

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

APOLLO

GUIDANCE, NAVIGATION AND CONTROL

R-599

DIGITAL SIMULATION MANUAL

by

Pieter Mimno

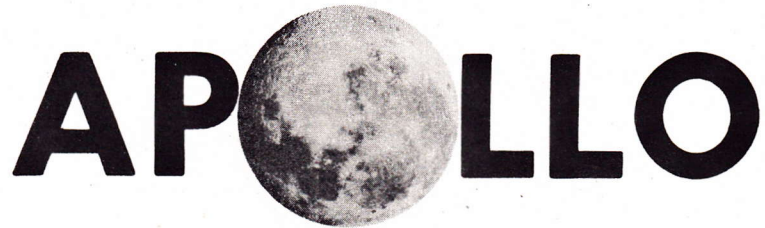
JANUARY 1968

MIT

CAMBRIDGE 39, MASSACHUSETTS

**INSTRUMENTATION
LABORATORY**

EVLES



GUIDANCE, NAVIGATION AND CONTROL

Approved: R. H. Battin Date: 1/26/68
R. H. BATTIN, DIRECTOR, MISSION DEVELOPMENT
APOLLO GUIDANCE AND NAVIGATION PROGRAM

Approved: David G. Hoag Date: 29 Jan 68
DAVID G. HOAG, DIRECTOR
APOLLO GUIDANCE AND NAVIGATION PROGRAM

Approved: Ralph R. Ragan Date: 29 Jan 68
RALPH R. RAGAN, DEPUTY DIRECTOR
INSTRUMENTATION LABORATORY

R-599

DIGITAL SIMULATION MANUAL

by

Pieter Mimno

JANUARY 1968

MIT

**INSTRUMENTATION
LABORATORY**

CAMBRIDGE 39, MASSACHUSETTS

COPY # _____

ACKNOWLEDGEMENT

This report was prepared under DSR Project 55-23870, sponsored by the Manned Spacecraft Center of the National Aeronautics and Space Administration through Contract NAS 9-4065 with the Instrumentation Laboratory, Massachusetts Institute of Technology, Cambridge, Massachusetts.

The author wishes to acknowledge the assistance of members of the Digital Simulation Group of the MIT/IL Apollo Program in the preparation of this manual.

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

DIGITAL SIMULATION MANUAL

ABSTRACT

The digital simulation manual has been written to fulfill a need for comprehensive documentation of the digital simulator used at the MIT/IL to verify mission programs for Apollo. The manual is intended to be used both as an introduction to the digital simulator and as a reference document for users of the simulator. Included in the manual are an introduction to the simulator, a description of the structure and operational modes of the simulator, a summary of the models used to simulate the spacecraft and its environment, and a complete user's guide to simulation input cards, rollbacks and edits.

Digital Simulation Manual

Revision 1, January 1968

Table of Contents

Chapter 1 Introduction

	Page
1. A Preface	1-1
1. B Introduction to Digital Simulator	1-3

Chapter 2 Description of Digital Simulator

2. A General Description	
2. A. 1 Role of the Simulator	2-1
2. A. 2 Structure of the Digital Simulator	2-6
2. B AGC Instruction Simulator	2-9
2. C Communicator	2-13
2. D Environment	
2. D. 1 Structure of Environment	2-14
2. D. 2 List of Environment Routines and Data Files . . .	2-22
2. D. 3 Description of Environment Routines	2-27
ENVCNTRL	2-28
INPUTPRG	2-29
UNIVERSE	2-30
PRELUNCH	2-35
SATURN	2-38
CSMBLK2	2-41

	Page
LEM	2-48
LVCBLK2	2-55
IMUBLK2	2-57
OPTICSB2	2-58
RRADAR	2-59
TARGET	2-62
ASTRNAUT	2-63
LMP1	2-67
DISCONT	2-69
SIMSETUP	2-70
ASTXMCI	2-71
2. D. 4 Initialization of Environment	2-72
2. D. 5 Sequencing of Simulator Programs	2-75
2. D. 6 Coordinate Systems and Transformation Matrices	2-77
2. D. 7 Specification of Vehicle and Stable Member Orientation	2-86

