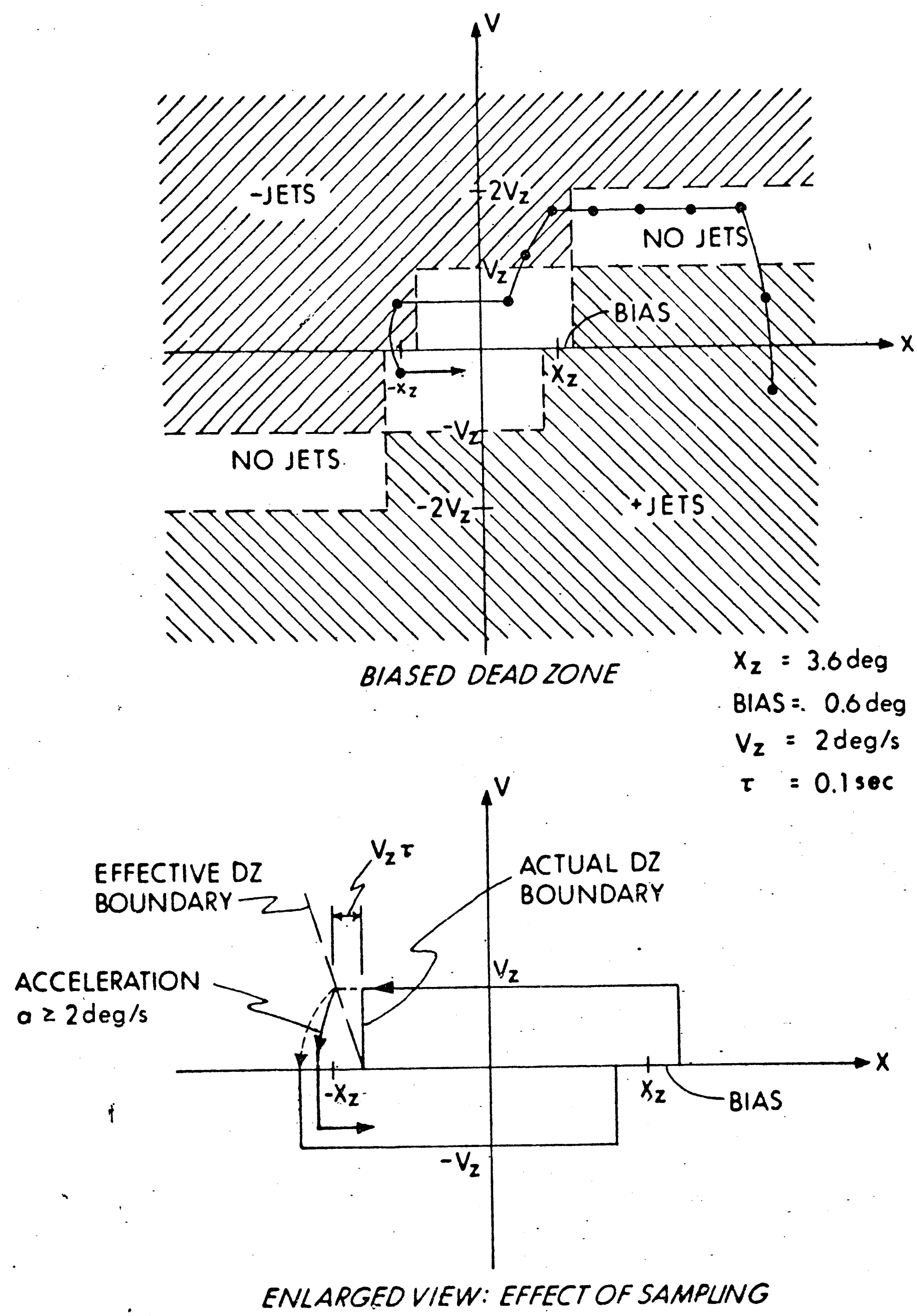


Figure D. 4-2 Exoatmospheric Phase-Plane Logic

When $|\alpha|$ becomes less than 45 deg, the roll-axis logic changes over to the 2-sec predictive phase-plane logic described below for the atmospheric DAP.

The atmospheric logic is such that when the pitch rate, q, exceeds 2 deg/sec at a DAP update, the proper q jet is fired; for yaw coordination and damping, when the combination $r - p \tan \alpha_t$ exceeds 2 deg/sec, the proper yaw (r-axis) jet is fired. For both pitch and yaw, if the rate is less than 2 deg/sec, the jet is turned off. As with the exoatmospheric DAP, the jets are changed only at DAP updates.

Unlike the pitch and yaw axes, the roll axis is controlled by a 2-sec predictive DAP, which becomes active during the exoatmospheric mode, when $|\alpha|$ became less than 45 deg; thus, it is already operative when the atmosphere is encountered. Every 2 sec during the entry phase, entry guidance provides a roll command R_c. To do this, it examines current vehicle position and velocity and also landing-point position, decides on the proper orientation of the lift vector, and generates the commanded roll attitude necessary to achieve that orientation. The roll autopilot uses the command R_c, the present roll attitude R, and roll rate p to generate firing times for the jet thrusters. In general, three time intervals are generated each 2 sec—two are thrust durations, and one is quiescent duration.

The vehicle roll attitude, in response to applied roll RCS torque, is modeled adequately by considering only the torque due to moment of inertia, thus permitting use of a phase plane wherein motion can be described using only straight lines and parabolas. The (X, V) phase plane of Fig. D.4-3 is used where the roll attitude error is $X = R_c - R$, and the attitude rate is $V = \dot{R}$. When the CM is at the commanded attitude, both X and V are zero. Note that roll rate p is related to $\dot{R}$ by $\cos \alpha$ in a coordinated roll. The inclusion of the $\cos \alpha$ term ensures a coordinated roll such that $p \rightarrow 0$ as $X \rightarrow 0$. The X axis is selected for the $\cos \alpha$ term; consequently, $V = p$ and $X = (R_c - R)\cos \alpha$ in the following discussion. This choice, though originally made for simplicity so that only the error X depends on allowing the velocity limits and deadzone limits to be expressed in units of body rate, is also desirable when platform misalignment is considered. In this case, as the state-vector error begins to accumulate, errors are introduced into the indicated body-attitude angles such that the indicated α can go through 90 deg. In this region, the $\cos \alpha$ in the DAP approaches zero and turns off the roll-axis attitude control. Since, in this region, the effect of p on indicated roll attitude R is insignificant, the DAP is not used to

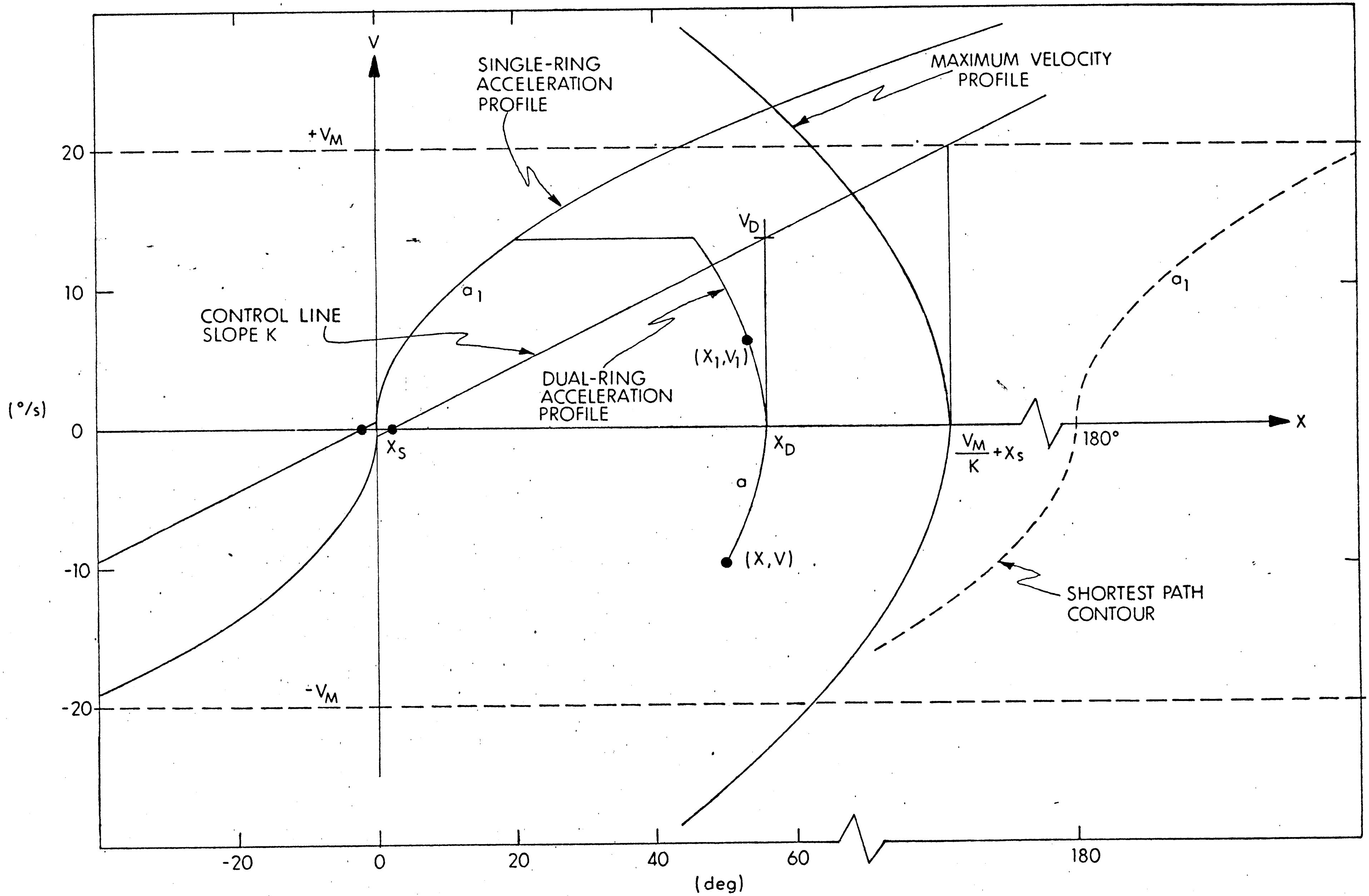


Figure D. 4-3   Hybrid-Gain Roll-Attitude Phase Plane generate roll rates, and the CM will drift at deadzone values of p. When α increases further, $\cos \alpha$ becomes nonzero, and the DAP resumes control. The sign reversal of $\cos \alpha$ maintains stability. When the DAP is used in the exoatmospheric mode, inclusion of α allows a coordinated roll maneuver to be generated, even though α is changing.

The roll phase-plane logic illustrates two important design considerations. The first feature of the control technique is the use of a line of slope K to determine the drift rate at which an error is reduced. A fundamental and motivating advantage of this method is that fuel consumption (ΔV) becomes somewhat proportional to error, in that the CM responds rapidly to large error and more slowly to small error. Slope K is chosen to be as small as is consistent with the rate of response needed by the guidance during entry. Also, the straight line simplifies the equations for the jet-firing times and also allows the DAP to follow ramp inputs efficiently by establishing the necessary rate. The second feature of the control technique is the construction of the predicted trajectory from the DAP update point to the origin, using parabolas of different acceleration, to minimize the sensitivity of the control system to actual jet-thruster acceleration. This latter design consideration ensures not only stable operation, but comparable transient response behavior—even in the presence of a possible error of 100 percent in the control authority. Such a contingency could occur as follows: To allow rapid crew detection of jet failure during entry, only one ring of thrusters is active at a time. However, should circumstances dictate the use of both thruster rings, the DAP is required to perform in a stable and comparable fashion—even though it has no knowledge of whether single or dual rings are operative. This requirement is met by imposing a hybrid phase-plane profile: the first thrusting interval assumes dual-ring acceleration; a nonthrusting drift interval follows; and then a second thrusting interval assumes single-ring acceleration.

Figure D.4-3 is a simplified illustration of the roll logic. The prediction is based on the typical trajectory drawn from the point X, V to the origin. Assuming an initial acceleration a, the X-axis intercept, X_D, is obtained and is projected on to the control line to yield the drift velocity V_D. Using dual-ring acceleration, a, and the velocity difference of V_D and V, the first firing time, t, is computed for the +p-jet thruster. The firing time t_2 for the -p-jet thruster uses acceleration a_1

(single-ring acceleration) and velocity V_D. The drift time t_D is determined from the velocity V_D and the error to be covered at this velocity. The procedure is the same for any point below the control line. For example, the point (X_1, V_1) in the figure yields the same X_D and hence the same V_D as above; however, because the velocity correction needed, $(V_D - V)$, is smaller, the first firing time t_1 is smaller.

If the point (X, V) is such that $(V_D - V)$ is negative, the first burn is omitted. If the point (X, V) lies to the right of the maximum velocity trajectory, V_D is defined to be V_M (20 deg/sec) to provide a roll-rate limit. Once the time intervals t_1, t_2 and t_D are computed, only the first 2 sec of the trajectory are implemented. Each subsequent 2-sec DAP update will compute a new trajectory.

Since the actual acceleration of the CM is always either a or a_1, the roll response differs from the predicted trajectory as follows:

First, consider the dual-ring response to this phase-plane logic illustrated in Fig. D.4-4. The actual initial acceleration equals the assumed. V continues to rise to the V_D computed from the X intercept, X_D. Because of the dual-ring acceleration, both t_1 and t_D were computed correctly from the beginning. However, at the point where the trajectory intersects the single-ring profile, t_2 is computed based upon half the prevailing acceleration. The trajectory goes barreling down, but is stopped—before it goes too far—by a regular 2-sec update. Since the trajectory is again below the control line, a new t_1 is computed, and the trajectory rises to a new V_D determined by the new X intercept; the trajectory drifts toward the ordinate until it again intersects the single-ring profile; and then this time it goes down twice the required distance because of the dual acceleration and the absence of an update. This process is repeated until the trajectory is securely within a deadzone (to be described later).

Second, consider the single-ring response to the phase-plane logic, illustrated in Fig. D.4-5. In this case, which is the nominal one, the actual acceleration is half the assumed. Each vertical rise is not only half what is predicted, but the drift period is entered at a lower velocity. By the third update, the trajectory is above the control line, and the trajectory drifts toward the single-ring acceleration profile. At intersection, the trajectory follows this profile directly to the origin, without overshoot.

