Prepared for the
GEORGE C MARSHALL
SPACE FLIGHT CENTER
Huntsville, Alabama

15 April 1975

Contract No. NAS 8-1400
MSFC No. ASTP DRL 686, Line Item 034
IBM No. 75W-00067

2

Program Verification Document for the ASTP Flight Program

(NASA-CR-143958) PROGRAM VERIFICATION DOCUMENT FOR THE ASTP FLIGHT PROGRAM (IBM Federal Systems Div.) 77 p HC $4.75 CSCL 22A G3/12
N75-33114 THRU N75-33125
Unclas 42228

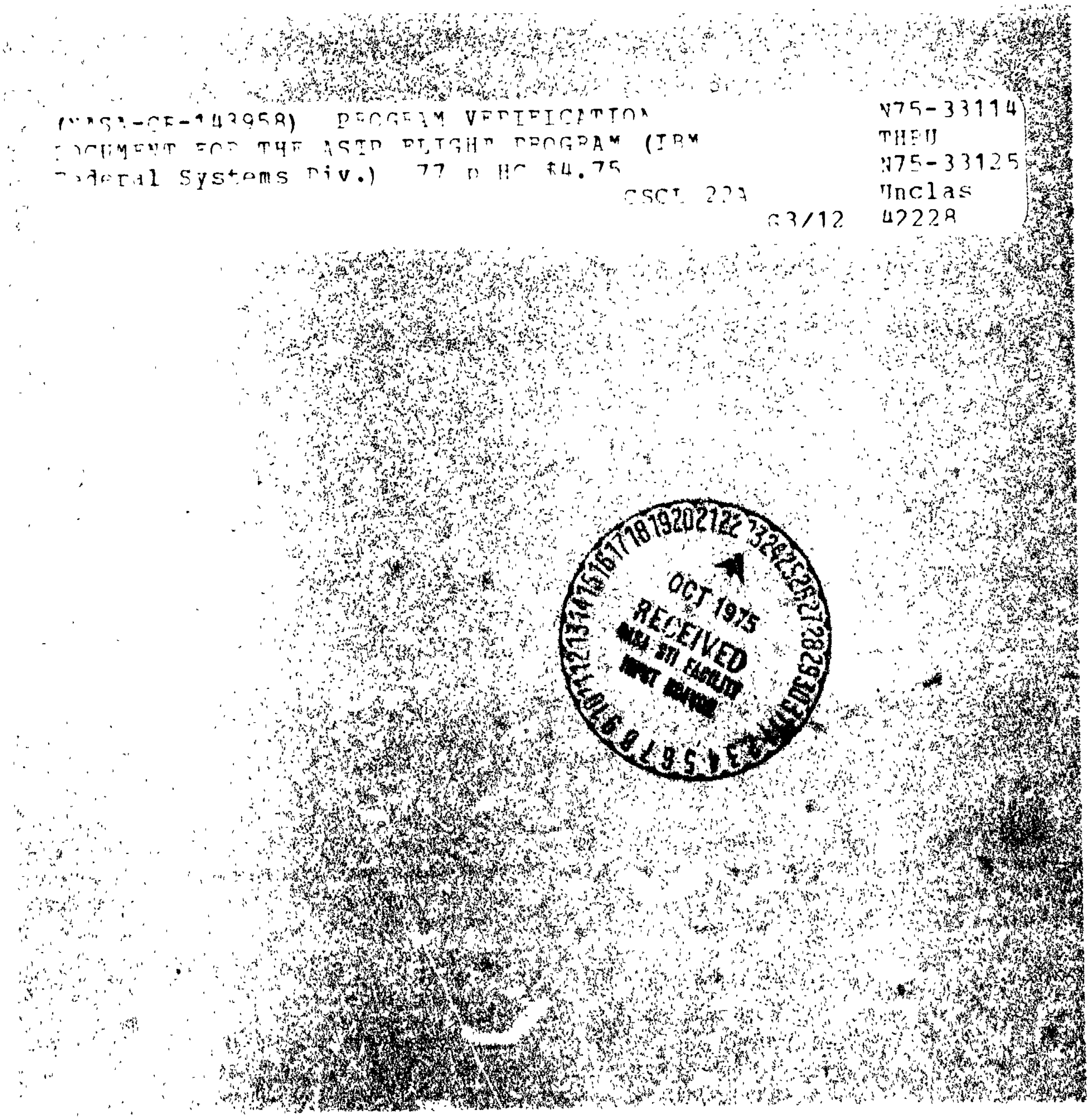

TABLE OF CONTENTS

Prepared for the
GEORGE C. MARSHALL
SPACE FLIGHT CENTER
Huntsville, Alabama

15 April 1975

Contract No. NAS 8-1400
MSFC No. ASTP DRL 686, Line Item 034
IBM No. 75W-00067

Program Verification Document for the ASTP Flight Program

Classification and Content Approval RW Draney

Data Manager Approval Stanley Rodzkowski

Program Office Approval

IBM Federal Systems Division, Space Systems Huntsville, Alabama

PREFACE

This Program Verification Document details the results of the ASTP flight program verification specified in the Program Verification Plan for the ASTP Flight Program, IBM No. 75W-00005 and revised by 75W-00032.

The ASTP flight program verification effort was a complete verification.

The independent simulator used in performing flight program verification is an all digital simulation utilizing an IBM System 370/Model 155 Computer. The simulator program mathematically models both the vehicle (6D) and the LVDC and is referred to as the 6D/LVDC simulator. A detailed description of the simulator is contained in Appendix B of the ASTP Flight Program Verification Plan (PVP).

Numerous support programs were used in verification by summarizing data for rapid analysis and by making independent calculations to aid in evaluation of flight program performance. Some programs, where applicable, are duplicated on different computer systems to ensure accessibility regardless of workload or machine down time. A description of each support program is contained in Appendix C of the PVP.

One of the primary means for determining the adequacy of the flight program was by an analysis of the S-IVB end conditions. Achieved LVDC terminal end conditions were calculated by an independent support program for each performance case which reached S-IVB cutoff. The achieved terminal end conditions were compared against the desired terminal end conditions and the differences are listed in Appendix D.

SECTION 1

GENERAL VERIFICATION DESCRIPTION

1.1 INTRODUCTION

This section contains a brief, general description of ASTP verification. The goal of the verification effort was to assure that the flight program, by meeting all mission requirements, will not be the limiting factor in achieving the mission, even in the presence of other vehicle failures. The discussion follows the outline in the Program Verification Plan for the ASTP Flight Programs (IBM No. 75W-00005).

1.2 TYPICAL MISSION REQUIREMENTS

The total program verification effort assures the accuracy and adequacy of the LVDC flight program and verifies that the final program meets mission requirements and conforms to program documentation. The results are also applied to guidance error analysis.

Verification methods and results of the verification effort performed on the three mission phases (prelaunch targeting, boost-to-earth orbit, and orbital operations) are discussed in the remaining sections of this document.

1.3 GENERAL FLIGHT PROGRAM REQUIREMENTS

1.3.1 Applicable Document List

The documents used directly in verification of the ASTP flight program are listed in Appendix A. The specification document was the LVDC Equation Defining Document (EDD) for the Saturn IB Flight Programs (Reference 1), revision J (ASTP mission), as modified by the ASTP Flight Program Change Requests (FPCRs). The Astronics System Handbook (Reference 2) was used during verification phases as an aid for interpretation of the specifications. The Programmer's Operating Manual (Reference 3) was used to interpret programming techniques.

1.3.2 Program Functional Requirements

The flight program's functional requirements to integrate the guidance and control system with the launch vehicle sequencing system were verified directly by analysis of many special logic checks designed for this purpose and indirectly by the correct overall program response to nominal and numerous perturbed conditions (performance cases). Discussions and results of the verification performed in each area are contained in the remaining sections of this document.

The program's recurring functions through the mission are:

- Navigation
- Guidance
- Attitude Control
- Launch Vehicle Sequencing
- Telemetry
- Programmed backups to specific hardware functions
- Command Processing
- Data Compression

Verification of these functions was accomplished in part by overall program performance, in part by special logic tests, and in part by specially designed independent digital programs (refer to Appendix C of the PVP) which analyzed flight program data.

1.3.3 Requirements Interaction

Verification of the interaction of function requirements was accomplished on every case run during the verification effort. Functions which require fixed repetition rates, such as the minor loop and the orbital guidance, were verified on special test cases. Other test cases, and performance cases which sequenced one-time events were checked to assure detection of each event within the specified time frame. To ensure that the variable repetition rate is always consistent with accuracy requirements, plots were generated showing each major loop computation cycle length. These plots were generated by the PLOTS portion of the SUPER support program for each performance case as described in Appendix C of the PVP.

1.4 PROGRAMMING GROUND RULES

Verification of adherence to programmed ground rule requirements was accomplished as outlined in the following subparagraphs:

1. Duplex computer operation in the flight mode was checked by an independent computer program which ensured that the simplex/duplex mode selection bit was on in the operand address of each Change Data Sector (CDS) instruction and in each Hop Constant (HPC). Assurance that the flight program was initiated in the duplex mode of operation was obtained by decoding the HPC executed as a result of the GRR interrupt.

2. Use of the Generalized Flight Program (GFP) concept and development of the GFP assembler have greatly minimized the effort required to add new requirements and delete old requirements.

Minimized verification effort and better quality control for program changes was achieved through development and use of the symbolic tape compare program described in Appendix C of the PVP.

3. Verification of program variable scaling was accomplished using two methods. Each time an accumulator overflow was executed within the 6D/LVDC simulator, all applicable data (instruction location, accumulator contents, etc.) was printed for subsequent analysis. All overflows were justified through analysis of the program listing. This verification method ensured that the assigned scaling can accommodate the maximum variable values experienced in both nominal and perturbed performance simulations. Attainment of maximum accuracy and program efficiency was verified through a comparison of the simulated vehicle (6D) state parameters with those computed by the LVDC flight program. The difference in these parameters (navigation errors) was within the accepted mission tolerances in all performance cases which had no perturbations to the inertial platform. The accuracy of the 6D is well established and thus provided an adequate check on the accuracy of the flight program parameter scaling.

4. Correct implementation of the algorithms used to compute trigonometric functions and the dot product routine was verified using the same methods described in subparagraph (3) of 1.4 and by checking algorithm outputs after forcing particular known inputs.

5. Flight Program listings were analyzed to ensure that all instructions where various priority levels of interrupts could destroy or alter existing data are interrupt protected.

N75 33116

SECTION 2

REFERENCE SYSTEMS AND TRANSFORMATIONS

2.1 INTRODUCTION

Several coordinate systems and transformation matrices are defined for the Saturn IB mission. The verification accomplished to assure correct computation of the many geometrical relationships of these functions is discussed in this section.

2.2 REFERENCE SYSTEMS

The flight program correctly represents and uses all the various three dimensional vector functions mentioned in this section in solving the equations of motion. Verification was accomplished by demonstrating the following:

(a) The program has the ability to correctly establish an initial vector coordinate system from external input data for use as a reference.

(b) The program logic is capable of properly executing the calculations necessary to transform vectorial parameters from one coordinate system to another.

(c) The presettings used for vector component calculations are valid.

Item (a) above was verified by the flight program initialization case described in Section 3. Items (b) and (c) were verified as described below.

2.3 TRANSFORMATION MATRICES

Vectors calculated by the flight program are correctly transformed from one coordinate system to another by operating on the known vector with the appropriate transformation matrix. Verification of the transformation matrices are accomplished as discussed in the following paragraphs.

2.3.1 [MSG] Matrix

The [MSG] matrix correctly transforms vectors from the S-system to the G-system. Since the [MSG] matrix and the [MG4] matrix are multiplied to form the [MS4] matrix, verification of the [MS4] matrix (see paragraph 2.3.6) indirectly verified the [MSG] matrix.

2.3.2 [MG4] Matrix

The [MG4] matrix correctly transforms vectors from the G-system to the 4-system. See the discussion above for the [MSG] matrix.

2.3.3 [MBS] Matrix

This matrix is used in the acceleration profile computations in the simulation flight mode only. Since this matrix is not used in the flight mode, no verification was performed.

2.3.4 [MBS$_a$] Matrix

The [MBS$_a$] matrix correctly transforms vectors from the B-system to the S-system using the rotation through average gimbal angles. Verification was accomplished by ensuring correct backup velocity calculations and checking individual matrix elements.

2.3.5 [M4V] Matrix

The [M4V] matrix correctly transforms vectors from the 4-system to the V-system. This matrix is used to transform position and velocity vectors for terminal guidance calculations; accurate guidance end conditions (see Appendix D) and checking the individual elements verified the [M4V] matrix.

2.3.6 [MS4] Matrix

The [MS4] matrix correctly transforms vectors from the S-system to the 4-system. Accurate guidance end conditions on each performance case (see Appendix D) verified the [MS4] matrix since this matrix is used in calculating position ($\overline{R}_4$) and velocity ($\overline{V}_4$) which are used in the active guidance routines. Individual elements were checked to verify correct implementation of the matrix.

2.3.7 [MGA] Matrix

The [MGA] matrix correctly transforms vectors from the G-system to the A-system. Correct telemetry station acquisition and loss calculations and checking individual elements verified correct implementation of this matrix.

2.3.8 [MSV] Matrix

The [MSV] matrix is obtained by multiplying the [MS4] by the [M4V] matrix. IGM parameters are calculated in the V-system as a function of navigation parameters in the S-system. Accurate guidance end conditions (see Appendix D) and checking individual elements verified correct implementation of this matrix.

2.3.9 [MEG] Matrix

This matrix is not actually implemented in the flight program and therefore, verification is not required.

2.3.10 [MES] Matrix

This matrix is not actually implemented in the flight program and therefore, verification is not required.