Chapter 3 User's Guide to Simulation Input Cards

3. A Preface	3-1
3. B Simulator Control Cards	
3. B. 1 360 Job Control Cards	3-2
3. B. 2 360 Job Step	3-2
3. B. 3 MAC Run Step	3-3
3. B. 4 AGC Instruction Simulator Director Cards	3-4
3. C AGC Instruction Simulator Special Request Cards	
3. C. 1 Formats	3-7
3. C. 2 Addressing and Bank Notation	3-8
3. C. 3 Glossary of Special Requests	3-10
3. C. 4 Description of Special Requests	3-14
3. C. 5 Glossary of Special Request Conditions	3-31
3. C. 6 Description of Special Request Condition Cards	3-32

	Page
3. D Structure of Input Deck for Benchtest Runs	3-34
3. E Structure of Input Deck for Astronaut Only Runs	3-37
3. F Structure of Input Deck for Flight Test Runs	3-41
3. G Structure of Input Deck for Environment Only Runs	3-46
3. H Environment Input Cards	
3. H. 1 Ground Rules for Input Data Cards	3-49
3. H. 2 Format of Environment Input Data Cards	3-51
3. H. 3 Summary of Input Data Card Requirements	3-52
3. H. 4 Input Cards for INPUTPRG	3-55
3. H. 5 Input Cards for UNIVERSE	3-57
3. H. 6 Input Cards for PRELUNCH	3-59
3. H. 7 Input Cards for SATURN	3-60
3. H. 8 Input Cards for CSMBLK2	3-62
3. H. 9 Input Cards for LEM	3-66
3. H. 10 Input Cards for LVCBLK2	3-72
3. H. 11 Input Cards for IMUBLK2	3-74
3. H. 12 Input Cards for OPTICSB2	3-77
3. H. 13 Input Cards for RRADAR	3-78
3. H. 14 Input Cards for LRADAR	3-80
3. H. 15 Input Cards for TARGET	3-81
3. H. 16 Input Cards for ASTRNAUT	3-82
3. H. 17 Input Cards for LMP1	3-99
3. H. 18 Input Cards for DISCONT	3-100
3. H. 19 Input Cards for SIMSETUP	3-101
3. H. 20 Input Cards for Environment Only Runs	3-105

Chapter 4 User's Guide to Rollbacks

	Page
4. A Introduction to Rollback Concept	4-1
4. B Restarting a Simulation Via a Rollback	4-2
4. C AGC Instruction Simulator Special Requests for Rollback . .	4-4
4. D Environment Input Cards for Rollback	4-5

Chapter 5 User's Guide to Edits

5. A Introduction to Edit Capability	5-1
5. B Edit Runs	
5. B. 1 Tape Labeling	5-3
5. B. 2 Tape Disposition	5-5
5. B. 3 Facilities for Editing AGC Dumps	5-6
5. B. 4 MAC Input Facilities	5-12
5. B. 5 Interpretive Tracing Options	5-18
5. B. 6 Time Selection Options	5-20
5. B. 7 Composition of Edit Input Decks	5-21
5. C Edit Logic	
5. C. 1 AGCEDIT	5-23
5. C. 2 MACEDIT	5-23
5. D Summary of Edit Control Cards	5-25
5. E Listing of EDIT.CONTROL program	5-28

	Page
<u>Appendix A</u> Call File Map	A. 1
<u>Appendix B</u> Data File Map	B. 1
<u>Appendix C</u> Summary Print (ENVSUM) Description	C. 1
<u>Index</u>	I. 1

List of Figures

	Page
2. A. 1 Elements of a Digital Simulation	2-3
2. A. 2 Modes of Operation of Digital Simulation	2-7
2. A. 3 Unmanned LM Mission	2-15
2. A. 4 Manned Lunar Mission	2-18
2. A. 5 Full Complement of Simulation Programs	2-21
3. D. 1 Structure of Bench Test Simulation Input Deck	3-35
3. E. 1 Structure of Astronaut Only Simulation Input Deck	3-38
3. F. 1 Structure of Flight Test Simulation Input Deck	3-42
3. G. 1 Structure of Environment Only Simulation Input Deck	3-47

CHAPTER 1

Digital Simulation Manual

1. A. Preface

This re-release of the Digital Simulation Manual is intended to be used as the primary reference manual for digital simulations on the IBM 360. The manual incorporates in one volume an expanded version of the present Digital Simulation Manual together with an updated revision of the Apollo Digital Simulation User's Guide (ADSUG). Contained in this manual is a detailed description of the organization and structure of the Apollo Digital Simulation as well as a complete Users Guide to Simulation input cards. The manual includes primary reference documentation on AGC special requests, Rollbacks, Edits, Environment input cards, and models used to represent flight hardware. It is intended that an AGC programmer should be able to use the information in this manual to set up a simulation deck, run it, interpret the on line output, and obtain post run edits.