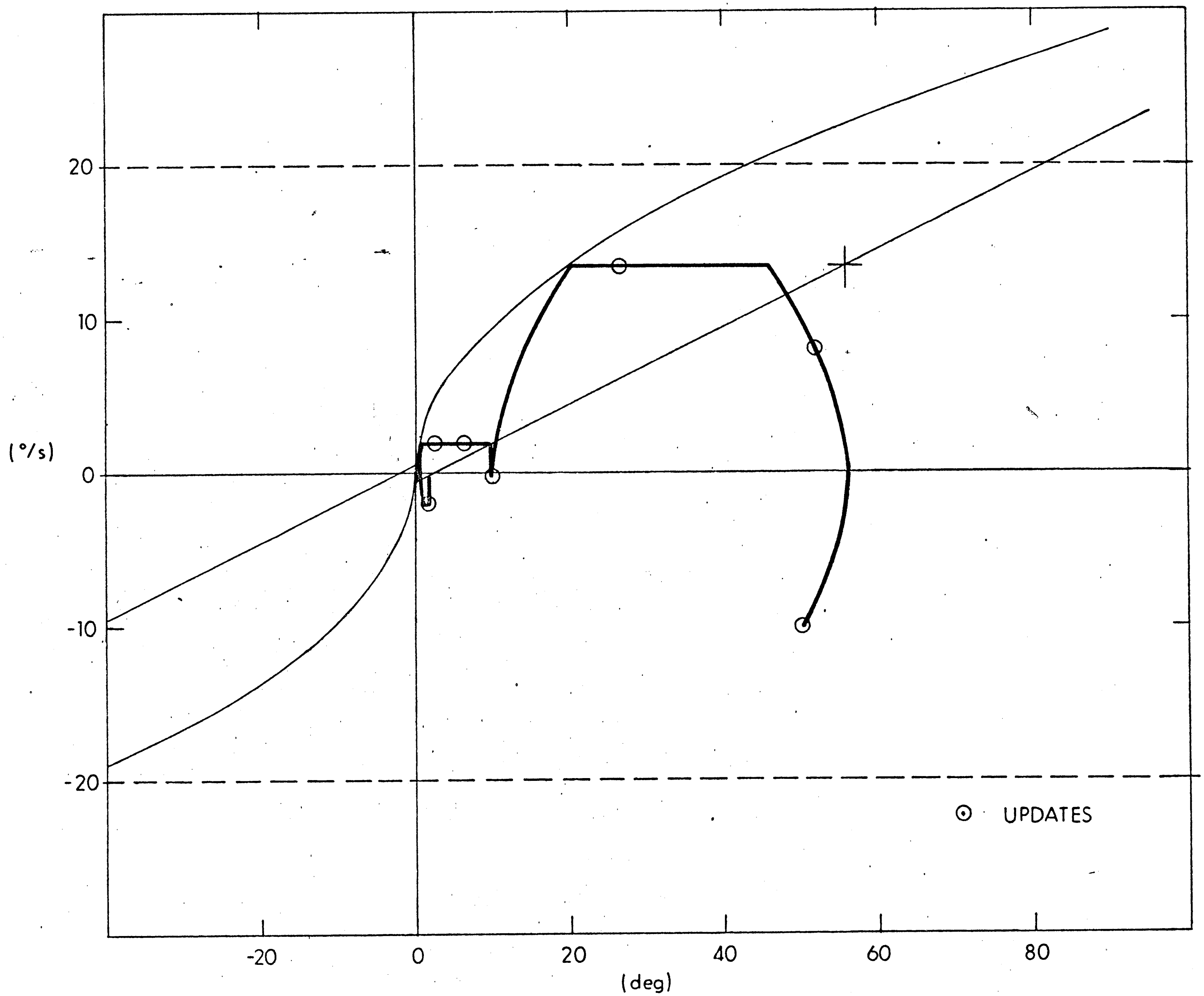


Figure D. 4-4 Dual-Ring Response to Roll-Attitude Phase Plane

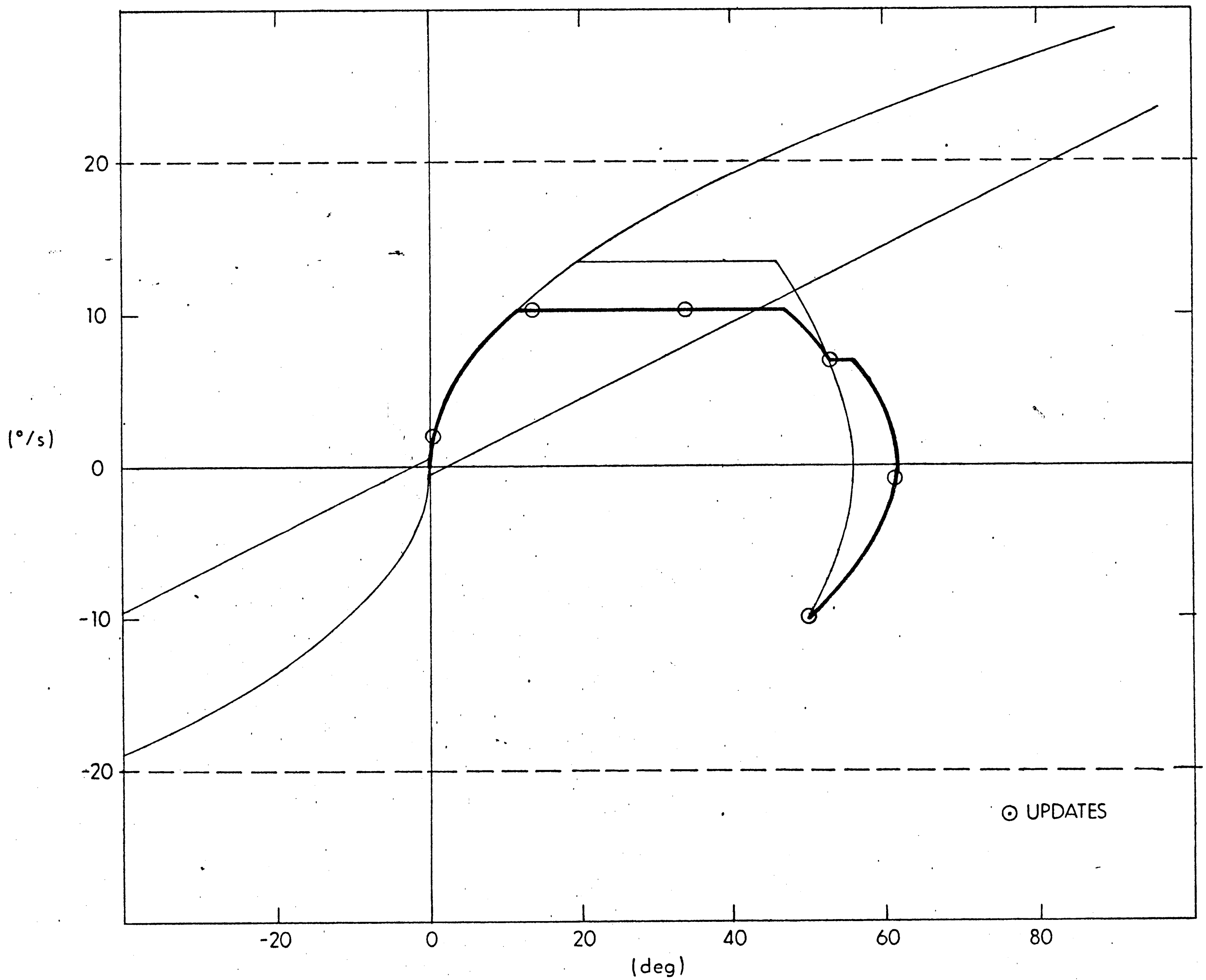


Figure D. 4-5 Single-Ring Response to Roll-Attitude Phase Plane

Figure D.4-6 illustrates the transient response of the DAP with predicted, dual-ring and single-ring conditions. Despite the wide variance in conditions, the DAP produces comparable solution times. Furthermore, simulations and actual flight* experience show that the fuel penalty for using dual rings is only about 20 percent—rather than the 100 percent one might anticipate. Figure D.4-6 also clearly illustrates that both dual-ring and single-ring response are slower than the predicted, but this is the price paid for indifference to actual acceleration. However, what is bought is stable operation over a two-to-one acceleration variation—a property that does not exist if both predicted thrust intervals are based on the same acceleration.

D.4.2.1 Shortest-Path Logic

As mentioned earlier, the roll DAP normally chooses the smaller of the two possible angles through which to roll to null the roll error. This is performed by using the shortest-path contour shown in Fig. D.4-3 at 180 deg and the corresponding contour at -180 deg.

The shortest angular path test consists of determining whether the point (X, V) lies within or without the contours at ± 180 deg. Any point (X, V) inside the contours considers the origin as its terminal point. Points outside these contours consider ± 360 deg as their origin. Such points X are shifted by -360 sgn(X) and thereby appear inside the contours as far as the phase-plane logic is concerned. Such a shift is necessary since, physically, -360, 0 and 360 are the same attitude. Furthermore, it is necessary that the contour dividing the regions of stable nodes be dynamic, rather than geometric—otherwise, there exists a region between the dynamic and the geometric contours where points will initially head for the origin as node; but the trajectory will carry across the geometric contour and then head for 360 as the proper node. This results not only in taking the longer path, but also in taking a longer time to do so.

D.4.2.2 Buffer Zone and Deadzone of the Roll-Attitude Phase Plane

As shown in Fig. D.4-7, the DAP has a deadzone at the origin, for the purpose of eliminating a high-frequency limit cycle. Its shape was chosen to provide a

---

*Apollo 7 flew a dual-ring reentry.

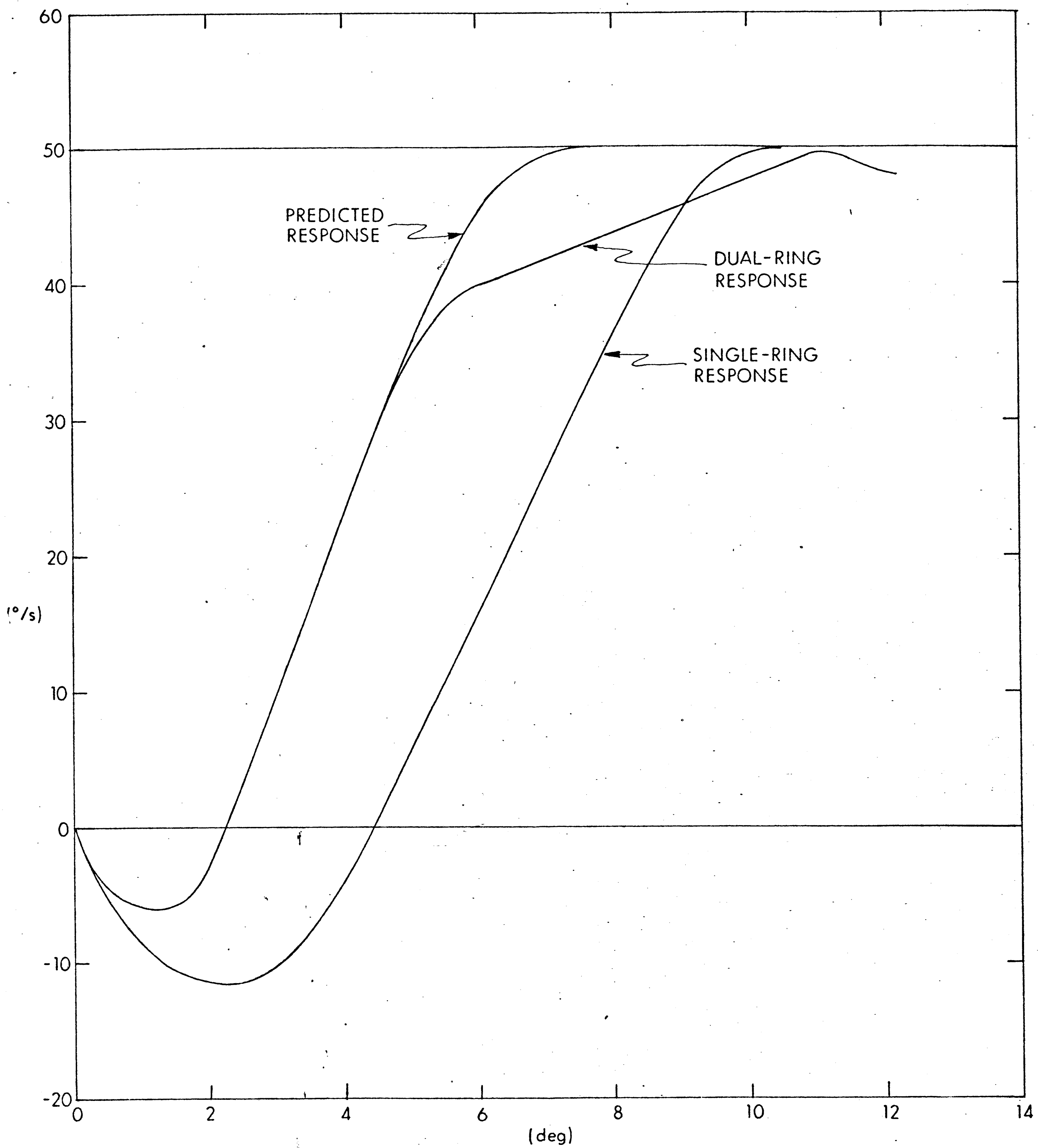


Figure D. 4-6 Transient Response of Roll-Attitude Phase Plane for Dual-Ring and Single-Ring Conditions

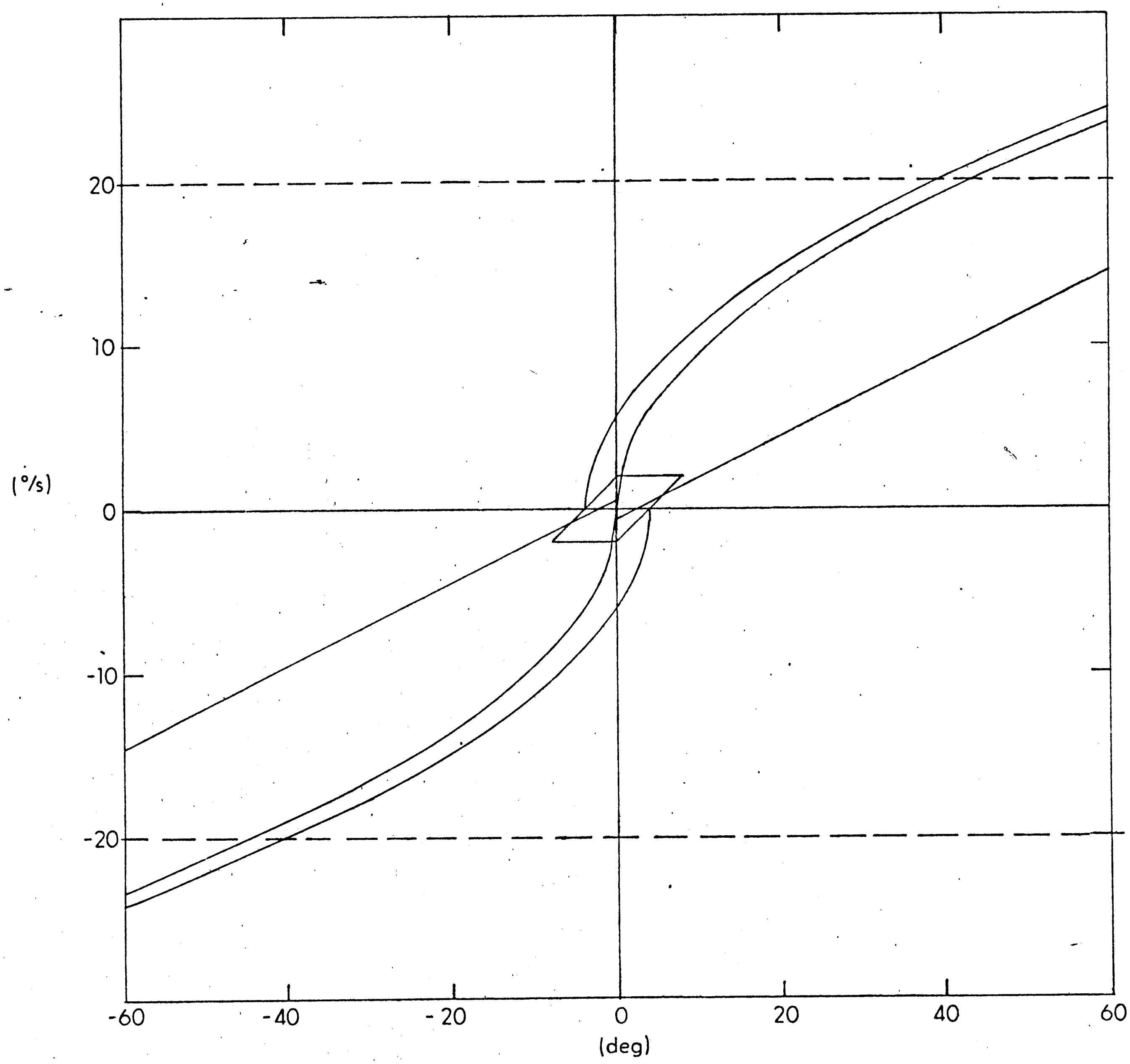


Figure D. 4-7 Buffer Zone and Deadzone of the Roll-Attitude Phase Plane smaller limit-cycle amplitude for a given zone width. Also, the terminal trajectory has a band or buffer zone along which it channels trajectories to the deadzone. The buffer zone overcomes the effect of noise in body-rate measurements and deviations in the actual acceleration from the nominal DAP design value, a_1.

D.4.3 Entry DAP Displays

The DSKY displays associated with the entry mission-control programs were mentioned briefly in Section C.4.2. Additionally, the autopilot provides the following displays to the FDAI attitude-error needles:

Exoatmospheric DAP	The three attitude errors, $R_c - R$, $\beta_c - \beta$, and $\alpha_c - \alpha$, are presented each 0.2 sec. The error used by the autopilot to fire the roll jets is displayed on the roll-error needle, the error used to fire the pitch jets on the pitch-error needle, and the error used for the yaw jets on the yaw-error needle. If $\lvert\alpha\rvert > 135$ deg, the roll DAP does rate damping only, and the roll-error needle is zeroed every 2 sec.
Atmospheric DAP	The only presentation is roll error, corrected for shortest angular path, on a 0.2-sec basis. The pitch- and yaw-error needles are not driven and are at null.

To avoid hitting the needle limits, the maximum deflection allowed is 67.5 deg for roll and 16.875 deg for both pitch and yaw.

D.4.4 Manual Override

No provision exists for the use of manual controls, i.e., hand controllers, in the primary GN&C system during entry; consequently, if the astronaut chooses to perform a manual maneuver, e.g., to avoid gimbal lock, he must switch to the backup control system to override the primary system. In this event, the Entry DAP merely ignores the override and continues to provide FDAI attitude-error displays and jet commands based upon prevailing CM attitude and rates. Hardware in the backup system prevents the GN&CS jet commands from reaching the solenoid drivers until the override is terminated. Consequently, during a manual override, the Entry DAP remains continually prepared (whenever GN&CS operation is restored) to resume control with an uninterrupted knowledge of the trajectory and prevailing conditions.

D.5 AGC Takeover of Saturn Steering

Prior to the flight of Apollo 10, MIT was requested by NASA to provide a boost-takeover capability in the event of a Saturn-Launch-Vehicle (LV) stable-platform failure. The task consisted of supplying an attitude reference and certain guidance information to ensure that, for a failure anytime after liftoff, the spacecraft could achieve earth orbit with help both from the Command Module AGC and from the crew.

For a takeover, the Saturn Instrument Unit (IU) must first sense that its platform has failed, and must call for the AGC backup mode by lighting the LV guidance-failure light in the cockpit. Once notified of a failure, the crew switches the LV guidance to GN&CS control.