SECTION 3

PREPARE TO LAUNCH

N75 33117

3.1 INTRODUCTION

This section describes verification of the preflight prepare-to-launch/flight program interfacing, targeting load, and flight program initialization. Since there is no requirement in flight program verification to verify the error paths and logic implementation within the preflight routines, these were not verified.

3.2 TARGETING LOAD

The digital command system (DCS) is the primary method for loading targeting data into the LVDC. The target load command itself was not verified as it is used in the preflight mode only (see Section 10.4.12). The targeting load command was issued during all time periods of flight mode (boost and orbital) to verify that the command was not enabled by the flight program. A flight program patch was then used to enable the targeting load command during the orbital flight mode. A targeting load command was issued followed by a memory dump command to verify correct storage of the targeting parameters. Correct utilization of the targeting load data was verified.

3.3 PREFLIGHT PREPARE-TO-LAUNCH

Verification of all of the requirements for computations and system checks described in this paragraph was accomplished using the 6D/LVDC simulator. The 6D/LVDC simulator monitors all error indications from the prepare-to-launch routine and halts sequencing if errors exist.

3.3.1 GMT Synchronizing

Real-time accumulation in the prepare-to-launch mode was checked by comparing the value of accumulated time from the LVDC with the time in prepare-to-launch as computed by the 6D/LVDC simulator. The handling of this time as it is passed from the prelaunch routines to the flight program was checked by a trace through phase I initialization. Platform gimbal angle readings made in prelaunch are also passed over to the flight program and were verified in the same manner.

3.3.2 Azimuth Laying Support

Preflight azimuth computations were verified by comparing the telemetered azimuth with independent computations obtained using identical input quantities to both programs.

3.3.3 Variable Data Tape

A visual inspection of the flight program listing was made to verify that the specified locations are reserved for each parameter. The nominal simulations run were made using the data on the variable data tape for the July 15 launch date.

3.4 FLIGHT PROGRAM INITIALIZATION

A program trace through phase I initialization was used to verify that Time Base 0 (TB0) was started properly, the accelerometers were read, the descending node of the desired orbit was calculated, and the remaining flight quantities were initialized. The trace verified that upon completion of flight program initialization, the LVDC/LVDA Firing Commit Enable Discrete Output (DO12) was set and the boost mode calculations began. The MS4 and MSG matrices were verified as described in Section 2. Maneuver angles from the variable data tape (patched to include all four quadrants) were traced to insure correct handling by the sine/cosine routines during initialization.

N75 33118

SECTION 4

BOOST NAVIGATION AND GUIDANCE

4.1 INTRODUCTION

This section discusses the effort involved to assure that the boost mode of the flight program correctly provides navigation and guidance functions for the vehicle during periods when significant, measurable accelerations exist. Many test cases and performance cases were made, testing the ability of the navigation and guidance routines to properly compensate for several perturbations. Various plots of parameters of special interest were made on all performance cases, which facilitated a check on each calculation (each boost major loop) of these parameters.

4.2 ACCELEROMETER DATA DESCRIPTION

Each step in the processing of the platform accelerometer readings, from the read to the computation of vehicle acceleration, was checked separately. Some steps were checked directly, by test cases, others were checked indirectly, by comparison to corresponding results in the 6D simulator. Each step correctly performs its function under both nominal and perturbed conditions.

4.2.1 Accelerometer Data Format

The flight program reads the inertial platform accelerometers and separates the obtained bit configuration into the A and B optisyn readings. The differences between the current and past values for each reading, ΔA and ΔB, are correctly computed and represent the measured velocity change during the last computation cycle to the nearest 0.05 m/s. Verification of the correct program utilization of the A-optisyn readings was verified by hand computations on accelerometer test cases and by a favorable comparison (within accepted mission tolerances) between 6D and LVDC navigation parameters on all other cases. The B-optisyn readings were verified by simulating an A-optisyn failure and checking that the navigation parameters continue to compare within accepted mission tolerances.

4.2.2 Accelerometer Data Processing

The following paragraphs discuss the verification accomplished to assure the correct execution of the disagreement, zero and reasonableness tests on the velocity data changes.

4.2.2.1 Disagreement Test

The disagreement test correctly selects the measured change (ΔA or ΔB) closest to the expected change for zero and reasonableness testing. Logic tests which forced differences between the delta's of -3, -2, -1, +1, +2, and +3 in each axis, were run to verify that the proper channels were selected and that the appropriate mode code bits were set.

4.2.2.2 Zero Test

The zero test correctly determines the acceptance or rejection of those velocity changes of one pulse or less depending upon time, vehicle attitude, and expected thrust misalignment. Special logic tests were run which forced ΔA and ΔB to be +1, 0 and -1 at times during boost when a zero change was acceptable and at times when a zero change was unacceptable. Tests were also run which forced ΔA and ΔB to be +2 and -2 at times when a zero reading was unacceptable. Use of the zero readings and the backup velocity parameter for navigation purposes was verified by hand computations.

The test which determines if the vehicle is within a 2° band of the plumb-line coordinate system axis was verified by checking that the zero readings were accepted when the expected changes were less than A_{CO} (the estimated uncertainty in the expected change computation due to thrust misalignments) and that the zero readings were rejected when the expected changes were greater than A_{CO} and the zero test is enabled. Other tests were made involving outboard engine failures to verify proper changes to the parameter used in the zero test logic for expected thrust misalignment (D. VSND). Mode Code 24 bits were properly set for each case.

Other cases were run which monitored various parameter and "flag" changes throughout the mission to verify the bypassing of the zero test and the additional zero test logic during the particular times specified in the Events Sequence Timeline table.

There were no unexpected zero test failures in any of the cases simulated.

4.2.2.3 Reasonableness Test

The reasonableness test correctly determines if the velocity change selected by the disagreement and zero tests falls within a band of ±50% of the expected change enlarged by a reasonableness test constant (RTC) multiplied by ΔT. Verification of the reasonableness test was accomplished by forcing the selected change to be just within and just without the reasonable limits for both increasing and decreasing changes. Hand calculations were performed to verify the use of the backup velocity in place of the unreasonable changes. Proper bit settings in MC24 were also verified for each condition.

The RTC values are changed as specified in the Event Sequence Timeline table. This verification was accomplished by a series of cases which monitor various parameter changes throughout the mission.

There were no unexpected reasonableness test failures in any of the cases simulated.

4.2.3 Velocity Accumulation

The velocity accumulations in the flight program are correctly converted from pulses to platform velocity components $\dot{X}_m$, $\dot{Y}_m$, and $\dot{Z}_m$ or, (in the case of accelerometer failures), from the backup acceleration parameter to platform velocity. Verification was accomplished by comparing actual (6D) velocity components in the platform system to the flight program velocity accumulations or, (in the case of accelerometer failures, by hand computations.

4.2.4 F/M Calculations

The vehicle acceleration (F/M) is correctly calculated by dividing the root sum square of the difference between the present and past values of $\dot{X}_m$, $\dot{Y}_m$, and $\dot{Z}_m$ by the length of the previous computation cycle. Verification was accomplished by comparing the actual (6D) F/M with the flight program F/M on all cases.

During periods of flight in which the F/M computations may be unusually noisy, the preset constants for F/M are used as specified in the Event Sequence Timeline table and the Presetting table. These F/M constants were verified by a series of cases which monitor various parameter changes throughout the mission. The accuracy of these presettings was verified by comparing the actual value of acceleration at thrust buildup to the preset value.

4.2.5 M/F Smoothing Calculations

The initialization of the smoothed reciprocal value of the measured vehicle acceleration $(M/F)_s$, and all of the independent variables used in the calculation of $(M/F)_s$, was verified by the cases which monitor parameters such as those for changes. The initialization value, magnitude and rate of change limits, and the time to start $(M/F)_s$ calculations were verified to the specifications in the Presetting table and Events Sequence Timeline table of the EDD.

4.2.6 $(F/M)_c$ Acceleration

Backup acceleration $(F/M)_c$ is correctly computed from prestored force, mass, mass flow rate, and computed Performance Factor values. The precision of the backup acceleration profile was verified, via plots, by comparison to the actual acceleration profile for nominal flight. The value of $(F/M)_c$ during TB0 and the specified values of force, mass, and mass flow rate used to compute the backup acceleration were verified by the cases which monitor various parameter changes throughout the mission.

The backup velocity is correctly resolved through the measured gimbal angles. Each component is used correctly for the computation of the boost navigation parameters on all accelerometer failure cases. A failure was forced in each axis through all periods of boost with one case for each axis to check the navigational accuracy of using the backup acceleration.

The backup acceleration profile is correctly adjusted for S-IB engine failures. The backup force (F_b), backup mass flow rate $(\dot{M})$, and Performance Factor are adjusted by SIBEOB for the first detected S-IB engine failure. Verification was accomplished by a series of engine failure cases during different time segments.

4.3 BOOST NAVIGATION

During boost, the flight program correctly determines the vehicle position and velocity relative to the plumbline coordinate system (S-system). The following paragraphs discuss the effort to assure correct implementation of the trapezoidal integration scheme and the gravitational acceleration model used to compute these navigation parameters.

4.3.1 Integration

The trapezoidal integration scheme implemented in the flight program correctly determines the vehicle position and velocity from initial conditions, accelerometer data and earth gravitation. The basic tool used for verification of the boost navigation functions of the flight program is a comparison between the 6D and LVDC navigation parameters. The 6D uses essentially the same navigation equations that are specified for the flight program but has greater accuracy due to a faster integration rate, additional harmonic terms in the gravitation acceleration model, double precision floating point arithmetic, and the quantization level of the LVDC accelerometer readings (one pulse represents 0.05 m/s).

4.3.2 Gravitational Acceleration

The potential function of the earth that is used to model gravitational acceleration is correctly calculated during the boost periods of flight.

The same gravitational acceleration model is used for both the flight program and the 6D simulator. During the boost phases, the 6D simulator, however, uses the third and fourth terms in the potential expansion whereas the flight program uses only the first two. The calculations for gravitation acceleration were verified indirectly by a favorable comparison (within the accuracy allowed) of the navigation parameters.

4.4 BOOST GUIDANCE

The flight program properly guides the launch vehicle to the desired orbital conditions by generating the necessary steering commands. Verification of the guidance functions was accomplished as discussed in the following paragraphs.

4.4.1 First Stage Guidance

The roll maneuver to align the launch vehicle along the specified azimuth and the preprogrammed time tilt guidance, including adjustments for engine failures, are all executed properly. Verification of these functions is discussed in the following paragraphs.

4.4.1.1 Roll Maneuver

The roll guidance command is correctly initialized to the last roll platform gimbal angle computed in the Prepare-to-Launch routine (see Section 3). This initial command is held (frozen) until the vehicle has cleared the launch tower (the vehicle has increased its position along the vertical plumbline (X_S) axis GANTRY meters), or until the time from liftoff is greater than or equal to the specified time guard, at which time the roll guidance command is properly set to zero.

Verification of the correct command at tower clearance and at the time guard was accomplished. Proper setting of mode code bits was verified on all cases.

4.4.1.2 Time Tilt Pitch Guidance

The time tilt pitch computations begin under the identical requirements as the initiation of the roll maneuver. The initiation time was verified as discussed in the previous section.

The pitch guidance commands during the first stage burn, which are designed to yield a gravity term biased for expected winds, are computed correctly as functions of time from prestored polynomials. The 6D/LVDC simulator, which has the capability of simulating the expected winds for the specified launch month, was used to verify the pitch commands. From the initiation of time tilt pitch guidance until the tilt arrest time, the guidance commands in the minor loop were compared against a time tilt pitch profile. The two compared within the specified tolerance.

4.4.1.3 Time Tilt Yaw Guidance

The yaw guidance commands are correctly computed from tables as a function of time (T_c). For this mission, the yaw table is set to zero. The yaw table logic was verified by inputting known tables and independently solving equation 4.4.2.1 (Reference 1) with these known tables, ensuring that the yaw command profile was within a specified band of the tabular profile.

4.4.1.4 Engine Out Guidance Modifications.

The guidance commands are correctly adjusted for S-IB engine failures. Verification was accomplished by a series of engine failure cases during each segment of the engine failure freeze time function and during each segment of the tilt arrest time bias function. Proper engine out detection, adjustments to the backup acceleration $(F/M)_c$, mode code bit settings, and adjustments to D.VSND were also verified.

4.4.2 Iterative Guidance Mode (IGM)

Correct implementation of the active guidance scheme in the flight program was verified as discussed in the following paragraphs.

4.4.2.1 General Description of IGM

The IGM scheme correctly performs its two general functions: IGM guidance computation and IGM phasing. Both functions are dependent upon numerous variables, most of which change considerably with vehicle perturbations. Consequently, the guidance scheme was verified indirectly by looking at the overall program performance. The basic verification tool used to analyze overall program performance was a set of plots including all significant intermediate IGM parameters, all phasing parameters, and all guidance parameters for each performance case. The graphs were plotted by computer as a function of time from GRR and contains one value for each computation cycle.

The parameters listed in the PVP were plotted for IGM analysis for each performance case. Definitions of the parameters are covered in the EDD.

The plots of these parameters were analyzed by an examination of their corresponding mathematical equations (see Section 13 of the EDD). Each inflection point, "spike," or other unusual action of the parameter was explained by checking that it reacted in the manner dictated by the appropriate equation. The effects of IGM phasing, artificial tau modes, thrust level changes, and performance perturbations on each parameter were studied to ensure proper response.

4.4.2.2 Basic IGM Guidance Calculations

The IGM guidance commands required to steer the vehicle into the desired orbit are calculated correctly during each computation cycle. These guidance commands were checked by a very close critique of some of the aforementioned plots.

All preset guidance parameters (those which are assigned an initial value) are initialized correctly. These presettings were verified by observation on the nominal and/or performance cases or by the special cases which check various parameters for changes throughout the mission.