The manual consists of the following major sections:

1. Introduction to Concept of Digital Simulator
2. Organization of Digital Simulator -
data flow, dynamical models, modes of operation.
3. Simulation Input Cards - AGC Special requests, Environment input cards.
4. Rollbacks - options available.
5. Edit capability - AGC dumps, MAC editing, Interpretive trace.
6. Output from Simulation - options available, interpretation of output.
7. Appendices - Summary print definitions, Data File map, Call File map.

The user should note that the documentation in this manual applies only to IBM 360 simulations. Programmers who wish to utilize the Honeywell-1800 should continue to use ADSUG, Revision 4, July, 1966 and Revision 2 of the Digital Simulation manual dated March 3, 1967.

Examples of run decks for use on the IBM 360 are not included in the present Digital Simulation Manual since it is known that the format of various Control Cards will change. A separate manual containing example run decks will be issued periodically, and users will be apprised of important changes to the structure of run decks through individual memos. In addition, listings of up to date run decks and example decks will be maintained on a bulletin board in the Assembly Control Room. It is intended that this bulletin board will be the primary medium of day to day communication between AGC programmers and the Digital Simulation Group. AGC users are urged to review periodically that their run decks conform to the example decks listed on the Assembly Control Bulletin Board.

1. B. Introduction to Digital Simulator

The function of the M. I. T. Apollo Digital Simulator is to provide a rigorous flight environment for the testing of Apollo Guidance Computer programs. The Simulator is basically a tool which an Apollo programmer may use to validate his program in test situations ranging from short diagnostic analyses to full scale mission simulations running from Prelaunch through Entry.

The Simulator is entirely digital - no flight hardware of any sort is incorporated in the simulation, and elements of the Simulator are represented by programs on a general purpose digital computer. At present, all portions of the Simulator have been implemented on an IBM System 360, Model 75 computer. Since the Simulator exists in the purely abstract form of a number of computer programs, its usefulness is closely tied to the precision with which its programs represent the detailed functioning of the real hardware and the flight environment. Considerable care was taken to ensure that the simulated AGC and the real AGC function identically, and that all aspects of the Apollo hardware and flight environment that interact significantly with the AGC were simulated. Finally, in order to enhance the usefulness of the Simulator as a diagnostic tool, a large number of trace, dump, and patch options were built into the Simulator and an extensive post run edit capability was developed.

In structure, the Digital Simulation is composed of two logically independent sections interconnected by an interface routine. One major section of the Simulator consists of a large Assembly Language computer program, known as the AGC Instruction Simulator, which exactly simulates the operation of the Apollo Guidance Computer (AGC), both in storage layout and in detailed arithmetic and logical operation. The other major portion of the Simulator is made up of a number of MAC coded subroutines which, together,

simulate all relevant aspects of the hardware and flight environment which the AGC operates within. This Environment ranges from counters connected to the AGC, through relays, servos, actuators, engine, spacecraft dynamics, optics, astronaut interactions, atmospheric and gravity effects, on up to the motions of celestial bodies. Almost every aspect of the environment which could conceivably interact with the AGC has been simulated by computer programs.

During a simulation run, the AGC Instruction Simulator (that portion of the Digital Simulator that represents the AGC) advances through the AGC program instruction by instruction, simulating the micro operations performed by the AGC in executing each instruction. After each instruction cycle, the state of the simulated computer, in terms of instruction sequencing, contents of erasable storage, interrupt activity, clock incrementation, etc., is identical to the state of an actual AGC executing the same program.

Occasionally in the course of advancing through the AGC program, the AGC Instruction Simulator encounters an instruction which refers to an input or output operation, such as the reading of an input counter or the setting of an output discrete. A program known as the Communicator examines all such input-output references and determines whether immediate interaction with the Environment is required by the specific action of the AGC. In general, the Communicator functions to decouple the Instruction Simulator and the Environment by filtering out unnecessary interaction.

Output discretes generated by the AGC are filtered by the Communicator and passed to the simulated Environment which then reacts to the stimuli in a closed loop fashion. Specifically, the AGC generates output control signals, such as pulse trains and output channel discretes, which the Environment responds to by generating AGC counter increments and input discretes.