With the LV guidance failure light illuminated and the LV guidance switch thrown, the computer can send attitude-error signals to the Saturn autopilot. It is significant that with this procedure, takeover cannot be effected unless the IU initiates the action. Simulation testing has shown that this simple backup scheme can place the spacecraft safely into earth orbit with errors in apogee and perigee of no greater than 10 to 15 nmi.

D.5.1 Generation of Guidance Commands

For first-stage flight, the IU commands an open-loop attitude profile that pitches the vehicle about 60 deg from the vertical in less than 3 minutes. To provide backup here, the AGC calculates a sixth-order polynomial fit to the desired pitch profile. The attitude-error signals are then the difference between these pitch commands and the attitude feedback from the IMU. During takeover, the delay between the failure detection and the first AGC command may be several seconds; however, even in the region of maximum dynamic pressure, transients in attitude, angle-of-attack, and engine angle do not produce excessive structural loads. In addition to the pitch maneuver, the IU normally commands a yaw and a roll maneuver. The yaw maneuver provides additional clearance from the launch tower, and the roll maneuver rotates the vehicle from the launch-pad azimuth to the down-range azimuth, an angular change of about 18 deg. In the backup mode the AGC neglects the yaw maneuver, but it does command a constant roll rate to achieve the desired down-range azimuth.

For second- and third-stage flight, the IU provides a closed-loop guidance scheme to achieve the desired orbit parameters. However, computer-storage limitations preclude such a scheme for AGC backup; thus the task of guiding the vehicle into orbit was given to the crew. This requires that the pilot compare DSKY displays of altitude, altitude rate, and velocity against nominal values tabulated on a card. Using this information, as well as the attitude display, the pilot can then use the Rotational Hand Controller in a rate-command mode to fly the desired trajectory. The manual mode is enabled by a keyboard entry.

D.6 LM Autopilot

D.6.1 Integrated Design

The Lunar Module DAP provides attitude control of the LM spacecraft during both coasting and powered flight. The autopilot is designed to control three spacecraft configurations: LM descent, LM ascent, and CSM-docked. The modules comprising these configurations are shown in Fig. D.6-1.

Torques for attitude control may be generated by the Reaction Control System and by the Descent Propulsion System. The LM Reaction Control System employs 16 jets mounted in clusters of four on outriggers equally spaced around the LM ascent stage. Each jet has a thrust of 100 lb. The Descent Propulsion System (DPS) has a single engine throttleable from a maximum thrust of 10,000 lb down to 12 percent of the maximum thrust. This engine is mounted in a gimbal system with actuators; thus the angle of the thrust vector relative to the spacecraft center of mass can be controlled. The actuators can change the engine angle at the constant rate of 0.2 deg/sec. The Ascent Propulsion System has a single 3500-lb engine, mounted rigidly to the ascent stage. Since the thrust vector of this engine cannot be rotated to pass through the spacecraft center of mass, attitude control during powered flight in the ascent configuration must be maintained by use of the RCS jets.

The LM DAP is an integral part of the LM Primary Guidance, Navigation and Control System. Inputs to and outputs from the LM guidance computer which are associated with the control function (the autopilot) are shown in Fig. D.6-2. The autopilot directly commands the firing of each thruster of the RCS, and the

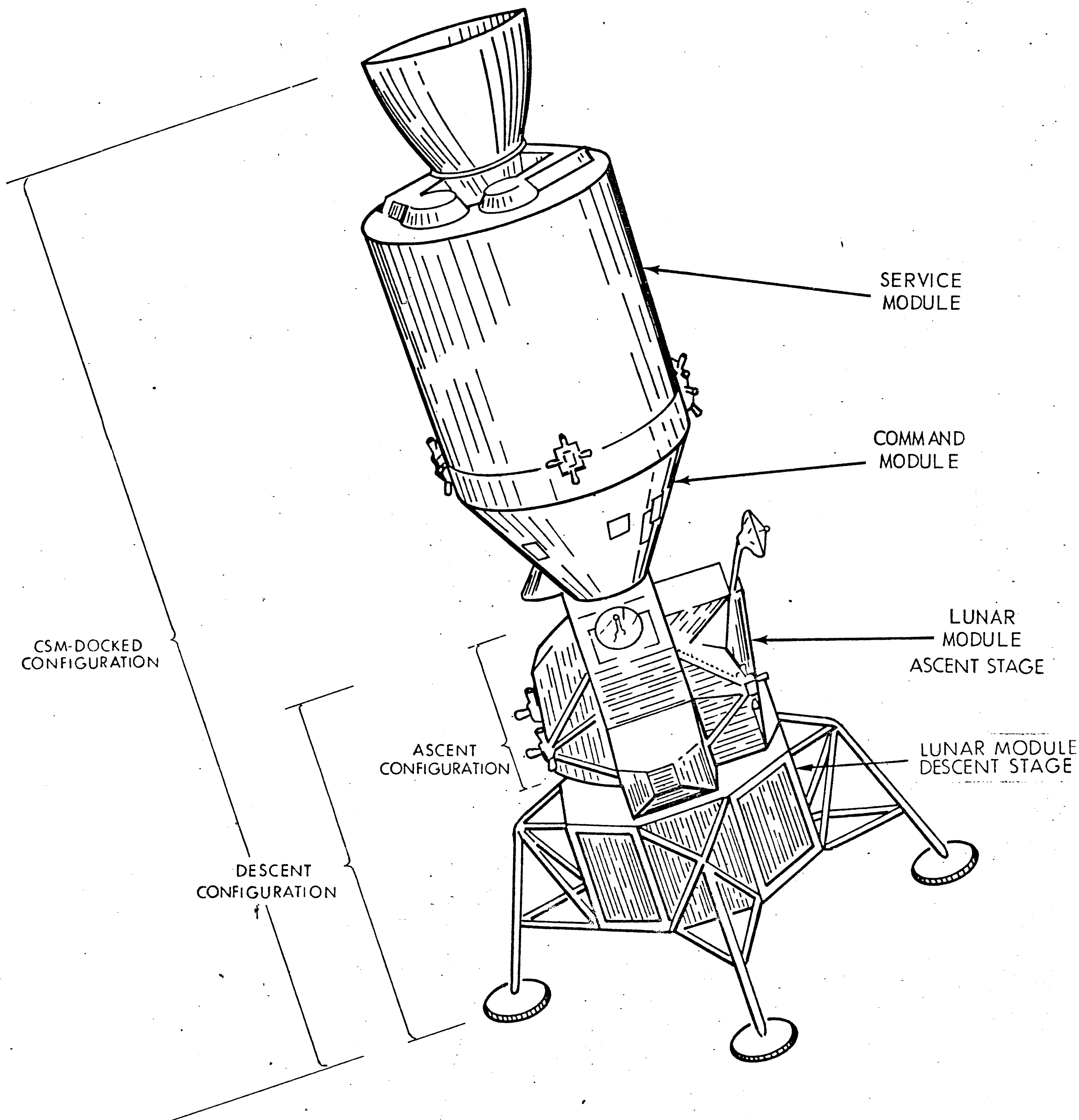


Figure D. 6-1 Spacecraft Configurations Controlled by the Lunar Module Autopilot

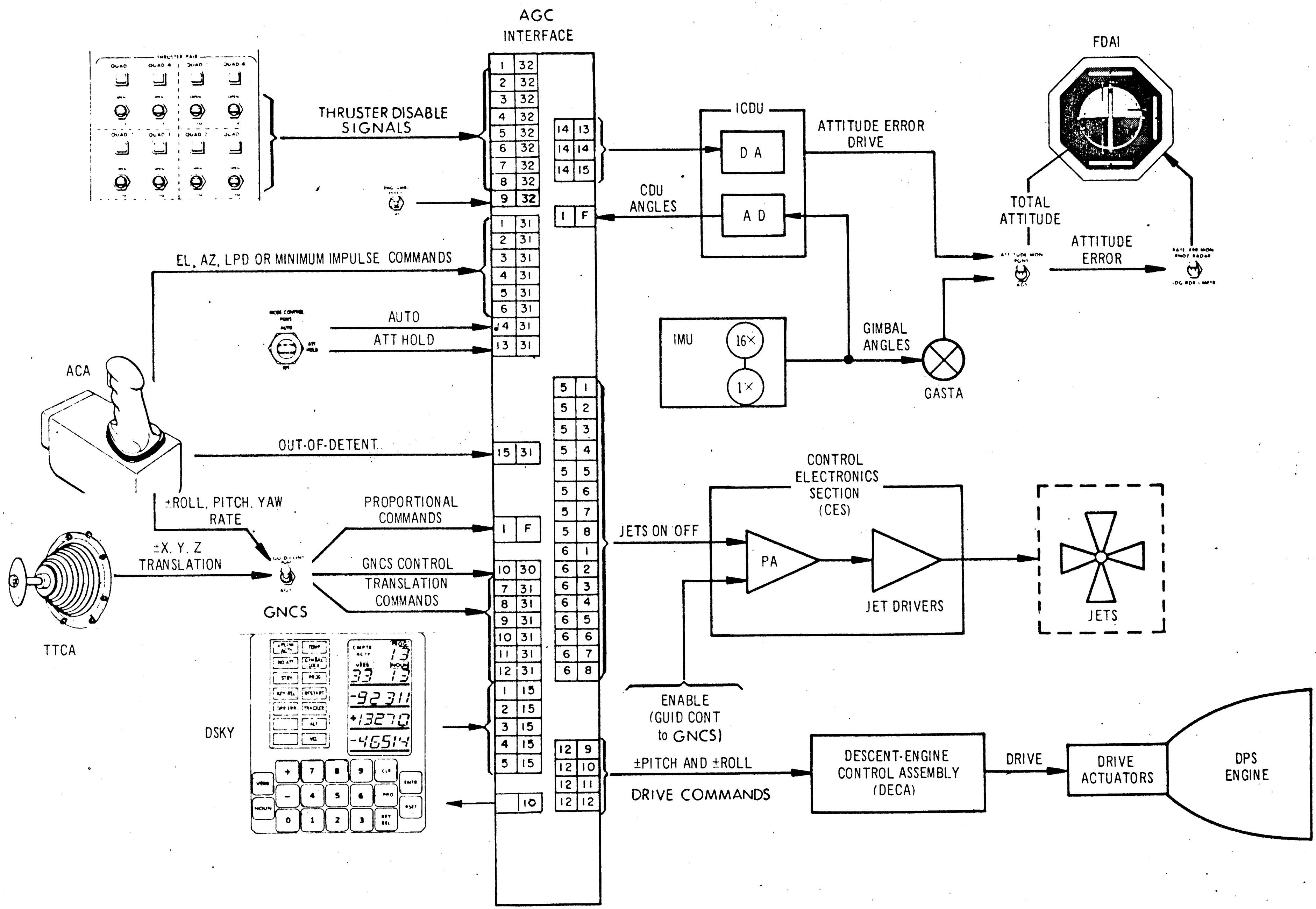


Figure D. 6-2 LM DAP/AGC Interfaces

movement of the descent-engine Gimbal Trim System (GTS). In addition, computed attitude errors are displayed on the FDAI. Among the inputs to the LM DAP are the following:

Measurements of the spacecraft attitude with respect to inertial space, as indicated by the three gimbal angles of the gyro-stabilized platform of the IMU.

Hand-controller signals for providing both manual rotational-control commands and manual translational-control commands.

Mode-switch discretes for selecting the autopilot control modes.

Eight thruster-pair disable-switch discretes for modifying the RCS selection logic and optimizing the autopilot performance in the presence of failed jets.

Keyboard inputs for specifying autopilot control parameters, such as angular deadbands, maneuver rates, hand-controller functions, and spacecraft-mass properties.

Internal steering commands for providing automatic attitude control in both coasting and powered flight.

An internal discrete for providing automatic ullage (+X translation) prior to main-engine ignition.

Internal discretes for switching the autopilot control modes and configurations.

The LM autopilot provides various control modes for coasting and powered flight. For coasting flight, the following modes are available (as discussed in Sections D.6.2 and D.6.3):

Attitude-hold mode, in which the inertial attitude is held constant.

Automatic attitude-maneuver mode, in which the vehicle rotates at a constant angular velocity from some initial attitude to some desired final attitude.

Rate-command/attitude-hold mode, in which the vehicle rotation rate is brought to the desired rotation rate indicated by the Rotational Hand Controller.

Minimum-impulse mode, in which single small firings of the RCS jets are commanded in response to each deflection of the Rotational Hand Controller.

These modes are available during powered flight of the Lunar Module (as discussed in Sections D.6.4, D.6.5 and D.6.6):

> Automatic steering, in which the vehicle follows attitude commands provided by the guidance equations. This is the mode which is used during most of the powered descent to the lunar surface.
>
> Rate command/attitude hold, which has the same characteristics as in coasting flight. This is the mode which is selected by the LM pilot during the final phase of the lunar landing to guide the vehicle manually over what could be a rocky terrain.
>
> X-axis override mode, a combined automatic and manual mode in which the LM pilot controls the vehicle attitude about the X-axis (the thrust direction) by manual rate commands, while the direction of the thrust vector is maintained automatically in response to the guidance commands.

D.6.1.1 Design Approach and Structure of the Autopilot

Certain performance requirements guided the design of the LM autopilot. Attitude control must be maintained with a minimum expenditure of RCS propellant. Concomitantly, the number of RCS firings must be minimized to achieve greater thruster reliability. Attitude control must be maintained even in the presence of a disabled single jet, quad or RCS system (8 jets). The DAP must be stable in the presence of bending modes in the CSM/LM docked configuration, slosh in all configurations, and transients due to ignition, abort stage, transfer from the backup Abort Guidance System and switching of DAP modes. Finally, attitude control must be essentially unaffected by off-nominal vehicle, thruster and sensor characteristics.

In its most general form, the requirements for the LM DAP posed a formidable multi-input, multi-output problem: control logic had to be synthesized to relate the measurement of the three-axis spacecraft attitude with the firing of 16 RCS thrusters and the gimballing of the descent engine about two axes. The design approach employed for the LM autopilot was to separate the total synthesis problem into a set of smaller design problems. Accordingly, four major subsections of the LM DAP were defined, each of which could be approached somewhat independently:

1. Certain DAP parameters are functions of vehicle mass and can be expected to change slowly; others need to be computed infrequently. These

calculations are done in a DAP subsection called 1/ACCS (so named because the reciprocals of accelerations are computed), which is executed every two seconds during powered flight and after major transients (such as LM-configuration changes). Its major outputs are the estimates of RCS jet-control authority (angular-rate change induced by a jet firing), Gimbal Trim System control authority (angular-acceleration changes induced by a trim gimbal drive), and related quantities. GTS control can be executed under direction of 1/ACCS when the control mode requires only infrequent reevaluation of GTS drive status.

2. The <u>state* estimator</u> estimates the spacecraft's angular-velocity and angular-acceleration vectors from sampled measurements of the vehicle attitude. The state estimator was designed with the recursive structure of a Kalman filter. However, nonlinear logic was incorporated to reject small-amplitude high-frequency disturbances from vibration and the CDUs; these are recognized and incorporated into the estimates only when they exceed certain threshold magnitudes. Filter gains were chosen to make the rate estimate respond rapidly to changes, and cause the offset angular-acceleration estimate to respond much more slowly. In this way, rapid maneuvers can be performed accurately, satisfying the astronaut's need for quick, flexible, and reliable response during manual control—particularly for emergency maneuvers. On the other hand, the slow acceleration estimate is largely insensitive to slosh oscillations, responding with a greatly damped oscillation in the presence of slosh. In the case of CSM/LM docked powered flight with the LM active, the rate- and acceleration-estimator filter gains are reduced even further to additionally buffer the estimates from the disturbance induced by bending oscillations. This state estimator derives the angular velocity and angular acceleration of the LM, based only on measurement of spacecraft attitude and assumed control response, thus demonstrating that rate gyros are not required sensors for this autopilot.

3. The <u>RCS control laws</u> fire the RCS jets in response to the vehicle-attitude state, the attitude commands, and the translation commands. The RCS control

*The word "state" used in this section refers to the spacecraft's attitude, attitude rate (and sometimes angular acceleration).

laws in the LM-alone configuration employ parabolic switch curves in their phase-plane logic. The critical parameters in the RCS control laws are adapted in response to the varying moment of inertia of the spacecraft and the bias angular acceleration due to the thrusting main engine. The control-law design permits rapid response to commands with a minimum of jet firings. In coasting flight, steady-state attitude control is maintained with a minimum-impulse limit cycle. In ascent powered flight, a larger-pulse low-frequency limit cycle is employed to hold attitude against the bias angular acceleration. The RCS control laws maintain satisfactory attitude control of even the lightest ascent configuration* even though the control action is reevaluated at most 10 times per sec. Satisfactory control is possible because the RCS control laws compute the exact firing time required to achieve a desired rate change.

4. The <u>trim-gimbal control laws</u> drive the orientation of the descent engine about its two axes in response to the vehicle-attitude state and attitude commands. A third-order minimum-time control law is used to control the vehicle attitude by means of the thrusting descent engine. This permits attitude control often without the assistance of the RCS jets.

Within the RCS control laws and the trim-gimbal control laws, further design simplifications were made by separating these control laws into logically distinct control channels. The choice of these control channels followed from the natural control axes of the LM, shown in Fig. D.6-3.