The final test of the accuracy of the guidance calculations is evidenced by the close agreement of the achieved (LVDC) terminal conditions on all performance cases to the desired conditions (see Appendix D).

4.4.2.3 Basic IGM Phasing

IGM phasing calculations and logic properly determine what parameters represent vehicle performance and correctly sequences IGM calculations. The initialization of the smoothed reciprocal value of the measured vehicle acceleration, $(M/F)_s$, and all of the independent variables used in the calculation of $(M/F)_s$ were verified by the cases which monitor these parameters. The initialization values and the times to start $(M/F)_s$ calculations properly conform to the specifications in the Presetting table and Events Sequence Timeline table of the EDD.

IGM phase initiation times, mode code bit settings, artificial tau phasing, χ-steering initiation time, and the high speed loop initiation time are all executed properly. Verification was accomplished by close observation of the applicable data printed each computation cycle on every 6D/LVDC performance case. The effects of all the above phasing on the total IGM guidance was checked on each performance case via plots.

4.4.2.4 Terminal Steering and Cutoff

The initiation time of $\tilde{\chi}$-steering and the zeroing of the position correction terms during χ-steering was verified on each performance case.

The S-IVB cutoff prediction correctly enables the vehicle to attain the velocity required for injection into the desired orbit. The calculations for and the issuance of the S-IVB velocity cutoff command in the high speed loop were verified on the performance cases by checking the exact time of issuance of the S-IVB velocity cutoff command in the high speed loop were verified on the performance cases by checking the exact time of issuance of the switch selector command. An independent program, which uses flight program position and velocity data from the high speed loop and the S-IVB cutoff switch selector command time, calculated the achieved cutoff velocity by extrapolation. This velocity compared favorably with the desired cutoff velocity on all performance cases.

An additional check on the S-IVB fine cutoff prediction scheme was accomplished by a comparison of the achieved terminal conditions with the desired terminal conditions. These conditions are derived by extrapolating the state vector (radius and velocity components) at cutoff to the time that terminal velocity is reached considering the velocity bias (ΔV_b) for thrust tailoff. The differences are shown in Appendix D and the close comparison indirectly verifies the adequacy of the fine cutoff scheme.

The start time of the high speed loop (HSL) was verified on every performance case considering the time bias ΔT_b. The velocity guard logic for the high speed loop was verified by program trace.

4.4.3 Guidance Reference Failure (GRF)

Guidance reference failures were forced in several cases to verify that DO4 and DO6 are set, bit 14 of Mode Code 27 is set, attitude error commands are frozen and DI9 (spacecraft control of Saturn) is checked. After DI9 is set, the program zeroes the attitude error command and sets bit 15 of Mode Code 27.

Special GRF cases were run forcing a GRF prior to and during the high speed loop (HSL) to demonstrate the following:

- In response to a GRF before HSL initiation, entrance to the HSL was inhibited and DI9 tests were enabled.
- In response to a GRF during the HSL, the GRF was recognized, and the HSL continued until cutoff.

4.4.4 Steering Misalignment Correction (SMC)

The steering misalignment correction (SMC) properly compensates for errors caused by the misalignment of the thrust vector with the vehicle longitudinal axis. The calculations for the SMC terms were verified by hand calculations made at various points in the flight (where SMC computations are active). Good terminal conditions on the nominal and the performance cases indirectly verified the SMC calculations.

The correct setting and resetting of bit 10 in MC25, the nominal sequencing of the SMC computations, and the inhibiting of SMC computations for program limiting and active hardware failures are performed correctly.

4.4.5 Chi Computations

Steering commands (χ_{y4} and χ_{z4}) computed in IGM are properly converted into the platform gimbal system and the SMC terms are correctly added to the guidance commands. Assurance that the yaw command is limited to a maximum magnitude of 45° was obtained by a special trace of program logic.

SECTION 5

N75 33119

ORBITAL NAVIGATION AND GUIDANCE

5.1 INTRODUCTION

This section describes the verification performed for the following orbital functions:

- Orbital flight mode initialization
- Orbital navigation
- Orbital guidance
- Telemetry acquisition

Verification of other orbital functions is discussed in the appropriately named sections.

5.2 ORBITAL PROCESSING RATES

The flight program's ability to accomplish the processing required for each orbital function at the specified rates was verified utilizing the output from the 6D/LVDC simulator. The minor loop and minor loop support cases described in Section 8 were used to verify correct implementation of the timing logic for these two modules. Once this was accomplished, only the time controlling constants used by these modules required further investigation. Simulations made across each area of constant replacement were used to verify correct processing rate control for the minor loop and minor loop support modules. The execution rate for orbital guidance computations was verified using 6D/LVDC simulator data. These cases were also used to check the processing rates for position extrapolation, orbital navigation, discrete processing, gimbal angle read, and telemetry acquisition and loss determination. Verification of these cases was preceded by a flight program coding check to ensure that the guidance and navigation parameters are telemetered when computed. Event sequencing and interrupt processing during orbital flight phases were indirectly verified by the minor loop cases, a check on correct switch selector command issuance, and the DCS interrupt tests described in Section 10. Adherence to the Event Sequence Timeline in Reference 1 was ensured for each orbital function.

5.3 ORBITAL INITIALIZATION

Proper parameter initialization for orbital navigation calculations was verified by analyzing a program listing and by utilizing the navigation error computation features incorporated in the 6D/LVDC simulator. The differences between the flight program computed values and the parallel and independently computed 6D program values were printed for each computation cycle. Plots of the error functions across the orbital initialization period were used for detection of possible errors in navigation component initialization and scaling.

5.4 ORBITAL NAVIGATION

Implementation of the orbital navigation scheme was verified using the concept that if the end result is good, then all of the individual considerations are correctly accomplished. However, a flight program trace of the applicable modules was used as an additional means of validating the computations employed in the orbital navigating scheme. The 6D/LVDC simulator compared the navigation parameter calculations of both computing systems as described in paragraph 5.3. Analysis of this data yields an accuracy figure for each flight phase. The 6D simulator was thus the reference against which the flight program implementation was checked since the accuracy of the 6D is well established. The following paragraphs discuss the applicable 6D simulator's program areas used for orbital navigation verification.

5.4.1 Integration

Integration of the equations of motion within the 6D program was accomplished at the end of each flight program minor loop resulting in navigation parameter update approximately every 100 milliseconds in the orbital flight phase. The digital integration is a forward trapezoidal scheme. Use of this integration rate provides an accurate reference for comparison to the flight program scheme of navigation parameter update every eight seconds utilizing a midpoint predictor in a modified Scarborough technique.

5.4.2 Acceleration Models

The mathematical models used to compensate for the effects of gravity and drag accelerations on the vehicle are identical in both the 6D and flight programs for orbital simulation periods. Since the 6D model implementations have been proven, only the input data needed further investigation.

5.4.2.1 Gravitational Acceleration Model

The equations for the gravitation acceleration model require vehicle position components as inputs. Since these values are computed by the 6D simulation, no further verification was required.

5.4.2.2 Drag Acceleration Model

The equations for the drag acceleration model use input parameters computed within the simulator. No verification effort was required in this area.

5.4.3 Navigation Update

At the specified navigation update time, T0+NUPTIM, of every accepted DCS command, the established processing sequence was correctly performed. Subsequently, the guidance passes and the navigation passes were scheduled at one and eight second intervals, respectively, after the specified navigation update execution time. Verification of the navigation update capability is also discussed in Section 10.

5.4.4 Delta V Monitoring

An accelerometer error input to the 6D/LVDC simulator was used to cause the desired acceleration to be read each time the accelerometers are read in Time Base 4 and Time Base 5. The trace capability of the 6D/LVDC simulator was used to verify that the accelerometer data was processed correctly.

5.4.4.1 Delta V Monitoring During Time Base 4

During Phase II Initialization, the measured velocity components are set to zero and the accelerometers are read. The accelerometers are read each second thereafter and the change in the measured velocity is accumulated in the storage locations designated for the measured velocity components. The velocity components are telemetered each time the accumulations are made. The velocity is accumulated from the "A" channel only for each of the three accelerometers. No disagreement, zero, nor reasonableness testing was performed on the accelerometer data and the data was not used for navigation during this time period.

5.4.4.2 Delta V Monitoring During Time Base 5

At $T^5+0.0$, the measured velocity components are reset to zero and the accelerometers are read. The accelerometers are not read again until $T^5+31.0$ seconds, but are read every second thereafter. Delta V Monitoring processing in Time Base 5 includes all the processing done in Time Base 4. If Time Base 5 has started, a test is made on DELVR. If DELVR is zero, no further action is taken on that pass; but if DELVR is non-zero, the total change in measured velocity is calculated by taking the square root of the sum of the squares of the velocity components. When the total change in the measured velocity becomes greater than or equal to DELVR, the safing sequence is initiated. DELVR is set to zero when the safing sequence is started, regardless of the condition under which the safing sequence is started.

Delta V Monitoring was verified in conjunction with the S-IVB/IU Deorbit DCS Command, Section 10.4.13.

5.5 ORBITAL GUIDANCE

Vehicle attitude control during orbital flight period was verified using the outputs of the 6D/LVDC simulator. Attitude computation rate verification is covered in paragraph 5.2. Each of the four basic maneuver types was thoroughly exercised individually and a series of maneuver combinations executed to test for possible undesirable interaction. Preprogrammed maneuvers were checked for correct initiation and termination in each nominal simulation. Nominal cases were designed to include all the programmed maneuvers defined for the mission. Perturbations to the nominal attitude timeline were designed to test all realistic maneuver combinations of preprogrammed, DCS, and spacecraft initiated vehicle attitudes. Mode Code indications and guidance commands were monitored in each simulation to verify correct implementation. The following paragraphs describe the verification requirements applicable to each maneuver type. In all cases, the computed vehicle attitude derived from the platform gimbal angles was checked to ensure that vehicle control was within the control system (APS) deadband. Time Base 5 was started during each possible maneuver type to verify that maneuver in progress is maintained when the new time base is initiated.

5.5.1 χ-Freeze Maneuver

Proper command holding of the last computed value during each χ-freeze maneuver was verified through analysis of the chi components printed for each pass of the 6D/LVDC simulator.

5.5.2 Inertial Attitude Hold Maneuver

Verification of the inertial hold maneuver consisted of ensuring that the specified angles in the platform coordinate system replaced the χ commands upon maneuver initiation and that the commands were held constant until the next maneuver. Gimbal angle rate of change and attitude response time were also analyzed.

5.5.3 Track Local Reference Maneuver

The guidance commands computed for each track local reference maneuver were verified using an independent computer program. This guidance check program uses flight program computed navigation and guidance parameters to calculate and print the vehicle attitude offset from a plane parallel with the local horizontal. Possible attitude computation errors are easily detected using this program.

5.5.4 Inertial Hold of Local Reference Maneuver

The track local reference guidance check program was used to verify that the flight program correctly computed the guidance commands at maneuver initiation. Ensuring that these guidance commands are held constant for the duration of the maneuver verifies that the maneuver is implemented properly.

5.5.5 Control Switchover Capability

Simulations were made across TB4 with DI9 on to ensure that the flight program logic checks for the spacecraft (S/C) control indication as a function of the "S/C Control of Saturn Enable" switch selector command. Interaction of the S/C control capability with other orbital maneuvers was tested through a series of 6D/LVDC simulations. S/C control was taken and returned across and during each maneuver in the nominal attitude timeline. Perturbed maneuver combinations are also tested for all realistic configurations. S/C control was also taken and returned after an Execute Generalized Maneuver DCS command was received but before implementation. All combinations of setting and resetting DI9 with the various maneuvers possible with the Execute Generalized Maneuver DCS command were verified. Flight program response was correct for each combination of maneuver and DI9 condition.

Correct attitude calculations upon vehicle return of control to the IU were verified in each case using the guidance parameters telemetered in conjunction with independent calculations. Mode Code 27 indicator bit implementation was checked.

5.5.6 Guidance Reference Failure (GRF)

Attitude command modifications as a function of GRF detection and S/C control (DI9) was verified in a test series designed to check program performance in each time base and flight mode. Assurance that the ladder commands are frozen upon detection of GRF and zeroed when DI9 is detected was obtained by a program trace of the logic. Attempts to change the ladder commands was accomplished by issuance of DCS navigation updates with suitable parameters.

5.5.7 DCS Commanded Functions

Verification of the DCS commands capable of altering the nominal attitude timeline is described in Section 10.

5.5.8 Variable Data Tape Maneuvers

The start times for maneuvers 4, 5, 6, and 7 and the attitudes with respect to local horizontal and the desired roll angle for maneuvers 4 and 6 are obtained properly from the locations defined for these parameters on the variable data tape. The snap and trace capability of the 6D/LVDC simulator was used to verify that the data was obtained from the variable data tape locations and implemented correctly in the orbital guidance calculations.

5.6 TELEMETRY ACQUISITION AND LOSS

The flight program correctly determines the launch vehicle position with respect to the earth oriented telemetry receiving stations. Simulations of four revolutions were made at the nominal launch azimuth.

5.6.1 Acquisition and Loss Calculations

The acquisition and loss calculations correctly determine whether or not the vehicle is in range of a telemetry station. These calculations were verified by comparing the telemetered acquisition and loss times against the results of an independent program.

5.6.2 Acquisition and Loss Sequence

The alternate switch selector sequences required for the mission were shown to be correctly commanded through analysis of telemetry data obtained from the simulations.

SECTION 6

There is no requirement for Section 6. This section is included to be compatible with the Saturn IB EDD.

N75 33120

SECTION 7

TIME BASES, DISCRETES, AND INTERRUPTS

7.1 INTRODUCTION

This section discusses the effort performed to assure proper flight program handling of time bases, discretes, and interrupts.