By maintaining a high degree of similarity between the simulated and the real AGC-Environment interface, it is possible to subject the simulated AGC to computational loads and dynamic situations which closely approximate the conditions of a real mission. Precision in this comparison of simulated versus real AGC performance is degraded primarily by inaccuracies in AGC or Environment models. Some of these inaccuracies are deliberate, representing a compromise between fidelity and computational speed, while others are largely unknown and difficult to evaluate.

In a typical Flight test simulation run, the AGC programmer sets up a computer run deck which:

- a. Selects the Environment programs and files required to simulate the specific test flight condition.
- b. Selects a specific AGC program to be tested.
- c. Initializes the AGC program to start in a desired manner.
- d. Initializes the Environment (vehicle position, velocity, attitude, and internal status) to a desired configuration.
- e. Starts the AGC Program at a specified location.
- f. Observes the resultant interaction between the AGC and the Environment.

The user has available to him numerous output options, traces, dumps, and a flexible edit capability which he may use to analyze the performance of the AGC program in his specific test situation. If errors are found in the AGC program, the program is reassembled and the simulation run is performed again. Generally, the debugging of an AGC mission program proceeds by testing individual elements of the AGC program on the Simulator separately, and then gradually merging the elements into a working program.

In the following section, the structure of the Simulator is described in greater detail, first in block diagram form, and then in presentations of each major section of the Simulator - the AGC Instruction Simulator, the Communicator and the Environment. The models incorporated in each simulated hardware complex are described in considerable detail to enable AGC users to make individual engineering judgements on the adequacy of the simulated models for his particular test requirements. Following this extensive introduction to the Simulator is the Guide to Simulator Input cards, rollbacks, edits, and outputs.

CHAPTER 2

Description of the Digital Simulator

2. A General Description

2. A. 1 Role of the Simulator

The M. I. T. Apollo digital simulation is a primary design tool used to develop programs for the on-board Apollo Guidance Computer in the command module and in the lunar excursion module. The AGC programmer uses the simulation in early stages of program development to debug preliminary coding. The program under test is executed in a simulation, and by means of various diagnostic tools the programmer can determine where errors exist. In later stages of program development, the Simulator can be used to verify the adequacy of the various navigation, guidance, and control programs as these programs are exercised in a flight environment. By flying the various mission phases in simulation, the user is able to check in detail the computer control of the vehicle. The implementation of specific guidance control laws on the AGC often leads to problems with scaling, job sequencing, or timing. These problems may be uncovered in simulation and often result in redesign of some of the control algorithms. The exact closed loop simulation of the AGC interacting with the detailed simulation of the vehicle tests the adequacy of the steering and autopilot design in many ways that are not possible through analysis alone. In the final stages of program development, the simulation may be used to generate long verification runs which demonstrate the full mission capability of the program in the guidance computer.

In the design and implementation of the Digital Simulator, compromises had to be made between the desire to model every feature in the spacecraft or flight environment, and the desire to produce a simple fast simulation. In general, an attempt was made to identify every factor which would interact with the on-board computer to control the environment properly. However, if every factor were modeled, the resulting simulation

would require excessive development time to create and debug, excessive learning time for the user to become competent, and excessive running time to execute a single simulation. Thus, the overall design concept was to develop the least complex model which will properly exercise the on-board program.

The Simulator, as it finally evolved, is a complex, highly flexible tool. A large number of options have been developed which enable users to apply the Simulator to such diverse tasks as high accuracy thrust misalignment parameter studies, or low accuracy, but high speed program sequencing problems. One of the main functions of this manual is to acquaint AGC programmers with the Simulator options available to them and to provide sufficient information to enable individual users to utilize the Simulator in the most efficient manner.

Running A Digital Simulation

The basic computer program being developed for a particular mission is assembled and stored on the disc file of the IBM 360, model 75 computer. The programs comprising the Digital Simulator are also stored in the disc file. An AGC programmer tests his portion of a mission program assembly by submitting a simulation run deck to the IBM 360, as illustrated in Fig. 2. A. 1. The user specifies in his input deck the specific revision of a mission assembly he wishes to exercise, as well as information required to initialize the AGC program and the flight environment, if any. He may wish to alter some of the coding in the AGC program through special requests to the simulator. In addition, the user normally specifies the initial contents of certain erasable memory locations and may set up instruction traces and erasable memory dumps for diagnostic purposes. The user also has the capability of dumping the state of the vehicle and of the flight environment if he desires. Thus, for example, the accuracy of the on-board navigation can be compared with the simulated position of the