The descent engine may be gimballed under computer control about the pitch (Q) axis and the roll (R) axis. Therefore, the descent-engine trim-gimbal control laws have been separated into two channels (Q and R). The computation of the proper trim-gimbal drive for each channel is based on independent single-plane control laws.

The RCS jets mounted on the LM ascent stage are skewed 45 deg away from the spacecraft's coordinate frame (in the QR plane) to avoid jet impingement on the pads. This rotational jet-coordinate frame is designated the P, U and V axes. The

---

*In the lightest ascent configuration, single-jet firing for 0.1 sec (one DAP pass) could produce an attitude-rate change of 1.5 deg/sec.

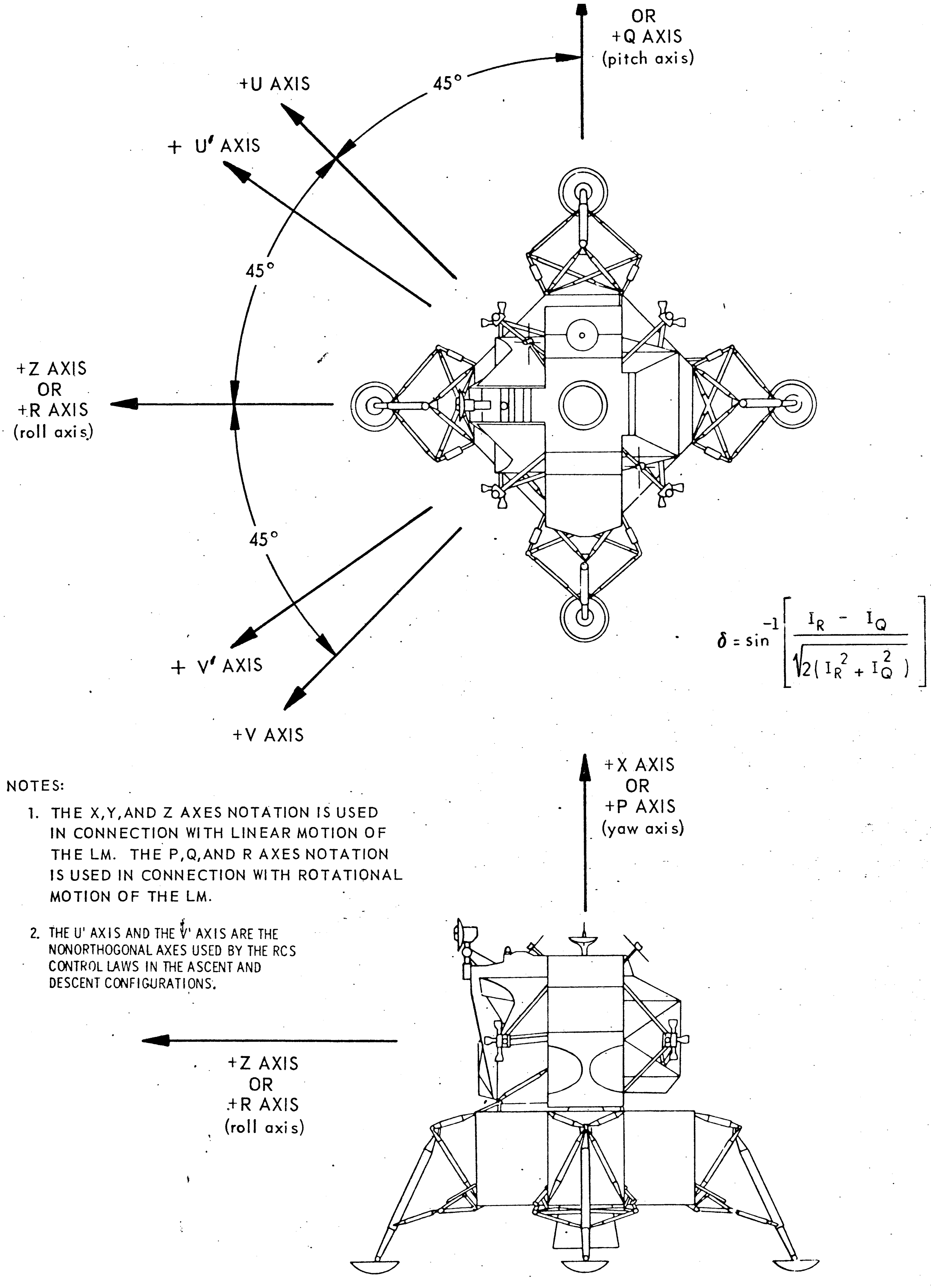


Figure D.6-3 The Control Axes of the LM

locations and orientations of the RCS jets are such that if the spacecraft center of gravity lies near the geometric center of the 16 RCS jets, then:

a. The eight jets that thrust only in the Y or Z directions produce torques about the P axis only. Accordingly, these jets are termed the P jets.

b. Four of the jets that thrust in the ±X direction produce torques about the U axis only. Accordingly, these jets are termed the U jets.

c. The other four jets that thrust in the ±X direction produce torques about the V axis only. Accordingly, these jets are termed the V jets.

Due to the existence of significant cross-inertia between the U and V axes, cross-coupled acceleration between the U and V axes is introduced whenever a U jet or a V jet is fired. To decouple the RCS control channels and thereby reduce the number of RCS firings and hence propellant consumption, a nonorthogonal set of control axes called U' and V' was introduced. The U'- and V'-axis directions are determined as follows: first, cross-coupling between the P axis and any axis in the Q,R plane is assumed negligible and hence is ignored—therefore, the U' and V' axes are constructed to lie in the Q,R plane and correct for cross-coupled acceleration in that plane only; second, acceleration vectors are produced by applying a torque around U and V, as illustrated in Fig. D.6-4; third, perpendiculars are drawn to each of these acceleration vectors to produce the new U' and V' axes. It can be seen that if an RCS torque is applied about U, there will be a component of acceleration along U' and no component of acceleration along V'. Similarly, an RCS torque applied about V will produce no component of acceleration along U'. Therefore, if a U-axis torque is commanded to achieve U'-axis control and a V-axis torque is commanded to achieve V'-axis control, no cross-coupled acceleration will result. The angle δ by which the U' and V' axes are skewed away from the U and V axes is computed from the Q, R moments of inertia.

D.6.2 Manual Modes of the LM DAP

During certain critical phases of the Apollo mission, LM attitude is manually controlled by the LM crew. To achieve a precisely defined attitude, such as that required for initial thrust-vector positioning, an automatic maneuver is usually more

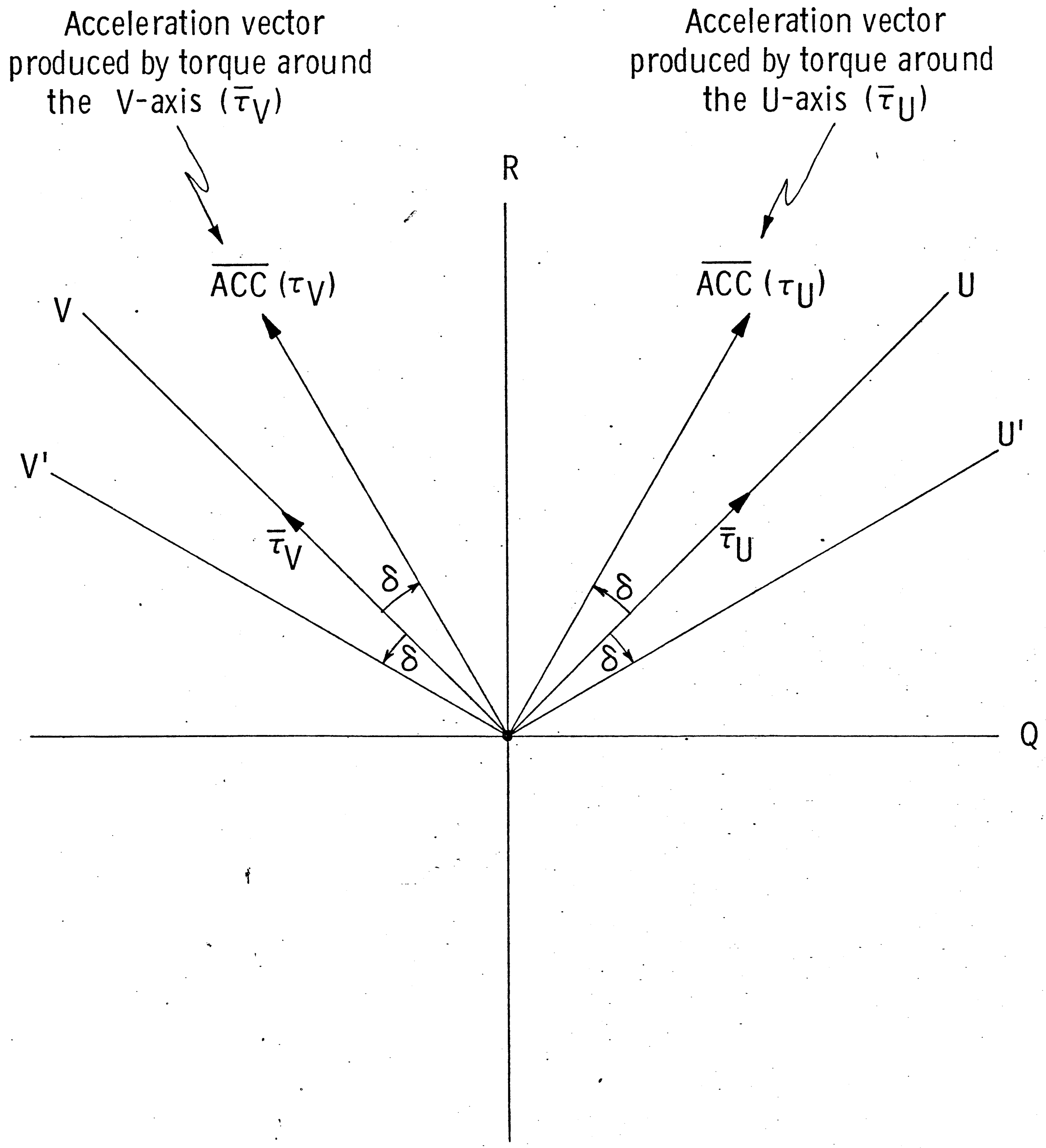


Figure D. 6-4 Nonorthogonal LM U', V' Axis System efficient. For less precisely determined tasks, however, manual control is better. For instance, station keeping and gimbal-lock avoidance can best be performed in a manual mode. In such instances, the pilot's ability to perceive the state of his vehicle and to select alternatives to the nominal control is clearly an asset.

In the GN&CS, two manual modes are implemented through the use of a Rotational Hand Controller. Two such controllers are available (although they cannot be used simultaneously), one for each of the astronauts onboard the LM. The Minimum-Impulse mode provides a single, 14-msec thruster pulse* each time the controller is moved out of detent; each pulse results in an angular-velocity increment whose magnitude is a function of vehicle inertia.

The second—and by far, the more significant—manual mode is Rate Command/Attitude Hold, which incorporates a number of features that enhance the rapidity and precision of control response. This closed-loop mode provides angular rates which are a function of the degree of RHC deflection. The remainder of this section will deal exclusively with the evolution of the Rate Command/Attitude Hold mode.

D.6.2.1 Rate-Command/Attitude-Hold Mode

Development of the Rate-Command/Attitude-Hold mode has been both lengthy and complex. The LM AGC program for Apollo 9 (SUNDANCE) contained manual rate-command logic which was a digital realization of earlier reaction-control rate-command systems. This logic provided rate command within the resolution of a rate-error deadband when the RHC was held out of detent. Automatic attitude hold was maintained when the controller was in detent.

In the manual rate-command logic of SUNDANCE, RHC output was scaled to a maximum commanded rate of 20 deg/sec in "Normal" and 4 deg/sec in "Fine". Normal would be used for a lunar landing, in which high maneuver rates might be necessary; Fine would be used for all other operations, where a premium would be placed on precision and fuel-saving low rates.

---

*The RCS jets require a minimum pulse width of about 14 msec to ensure a proper mixture ratio in the combustion chamber.

The SUNDANCE program provided acceptable manual control for the LM-alone configuration, as evidenced both by simulated and actual earth-orbital flight. But improvements in the manual rate command were deemed necessary to meet the following objectives for an actual lunar landing (program LUMINARY):

a. to reduce drift about uncommanded axes
b. to provide more precise rate control
c. to assure positive return to attitude-hold mode after rate commands
d. to make manual rate command available for coasting flight in the CSM-docked configuration
e. to reduce the on-time of the +X firing thrusters during the lunar landing.

The following sections discuss each of these objectives in terms of the LM DAP manual rate-command mode.

D.6.2.1.1 Reduction of Drift

Although SUNDANCE's rate-command-with-deadband was an acceptable mode from a handling-qualities standpoint, it was open-loop for small, secular errors. Because the thruster switch curves were independent of attitude error, it was possible for the spacecraft to have an uncorrected drift rate just barely within the deadband.

Two factors complicate the drift problem. If the controller were out-of-detent about any axis, all three axes used the manual logic. Consequently, the spacecraft could drift about uncommanded axes (up to nearly 2 deg/sec with normal scaling). Also, a bias acceleration could cause the phase point to chatter along the switch curve. Sampling and state-estimation delays compounded this drift. To limit drift, attitude errors were incorporated in LUMINARY's control computations.

D.6.2.1.2 Precise Rate Control

The rate deadband determines the resolution of rate control. Targeting jet-on time for zero rate error and using inertial and bias acceleration estimates often result in rate-step response with errors initially smaller than the deadband. Such precision cannot be guaranteed, however, because of error in knowledge of vehicle rate or of control authority. Once the rate error is within the deadband at a sampling instant, firing ceases. A heavy configuration requiring more than a 0.1-sec firing for a rate change equal to the deadband will never have rate error nulled entirely, and firing may stop just within the deadband for lighter vehicles (depending on initial rate error). Furthermore, uncertainty in command response extends to twice the error-angle deadband width, measured from zero. If the rate error is just barely within the negative limit and a positive change is requested, the rate error must traverse the entire deadzone before a firing occurs. To obtain precise rate control, LUMINARY's manual rate command applies integral compensation.

Tightening the rate loop alone proved insufficient to improve the pilot's estimation of handling qualities in the lunar-landing task. (It was still virtually impossible to achieve small attitude changes during simulations.) The difficulty lay in the small amount of deflection required to obtain RHC output and in the sensitivity of the controller. (During simulations, pilots "felt" or "heard" the detent switch click, yet, depending upon the particular RHC, up to 1.5 deg additional deflection was necessary to obtain any output signal.) With small, smooth hand motions, small attitude changes clearly are difficult to command, since the pilot cannot predict when the voltage buildup begins. For LUMINARY, therefore, RHC sensitivity was modified to resolve these difficulties.

D.6.2.1.3 Return to Attitude-Hold Mode

Once the RHC is returned to detent, control should be passed from manual rate command to attitude hold positively and with a minimum transient. The latter requirement is met if rates are damped before the switch to automatic attitude hold; if the rate error were large, the attitude-hold phase-plane logic could command oscillatory response in seeking to null the attitude error in minimum time, consuming RCS propellant unnecessarily. Small rate error is not imperative for mode change, however. Return to automatic control should be assured whenever the RHC is returned to detent, even if any or all components of angular rate fail to damp within a short time. Once damping about an axis has reduced the rate error to a small value, that axis should be considered to have passed the damping test. Chattering about more than one axis can delay—and possibly prevent—return to attitude hold. In the SUNDANCE logic, the requirement for the return, after the RHC is returned to detent, was that all rate errors be less than the rate deadband simultaneously. Phase-point chattering out-of-phase will fail this test. If this occurs as a result of an undetected jet failure or a mass uncertainty, the only ways to return to attitude hold are to inform the computer of the failure or to momentarily switch the DAP mode. In LUMINARY, rates about all axes are normally damped before the switch to automatic mode; however, the return to automatic mode is forced if damping has not been completed by the end of a brief interval.

D.6.2.1.4 Availability for CSM-Docked Configuration

CSM-docked rate command was not a requirement of SUNDANCE. To accommodate the use of this mode for coasting flight in Apollo 10, the minor required changes were made in LUMINARY.

D.6.2.1.5 Reduction of +X-Thruster On-Time

Simulation of an early version of LUMINARY uncovered an excessive total on-time of the RCS thrusters during manual landing simulations. The additional RCS propellant usage was discomforting, but the primary concern was the cumulative heating of the descent stage which would be caused by exhaust impingement of the down-firing (+X) RCS thrusters*. Inhibiting the +X jets for small rate errors was proposed as a solution; however, the resultant deterioration in handling qualities was unacceptable. In simulations, pilots were forced to use larger rates more often, bringing the +X jets back into use. As a result, actual mission savings were unpredictable. This was one indication that handling qualities were at the base of the problem. Prior research indirectly indicated that improved handling qualities, through reduced controller sensitivity, might alleviate the problem. This proved to be the correct solution; thus, to minimize RCS on-time in manual control, handling qualities were optimized.