7.2 TIME BASES

Time bases are used to conveniently reference flight program events to some key mission event. Proper initiation of all time bases by recognition of the primary signal as well as the backup signal was verified as outlined in the following paragraphs. As an assurance that time bases are initiated only within the correct time frames, each primary and backup signal was forced prior to the time base enable time and forced again just after the time base initiation time. Each time base initiation time was verified by checking that TI was updated to within 2 ms of the time that the flight program recognized the required signal(s) for starting the time base.

7.2.1 Time Base 0 (Guidance Reference Release)

TB0 is correctly initiated by the GRR interrupt (INT7) from the launch sequencer. Verification of the proper initiation of TB0 was accomplished by sequencing the flight program through the preflight routines and checking that, in response to INT7, the time in the time base (TB) and the time of time base initialization from GRR (TI) were initialized correctly, that the correct mode code bit was set and that phase I initialization was performed as specified in the Event Sequence Timeline table and the Presetting table.

7.2.2 Time Base 1 (Liftoff)

TB1 is used for sequencing during S-IB stage until detection of the S-IB low level sensors dry interrupt.

TB1 is initiated properly by receipt of either liftoff discrete (DI7 or DI24). Failure to liftoff (by inhibiting DI7 and DI24) was sequenced to verify correct program action at T0+150.

Correct monitoring of DI7 and DI24 was verified by setting each independently and together prior to the enable time and utilizing a program trace to demonstrate that each is checked properly from the enable time until TB1. Proper initiation of TB1 was verified by demonstrating that TB and TI are correctly updated, that the proper mode code bit is set and that TB1 events are executed as specified in the Event Sequence Timeline table.

7.2.3 Time Base 2 (S-IB Low Level Sensors Dry)

TB2 is used for sequencing from S-IB low level sensors dry until S-IB outboard engine cutoff. The signals for initiating this time base are the S-IB low level sensors dry "A" interrupt (INT2) and the S-IB low level sensors dry "B" interrupt (INT6). Correct initiation of TB2 was verified by checking the program's response to INT2 and INT6. Each interrupt was inhibited to verify correct initiation of the time base by receipt of the other signal. Both INT2 and INT6 were set prior to TB2 enable to verify that TB2 is enabled properly.

Verification of the downrange velocity constraint was accomplished by forcing the downrange velocity to be less than 500 m/s and checking that TB2 was not started and the program set the ladder outputs to zero and entered a one-instruction loop. Correct bypassing of the velocity constraint was verified by starting TB1 without actually lifting off, forcing GRF, and demonstrating that TB2 was properly initiated.

7.2.4 Time Base 3 (S-IB Outboard Engines Cutoff)

TB3 is used for sequencing during S-IVB stage. The primary signal for initiating the time base is the S-IB outboard engines cutoff "A" interrupt (INT5) and the backup signal is the S-IB outboard engines cutoff "B" discrete (DI23).

Correct initiation of TB3 by the primary signal was verified by checking the program's response to INT5. INT5 was inhibited to verify correct initiation of the time base by receipt of DI23. Both INT5 and DI23 were set prior to TB3 enable to verify that TB3 was enabled properly. Also, both INT5 and DI23 were inhibited. TB3 was properly initiated on the time backup at the correct time by the flight program.

7.2.5 Time Base 4 (S-IVB Cutoff)

Time Base 4 (TB4) is used for sequencing after S-IVB cutoff. There are four indications which are checked to start TB4; the detection of any two of which will initiate the time base. These four indications are listed below:

- S-IVB Engine Out "A" (DI5)
- S-IVB Engine Out "B" (INT4)
- S-IVB Cutoff switch selector issued by the LVDC
- Velocity change of less than 1 m/s measured by the inertial platform over last boost major loop

Verification of the proper initiation of TB4 was accomplished by forcing inhibits on all combinations of these four conditions and checking that, in response to at least two of the conditions being present prior to TB4 enable, the proper mode code bit was set, TB and TI were correctly updated and the TB4 events were executed as specified in the Event Sequence Timeline table. Single indications were forced to verify that initiation of TB4 requires two S-IVB cutoff indications.

7.2.6 Time Base 5 (S-IVB/IU De-orbit DCS Command)

Time Base 5 is used for sequencing of LOX and LH_2 dump, and a vehicle safing sequence. Time Base 5 must start at $T4+60T_{DSS}$, where T_{DSS} is specified by a DCS command. To prevent premature starting of TB5, it must be inhibited until $T4+T5_{GRD}$.

Proper initiation of TB5 was checked by issuing T_{DSS} to start TB5 prior to and after TB5 is enabled. In the former case, the DCS command was rejected by the flight program, and a DCS error code issued. In the latter case, proper response was verified by observing that bit 7 of Mode Code 27 was reset after being set by the DCS command. DCS time updates were also issued to ensure that they did not affect the Time Base 5 start time.

7.3 DISCRETE OUTPUTS

Discrete outputs are directly dependent upon the flight program and are used to affect external equipment. Of the thirteen bit positions in the discrete output register, only five are used in flight. The proper setting of these discrete outputs was verified as discussed below.

7.3.1 DO1: Reset Command Decoder

This discrete output is an indication to the command decoder that a DCS word has been read and found to be valid by the flight program. Issuance of this DO provides a computer reset pulse (CRP) that resets the command decoder. The correct setting of DO1 was verified by demonstrating that, in response to a valid DCS word, the DO is set to a logic 1. Invalid DCS commands were issued to verify DO1 remains a logic 0 in response to rejected commands.

7.3.2 DO4 and DO6: Guidance Reference Failure

DO4 and DO6 are redundant indications of guidance reference failure (GRF). Verification of the proper setting of these discrete outputs was accomplished by demonstrating that DO4 and DO6 are set to a logic 1 in response to a GRF.

7.3.3 DO12: LVDA/LVDC Firing Commit Enable

The flight program sets this discrete output to signify a ready-to-launch condition after receipt of GRR. The correct setting of DO12 was verified by demonstrating that the DO is set upon completion of normal flight program initialization and is reset at T1+0.

7.3.4 DO13: LVDA/LVDC Firing Commit Inhibit

This discrete output is interlocked with the launch sequencer such that its receipt forces a countdown recycle before launch is possible. DO13 is set by hardware, not by the flight program.

7.4 DISCRETE INPUTS

Discrete inputs are hardware dependent signals originating outside the flight program which control flight sequencing functions by affecting program flow. The flight program's response to each signal was verified as outlined in the following paragraphs.

As an assurance that the discretes are honored only in the proper time frames, each discrete was forced in the following intervals:

- Before the discrete is enabled
- After the discrete has been detected
- After the discrete has been disabled

The correct sampling rate of the discrete input register (DIR) was verified utilizing the selective print capability of the 6D/LVDC simulator.

7.4.1 DI1: RCA-110A Sync

DI1 is used only in the ground routines and there is no requirement for the flight program to monitor this discrete. Verification that DI1 is never monitored was accomplished by activating DI1 during several phases of the mission and checking that no program reaction results.

7.4.2 DI2: Command Decoder OM/D "A" and Command Decoder OM/D "B"

This discrete correctly indicates to the LVDC whether a DCS command is a mode or data command. The correct program response to this discrete was verified by issuing a DCS command and demonstrating that the program sets up for a mode command in response to DI2 being a logic 1 and sets up for a data command in response to DI2 being a logic 0.

7.4.3 DI3: Spare

DI3 is a spare and was not monitored by the flight program. Verification was accomplished by forcing the discrete during several parts of the mission and checking that no program reaction occurred.

7.4.4 DI4: Spare

Verification was the same as for DI3.

7.4.5 DI5: S-IVB Engine Out "A"

DI5 indicates that the S-IVB engine is out and is one of the conditions which initiates TB4. The correct program response to this discrete was verified by forcing the discrete and utilizing a program trace to demonstrate that the presence of DI5 is noted as one of the conditions for starting TB4. DI5 and each other condition for starting the time base were forced to verify all logic paths.

7.4.6 DI6: Spare

Verification was the same as for DI3

7.4.7 DI7: Liftoff "B"

This discrete indicates that liftoff has occurred. The proper program response to DI7 was verified by demonstrating that TB1 is initiated by recognition of this discrete. A program trace was used to verify that DI7 is detected within the specified time.

7.4.8 DI8: Spare

Verification was the same as for DI3.

7.4.9 DI9: S/C Control of Saturn

This discrete is a signal from the spacecraft to indicate to the LVDC that the spacecraft has taken control of the flight control computer (FCC) and that the LVDC outputs to the FCC are not being accepted. Correct program response to the presence of this discrete was verified by forcing DI9 and demonstrating that the proper mode code bits are set and that the ladder outputs are zeroed by maintaining the χ's and minor loop χ's equal to the gimbal angles and by setting the minor loop χ rates to zero. The correct interrogation of this discrete by the flight program was verified by using a program block trace to demonstrate that DI9 is monitored once per BML from T4+5.0 seconds to orbit initialization and once per second from then until T5+0. Verification of the detection of DI9 in combination with a GRF is discussed in paragraph 5.5.6.

Verification of the correct program response to the deactivation of DI9 is discussed in paragraph 5.5.5.

7.4.10 DI10: Coolant Thermal Switch #1

DI10 and DI11 indicate that the temperature of the environmental control system (ECS) coolant is above the selected control temperature. The verification of the correct program response to these discretes was accomplished by setting and resetting combinations of DI10 and DI11 at various times during the flight and checking that the proper switch selectors are issued.

The correct sampling rate of DI10 and DI11 was verified by demonstrating that these discretes are checked every 300 seconds beginning at T0+TM seconds.

Correct inhibiting of the water control valve logic was verified by demonstrating that DI10 and DI11 are permanently ignored by the flight program following the Water Control Valve Logic Inhibit DCS command and that bit 18 of MC27 is reset.

7.4.11 DI11: Coolant Thermal Switch #2

See DI10.

7.4.12 DI12: S-IB/S-IVB Separation

This discrete indicates that the S-IB and S-IVB stages have separated. There is no requirement for the flight program to monitor this discrete. Verification that DI12 is never monitored was accomplished by checking that no program reaction results when DI12 is reset.

7.4.13 DI13: Spare

Verification was the same as for DI3.

7.4.14 DI14: S-IB Outboard Engine Out

This discrete indicates that at least one S-IB outboard engine is out. Correct adjustments based on the detection of DI14 in TB1 were verified by forcing an S-IB outboard engine failure and checking that the time tilt calculations were properly modified, that the backup acceleration, $(F/M)_c$, was correctly adjusted, that the correct mode code bits were set and that SIN(6°) was substituted for SIN(2°) in the zero test computation. Correct adjustments based on the detection of DI14 in TB2 were verified by forcing an S-IB outboard engine failure and checking that SIN(6°) was substituted for SIN(2°) in the zero test computation, and that the proper mode code bits were set. Verification that erroneous indications of an S-IB outboard engine failure are handled properly was accomplished by demonstrating that, in response to forcing DI14 without an engine failure, the discrete is detected the same as for a valid S-IB outboard engine failure. A program block trace was utilized to verify that this discrete is checked once per BML, until detection, during the time interval from $T1+T_{S1E0}$ seconds until TB3.

7.4.15 DI15: S-IB Inboard Engine Out "B"

DI15 indicates that at least one of the S-IB inboard engines is out. Correct program adjustments based on the detection of this discrete in TB1 were verified by forcing an S-IB inboard engine failure and checking that the time tilt calculations were properly modified, that the backup acceleration, $(F/M)_c$, was correctly adjusted, that the correct mode code bits were set and that SIN(6°) was substituted for SIN(2°) in the zero test computation. Correct adjustments based on the detection of DI15 in TB2 were verified by forcing an S-IB inboard engine failure and checking that SIN(6°) was not substituted for SIN(2°) in the zero test computation, and that the proper mode code bits were set.

Verification that erroneous indications of an S-IB inboard engine failure are handled properly was accomplished by demonstrating that in response to forcing DI15 without an engine failure, the discrete was detected the same as for a valid S-IB inboard engine failure. A program block trace was utilized to verify that this discrete is checked once per BML, until detection, during the time interval from $T1+T_{S1E0}$ seconds until TB3.

7.4.16 DI16: Prepare for Guidance Reference Release

DI16 is used only by the ground routines and there is no requirement for the flight program to monitor this discrete. Verification that DI16 is never monitored was accomplished by checking that no program reaction results from DI16 being activated.

7.4.17 DI17: Spare

Verification was the same as for DI3.

7.4.18 DI18: Spare

Verification was the same as for DI3.

7.4.19 DI19: Spare

Verification was the same as for DI3.

7.4.20 DI20: Spacecraft Initiation of S-IVB Engine Cutoff

There is currently no requirement to check DI20. Verification of program response to DI20 was the same as for DI3.

7.4.21 DI21: Spare

Verification was the same as for DI3.

7.4.22 DI22: Spare

Verification was the same as for DI3.

7.4.23 DI23: S-IB Outboard Engines Cutoff "B"

DI23 is the backup signal for starting TB3. The correct program response to this discrete was verified by forcing the discrete and using a program trace to demonstrate that DI23 will start TB3 properly.

7.4.24 DI24: Liftoff "A"

This discrete indicates that liftoff has occurred. The proper program response to DI24 was verified by demonstrating that TB1 is initiated by recognition of this discrete. A program trace was used to verify that DI24 is detected within the specified time.

7.4.25 DIS1-DIS8: Spares

Verification was the same as for DI3.

7.5 INTERRUPTS

The LVDC has a feature which permits interruption of the normal program to free the computer for priority tasks. When such a priority task arises, an interrupt is generated. This transfers control to a special routine upon the completion of the instruction then being executed.

Of the twelve interrupts, nine are external and three are provided for functions internal to the LVDC. The correct program response to each interrupt was verified as discussed in the following paragraphs.

As an assurance that an interrupt is honored only in the proper time frames, each interrupt was forced during the following intervals:

- Prior to the specified enable time
- After the interrupt has been honored
- After the interrupt has been disabled

7.5.1 INT1: Command LVDA/RCA-110A Interrupt

This signal is used in the preflight routines and there is no requirement to process it by the flight program. Verification that this interrupt is not processed

[redacted] [redacted]

was accomplished by activating the interrupt during several parts of the mission and checking that no program reaction results.

7.5.2 INT2: S-IB Low Level Sensors Dry "A"

This interrupt indicates that the propellant level in either the S-IB fuel tanks or LOX tanks has dropped below a given level. The initiation of TB2 in response to the activation of INT2 verified correct flight program response to this interrupt.

7.5.3 INT3: RCA-110A Interrupt

See discussion for INT1.

7.5.4 INT4: S-IVB Engine Out "B"

INT4 indicates that the S-IVB engine is out and is one of the conditions which initiates TB4. The correct program response to this interrupt was verified by forcing the interrupt and utilizing a program trace to demonstrate that the presence of INT4 is noted as one of the conditions for starting TB4. INT4 and each other condition for starting TB4 were forced to verify all logic paths.

7.5.5 INT5: S-IB Outboard Engines Cutoff "A"

This interrupt indicates that the propellant in the S-IB fuel tanks has depleted and is the primary signal for initiating TB3. The correct program response to INT5 was verified by demonstrating that TB3 is initiated by detection of this interrupt.

7.5.6 INT6: S-IB Low Level Sensors Dry "B"

Verification was the same as for INT2.

7.5.7 INT7: Guidance Reference Release (GRR)

This interrupt is initiated by the launch sequencer and indicates that the stabilized platform has been released. Correct processing of this interrupt was verified during the preflight prepare to launch (PTL) mode by demonstrating that TB0 is started by detection of this interrupt.

7.5.8 INT8A and INT8B: Command Decoder Interrupts "A" and "B"

This interrupt indicates to the LVDC that a DCS command has been received by the command decoder. The correct response with respect to INT8 was verified by demonstrating that the program sets up to process a DCS command when this interrupt is activated.

7.5.9 INT9: TLC-Simultaneous Memory Error

This interrupt indicates either a memory parity error or a memory drive current check failure. Correct program response to INT9 was verified by forcing the interrupt at various mission times to test for execution of the specified recovery modes. In each case, correct telemetry coding was verified, proper DOR & ICR reset checked and EMR accumulation ensured through analysis of 6D/LVDC program execution traces of the TLC module.

7.5.10 INT10: Spare

INT10 is a spare and there is no requirement for the flight program to process this interrupt. Verification that the program does not react to this interrupt was accomplished by forcing INT10 and checking that no program response occurs.

7.5.11 INT11 and INT12: Internal Function of the LVDC

These two interrupts are provided to the LVDC for functions internal to the computer. INT12 is referred to as the Timer 1 interrupt and is used to control the execution of priority modules which require operation at an exact time or at a precisely cyclic rate. INT11 is referred to as the Timer 2 interrupt and is used to control the execution of priority modules of a lower order than those under Timer 1 but which also require operation at a specific time or rate as precisely as possible.

The correct program response to INT11 and INT12 was verified indirectly by verification of the functions which they control; namely, for INT12, minor loop, switch selector processing, and liftoff search; and, for INT11, Phase I and II time update, events processor, time tilt guidance, and Phase II control.

N75 33121

SECTION 8

LAUNCH VEHICLE ATTITUDE CONTROL

8.1 INTRODUCTION

The flight program maintains correct vehicle attitude control via the minor loop and minor loop support modules by generating vehicle attitude error signals. This section describes the verification of the minor loop and minor loop support modules of the flight program. Logic was checked during boost and orbit (unusual timing situations necessitate the additional orbit checks) and the constants used by these modules checked at every point during the flight. Since the same logic instructions are used throughout the flight with only the constants changing, verification of the logic at two points and the constants at all points verified the logic throughout the flight.

8.2 MINOR LOOP

The minor loop properly samples platform gimbal angle data, evaluates this data, and computes and issues attitude error commands. Verification was accomplished by exercising the various logic paths and checking the limits, constants, and execution timing.

8.2.1 Gimbal Angle Data

The fine gimbal angle resolver is initially selected in each axis. Backup resolvers are properly selected in each axis when fine gimbal failures are forced as described in the following paragraphs. The gimbal reading bit pattern was correct in all cases.

Verification of resolver reading initialization for both fine and backup configuration is discussed in paragraph 3.2.2.

8.2.2 Gimbal Data Processing

The gimbal resolver readings correctly undergo several validity tests before they are used to compute vehicle attitude and attitude error commands. The following paragraphs describe the verification of the logic used to detect erroneous gimbal angle readings.

8.2.2.1 Disagreement Bit Processing

The counters are correctly determined to be in disagreement whenever the A and B readings disagree by more than +3 or less than -4 bits. Verification runs were made with the following combinations of differences between the A and B counter readings and the state of the disagreement bit (DGB):

(a) Positive and negative differences just within tolerance and the DGB on,

(b) Zero difference with the DGB on,

(c) Positive and negative differences just out of tolerance with the DGB off, and

(d) Positive and negative differences just out of tolerance with the DGB on.

The flight program properly determines the valid and invalid disagreement bit and keeps account of the disagreement bit circuitry failures.

After a valid disagreement had been detected, the counters were perturbed by simulating in and out of tolerance counter readings when the flight program addressed the counters for incrementation at a known frequency. The following combinations were used to verify the program's capability of detecting and compensating for counter malfunctions.

(a) A and B counters within tolerance.

(b) A and B counters out of tolerance.

(c) A counter within and B counter out of tolerance.

(d) B counter within and A counter out of tolerance.

The proper counter reading is selected for reasonableness testing in each case and the program keeps account of the number of counter failures. Two failures of either counter in the specified time results in the disagreement bit processing being permanently bypassed and the other counter selected. Failure of both counters twice in the specified time results in the guidance reference failure discretes being set.

When a valid disagreement is detected and both counters are within tolerance, the flight progra n selects the A counter reading for reasonableness testing. If the A counter reading is found reasonable, a B multiplexer failure is assumed and the B multiplexer failure counter is incremented. If the A counter reading is found unreasonable, an A multiplexer failure is assumed and the A multiplexer failure counter is incremented. If the A multiplexer is in error twice or the B multiplexer is in error five times in the specified time, disagreement bit processing is permanently bypassed and the counter corresponding to the other multiplexer is permanently selected for reasonableness testing.

All program logic was implemented properly and all mode code bits were set correctly.

8.2.2.2 Reasonableness Tests

Both reasonableness tests, unreasonable zero and unreasonable change are performed properly on the selected gimbal angle readings. Verification was accomplished as discussed in the following paragraphs.

Readings were simulated that forced the attitude error signals to values less than, equal to, and greater than (in both positive and negative directions) the zero test constant in each axis. The program properly selects the reasonable readings to update the vehicle attitude angle and rejects the unreasonable readings.

Gimbal angle readings were forced on both sides of the resolver reasonableness test constant. The resolver overflow test was also exercised in the same manner. The program properly determines the reasonable and unreasonable readings.

Reasonableness test verification was accomplished for both fine and backup resolvers in each axis. The counters that keep track of the error rates are incremented and reset properly.

8.2.2.3 Minor Loop Error Telemetry

The minor loop error word is telemetered in the correct format and at the appropriate time in all cases. Verification was accomplished by obtaining a printout of the minor loop telemetry word in each of the minor loop verification cases.

8.2.3 Attitude Error Commands

The equations used to compute the minor loop attitude commands are implemented properly. Verification was accomplished by using independent calculations for each axis. Extreme desired vehicle attitudes were forced, but the attitude command magnitude and rate were limited properly.

The error monitor register indication of circuitry failure was forced to verify the selection of the proper ladder channel. The channels were selected properly and the counter that keeps track of circuitry failures was incremented and reset properly.

8.3 MINOR LOOP SUPPORT

The minor loop support functions properly compute attitude change increments and the coefficients for the gimbal-to-body transformation to be used in the minor loop. Verification of the minor loop support functions is described below.

8.3.1 Attitude Increments

The desired attitude commands are computed properly each minor loop by adding the fixed increments to the previous attitude command values. Computation of the attitude increments for use by the minor loop module was checked for each platform axis with independent equation solutions. Proper attitude change magnitude limiting was verified through analysis of the minor loop and minor loop support telemetry when large desired attitude commands were given. Correct bypassing of the steering misalignment correction (SMC) terms during iterative guidance mode (IGM) of flight was verified by performance cases containing perturbations which caused the attitude command increments to exceed the magnitude limits.

8.3.2 Gimbal-to-Body Transformation

The calculations required to obtain the predicted roll and yaw average attitude angles for compensation of the changing relationship between the vehicle body and the inertial platform were executed properly. Verification was accomplished through independent solutions of the equations.

The special logic required for average roll attitude angle determination was executed properly. Verification was accomplished by forcing the difference between predicted and actual angles to values that lie on either side of the crossover magnitude (180°). Coefficients needed by the minor loop module for transformation of the attitude errors from the gimbal coordinate system to the vehicle coordinate system were calculated independently and compared to those values computed by the flight program.

8.3.3 Loss of APS Attitude Control Test

The X and Z attitude control tests were enabled properly and correctly detected attitude errors which exceeded the specified test constants. The ladder magnitude limit was set to twelve degrees when an X or Z attitude control failure was determined provided the nominal timeline did not specify a larger magnitude limit for an axis. Correct priority between the ladder magnitude limit following an attitude control test failure, DCS Ladder Magnitude Limit command, and nominal timeline changes was verified. The APS attitude control test is correctly disabled after detection of an attitude control failure or after issuance of the DCS Ladder Magnitude Limit command.

N75 33122

SECTION 9

SWITCH SELECTOR PROCESSING

9.1 INTRODUCTION

Switch selector commands are correctly issued under program control to provide sequencing signals to the launch vehicle hardware. This section discusses the verification accomplished to assure proper program execution of the switch selector commands.

9.2 COMMAND EXECUTION OPERATIONS

All the separate program operations required to properly issue one switch selector command are executed properly. Verification of the correct execution of each operation was accomplished as outlined below.

9.2.1 Sequence of Operation

The sequence of operation refers to the order of execution and timing requirements of each step necessary to activate a switch selector command. Correct execution of each step and maintenance of the minimum timing requirements between steps were verified through analysis of a series of 6D/LVDC intermediate timing cases. Nominal and perturbed switch selector feedback conditions were simulated to verify correct processing of all realistic sequencing and timing combinations.

9.2.1.1 Hung Stage Test

The hung stage test correctly checks the address feedback to assure all inputs are zero. The proper execution of this test was verified by forcing the address feedback to be non-zero and checking the program's response. All switch selectors from GRR to T4+100 were tested to verify that the hung stage test was made prior to issuing commands which require a forced reset for hung stage conditions. Sequencing from GRR to T4+100 includes all of the types of switch selectors which require the hung stage test.

9.2.1.2 Stage and Address

After the hung stage test, the flight program correctly issues the stage and address. This operation was verified by comparing the issued stage and address against the expected stage and address and also by checking that the correct telemetry was issued.

9.2.1.3 Address Verification

After issuance of the stage and address, the switch selector feedback is properly checked. This operation was verified by forcing feedback errors and checking that the program executed the proper corrective action including mode code bit setting, internal control register bit resetting, and telemetry when required. One bit feedback error, all bit feedback error, and all zero feedback were forced to verify all possible logic paths involved with this operation.

9.2.1.4 Read Command

The read command correctly activates the logic on the switch selector to produce the commanded output. This operation was verified by comparing the stage, address and associated real-time clock reading telemetry after read command issuance with the output from the simulated 6D switch selector register.

9.2.1.5 Reset Read

The reset of the read command is correctly issued no less than 25 ms after the read command is issued. This operation was verified by the series of 6D/LVDC switch selector cases by checking that the read command for each switch selector is not reset until the specified interval (25 ms) had elapsed.

9.2.2 Termination of a Command Sequence

A command which is in progress can be properly terminated by issuance of a forced reset. Verification was accomplished by forcing the conditions which require a forced reset (by a TLC, a hung stage failure or by feedback errors) and observing that the proper termination was completed.

9.3 TIMING

The timing requirements for all switch selector commands are properly satisfied. Switch selector timing was verified by comparing the times of all switch selector read commands, for the nominal sequence and all alternate sequences, with the issuance times specified by the Flight Sequencing table. All switch selector commands were issued within the specified tolerance.

9.4 PRIORITIES

When requirements for simultaneous issuance of switch selector commands occur, the commands are issued correctly with the following priority:

A. Class I alternate sequence

  1. S-IVB cutoff switch selector

B. Class II alternate sequence

None defined for this mission

C. Nominal flight sequence

D. Class III alternate sequence

1. Generalized switch selector
2. Coolant valve

E. Class IV alternate sequence

1. Telemetry station acquisition sequence

2. LOX depletion dump start sequence

3. LOX depletion dump stop and LH2 depletion dump start sequence

4. Start safing sequence

The hierarchy of these switch selector commands was verified by forcing the requirements for simultaneous switch selectors, where the possibility exists for interference, and then checking that the correct priority was followed.

There were two simulations made for each interaction. The first simulation was designed such that the sequence of operations for the lower priority switch selector would be in progress when the requirements for issuing the higher priority switch selector were introduced. In each of these tests, the lower priority sequence was correctly interrupted and replaced by that of the higher priority sequence. Depending upon specifications, the sequence of operations for the lower priority switch selector would be re-entered and the command correctly issued or the sequence of operations would be correctly terminated. The second simulation was the sequence for the higher priority switch selector in progress when the conditions for issuing the lower priority switch selector were introduced. In these tests, the higher priority sequence would not be interrupted and the command would be properly issued. The lower priority sequence was then, depending upon specifications, either re-initiated and the command correctly issued or correctly terminated.