D.6.2.1.6 RHC Scaling

The sensitivity of commanded rate to RHC deflection is the most important manual-control parameter, once rotational-control acceleration ("control power") is fixed. A range of controller sensitivities that provides stable human-pilot loop

*The later addition of jet-plane deflectors (see Section D.6.5) somewhat alleviated this problem, but a heating constraint still exists for the deflectors.

closures can often be defined; as a consequence of the pilot's adaptive ability, however, optimization of the sensitivity within that range is a subjective process. Choice of scaling can be affected by control power, vehicle and control-system dynamics, external disturbances, the control task, and the individual pilot's ability to perceive and react.

It has been found that reduced controller sensitivity has a striking effect on the consumption of RCS propellant: there is a monotonic reduction with decreasing sensitivity. A reduction of maximum commanded rate (MCR) from SUNDANCE's 20 deg/sec to 14 deg/sec produced improved handling qualities, according to several test pilots. (The emphasis in these tests were placed on accuracy in flying to a designated site and on reducing the +X-jet on-time.) Moreover, handling qualities continued to improve as the MCR was reduced to a final value of 8 deg/sec. Reduced MCR improved jet-on time and RCS fuel consumption—and also landing-point accuracy.

In spite of the improvements resulting from reduced controller sensitivity, one conflict remained: reduced sensitivity made small rates and small angle changes easier to obtain, but there was concern that the MCR was insufficient for emergency conditions. A 20-deg/sec MCR was deemed mandatory by the astronauts. The solution adopted is nonlinear scaling of the RHC output.

D.6.3 Coasting Flight

The LM DAP has four coasting-flight modes which may be utilized in the LM-ascent, LM-descent or CSM-docked configurations. These modes are Rate Command/Attitude Hold, Minimum-Impulse Command, Attitude Hold and Automatic Maneuvering. Each mode controls vehicle attitude with the 16 RCS jets located on the LM ascent stage. Since Rate Command/Attitude Hold and Minimum-Impulse Command are discussed in Section D.6.2, this section discusses only the coasting-flight performance of the LM DAP in the Attitude-Hold and Automatic-Maneuvering modes.

D.6.3.1 Attitude-Hold Mode

The Attitude-Hold mode stabilizes the spacecraft about each of the Inertial Measurement Unit's reference axes to hold the spacecraft to within a specified deadband. The principal design objectives included minimization of RCS propellant consumption, minimization of the number of RCS jet firings, acceptable operation in the presence of detected and undetected RCS jet failures, and rapid recovery from large attitude-error and attitude-rate excursions.

D.6.3.1.1 Ascent and Descent Configurations

In the ascent and descent configurations, the number of jet firings and the RCS propellant consumption are both minimized when a minimum-impulse limit cycle is attained about each of the three control axes. (A minimum-impulse limit cycle about a given axis in the absence of disturbing torques is defined as a limit cycle in which a single torque impulse of the smallest available duration, i.e., 14 msec RCS firing, reverses the attitude rate whenever the attitude error drifts out of the deadband.) Although only one jet is fired for each U- or V-axis minimum-impulse torque correction, the displacement of the vehicle center of gravity from the RCS jet plane necessitates the use of two jets fired as a force couple for each P-axis minimum-impulse firing to prevent P-axis firing from disturbing U- and V-axis limit cycles. The RCS control-law phase plane for ascent and descent coasting flight, as illustrated in Fig. D.6-5, was designed for minimum-impulse limit-cycle operation in the steady state. For example, when the state is in the coast zone, Zone 4, with a small positive error rate, $\dot{E}$, the error E increases until Zone 3 is entered. When the state is in Zone 3, the LM DAP commands a 14-msec minimum-impulse RCS firing, which induces a small negative rate and causes the state to reenter Zone 4. The error then decreases until Zone 3 is reentered on the other side of the phase plane. At this point, another minimum-impulse firing induces a small positive error rate and completes the cycle. In coasting-flight minimum-impulse limit cycles, the maximum attitude errors are determined by the deadband size; the maximum attitude-error rates are determined by the control authority of the RCS jets; and the limit-cycle frequency is a function of both the control authority and the deadband size.

The LM DAP was designed to provide acceptable control of the ascent and descent configurations in the presence of detected and undetected jet failures. When jets are disabled due to detected failures, the LM DAP jet-selection logic modifies the selection of jets to exclude those which have been disabled. Single undetected jet-off failures cause rate and attitude undershoot when selected for attitude control,

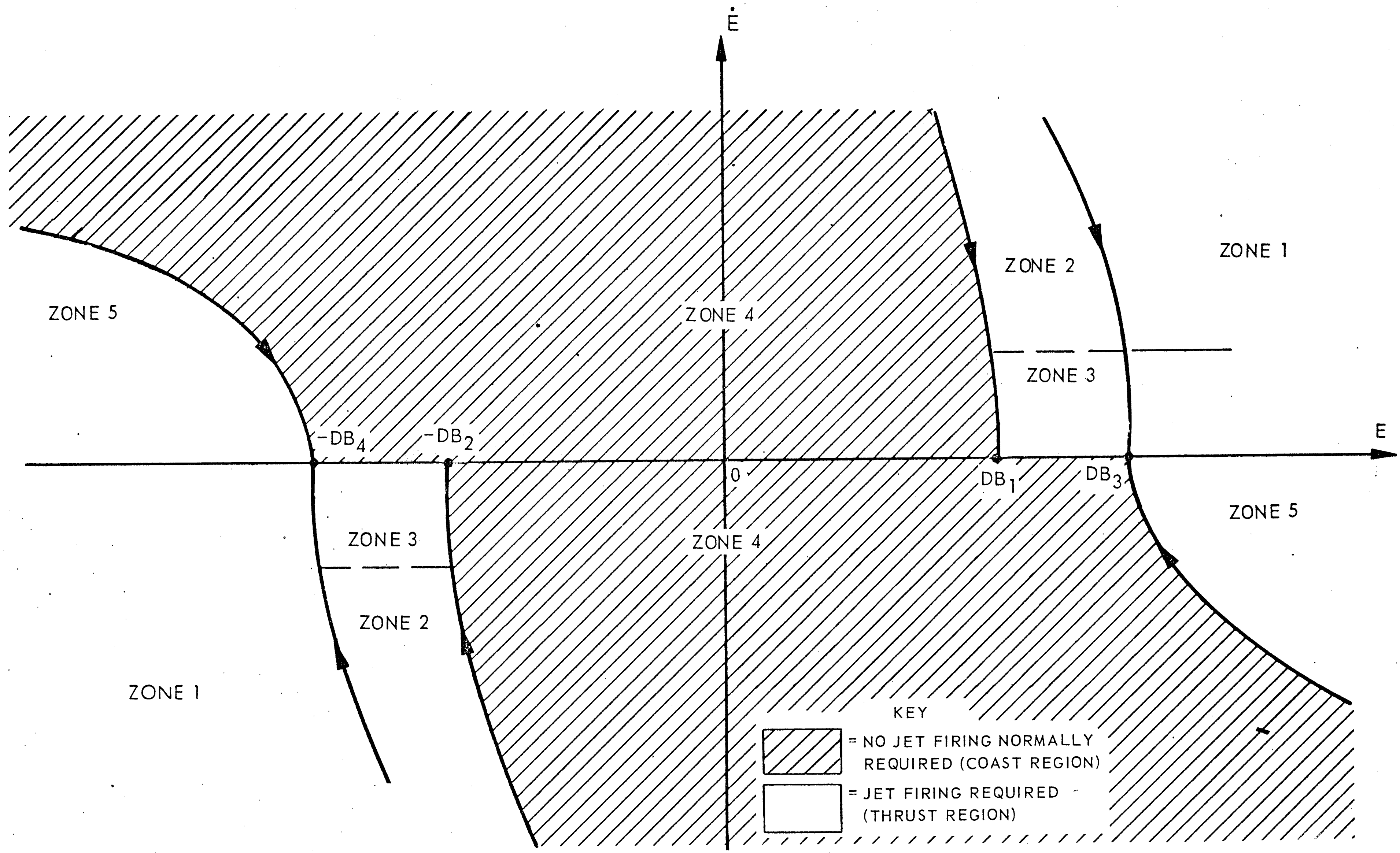


Figure D. 6-5 RCS Control-Law Phase Plane for Coasting Flight (Ascent and Descent Configurations) but result in no fuel penalties. Simulation results indicate that minimum-impulse limit cycles are attained by the LM DAP with any single jet pair disabled and with any single failed-off jet undetected. Undetected jet-on failures degrade the LM DAP performance considerably; however, the degradation lies within the limits NASA deems acceptable.

Rapid recovery from large attitude errors and error rates was provided by the Zone 1 and Zone 5 logic of the RCS control law (FINELAW), illustrated in Fig. D.6-5; and by the special control law (ROUGHLAW) for very large attitude errors and error rates illustrated in Fig. D.6-6. If the state is in Zone 1 or Zone 5 of the FINELAW phase plane, the DAP commands the jets to fire until the state crosses one of the Zone 4 boundary-target parabolas and enters the coast zone. For example, for large negative attitude errors and error rates, the Zone 1 logic would cause positive-torquing jets to fire until the state passed through Zone 5, and crossed the target parabola to enter Zone 4 with a positive error rate. The attitude error would then increase, causing the state to drift across Zone 4 until Zone 2 was entered. The Zone 2 logic would cause the jets to torque negatively until the state had crossed Zone 3 and entered Zone 4, again with a very small negative error rate. The attitude error would then slowly decrease, causing the state to drift across the coast zone into Zone 3. In Zone 3, a single minimum-impulse jet firing would occur, producing a small positive error rate and initiating a minimum-impulse limit cycle.

If the attitude-error magnitude exceeds 11.25 deg or the error rate exceeds 5.625 deg/sec, the ROUGHLAW control-law phase plane in Fig. D.6-6 applies. If the state is in Zone A or Zone D, jets are fired until the rate magnitude is 6.5 deg/sec. No jets are fired by the DAP as the state drifts across Zone C. When Zone B is entered, the jets are fired continuously until the FINELAW region is entered.

D.6.3.1.2 CSM-Docked Configuration

Since the CSM-docked LM DAP was intended for use only as a backup system, the simple control law illustrated in Fig. D.6-7, which differs considerably from the ascent- or descent-configuration phase planes, was employed merely to minimize

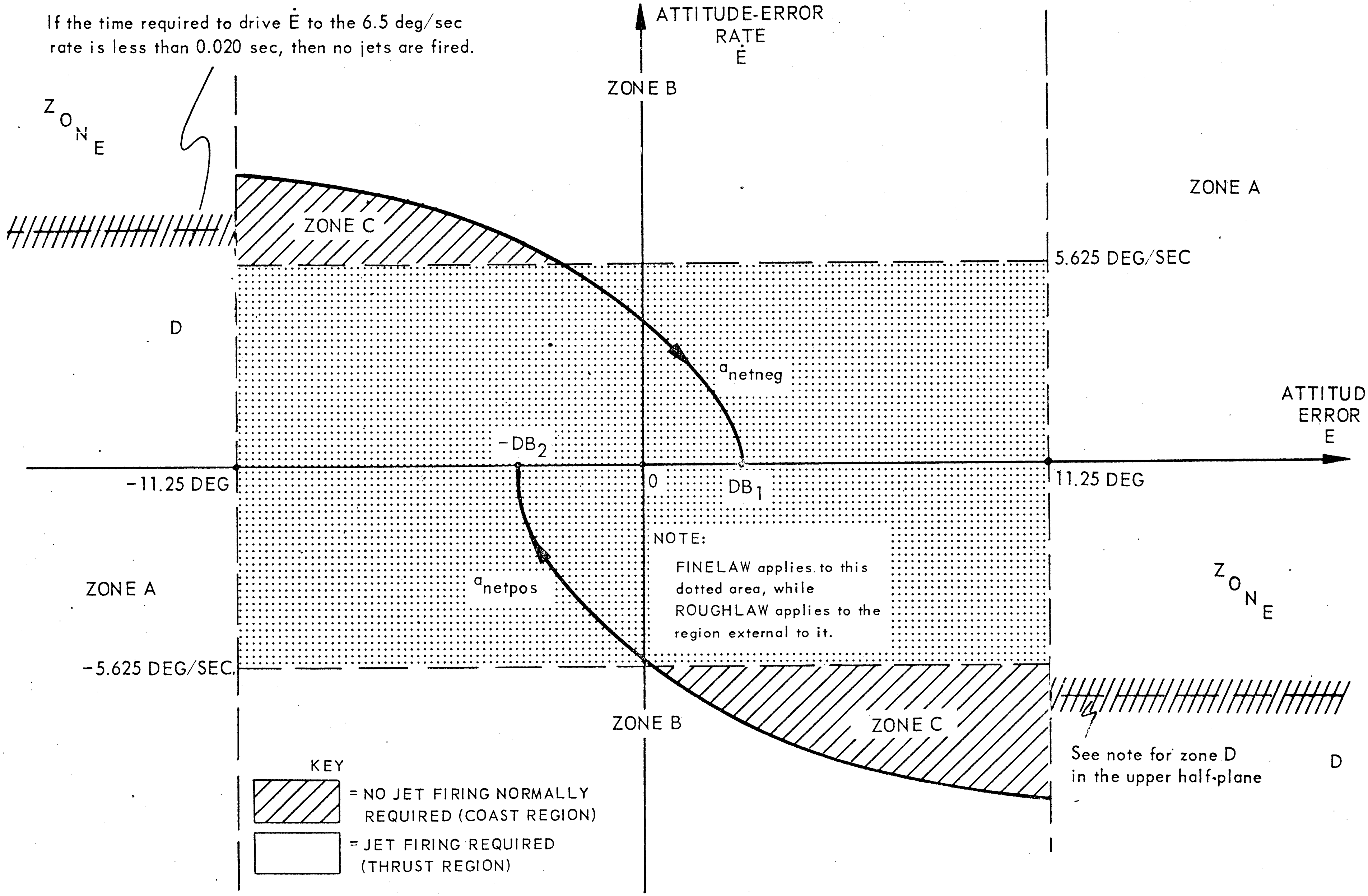


Figure D. 6-6 RCS Control-Law Phase Plane for Large Attitude Error and Error Rates (Ascent and Descent Configurations)

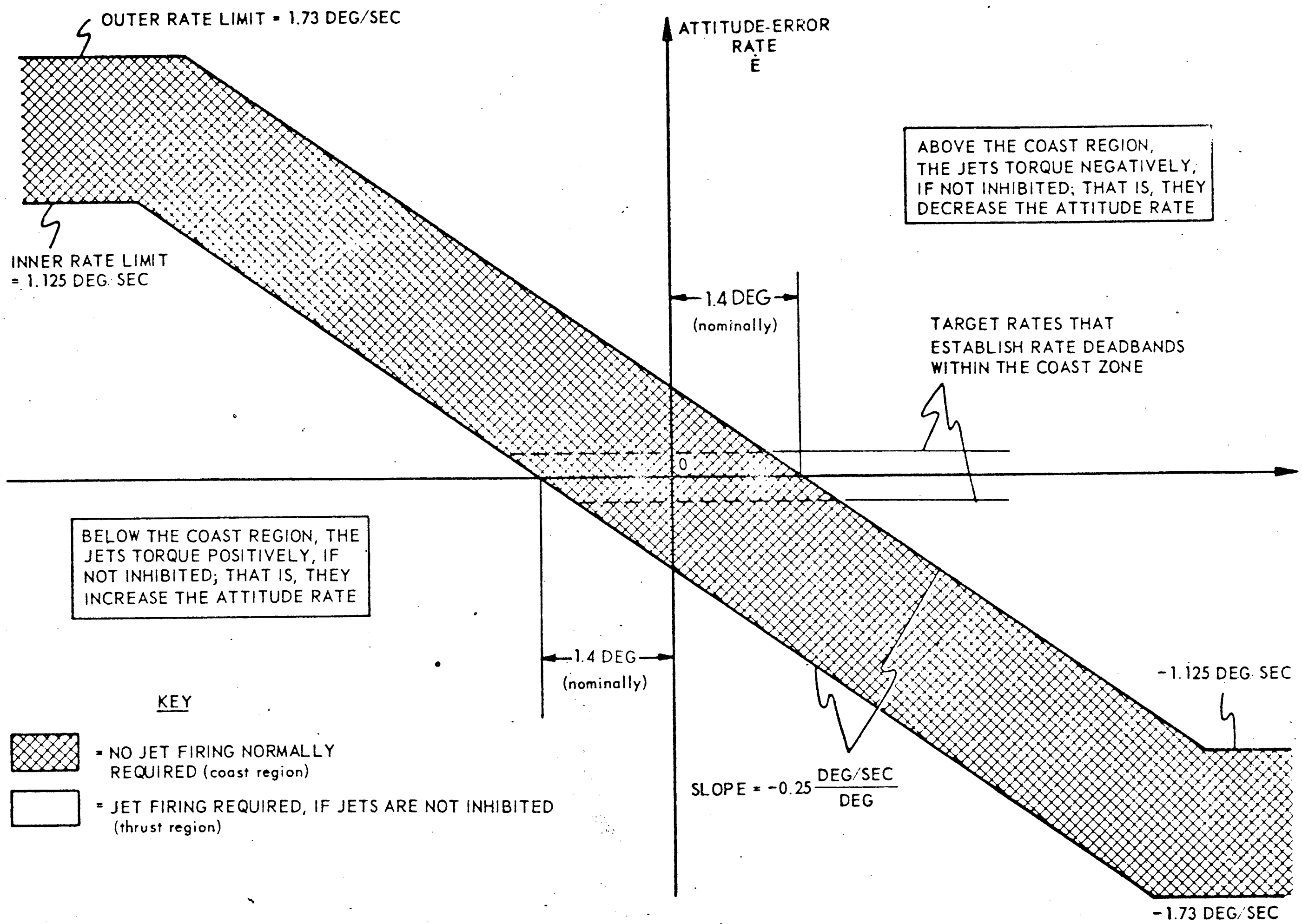


Figure D. 6-7 Control-Law Phase Plane (CSM-Docked Configuration) coding requirements for this configuration*. Each 0.1 sec, the DAP determines the location of the rotational state in the control law and turns jets on or off for a full DAP cycle. If the state is outside the cross-hatched coast zone, the jets are fired until the coast zone is entered. When the state is in the coast zone with the jets still firing, the jets remain on if the trajectory has not crossed the error-rate target (E=0), and are turned off if the trajectory has crossed the error-rate target. If the state is in the coast zone with the jets off, the jets remain off until the state drifts out of the coast zone. Since, in the CSM-docked configuration, the shortest jet-firing time commanded by the LM DAP is 0.1 sec and, in the absence of disabled jets or X-axis translational commands, jets are always fired in pairs about each control axis, minimum-impulse limit cycles cannot be attained. The smallest RCS propellant consumption and the least number of RCS jet firings are obtained when each limit cycle consists of one positive- and one negative-torquing two-jet, 0.1-sec firing as illustrated by the fine-line trajectory in Fig. D.6-8. Due to the large inertia in the CSM-docked configuration, the peak limit-cycle error rates attained in this type of limit-cycle are comparable to the peak error rates obtained in a minimum-impulse limit cycle in the lightest descent configuration.

Acceptable performance of the LM DAP in the CSM-docked configuration with disabled jets is achieved (for most situations) by the same means as in the ascent and descent configurations. That is, the jet-selection logic is automatically modified to select only jets which have not been disabled. Undetected jet failures and, for some mass loadings, disabled -X thrusting jets, however, created challenging problems which were unique to the CSM-docked configuration. In tests of early LM DAP designs, it was discovered that an undetected jet-on failure fixed the vehicle state at one of the coast-zone boundaries of the RCS control-law phase plane, requiring rapid on-off pulsing of the jets to maintain attitude control. Since the jet-pulsing frequency under these circumstances was often close to the natural frequencies of the vehicle bending modes, large bending oscillations could develop. In many of the cases tested, the magnitude of these oscillations became large enough to cause the state estimate to move from one side of the RCS control-law coast zone to the other, resulting in alternate positive and negative torquing of the jets at the bending

---

*The LM-alone control law could not be employed because the large inertias in the CSM-docked configuration caused scaling problems.

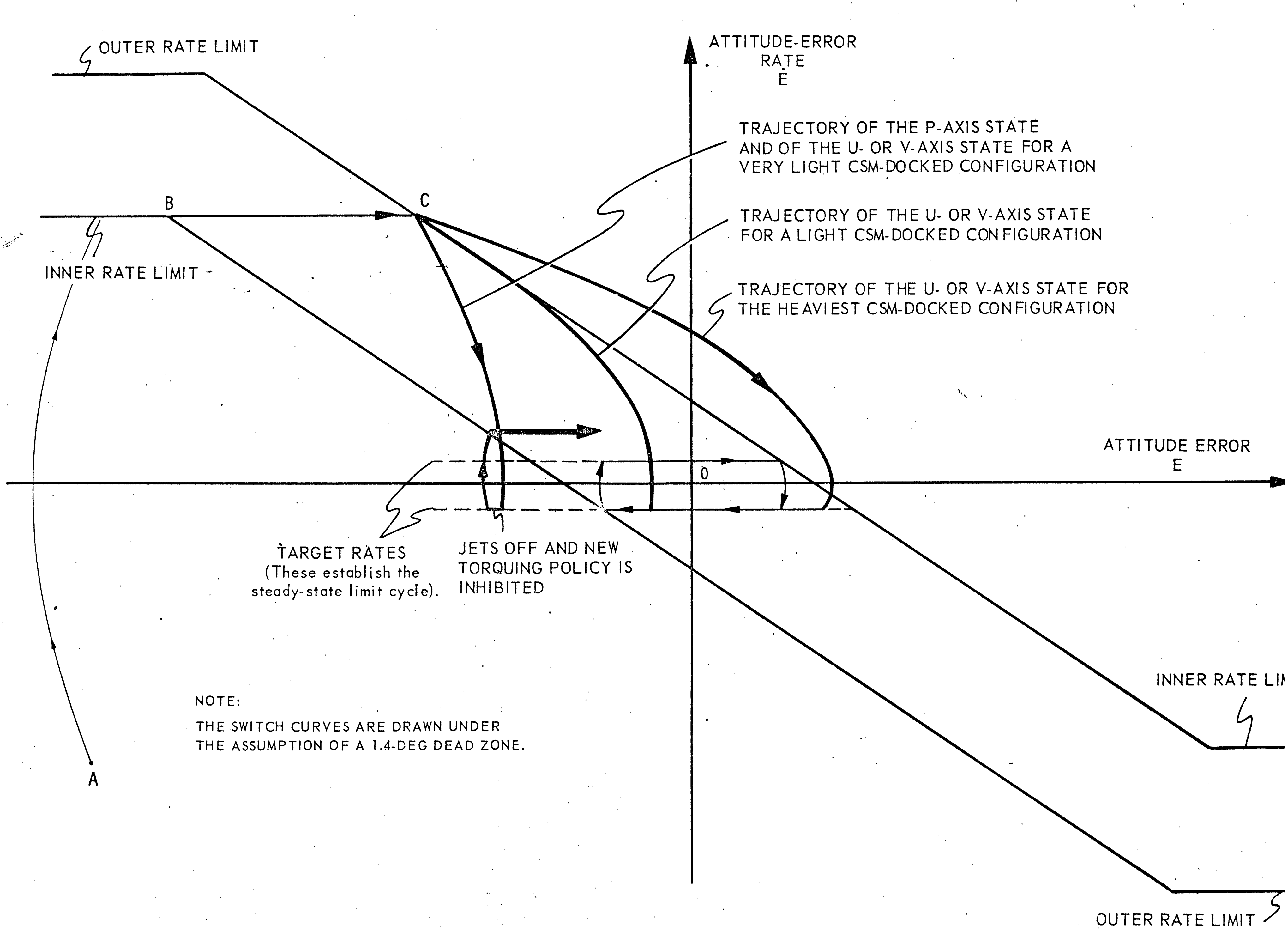


Figure D.6-8 Typical Error-State Trajectories of the CSM-Docked Configuration frequency. Due to sampling lags and lags in the state estimator, these jet firings were in phase with the bending oscillations and would sustain them even after the failed-on jet had been detected and turned off. Such bending instabilities were highly undesirable, since they produced forces significantly larger than the maximum load capability of the LM/CSM docking tunnel. Consequently, the LM DAP was modified to improve the bending stability in the CSM-docked configuration. Two approaches were utilized. The first approach reduced the bending excitation due to RCS jet firings by inhibiting all jet firings about an axis for a predetermined time interval each time the jets firing about that axis were turned off. This jet-inhibition scheme significantly reduced the bending excitation by ensuring that the jet-firing frequency was always lower than the resonant frequencies of the most significant bending modes. The second means of improving the bending stability in the CSM-docked configuration prevented the oscillations which did develop from becoming self-sustaining. With this approach, the jets were turned off and left off for a predetermined time interval each time a reversal of torquing direction was commanded about an axis in which jets were still firing. Simulation results and theoretical worst-case analyses indicated that these modifications reduced to a safe level the maximum bending-oscillation magnitudes and bending moments on the docking tunnel in the presence of a jet-on failure.

For some CSM-docked mass loadings, the presence of a disabled or failed-off -X thrusting jet can cause a serious control instability when the LM DAP attempts to control pitch or roll attitude. The problem can be explained using Fig. D.6-9. Typically, when a jet pair is selected to induce a commanded clockwise rotation about the cg, the 100-lb downward-thrusting jet impinges upon the jet-plume deflector, producing a force of 89 lb in the +X direction acting on the moment arm D_1 and a force of 59 lb perpendicular to the X axis acting on the moment arm D_2. The net moment due to the firing of the downward-thrusting jet is

$$M_{+X} = (89 \text{ lb})D_1' - (59 \text{ lb})D_2$$

which, for many mass loadings, can be negative, thus commanding the vehicle to rotate counterclockwise. Normally the -X thrusting jet can counteract this moment, but should it fail-off or be disabled, a grave instability results. Even in a normal situation, propellant consumption is excessive for the amount of net torque gained.

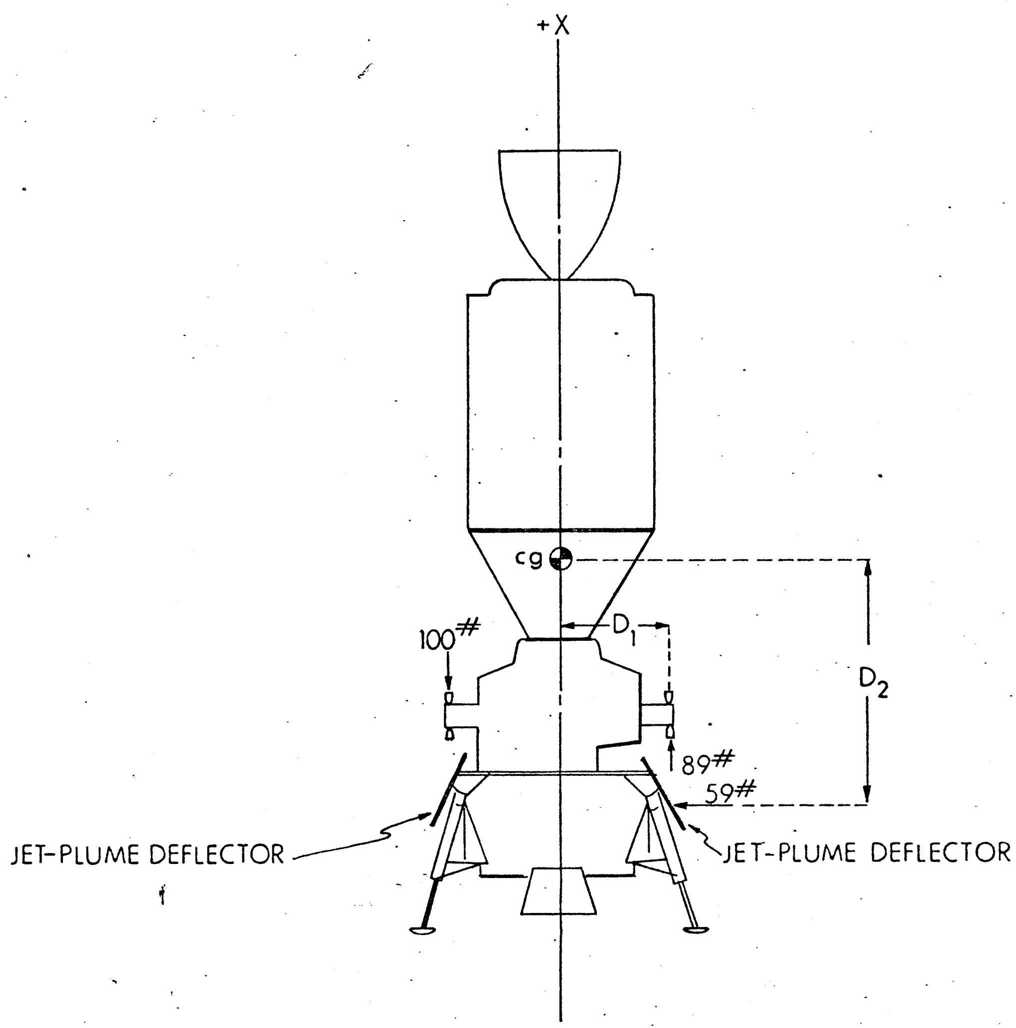


Figure D. 6-9 CSM/LM Docked Configuration with Jet-Plume Deflectors

To eliminate this problem, MIT proposed modification of the DAP jet-select logic such that downward-thrusting jets producing a net torque in the wrong direction are avoided. NASA rejected this software fix and instead used a procedural work-around in which the deflected jets are manually disabled when they produce a net torque in the wrong direction.

The CSM-docked control law illustrated in Fig. D.6-7 included provisions for rapid recovery from large attitude errors and rates. In Fig. D.6-8, for example, jets would be fired continuously to move the state of the spacecraft from its initial position at A to the inner rate limit at B. At B the jets would be turned off to allow the state to drift across the coast zone to C. At C, the jets would fire until the error rate reached zero. Usually, for U- or V-axis control, the state would then begin limit cycling. However, for U- or V-axis control of a very light configuration, or for P-axis control, the higher control authority would cause the state to undershoot to a point outside the coast zone. In such a case, the inhibition logic would prevent jet firings for a short time as the state drifted slightly. The jets would then torque in the opposite direction until the state entered the coast zone, after which a short firing would occur to produce a slightly negative error rate and initiate the limit cycle. (The extra two jet firings needed to recover from large errors and rates, in this case, are the penalty for using a single fixed phase plane to control all axes for all mass loadings in the CSM-docked configuration.)

D.6.3.2 Automatic-Maneuvering Mode

Since the main engine is not thrusting during coasting flight, automatic maneuvers required for certain mission functions (e.g., preburn alignments, rendezvous tracking, etc.) are performed using RCS jets. These maneuvers are controlled by the LM DAP according to a set of steering variables computed by the attitude-maneuver routine, KALCMANU. These steering variables affect only the attitude and rate errors used by the RCS control-law phase planes. A principal design objective of the Automatic-Maneuvering mode is accurate tracking of desired gimbal angles and desired spacecraft rates; in addition, the four design objectives of the Attitude-Hold mode cited in Section D.6.3.1 also apply to Automatic Maneuvering. In every other respect, the Automatic-Maneuvering mode is identical to the Attitude-Hold mode.

D.6.4 Descent Powered Flight

In the powered descent to the lunar surface, the LM DAP must provide considerable precision in attitude control, with rapid response to attitude deviations. Control must be precise because seemingly insignificant deviations from the requested (preprogrammed) trajectory can result in landing-point errors of several miles.

The terminal descent and touchdown of a lunar landing can be done completely automatically; however, some manual intervention is permitted to allow for late landing-site redesignation. This manual-override capability has priority over the automatic guidance requirements. In either the automatic or manual-override control modes, the descent powered-flight DAP can alter its control procedures and/or jet-selection logic in the unlikely event of an RCS jet failure—on or off, recognized or not—or a failure of the descent-engine Gimbal Trim System (GTS).

As discussed in Section 6.1.1, the design of the powered-flight portion of the LM DAP was approached through four overlapping pathways—estimation of slowly-varying parameters, state estimation, RCS jet selection and timing, and GTS-drive selection and timing. However, the GTS-drive selection is particular to descent powered flight and is presented here in greater detail.

The LM descent engine can be rotated about the Q and R axes at a constant rate of 0.2 deg/sec by the Gimbal Trim System, thus directing the thrust vector through the spacecraft center of mass. Consequently, the RCS jet-control system is relieved of the burden of a bias angular acceleration due to an offset thrust vector. This mode of operation is called the GTS acceleration-nulling mode. Further analysis showed that the GTS could also control the LM angular rates and attitude, but only when the flight conditions placed relatively mild requirements on the DAP. This option was implemented in the LM DAP and is called the GTS attitude-control mode. The time-optimal control law [illegible] modified in the LM application so that delays attributable to the 0.1-sec cyc[illegible]riod of the DAP and to the mechanical lags in the trim gimbals would not gener[illegible]n unacceptable large, steady-state, control-limit cycle.

The existence of two independent Q,R axis attitude-control laws (RCS and GTS) created the possibility of conflicting control torques being applied to the LM, so a set of criteria were specified defining the precedence of the two systems, in an effort to obtain maximum benefit from the GTS (saving RCS firings and fuel) while preserving the rapid response of the RCS system.

The RCS/GTS interface was organized to implement these specifications. RCS alone exerts control over LM attitude, using the RCS phase plane and coast zone, until GTS is declared usable by a ΔV monitor and the gimbal monitor. Every two seconds during powered descent, the ΔV monitor and the gimbal monitor verify that ignition has been achieved, that adequate thrust is present, that the astronaut is not signaling a GTS failure, and that the astronaut has not indicated the onset of ascent. When these specified conditions are met, GTS control is admissible.

During a nominal automatic lunar descent, the GTS alternates control with the RCS on successive DAP passes. Considerable care was exercised to ensure that conflicting commands and chatter were avoided. However, should the astronaut select a manual override of the automatic system, the GTS becomes limited to acceleration-nulling control.