N75 33123

SECTION 10

DIGITAL COMMAND SYSTEM

10.1 INTRODUCTION

The Digital Command System (DCS) correctly provides a limited real-time means of controlling specific flight program functions. Correct processing of DCS commands by the flight program was verified as outlined in the following paragraphs.

10.2 DCS WORD FORMAT

The flight program has the capability to correctly read and process the information stored in the command decoder register upon receipt of the ANDed interrupt bits. Also, correct interpretation of the ANDed orbital mode/data bits is performed properly. Verification was accomplished each time a DCS command was issued by checking that the command was recognized correctly by the flight program.

10.3 DCS COMMAND VERIFICATION

Upon detection of the command decoder interrupt (INT8), the flight program correctly reads the contents of the command decoder register and makes several tests on the DCS command before it is accepted for use. Correct flight program implementation of the checks required to establish the validity of the data received from the command decoder was tested using the methods described below. A generalized switch selector mode command was issued after T4+0 to ensure that the command decoder discrete output (DO1) is set and the DCS error counter is zeroed.

10.3.1 DCS Mode Command Verification

The flight program has the capability to correctly detect DCS mode command format errors. Verification was accomplished by performing the following DCS command verification tests. Running the perturbation in the order listed verified that the flight program tests were performed in the correct sequence.

1. True complement test: A sequence of mode commands requiring no data words was issued with combinations of correctly and incorrectly coded complement bits. Proper rejection, acceptance and error code 10 telemetry was executed.

2. Sequence bit test: A sequence of mode commands not requiring data words was issued with valid and invalid sequence bits. Correct rejection of invalid mode formats and error code 24 telemetry was performed.

3. Terminate command test: Terminate commands were issued following 1) a memory dump request, 2) a compressed data dump, 3) issuance of a mode command requiring data, and 4) during each DCS time frame. Proper bypassing of the specified tests was achieved.

4. Mode expected test: A DCS sequence containing modes which do and do not require data words was sent to the flight program. Correct rejection of invalid configurations and correct error code 20 telemetry was accomplished.

5. Memory dump or compressed data dump in progress test: A memory dump command and a compressed data dump command were issued while another memory dump and compressed data dump was in progress. Error code 64 telemetry was correctly issued.

6. Mission acceptance test: All possible DCS mode commands $(00 \text{ to } 77)_8$ were issued in each DCS time frame to ensure that modes not defined for the mission were correctly rejected and error code 14 was telemetered.

7. Time acceptance tests: All mission defined mode commands and associated data words were sent to the flight program during each DCS time frame. Correct acceptance and rejection as a function of the time frame was performed properly and error code 74 was correctly issued.

8. Pending generalized switch selector test: A generalized switch selector command was issued while a previously issued generalized switch selector command was waiting to be processed. The first command was issued at a time to cause the generalized switch selector to be delayed by a nominal tabled switch selector. The second command was rejected and error code 34 issued.

The simulator monitored the discrete output register to verify correct setting by the DCS routines. If DO1 was not set within 400 milliseconds after the DCS interrupt (INT^9), rejection of the command was assumed. Correct telemetry of DCS mode status indicators and error code words was performed properly on all simulation runs.

10.3.2 DCS Data Command Verification

When a DCS data command is received, the flight program correctly performs several tests before the command is accepted. Verification of these tests was performed in the following sequence to ensure that the flight program checks are in the specified order.

1. Data legal test: A generalized switch selector mode and three data commands (only two required) were issued to verify this requirement. The last data command was correctly rejected and error code 04 was issued.

2. True-complement test: A correctly coded mode command requiring data words was issued with various combinations of data commands with valid and invalid complement bit configurations and correct sequence bit format. The invalid data was correctly rejected and error code 44 was issued.

3. Sequence bit test: A valid mode word followed by data words with correct complement bit patterns but valid and invalid sequence bit values was sent to the flight program. The invalid words were correctly rejected and error code 60 was issued.

Telemetry requirements for DCS error messages and the data status indicator were verified in all simulation runs.

10.3.3 DCS Data Validation

The data for some mode commands required further testing after being received and formatted.

- Illegal memory dump test: Memory dump commands requesting data from non-existant modules and where the start module, sector and address was greater than the end module, sector, and address were commanded. The commands were properly rejected and error code 50 issued. Further verification of this test is described in paragraph 10.4.4.

- Valid time test: The program properly rejected a Navigation Update command and an S-IVB/IU De-orbit command with implementation time of less than 10 seconds in the future. A TB^5 start time of less than $T5_{GRD}$ was properly rejected. Error code 54 was correctly issued following these rejections.

10.3.4 DCS Error Message

Each time the program rejects a DCS command, an error message is correctly telemetered. Error telemetry format was verified in the simulations described above. Issuance of DCS error messages exceeding seven consecutive failures verified the implementation of the error counter and automatic terminate command initiation. Error Code 70 could not be verified since no execute alternate sequence command is defined for this mission.

10.4 DCS COMMANDS

The flight program correctly accepts and processes all the DCS commands defined for this mission. The following paragraphs discuss the tests used to verify the operation of these DCS commands.

10.4.1 Time Base Update

The time base update command correctly increments or decrements the time in the time base by an amount specified by an accompanying DCS data command. Both positive and negative values of maximum, minimum, and least significant magnitudes were tested for accuracy of time base time change. A positive time base update of maximum magnitude was issued immediately after a station acquisition to ensure that the calibration switch selectors were issued at maximum rate. A negative maximum update issued after T4+0 was used to verify that switch selectors cannot be reissued. Time base updates of maximum positive magnitude were given prior to each orbital guidance maneuver to verify that the times for the maneuvers are not changed. It was verified that a time base update is not accepted in TB5 and that the biased time base time is reset at TB5+0. A time base update was issued after the last tabled switch selector.in TB4 to verify implementation.

Multiple time base updates (both positive and negative) were issued to check correct bias accumulation. Also, a terminate command was issued to ensure that it had no effect upon update implementation. Bit 9 of Mode Code 27 was used to indicate correct time base update command implementation.

10.4.2 Navigation Update

The navigation update command correctly replaces the orbital navigation state vector with one supplied from the ground. The accuracy of the navigation state vector component replacement was verified with a navigation update with an implementation time (NUPTIM) greater than 10 seconds in the future. A navigation update with NUPTIM less than 10 seconds in the future was issued to verify rejection, automatic program initiated terminate, and error code 54 telemetry. Multiple navigation updates were sent to the flight program with NUPTIM1 more than 10 seconds in the future and NUPTIM2 less than 10 seconds in the future to ensure that the second update is rejected with no effect on the first. Multiple updates with NUPTIM2>NUPTIM1 > 10 seconds in the future were issued to verify that the last navigation update accepted replaces previous updates. Mode Code 27, bit 8, was correct on each update simulation run. A terminate command did not affect the navigation update. A memory dump verified proper storage of the updated parameters.

10.4.3 Generalized Switch Selector

A sequence of generalized switch selectors was issued in all DCS time frames to check for correct nominal operation. Error code 34 was generated by requesting generalized switch selectors at maximum rate during a flight period with high speed density tabled switch selectors. The flight program will not issue a generalized switch selector command if there is less than 500 ms until the next preprogrammed switch selector. Terminate commands issued after the second data word were used to verify the no effect requirement on switch selector issuance. Attempts to obtain complement switch selectors were made to ensure that all commands are treated as true address switch selectors and bit 8 of the switch selector address is ignored by the program.

The generalized switch selector is classified as a Class III alternate sequence and its priority was tested, where the possibility existed for interference, with other types of switch selectors. Further discussion of switch selector priority verification is contained in Section 9.4.

10.4.4 Memory Dump

This DCS command correctly causes the flight program to telemeter the memory locations specified by the accompanying data words. A memory dump command was issued for telemetry of a memory portion greater than 16 words (a block). It was verified that the first word of the block telemetered does identify the first memory word telemetered. A memory dump command including a portion of data requested from a non-existent module was issued to verify rejection of the command. The start module, sector and address must be less than or equal to the end module, sector and address requested. Correct implementation of memory dump commands requesting from one to fifteen memory locations was verified. A memory dump command was issued requesting data from an odd-numbered module to ensure that the telemetered data was from an even-numbered module. Terminate commands were issued during all portions of memory dump to verify that the terminate will stop the dump.

10.4.5 Terminate

In addition to the tests herein outlined, a terminate was tested with respect to all mode commands requiring data to ensure correct DCS routine resetting before all required data words had been accepted.

10.4.6 Execute Generalized Maneuver

The implementation of both types of generalized orbital maneuvers, inertial hold and track local reference, was verified by issuing the appropriate DCS mode command and the 20 DCS data commands.

Commands with T_{som} equal to zero and some time in the past were implemented within one computation cycle (after all data was received and formatted). Commands with T_{som} equal to a future time were implemented at the correct time.

A check was made to verify the correct usage of the three reference angles, Y_{ref}, Z_{ref}, and X_{ref}, by the flight program and the setting of the correct mode code bits. A memory dump was commanded to verify correct storage of the execute generalized maneuver parameters.

The generalized maneuver command remains in effect until further DCS action commands another executed generalized maneuver or return to nominal timeline. Correct setting of MC27 bits in conjunction with this DCS command was established.

Correct interaction of the generalized maneuver with D19 was verified in a test series which included both inertial hold and track local reference maneuvers.

With the spacecraft in control of Saturn (DI9 set), an Execute Inertial Hold Maneuver DCS command was accepted followed by a Return to Nominal Timeline DCS command. Upon the return of control to the instrument unit (DI9 reset) the flight program executed the preprogrammed Track Local Reference maneuver. Subsequently, control was switched to the spacecraft followed by an acceptable Execute Inertial Hold Maneuver. Upon the return of control to the instrument unit the flight program executed the commanded Inertial Hold maneuver. An execute generalized maneuver command was issued before a pending execute generalized maneuver and a return to nominal timeline DCS command was implemented to verify the replacement of the pending command by the current command. A terminate command was issued after the 20th valid data command was received to verify that it does not prevent the execute generalized maneuver command implementation.

A generalized maneuver command with a T_{som} scheduled to occur after the start of TB5 was issued and a memory dump commanded to verify that the storage locations of the generalized maneuvers are zeroed at TB5+0.

10.4.7 Return to Nominal Timeline

This DCS mode command provides the capability to return to the nominal timeline after other DCS action has been initiated to override the preprogrammed orbital attitude timeline. The time to return to nominal time (T_{RNTL}) is sent in five DCS data commands. A return to nominal timeline command was issued with T_{RNTL} in the future to verify the correct storage of T_{RNTL}, the zeroing of all bits in the location in which GOMTYP (generalized orbital maneuver type) is stored, except a 1 in bit 2, and the zeroing of memory locations containing the reference angles for a pending execute generalized maneuver. A command was issued with T_{RNTL} equal to zero and some time in the past to establish that the command was implemented within one computation cycle (after the data is received).

A return to nominal timeline command was issued before a pending return to nominal timeline and execute generalized maneuver command was implemented to verify the replacing of the pending command by the current command. After the fifth valid data command was received, a terminate command was issued to demonstrate that it does not prevent the return to nominal timeline command. The memory locations containing the data for the return to nominal timeline command are zeroed at TB5+0.

10.4.8 ECS Water Control Valve Logic Inhibit

Correct operation of this mode command was verified in the ECS water valve logic test defined in Sections 7.4.10 and 7.4.11.

10.4.9 Execute Maneuver

This DCS command will provide the capability to initiate a specialized orbital maneuver. This command is presently not defined; therefore, no verification was done on this command.

10.4.10 Execute Alternate Sequence

There are presently no DCS alternate sequences defined. No verification was performed on this command.

10.4.11 Targeting Load

This DCS command is used in preflight mode only; therefore, this command will not be verified as part of flight mode verification. It was verified that this command was not enabled during flight. Additional verification of this command is described in paragraph 3.2.

10.4.12 Ladder Magnitude Limit

A Ladder Magnitude Limit command was issued with the least significant value in the data word to verify the accuracy of the implementation of the command. A command with the maximum possible value was issued to verify proper limiting of the command.

This command was issued followed by an attempt by the nominal timeline to change the ladder magnitude limits to values less than and greater than those specified by the command. The program used the correct limits. The loss of APS attitude control test was failed after giving the Ladder Magnitude Limit command to verify that the limits are set to LML. This was followed by another command with limits smaller than LML to verify that the limits remain at LML. This portion was verified in conjunction with Section 8.3.3.

10.4.13 S-IVB/IU De-orbit

The DCS S-IVB/IU De-orbit command correctly begins TB5, performs the required initializations, and issues the specified alternate switch selector sequences.

The data commands are formatted correctly by the flight program. The maximum and minimum duration times specified by the data commands were verified. The start time of TB5 as specified in the DCS command is properly tested to ensure that the start time is greater than $T5_{GRD}$ and at least 10 seconds in the future. The program properly rejects the command, issues error code 54 and performs an automatic program initiated terminate if either of these tests is failed.

S-IVB/IU De-orbit commands were issued with TLDD=0, DELVR 0, and THDD=0. When TLDD is zero, the LOX and hydrogen dumps are bypassed and the safing sequence is scheduled to start at T540.0 seconds. When DELVR is zero, the velocity test is bypassed and the LOX and hydrogen dumps were performed for the duration defined by TLDD and THDD. When DELVR is non-zero, the dumps are terminated and the safing sequence started when the total measured velocity becomes equal to or greater than DELVR. When THDD is zero, the LOX dump is terminated by scheduling the safing sequence to start immediately following the LOX dump. Commands with various values of TLDD, THDD, and DELVR were issued to cause the velocity test to be satisfied during every possible segment of the LOX and hydrogen dumps. In every case, the velocity test is terminated when the safing sequence is scheduled.

The four quantities from the data commands are properly scaled and stored by the flight program. A de-orbit command, accepted before TB5 is started in response to a previous de-orbit command, correctly replaces the previous command. The de-orbit command is not accepted in TB5.

Mode Code 27, bit 7, is set and reset properly.

10.4.14 Compressed Data Dump

This DCS command provides the capability for commanding a dump of the compressed data tables.

Upon acceptance of this command, the flight program will dump the compressed data tables three times in their entirety. The compressed data dump will be stopped only by receipt of a terminate command. Other modes commanded during a dump are accepted except a memory dump or another compressed data dump which are rejected and error code [illegible]4 will be issued.

Verification of the data contained in the compressed data tables is described in Section 11.5.2.

10.4.15 Remove Inhibit on the Extraction Maneuver

The Remove Inhibit on the Extraction Maneuver command was issued prior to the nominal start time of Maneuver 6. Maneuvers 6, 7, and 8 were started at the nominal times. The command was issued after the nominal start times for Maneuver 6, 7, and 8. In each case Maneuver 6 was started immediately and Maneuvers 7 and 8 were delayed by the same amount of time that Maneuver 6 was delayed. Bit 13 of mode code 27 was set properly in every case. The nominal start time deltas are maintained unless ground command maneuvers or spacecraft control are interspersed with these maneuvers.

Multiple commands were issued in Time Base 4. The first command removed the inhibit on the extraction maneuver. The subsequent commands were properly accepted, but had no further effect. The command was issued in Time Base 5, but it was properly rejected and error code 74 was issued.

SECTION 11

REAL TIME TELEMETRY AND DATA COMPRESSION

11.1 INTRODUCTION

The flight program correctly provides telemetry and data compression. Verification of this activity is described in the following paragraphs.

11.2 TELEMETRY SYSTEM INTERFACE

The adequacy of flight program telemetry control and timing was proven by the analysis of Sim Lab runs of past programs through the use of telemetry.

11.3 IDENTIFICATION TAGS

The telemetry data is correctly issued using specific tags called PIO (process input/output) tags. The flight program controls only portions of the composite 40 bit telemetry word, the remaining parts being formed by the telemetry system hardware. Verification was accomplished by ensuring that a specific PIO tag and mode register setting identified the correct parameter and that the data was properly scaled for subsequent ground station reduction.

The correspondence between parameter, PIO tag and mode register setting was verified by checking the tabled data in the flight program against the tabled requirements in the EDD. A careful monitoring of changes to the EDD telemetry tables and symbolic tape compare of the implementation of these changes was used to ensure a fixed tag/quantity definition.

11.4 GENERAL REQUIREMENTS FOR LVDC AND LVDA TELEMETRY

LVDC telemetry correctly adheres to the general requirements specified by the EDD.

11.4.1 LVDC Telemetry

The 6D/LVDC simulator monitors the length of time between execution of telemetry PIO instructions; therefore, all 6D/LVDC runs were checked for an insufficiency of time (less than 4.25 milliseconds) between these instructions.

11.4.1.1 LVDC Regularly Scheduled/When Computed Telemetry

Correct implementation of the requirements for regularly scheduled/when computed telemetry was checked using the 6D/LVDC simulator. Navigation, guidance, accelerometer, IGM, mode code and special parameter telemetry was verified through analysis of the outputs obtained from simulation processing. For each flight phase, a comparison of the specified parameters and those monitored during the simulation was made to ensure that the flight program conforms to telemetry requirements.

11.4.1.2 LVDC On-Occurrence Telemetry

Telemetry indicating execution of a flight sequence event was verified through analysis of both nominal and perturbed simulations designed to cause the required condition. The format of data, identification codes and real-time clock readings were checked for discrete input and output registers, interrupt register, switch selector register, switch selector feedback register, and special event telemetry.

11.5 DATA COMPRESSION

Data compression specifications were verified using compressed data from nominal flight simulations and from a series of perturbations designed to test data table overflows, data dump rates, and compression of data for on occurrence events. The results of the data compression are discussed below and also in Section 10.4.14.

11.5.1 General Data Compression Requirements

The telemetered compressed data tables were checked by monitoring the time, identification code and data formats. The table length and the maximum compression period was verified by monitoring compressed data telemetry after table wraparound. The program correctly stores and telemeters all data types with the associated time.

11.5.2 Data to be Compressed

Time compressed (Group A), occurrence compressed (Group B) and amplitude compressed (Group C) data were verified using both nominal and perturbed simulations. Verification of each group is disucssed in the following paragraphs.

11.5.2.1 Group A: Time Compressed Data

Group A data was obtained from the nominal simulation. Fine and backup gimbal angle data and accelerometer data were checked to verify correct compression rates.

11.5.2.2 Group B: Occurrence Compressed Data

Group B data was forced in a series of simulations designed to verify correct storage of event data. Storage of discrete output register changes and both tabled and generalized switch selector data was ensured. TLC HOP constant compression was verified by checking the flight program logic flow.

11.5.2.3 Group C: Amplitude Compressed Data

Group C data was verified using both nominal and perturbed simulations to check compression of each function specified. Sample and storage rates for the functions of this group were checked by forcing system failures for MC24 and the Error Monitor Register and by setting and resetting various discrete input register bits.

11.5.3 Telemetry of Compressed Data

The compressed data tables are correctly telemetered three times whenever the compressed data dump DCS command is received. The issuance of the compressed data telemetry was verified by checking the mode register setting, the telemetry PIO tags and the sequence in which the compressed data table locations were selected for telemetry.

SECTION 12

PREFLIGHT TESTS

Verification of the preflight routines is not a requirement in flight program verification. It was verified that the non-repeatable sim flight and repeatable sim flight modes do not interact with the flight mode. Section I-3 describes additional verification performed in the preflight prepare-to-launch routines.

SECTION 13
ALGORITHMS

13.1 INTRODUCTION

This section describes the algorithms used in the flight program to approximate elementary functions and mathematical procedures. The implementation of these algorithms was checked by verifying that at least one, and in most cases, more than one function computed through the use of the algorithm is calculated properly.

13.2 SINE-COSINE ALGORITHM

The sine-cosine algorithm was verified by forcing input arguments in each quadrant and at 0°, 90°, 180°, and 270° and the results checked against tables. This algorithm is correctly used to obtain terms for the coordinate transformation matrices and verification was completed by checking the results of these transformations.

13.3 ARCTANGENT ALGORITHM

The arctangent algorithm was verified by inputting various values for the numerator and denominator arguments. This algorithm was tested in each quadrant and at 0°, 90°, 180° and 270° and the results checked against tables. This algorithm is used in the calculation of the range angle (ϕi), the desired vehicle pitch and yaw attitude (χy and χz) and the track local horizontal guidance commands upon return of control from S/C to IU. The verification of the calculation of these quantities completed the arctangent algorithm verification.

13.4 NATURAL LOGARITHM ALGORITHM

The natural logarithm algorithm was verified by inputting test arguments of 1, less than 1, greater than 1 and the two end points. Verification of these logarithms and the correct calculation of the intermediate IGM terms of velocity-to-be-gained (L_1 and L_3) ensured the proper implementation of this algorithm.

13.5 SQUARE ROOT ALGORITHM

This algorithm was verified by inputting positive, negative and zero arguments and checking the results for accuracy. The values for desired yaw attitude (χ_z), relative velocity (V_r), and position (R) were also checked to complete the verification.

13.6 INVERSE SQUARE ROOT ALGORITHM

The inverse square root algorithm is correctly implemented in the calculations for the inverse of the vehicle's radius from the center of the earth (1/R). Verification was accomplished by hand checking these calculations at initialization, during boost and during orbit.

13.7 VECTOR DOT PRODUCT

Several test cases were run using known vectors to verify the results from the vector dot product algorithm. Since the vector dot product is used in the rotation of gravity acceleration into the injection system and the gravity accelerations affect the guidance commands, correct IGM performance also indirectly verified the implementation of the vector dot product.

13.8 VECTOR CROSS PRODUCT

The transformation from one coordinate system to another is correctly accomplished by use of the vector cross product algorithm. The checks of these transformation matrices and the exercise of the algorithm using several known vectors assured the correct implementation of the cross product.

APPENDIX A

REFERENCES

1. LVDC Equation Defining Document for the Saturn IB Flight Program, IBM No. 70-207-0001.
2. Astrionics System Handbook - Saturn Launch Vehicles, MSFC No. IV-4-401-1, IBM No. 68-966-0002.
3. Programmer's Operating Manual - Saturn Launch Vehicle Digital Computer, Launch Vehicle Data Adapter and Programmable Test Controller, Contracts NAS8-11561 and NAS8-11562.
4. Flight Program Deviation Report Processing, IBM No. K10-70-131.
5. Program Verification Plan for the ASTP Flight Program, ASTP-DRL-686, IBM No. 75W-00005.

APPENDIX B
PERFORMANCE CASES

1. +5% thrust and mass flow rates in both stages
2. -5% thrust and mass flow rates in both stages
3. High stop PMR (HPMR) for duration of S-IVB
4. Low stop PMR (LPMR) for duration of S-IVB
5. All accelerometer hard failure (zero change), A channel
6. X accelerometer hard failure (zero change), both channels
7. Y accelerometer hard failure (zero change), both channels
8. Z accelerometer hard failure (zero change), both channels
9. Engine #1 out at T1+5
10. Engine #6 out at liftoff
11. Engine #4 out at T1+40
12. Engines #1 and #5 out at T1+100
13. Fine gimbal failure, fly on backups
14. +2° β_p bias, +1° θ_p bias
15. -2° β_p bias, -1° θ_p bias
16. S-IB engine #1 actuator hardover inboard at 10 seconds

APPENDIX C

PERFORMANCE CASE ANALYSIS

CASE #1 - +5% THRUST AND MASS FLOW RATE FOR BOTH STAGES

This case was accomplished by simulating a 5% increase in the nominal thrust and mass flow rate of both the S-IB and S-IVB stage. As a result of the increased mass flow rate, the S-IB stage propellant low level sensor was activated at TB1+126.43 seconds, prior to the associated low level sense interrupt being hardware and software enabled at TB1+127.86 seconds. Inboard engines cutoff occurred as scheduled at TB2+3.0 seconds, but fuel depletion occurred at TB2+3.20 seconds, prior to the associated interrupt and discrete input being hardware and software enabled. The enable did not occur until TB2+4.5 seconds. The S-IB outboard engine out discrete input (DI14) was detected causing the expanded zero test to be used in accelerometer processing for one computation cycle. S-IVB cutoff occurred at TCRR+566.96 seconds. The flight program correctly compensated for this increase in thrust as acceptable end conditions were achieved.

CASE #2 - -5% THRUST AND MASS FLOW RATE IN BOTH STAGES

This case was accomplished by simulating a 5% decrease in the nominal thrust and mass flow rate in both the S-IB and S-IVB stages. As a result, fuel depletion of the S-IB stage occurred at TB2+7.470 seconds, 7.937 seconds later than nominal. In addition, the S-IVB stage burned 31.738 seconds longer than nominal with fuel depletion occurring at TCRR= 640.339 seconds, 2.348 seconds before the calculated cutoff. Therefore, unacceptable end conditions were achieved.

CASE #3 - HIGH STOP PMR (HPMR) FOR DURATION OF S-IVB

This case was simulated by forcing a high propellant mixture ratio (HPMR) for the duration of the S-IVB burn. Due to the higher thrust and mass flow rate, S-IVB cutoff occurred at TCRR+582.832 seconds, 17.828 seconds earlier than nominal. Satisfactory end conditions were achieved, indicating that the flight program correctly compensated for the high PMR.

CASE #4 - LOW STOP PMR (LPMR) FOR DURATION OF S-IVB

This case was simulated by forcing a low propellant mixture ratio (LPMR) for the duration of the S-IVB burn. Due to the improper mixture ratio, the S-IVB fuel depleted at approximately TGRR+669.043 seconds. At fuel depletion, T_{3i}, was equal to 5.24 indicating that 5.24 seconds of additional burn time was required to reach cutoff. With the exception of the early depletion, the flight program correctly compensated for the low PMR.

CASE #5 - ALL ACCELEROMETER HARD FAILURE (A CHANNEL)

In this case the A-accelerometer channel in the X, Y and Z axes were frozen to zero at liftoff. Whenever $|\Delta A-\Delta B| > 2$, the channel nearest the expected reading was used in computing velocity and position in that axis and the appropriate Mode Code 24 bits were set. Since ΔA was forced to zero, the disagreement test was failed when $|\Delta B| > 2$. Whenever $|\Delta A-\Delta B| < 3$, the A channel reading was selected and the appropriate MC24 bits were reset. Therefore, when $|\Delta B| < 3$, a zero reading was used in computing velocity and position in that axis.

End conditions telemetered by the LVDC compared favorably with the desired end conditions. However, "actual" end conditions indicated by the 6D were perturbed sufficiently that the orbit attained was slightly undesirable. All flight sequencing was correct.

CASE #6 - X-ACCELEROMETER HARD FAILURE (BOTH CHANNELS)

In the computation cycle following GRR, the X-accelerometer A and B channels were frozen to force zero accelerometer readings, simulating failure conditions. The accelerometer zero test correctly detected the zero readings as failure indications, and set bits 2, 3, and 23 in Mode Code word 24. The SMC calculations were inhibited and the backup acceleration, $(F/M)_C$, was used to compute the velocity and position in the X-axis.

Whenever the accelerometer zero test was disabled or when $|\theta_p - 90°| < 2°$, the zero accelerometer readings in the X-axis were accepted and were used in computing position and velocity in the X direction. Bits 2, 3 and 23 of Mode Code 24 were reset and bit 10 of Mode Code 25 was set when the SMC calculations became active.