Within the GTS portion of the DAP, a flag is checked to determine whether the thrust vector has yet been brought to within one degree of the center of mass. Until this condition is met, acceleration nulling will be executed every two seconds under 1/ACCS. If that flag is set, GTS interrogates the jets before exercising control on every other DAP pass to see whether any RCS jets are firing. If RCS jets are firing, the jets retain primary attitude control, and GTS is temporarily limited to acceleration nulling, with one nulling drive being executed immediately from GTS if attitude control was being exercised by GTS on the preceding GTS DAP pass. If the RCS jet interrogation shows all jets off, GTS executes attitude control every 0.2 sec, controlling the angular acceleration, the rate errors, and the attitude errors as long as the spacecraft attitude remains within the RCS coast zone.

D.6.5 Ascent Powered Flight

The LM ascent configuration in powered flight requires a much higher duty cycle for RCS jet activity than does the powered-descent LM, since, unlike the DPS engine, the APS engine is rigidly mounted, unable to control the angle of the thrust vector relative to the spacecraft cg. Attitude control of the LM ascent configuration is achieved solely by the RCS jets commanded by an appropriate RCS control law.

Since the ascent engine produces only a small torque about the vehicle P axis (the engine is nominally canted only 1.5 deg away from the P axis), the control problem about this axis is relatively straightforward. Indeed, the RCS control law assumes that no disturbing torque exists about the P axis; the major control problem for the ascent LM, therefore, is one of controlling attitude about axes perpendicular to the vehicle P axis—in particular the autopilot U', V' axes.

D.6.5.1 Autopilot Single-Jet Control Boundary

To avoid diminishing the effective thrust of the ascent engine, an important requirement for powered-ascent control is that, whenever possible, only upward-forcing RCS jets be used. In the Q,R plane (which contains the U,V and U',V' axis systems) a locus of vehicle cg positions exists, within which, theoretically, no more than a single U-axis or V-axis RCS-jet firing would be necessary to maintain attitude control of the vehicle. This locus is called the theoretical single-jet control boundary. For cg positions lying on the single-jet control boundary, at least one upward-forcing jet must fire continuously to maintain control; and for the particular axis (U or V) about which the jet is firing continuously, the net torque produced about that axis by the ascent engine and by this continuously-firing jet is zero. For cg positions outside this boundary, control cannot be maintained unless at least one downward-forcing jet is used.

In describing the single-jet control boundary, it is useful to employ a quantity called the "effective" cg displacement from the thrust axis of the ascent engine. The components of the effective cg displacement from the thrust axis along the Q and R axes can be defined as follows:

$$cg_{Q(eff)} = \frac{\text{torque about the R axis produced by ascent engine}}{\text{ascent-engine thrust}}$$

$$cg_{R(eff)} = \frac{\text{torque about the Q axis produced by ascent engine}}{\text{ascent-engine thrust}}$$

In Fig. D.6-10, the locus of effective cg displacements which defines the theoretical single-jet control boundary is drawn in the Q,R plane. In addition, locations of the upward-forcing RCS jets is shown together with the U,V and U',V' axis systems.

In calculating the theoretical control boundary shown in Fig. D.6-10, it is assumed that the component of the ascent-engine thrust along the vehicle X axis has a magnitude of 3500 lb, and that a U jet produces a torque of $550\sqrt{2}$ ft-lb about the U axis, and a V jet produces a torque of $550\sqrt{2}$ ft-lb about the V axis. Under actual operating conditions, these assumptions do not hold exactly true, and some deviation from the theoretical control boundary can be expected. For the purposes of this discussion, however, it will be assumed that the single-jet control boundary can be represented as in Fig. D.6-10.)

During powered ascent, the autopilot determines whether the cg lies within the single-jet control boundary by estimating the net control acceleration produced by a single upward-forcing jet about the appropriate U' or V' axis. (Net control acceleration is the angular acceleration about a given control axis, U' or V', which results from the combined torques produced by the commanded RCS jet and by the ascent engine.) Due to the way in which the U', V' system was constructed, a net control acceleration of zero about the U' axis is equivalent to a net torque of zero about the U axis; and a net control acceleration of zero about the V' axis is equivalent to a net torque of zero about the V axis. As noted above, the determination of a zero net torque about the U or V axis is required to establish the single-jet control boundary; and thus the computation of net acceleration in the nonorthogonal U', V' system allows the autopilot to make this determination.

It is not desirable, however, for the autopilot to allow net control acceleration to become as small as zero. (This would mean continuous jet firing to merely maintain attitude.) When the autopilot determines that the net single-jet acceleration about a control axis is less than $\pi/128$ rad/sec^2, a decision is made to use two-jet control about that axis. (This number is determined from scaling considerations in the AGC to avoid overflow.) The autopilot, therefore, assumes a single-jet control boundary which lies within the theoretical boundary.

Nominal cg positions for the LM ascent configuration during powered ascent lie well within the single-jet control boundary and would be expected not to require two-jet firings about a control axis.

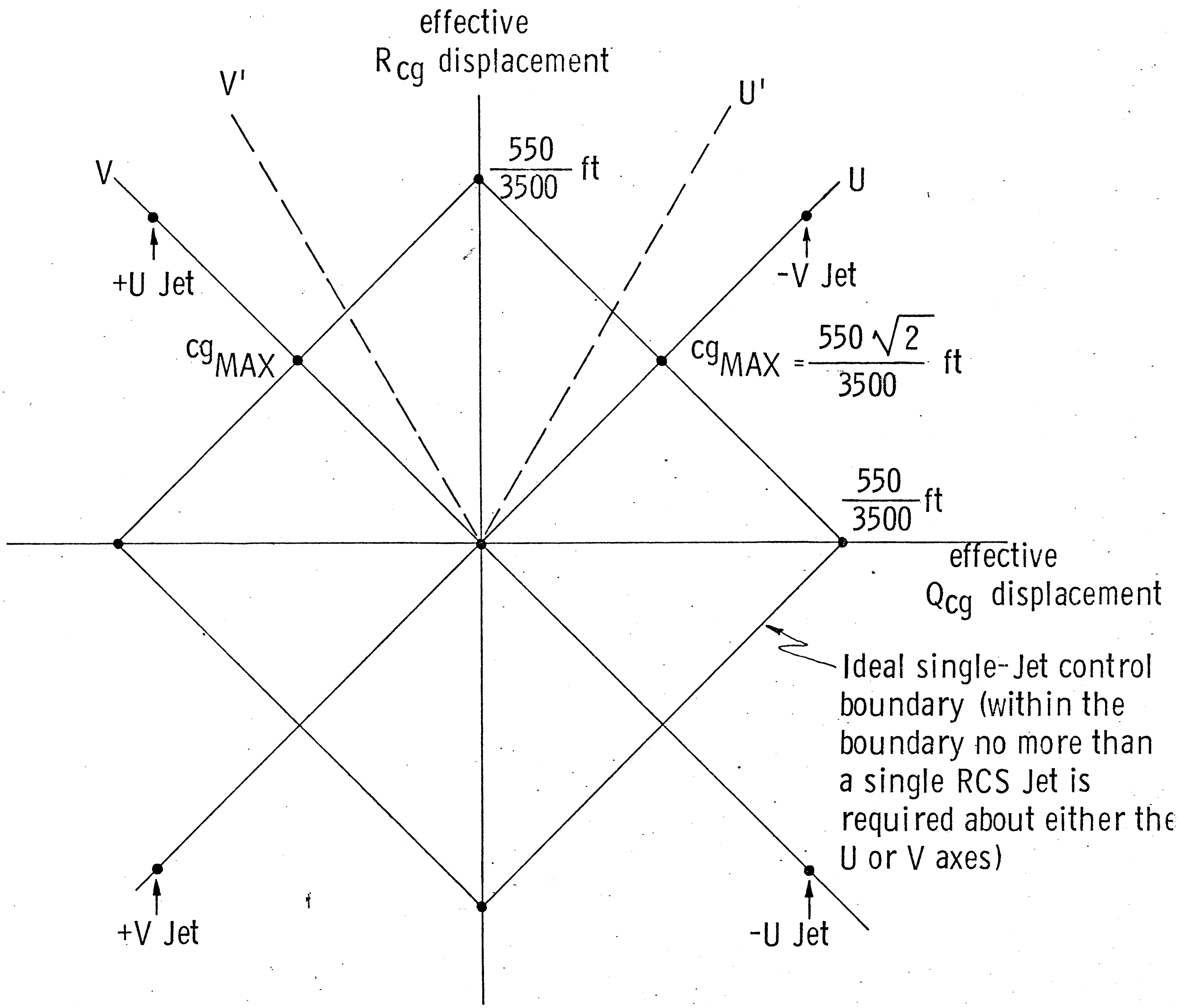


Figure D. 6-10 Ideal Single-Jet Control Boundary

During powered ascent, certain conditions (which may adversely affect guidance), such as an excessive attitude-error angle or error-angle rate, may require the use of two-jet control for the U and V axes, even for cg positions within the single-jet control boundary. The use of two-jet control is mandatory for sufficiently large errors of this type, even though the use of downward-forcing jets is ordinarily deemed undesirable. However, simulations show that in steady-state operation, error-angle excursions are sufficiently small to avoid two-jet control for all cg positions within the DAP single-jet control boundary.

D.6.5.2 Effect of Incorrect Knowledge of Inertia

The vehicle's moment of inertia is not used explicitly by the DAP. However, a priori knowledge of this quantity is assumed in the AGC subroutine which computes single-jet acceleration as a function of mass (for the P, Q and R axes). If the function is not accurate or if the computer's knowledge of mass is inaccurate, then this is equivalent to an error in knowledge of vehicle moment of inertia. Since the autopilot does not have a filter for estimating single-jet acceleration, no correction can be made to the value computed in the AGC; however, the DAP's recursive state estimator does estimate offset acceleration (offset acceleration is defined here as that part of the angular acceleration which cannot be explained by commanded jet firings). An important characteristic of the state estimator is that any error in the computed single-jet control acceleration tends to produce a compensating error in the state-estimator value for offset acceleration. (This compensation does not occur during coasting flight, since offset acceleration is not estimated in this case.) The result is that a compensating error is made in the DAP's computation of the net control acceleration about a control axis.

To test the effect of incorrect knowledge of moment of inertia on autopilot performance, all-digital simulations of the powered ascent LM were run in which extreme "mass-mismatches" existed (i.e., a large error existed in the AGC's knowledge of mass). For the purpose of this test, a special LM subprogram was used in which an effective cg displacement was selected of 2 in. from the thrust axis along the vehicle Z axis, as illustrated in Fig. D.6-11. In one simulation run, a mass-mismatch case was studied in which the DAP determined the mass of the LM ascent configuration to be 2775 kg and the actual mass was 4933 kg. As a

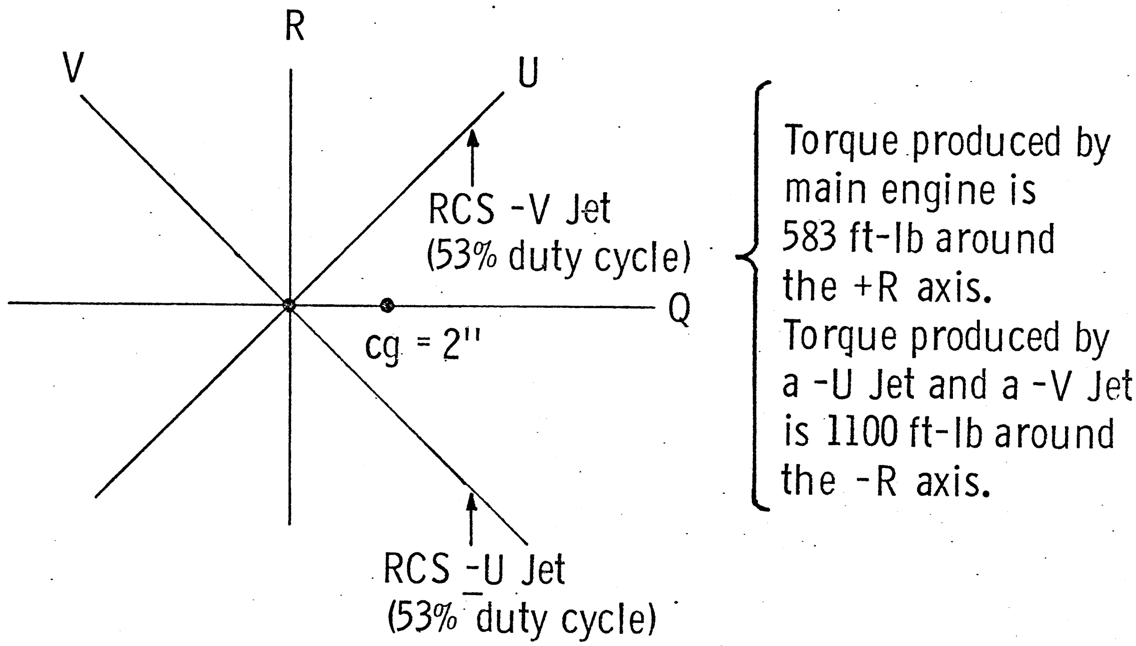


Figure D. 6-11 Effective cg Displacement for Simulations of Mass-Mismatch result, the DAP overestimated the single-jet accelerations about the P,Q and R axes. For example, at approximately 20 sec after ignition, the DAP overestimated the single-jet acceleration about the R axis by 9.1 deg/sec^2 for each of the two jets required to maintain control, for a total overestimate of 18.2 deg/sec^2 for the two jets. This error, however, was partially compensated for by an overestimate in offset acceleration about the R axis of 9.9 deg/sec^2. To illustrate the effectiveness of the autopilot in maintaining control for this case, a phase-plane plot (attitude-error rate vs attitude error) of a typical steady-state limit cycle is given in Fig. D.6-12. The phase plane is shown for the autopilot U' axis. Because of the errors in the DAP's knowledge of the RCS single-jet acceleration, the U' and V' axis directions determined by the DAP differed from the theoretical U' and V' directions; thus, acceleration cross-coupling existed between the DAP U' and V' control axes. The effect of this cross-coupling, however, was not a serious one. On the average, the extra firings required per limit cycle were 1.0 for the U axis and 0.7 for the V axis. Phase-plane attitude-error angle excursions were deemed reasonable. (A second simulation was run in which the DAP determined the mass of the LM ascent configuration to be 4933 kg, while the actual mass was 2775 kg. Again, as in the previous case, no serious complications resulted from cross-coupling. On the average, the extra firings required per limit cycle were 1.3 for the U axis and 0.8 for the V axis.)

D.6.5.3 Effect of an Undetected Jet Failure

The problem of undetected jet failure is similar to that of mass-mismatch, although there are important differences. The case of an undetected failed-on jet presents no special control problem for the autopilot. This is because the angular acceleration produced by the failed-on jet can be considered equivalent to an additional offset acceleration produced by the ascent engine. In either case, the DAP state-estimator filter estimates the acceleration as an offset acceleration, and no error is introduced into the DAP's knowledge of net control authority.

The more difficult case for the autopilot is that of an undetected failed-off jet. When this jet is commanded to fire, the state estimator "sees" the lack of response as resulting from an opposing step in offset acceleration. (The effect is similar to that of the mass-mismatch case where the DAP overestimates the single-jet accelerations about the P,Q and R axes.) If the DAP designates the failed-off jet for single-jet control, then control will temporarily be lost about that axis. This

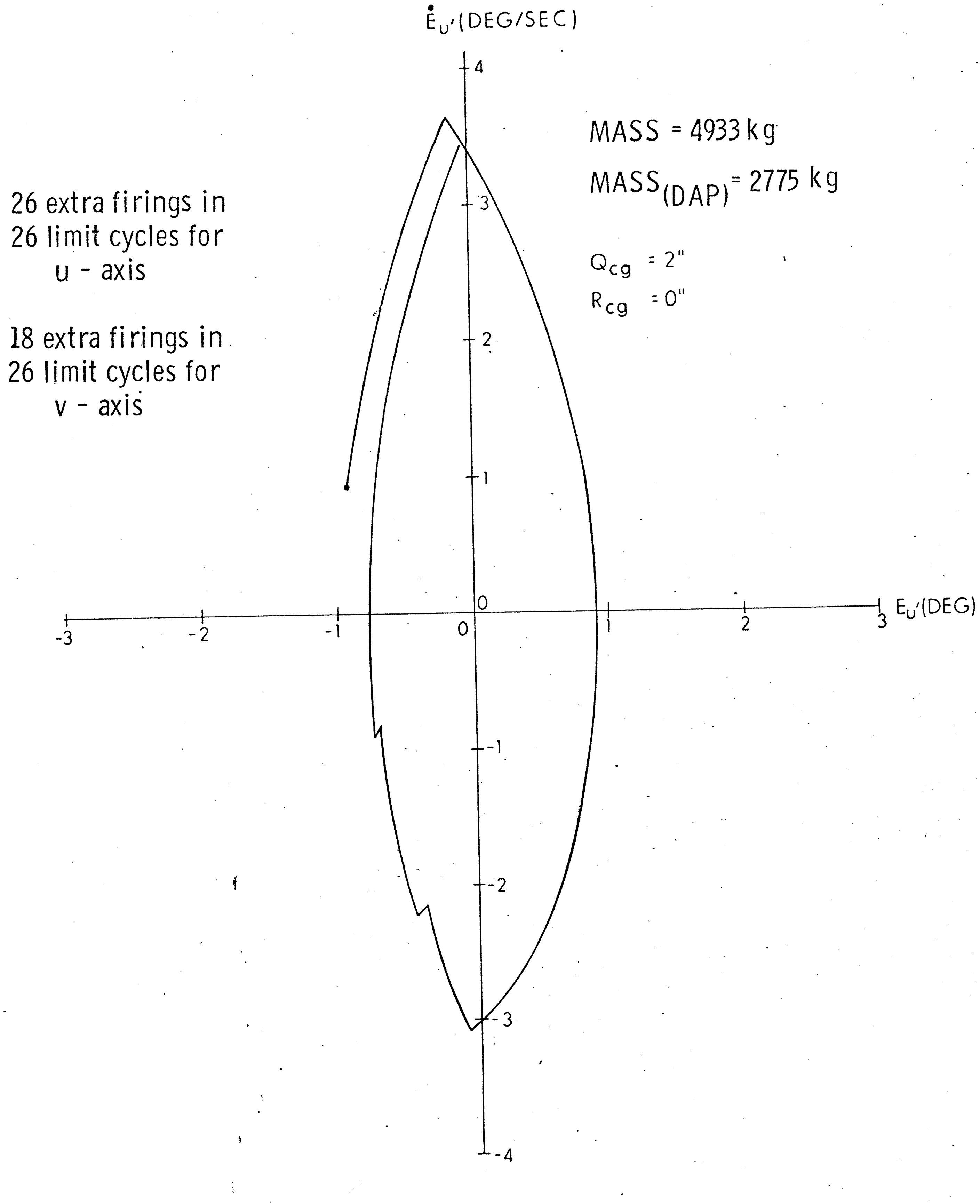


Figure D. 6-12 Limit-Cycle Behavior for Mass-Mismatch Heavy Vehicle–Powered Ascent loss of control will exist until the attitude error or attitude-error rate is sufficiently large to require mandatory two-jet control or until the estimate of net angular acceleration becomes sufficiently small that the autopilot concludes that the cg lies outside the single-jet control boundary. This situation is illustrated in Fig. D.6-13. The V' axis phase-plane limit cycles in this figure are taken from a digital-simulator run in which the effective cg displacement from the thrust vector is constrained to be along the vehicle U axis, with Q and R components of 0.5 in. On paths A-B, F-G and H-I, single-jet control (using the failed jet) is commanded. Thus, when the phase-plane state lies on these paths, the only angular acceleration acting about the V' axis is the V' component of offset acceleration, and control is temporarily lost. Paths B-C and G-H lie in a region of the phase plane which calls for mandatory two-jet control. On these paths, since one operable jet is firing, control of the vehicle is regained. On paths C-D and I-J, the estimated value of net angular acceleration is sufficiently small to cause the autopilot to conclude that the single-jet control boundary has been exceeded; two-jet control is then commanded. During periods of the limit cycle in which the failed jet is not commanded to fire (c.g paths D-F and H-K), the state estimator tends to reduce its offset-acceleration estimate for the V' axis toward the correct value. This causes the estimate of net single-jet angular acceleration to increase. At points E and K, the estimate of net angular acceleration has become sufficiently large to cause the autopilot to conclude that the cg again lies within the single-jet control boundary. It can be seen from the above that, for the case being tested, the autopilot alternates between periods in which it concludes that the cg lies within the single-jet control boundary and periods in which it concludes that the cg lies outside the single-jet control boundary. This results in "loose" limit-cycle activity, as can be seen in Fig. D.6-13. It will be noted that the existence of mandatory two-jet control regions in the phase plane does prevent unreasonably large attitude errors from building up. For a sufficiently large offset torque about an axis requiring a failed-off jet for single-jet control, the DAP will always determine that the offset acceleration lies outside the single-jet control boundary. The offset torque about the axis in question must exceed 275 2 ft-lb (half the torque produced by one RCS jet) for this to be true. Since single-jet control is never called for, in this case, reasonably tight limit cycles can be expected.

D.6.5.4 Velocity Errors

In general, the average attitude error for an autopilot phase-plane limit cycle is non-zero and will depend upon the values of offset acceleration and RCS control

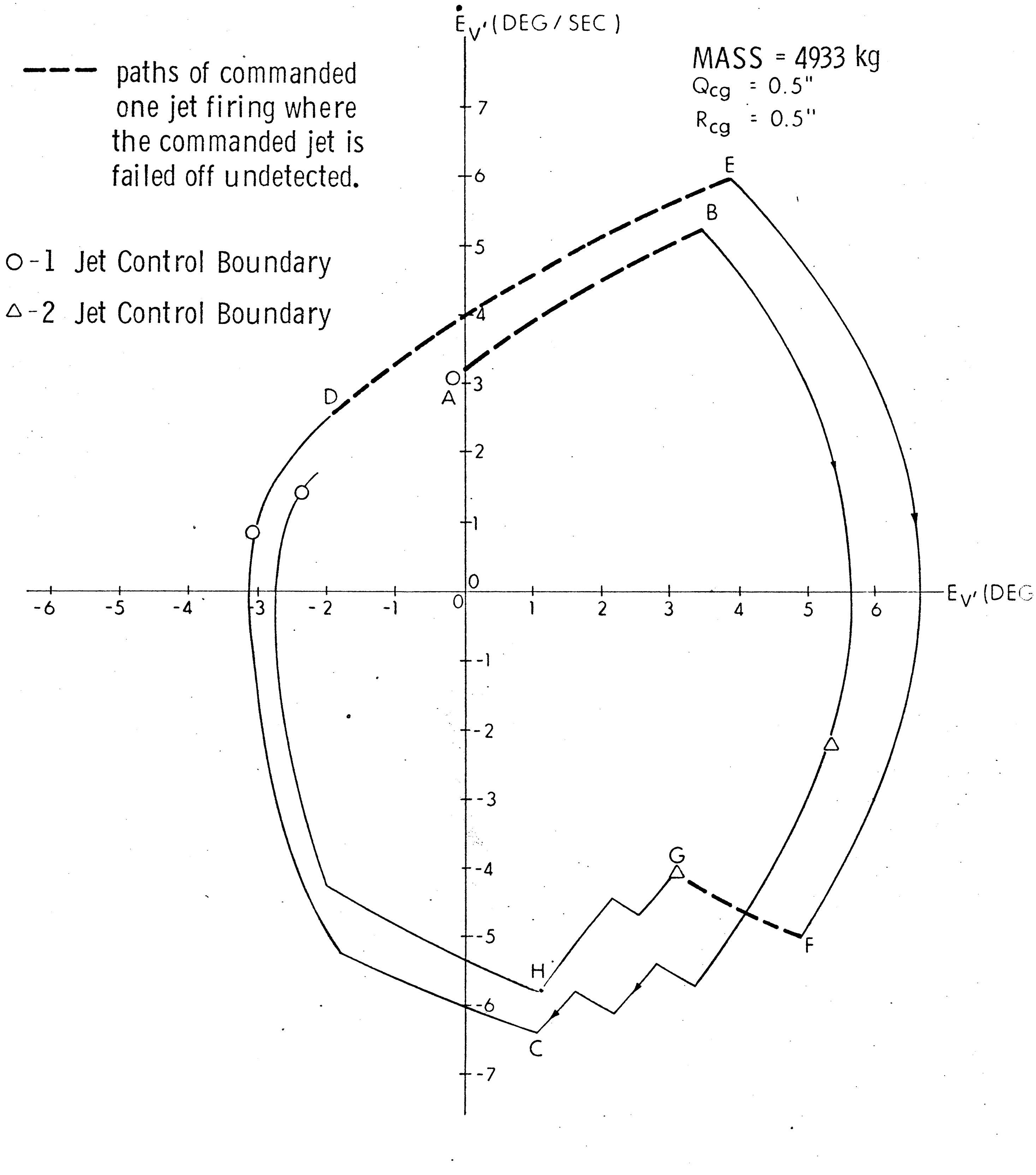


Figure D. 6-13 Limit-Cycle Behavior for -V Jet Failed Off (Undetected)-- Powered Ascent (Nonorthogonal Control Axes) acceleration. During each 2-sec guidance period, this average attitude error will tend to produce "cross-axis" velocity, i.e., a component of velocity normal to the commanded direction of the ascent-engine thrust vector. An additional source of cross-axis velocity is the inaccuracy in the knowledge (by the guidance law) of the direction of the ascent-engine thrust vector with respect to the vehicle axes. One of the functions of the FINDCDUW Guidance/Autopilot Interface Routine (see Section C.7) is to respond to a bias in thrust direction and to modify the commanded direction of thrust provided by the guidance law to correct the bias. The part of the FINDCDUW routine which performs this function is the Thrust-Direction Filter. In the presence of a fixed bias in the direction of the LM ascent-engine thrust vector with respect to the desired X-axis orientation, the Thrust-Direction Filter acts as a first-order filter with a time constant of 8 sec. In general, therefore, it can be expected that for a fixed bias in the direction of the thrust vector with respect to the DAP's desired vehicle X-axis direction, 24 sec will be required for the filter to correct for 95 percent of the bias.

The effects of autopilot bias and thrust-vector misalignment during powered ascent on velocity errors at engine cutoff have been tested on the All-Digital Simulator. The tests used the APS powered-flight guidance program, P42, with external ΔV guidance. A light ascent vehicle was assumed and ΔV was supplied as an input to the program. It was concluded from this study that the Thrust-Direction Filter was effective in reducing the cross-axis velocity resulting from either of the causes discussed above. Attitude-pointing errors produced by the cant of the ascent engine from the vehicle X axis are easily corrected, since the engine cant is a fixed bias throughout the run. Cross-axis velocity at cutoff for all of the runs was generally within 1 ft/sec. (In two test cases, however, cross-axis velocity exceeded 1 ft/sec. An extreme off-nominal cg close to the maximum cg displacement which can be controlled by the DAP produced a 2 ft/sec error at cutoff. In a case where the cg was constrained to lie along the thrust vector, a 1.25 ft/sec error resulted. In this latter case, a slow oscillation of the FINDCDUW thrust-axis direction results. This is a consequence of the flow limit-cycle period in the phase plane for the case in which the offset acceleration is zero. Since the limit cycle period is much larger than the 2-sec guidance period, the Thrust-Direction Filter follows the slow oscillation.)

D.6.6 CSM-Docked Powered Flight

The contingency of a CSM Service Propulsion System failure prior to the initiation of lunar landing would require that the LM descent configuration become the active spacecraft, pushing the CSM/LM configuration into a trajectory for the flight back to earth. Design considerations for LM DAP control of the CSM-docked configuration are complicated by radically altered control authorities and mass distributions and four physical constraints—bending and torsion at the docking terminal; slosh interactions between the two vehicles; RCS jet-plume impingement; and engine-on and throttling transients. These constraints are discussed below.

D.6.6.1 Bending and Torsion Constraints

Bending oscillations—and torsion, to a lesser degree—could jeopardize the structural integrity of the docking tunnel which joins the two comparatively rigid vehicles. Accordingly, appropriate care must be taken in the LM DAP design to ensure that the RCS jets (the principal instigators of bending excitation at critical frequencies) reinforce bending amplitudes.

D.6.6.2 Slosh Constraint

When the vehicles are in the CSM-docked configuration, consideration must be taken for the slosh modes imposed by the presence of both CSM and LM propellant. It must be verified that the LM DAP design ensures that the Gimbal Trim System (the principal agitator of slosh) does not cause slosh amplitudes to exceed acceptable levels.

D.6.6.3 RCS Jet-Plume Impingement Constraints

Thermal constraints imposed by RCS jet-plume impingement are even more severe in the CSM-docked configuration than in the LM-alone. (Section D.6.3 discusses these constraints in the context of LM coasting flight.) In the docked configuration, the LM RCS jets can impinge on both vehicles. The problem is further aggravated, because the larger pitch and roll inertias associated with the CSM-docked configuration necessitate longer jet-on times to achieve desired performance.

To combat the serious thermal constraint imposed by the jet-plume impingement, the crew can initiate, via the DSKY, LM DAP routines to inhibit the firing of U- and V-torquing jets during CSM-docked powered flight. In addition to this software approach, jet-plume deflectors were added to the LM; their effect upon LM DAP control of the CSM-docked configuration is discussed in Section D.6.3.

D.6.6.4 Constraints Related to Engine-On and Throttling Transients

Because all but the yaw-axis jets are inhibited during CSM-docked powered flight (thus preventing impingement on either the CSM or the LM jet-plume deflectors), the GTS must assume control of the spacecraft about the pitch and roll axes. Under these circumstances, engine-on and throttling transients become a serious GTS design consideration. These transients stem from three factors—computational underflow, initial mistrims of the descent-engine bell, and compliance of the descent-engine mount.

Until the computational-underflow problem was solved in the LUMINARY IA program (for Apollo 11), the scaling of the GTS control law was such that attitude errors could not be detected for the CSM-docked configuration at low thrust. Attitude errors could thus not be steered out during the ten-percent thrust period (26 sec) prior to throttle-up. A 40-percent manual throttle-up a few seconds after ignition avoided these errors*; at this thrust level, the more significant attitude errors could be detected by the GTS and steered out. Because it yielded superior performance, this intermediate manual throttle-up procedure was retained even after the computational-underflow issue was resolved.

Transients are also produced by mistrims of the descent engine prior to ignition. (A "mistrim" is here defined as a deviation of the engine-bell orientation.) Because the engine-bell gimbal drive rate is only about 0.2 deg/sec for the descent engine, substantial time is required to correct large initial mistrims, during which significant excursions in attitude error, rate and acceleration might occur.

---

*All four of these transient factors are kept to a minimum by this 40-percent manual throttle-up procedure.

The descent-engine mount has a thrust-proportional misalignment (compliance) associated with abrupt throttle changes. This association effectively introduces a mistrim at throttle-up—or any other throttle manipulation—with an effect comparable to that noted in the previous paragraph.

A NOTE ON SOURCES

By the time this record of MIT's Apollo software efforts entered the germination stage, many of the souls who had participated in those efforts had begun to disperse to other projects—both within and without the Draper Laboratory. Nonetheless, virtually all wanted to ascertain that that part of the history in which they played such important roles was finally, indeed, recorded. Some personally documented their accomplishments; others supplied bits and pieces that eventually interlocked to permit the construction of a unified whole. A considerable amount of the information recorded within these pages could be gleaned from documents that already existed*, but a surprising—indeed, exasperating—amount had never before been documented. For the latter, memories had to be tapped and taped; 57 transcribed interviews, countless conversations and mountains of notes bear testimony to the cooperation and enthusiasm which I encountered along this historical path.

Considerably more information was gathered than could be presented within any single cohesive text. But what I hope has remained is an insight into the team which carried the concept of Apollo software from a hopeful infancy, through an oft turbulent adolescence, to its magnificently successful goal. The Apollo software team was a heterogeneous, sometimes colorful lot, one which demonstrated two basic characteristics: competence and perseverence. The pressures imposed by the schedule sometimes revealed frailities, but more often demonstrated elemental strengths. The epoch which this history records was a significant time in all the participants' professional and personal lives, and, in recalling this period, no one felt dispassionate. Despite the crushing schedules, the fantastic amount of mental and physical exertion which the project came to demand—the goal which was to be reached seemed to energize us all.

Not every member of the Apollo software team contributed to this report, but a great many did, some extensively, some less so. In listing these persons below,

* A compendium of abstracts of all Laboratory reports pertaining to Project Apollo appears as an Appendix to Volume I of this report.

I stress that whatever kudos are due belong to us all; whatever deficiencies are contained in this report I acknowledge to be my own.

Robert R. Bairnsfather	Ivan S. C. Johnson	Charles A. Muntz
Nyles N. Barnert	Leonard B. Johnson	Peter E. Peck
Richard H. Battin	Malcolm W. Johnson	Alexander Penchuk
Lawrence J. Berman	J. Edward Jones	J. Marshall Reber
Hugh Blair-Smith	Paul F. Jopling	Jack C. Reed
Donald J. Bowler	Peter M. Kachmar	John R. Rhode
Timothy J. Brand	George R. Kalan	William M. Robertson
Edward M. Copps	Donald W. Keene	Richard J. Russell
Stephen L. Copps	James E. Kernan	Phyllis Rye
John M. Dahlen	Allan R. Klumpp	Robert W. Schlundt
Dana Densmore	Albrecht L. Kosmala	Norman E. Sears
C. Stark Draper	George J. Kossuth	Robert F. Stengel
Albert G. Engel	Gerald M. Levine	Gilbert F. Stubbs
Philip G. Felleman	Danial J. Lickly	John B. Suomala
Thomas P. Fitzgibbon	Frederick H. Martin	Milton B. Trageser
F. Keith Glick	Bruce J. McCoy	Joseph F. Turnbull
Richard D. Goss	Vincent A. Megna	John E. Vella
Alan I. Green	James S. Miller	Boyd A. Watson
Kenneth W. Greene	Pieter R. Mimno	Peter S. Weissman
Margaret H. Hamilton	Joseph P. Mori	William C. Widnall
David G. Hoag	John A. Morse	Craig C. Work
Albert L. Hopkins	Eugene S. Muller	Saydean Zeldin

I offer my personal thanks to all those who helped in creating this report, and I hope that the report itself will serve as a fond reminder of rewarding times past.

MSJ