As a result of using the erroneous zero readings when the zero test was disabled or when $|\theta_p - 90^\circ| < 2^\circ$ and using $(F/M)_c$ during all other periods of flight, large navigation errors developed in the X direction.

The X navigation errors caused slightly incorrect gravitational components to be used in the other axes thus causing small navigation errors in the Y and Z directions. As a result, the end conditions achieved were not suitable for comparison with nominal end conditions.

S-IVB cutoff occurred at TGRR+601.465 seconds, .804 seconds later than nominal.

All sequencing was executed properly and there were no deviations from specifications.

CASE #7 - Y-ACCELEROMETER HARD FAILURE (BOTH CHANNELS)

In the computation cycle following GRR, the Y-accelerometer was frozen in both the A and B channels, resulting in zero readings. The accelerometer zero test correctly detected the zero readings as failure indications. The zero accelerometer readings in the Y-axis were rejected whenever the accelerometer zero test was enabled and $|\theta \text{ yaw}| > 2^\circ$. Bits 4, 5 and 19 of MC24 were set, the SMC calculations were inhibited, and the $(F/M)_c$ backup profile was used to calculate position and velocity in the Y direction.

The zero accelerometer readings in the Y-axis were accepted and used in computing position and velocity in the Y direction whenever the accelerometer zero test was disabled or when $|\theta \text{ yaw}| < 2^\circ$. Bits 4, 5 and 19 of MC24 remained reset during the BML.

Large navigation errors developed in the Y direction as a result of using the erroneous zero readings when the zero test was disabled or when $|\theta \text{ yaw}| < 2^\circ$, and using $(F/M)_c$ during all other periods of flight.

The LVDC end conditions attained were good. The 6D end conditions were slightly perturbed as a result of the navigation errors, although they were still acceptable. This case demonstrated the capability of the flight program to handle a Y-accelerometer hard failure and still attain acceptable end conditions.

S-IVB cutoff occurred at TGRR+600.763 seconds which was near nominal cutoff time. All flight sequencing was correct.

CASE #8 - Z-ACCELEROMETER HARD FAILURE (BOTH CHANNELS)

In this case, the Z-accelerometer was failed to zero in the comp. cycle following GRR. Both the A and B channels were frozen to force zero readings.

The accelerometer zero test correctly detected the Z-accelerometer zero readings as failure indications. Whenever the accelerometer zero test was enabled and $|\theta_p| > 2^\circ$, the zero accelerometer readings in the Z-axis were rejected. The sign bit, bit 1 and bit 22 of MC24 were set, the SMC calculations were inhibited, and $(F/M)_c$ was correctly substituted for F/M. The resultant acceleration component is correctly used in the calculation of position and velocity in the Z direction. Whenever the accelerometer zero test was disabled or $|\theta_p| \leq 2^\circ$, the zero readings in the Z-axis were accepted and the sign bit, bit 1 and bit 22 of MC24 were reset.

Large navigation errors developed in the Z direction as a result of using the erroneous zero readings when the zero test was disabled and when $|\theta_p| < 2^\circ$ and using $(F/M)_c$ during all other periods of flight.

The S-IVB fuel depleted at TGRR+605.817 seconds, with T_{3i} equal to .86 seconds.

CASE #9 - ENGINE #1 OUT AT T1+5 SECONDS

Engine #1 lost thrust at T1+5 seconds and was properly detected at that time. The tilt freeze interval, ΔTf, was correctly computed to be 12.855 seconds and was implemented at Tc+30.417 seconds, with tilt arrest occurring at Tc+122.475 seconds. S-IB fuel depletion occurred at TGRR+178.867 seconds, 21.589 seconds later than nominal. The S-IVB stage depleted fuel at TGRR+627.303. T_{3i} at this time was approximately 3.74, indicating that 3.74 additional seconds of burn time would have been required to reach cutoff. The flight program properly corrected for the engine out and performed properly up to fuel depletion.

CASE #10 - ENGINE #6 OUT AT LIFTOFF

This case was accomplished by simulating the failure of S-IB engine #6 (inboard) at TB1+0.0 seconds. As a result, the tilt freeze interval, ΔT_f, was computed to be 13.875 seconds with tilt arrest, T_{ar}, during the boost major loop commencing at TB1+126.13, 10.8 seconds later than nominal. S-IB fuel depletion occurred at TGRR+178.18, 20.9 seconds later than nominal. S-IVB fuel depletion occurred at TGRR+626.69 seconds, with T3I-3.83 seconds. All flight sequencing was correct.

CASE #11 - ENGINE #4 OUT AT 40 SECONDS FROM LIFTOFF

This case was accomplished by simulating the failure of S-IB engine #4 (outboard) at TB1+40.03 seconds. Tilt arrest occurred during the boost major loop commencing at TB1+108.02 seconds (tilt freeze interval, ΔT_f, was zero). S-IB fuel depletion occurred at TGRR+173.43, which is 16.15 seconds later than nominal. S-IVB cutoff occurred at TGRR+620.46 seconds, 19.8 seconds later than nominal. Acceptable end conditions were achieved.

CASE #12 - ENGINES #1 AND #5 OUT AT 100 SECONDS FROM LIFTOFF

This case was accomplished by simulating the failure of S-IB engines #1 (outboard) and #5 (inboard) at TB1+100.06 seconds. The tilt freeze interval, ΔT_f, was calculated to be 0.0 and tilt arrest, T_{ar}, occurred at TGRR+125.34 seconds. S-IB fuel depletion occurred at TGRR+171.996, 14.7 seconds later than nominal. The S-IVB engine cutoff occurred at TGRR+618.50, 17.84 seconds later than nominal. The S-IVB burned only 3.12 seconds longer than nominal due to the two S-IB engines out. End conditions achieved were satisfactory.

CASE #13 - FINE GIMBAL FAILURE, FLY ON BACKUPS

This case was simulated by failure of the X, Y and Z fine gimbals so that the vehicle would fly using the backup gimbals. The flight program correctly responded to this condition, with S-IVB cutoff occurring only .08 second later than nominal at TGRR+600.74. Proper flight program response was demonstrated in that acceptable end conditions were noted with little change in vehicle performance.

CASE #14 - +2° β_p BIAS, +1° θ_p BIAS

This case was simulated by perturbing the pitch gimbal angle (θ_p) by 1° and the nozzle deflection (β_p) by 2°. The flight program correctly responded to this condition with S-IVB cutoff occurring at TGRR+602.93, 2.33 seconds later than nominal. Proper flight program computation of steering misalignment correction (SMC) terms was demonstrated in that acceptable end conditions were achieved, with the perturbation having little effect on flight program performance. The pitch SMC terms were significantly larger than nominal.

CASE #15 - -2° β_p BIAS, -1° θ_p BIAS

This case was simulated by perturbing the pitch gimbal angle (θ_p) by -1° and the nozzle deflection (β_p) by -2°. The flight program correctly responded to this condition with S-IVB cutoff occurring at TGRR+602.86, 2.20 seconds later than nominal. Proper flight program computation of steering misalignment correction (SMC) terms was demonstrated in that acceptable end conditions were achieved, with the perturbation having little effect on flight program performance. The pitch SMC terms were significantly larger than nominal.

CASE #16 - +2° β_y BIAS, +1° θ_y BIAS

This case was simulated by perturbing the yaw gimbal angle (θ_y) +1° and the yaw nozzle deflection (β_y) +2°. As a result, fuel depletion of the S-IB stage occurred at TB2=6.742 seconds, 0.014 seconds earlier than nominal. In addition, the S-IVB stage burned 2.339 seconds longer than nominal with cutoff occurring at TGRR=603.008 seconds. However, the flight program correctly compensated for these biases as acceptable end conditions were achieved.

CASE #17 - -2° β_y BIAS, -1° θ_y BIAS

This case was simulated by perturbing the yaw gimbal angle (θ_y) -1° and the yaw nozzle deflection (β_y) -2°. As a result, fuel depletion of the S-IB stage occurred at TB2=6.752 seconds, 0.024 seconds earlier than nominal. In addition, the S-IVB stage burned 3.408 seconds longer than nominal with cutoff occurring at TGRR=604.068 seconds. However, the flight program correctly compensated for these biases as acceptable end conditions were achieved.

CASE #18 - S-IB ENGINE #1 ACTUATOR HARDOVER INBOARD AT SEVEN SECONDS FROM LIFTOFF

This case was simulated by forcing the pitch and yaw actuators of engine #1 to a value of 8 degrees in a negative direction. As a result, fuel depletion of the S-IB stage occurred at TB2=6.801 seconds, 0.044 seconds later than nominal. In addition, the S-IVB stage burned 1.220 seconds longer than nominal with cutoff occurring at TGRR=601.919 seconds. However, the flight program correctly compensated for this failure as acceptable end conditions were achieved.

APPENDIX D

END CONDITIONS COMPARISON

TABLE I

PERFORMANCE CASE ERRORS

Flight Azimuth= 45.158°
(Desired-Achieved)

Parameter Units	ΔV_T M/S	ΔR_T M	$\Delta \theta_T$ Deg.	ΔI Deg.	$\Delta \lambda$ Deg.
Desired Values	7818.46	6528178.0	0.0	51.78	156.887
Case					
1	-.29	-7.64	-.0042	-.0024	-0.0036
2	S-IVB	Fuel Depletion			
3	.186	3.46	-.00083	-.0027	-.0039
4	S-IVB	Fuel Depletion			
5	.128	28.89	.0033	.0001	0.0
5-6D*	-11.82	-6445.6	-.1726	-.0108	-.0066
6	.068	11.38	-.00016	-.0034	-.0053
6-6D*	8.70	11788.4	-.2216	-.0082	-.0086
7	-.093	29.26	.00013	-.0169	-.0261
7-6D*	-.14	60.55	.00074	-.0317	-.0120
8	S-IVB	Fuel Depletion			
8-6D*	S-IVB	Fuel Depletion			
9	S-IVB	Fuel Depletion			
10	S-IVB	Fuel Depletion			
11	-.122	30.36	.0029	-.0006	.0010
12	.043	28.5	.0034	-.0013	.0022
13	.12	35.4	.0014	-.0006	-.0011
14	-.107	43.4	.0056	.0004	.0006
15	-.065	12.73	.00014	-.0002	-.00045
16	-.14	35.3	.003	-.001	-.002
17	-.32	37.8	.003	.001	.002
18	.08	20.1	.002	.001	.001

*Accelerometer failure cases, 6D data included for comparison.

TABLE II

NOMINAL TARGETING ERRORS
Flight Azimuth=45.158
(Desired-Achieved)

Parameter Units	ΔV_T M/S	ΔR_T M	$\Delta \theta_T$ deg	ΔI deg	$\Delta \lambda$ deg
Desired Values	7818.46	6528178.0	0.0	51.78	156.887
Case 1.1	-.008	35.9	.0055	-.0005	-.001

APPENDIX E

UNCORRECTED DEVIATIONS

ASTP-1 In TB2 if an accelerometer "zero" reading (Δ of 0 or 1 bit) occurs near S-IB IECO, the accelerometer zero reading may fail the zero test even if the reading should be acceptable.

If the SIB IECO occurs between the computation of the expected accelerometer change, V_F, and the computation of the thrust misalignment uncertainty in the estimated accelerometer change, A_{C0}, the value for F/M could be changed between the two computations. This is due to F/M being set to a constant at IECO. Thus, the comparison of V_F, with A_{C0}, would not be valid, possibly resulting in the accelerometer reading incorrectly failing the zero test.

ASTP-2 The Execute Generalized Maneuver DCS command may be enabled at a time different than $T_{SS}+2905.0$ in TB5. The enable for this command is scheduled, in the Time Base 5 module, by adding $T_{LDD}+T_{HDD}+63.5+2905.0$ seconds in TB5. This will be the correct time if the safing sequence is started after completion (full duration) of the LOX and LH_2 dumps specified in the S-IV/IU De-orbit DCS Command. This will <u>not</u> be $T_{SS}+2905$ if the safing sequence is started by one of the other three conditions stated in Note 6 of the switch selector table:

1. $T_{LDD}=0.0$
2. DELVR test
3. $T_{HDD} = 0.0$

APPENDIX F

NOMINAL SEQUENCE OF MAJOR EVENTS

ASTP 45.158 Degree Azimuth Nominal

Event	Nominal Time in Time Base (seconds)	Nominal Tim from GRR (seconds)
Guidance Reference Release	TB0+0.00	0.00
Time Base 1 Initiated	TB0+17.67	17.67
Pitch and Roll Initiation	TB1+8.79	26.46
Roll Maneuver Complete	TB1+57.47	75.14
Computer Switch Point #1	TB1+99.96	117.63
Computer Switch Point #2	TB1+100.17	117.84
Computer Switch Point #3	TB1+119.96	137.63
Tilt Arrest	TB1+128.11 (T_C+129.73)	145.78
Time Base 2 Initiated	TB1+132.87	150.54
S-IB Inboard Engines Cutoff	TB2+2.97	153.51
S-IB Outboard Engines Cutoff	TB2+6.74	157.28
Time Base 3 Initiated	TB2+6.74	157.28
S-IVB Ullage Ignition	TB3+1.06	158.34
S-IB/S-IVB Separation	TB3+1.26	158.54
S-IVB Engine Start Sequence Initiated	TB3+2.67	159.95
S-IVB Ullage Rockets Jettison	TB3+13.27	170.55
IGM Start	TB3+34.67	191.95
Computer Switch Point #4	TB3+41.96	199.24
SMC Start	TB3+60.07	217.35
Computer Switch Point #5	TB3+203.66	360.94
T_{1I}=0, Initiate PMRC	TB3+328.06	485.34
Begin Chi-Bar Steering	TB3+419.74	577.02
S-IVB Cutoff	TB3+443.38	600.66
Time Base 4 Initiated	TB3+443.60	600.88
Track Local Reference Start	TB4+15.11	615.99