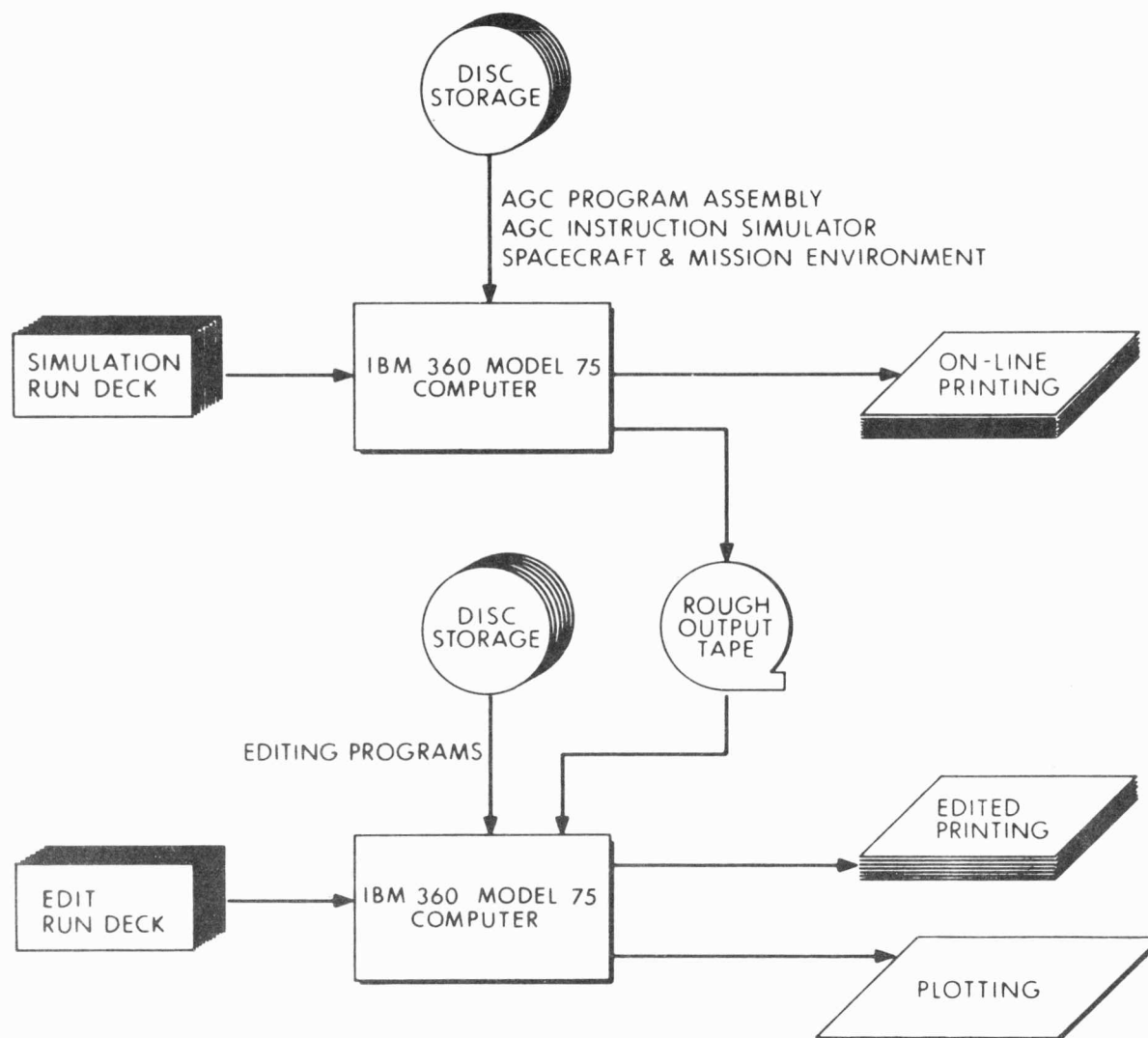


Figure 2. A.1 Elements of a Digital Simulation

spacecraft, or the precision of the autopilot - derived vehicle angular velocity can be compared with the simulated vehicle angular velocity. Additional special requests specify where in the coding to start the simulation and where or when to stop the simulation. A full list of definitions of these special requests is included in Chapter 3, Section C.

Input conditions must be specified for the flight environment. This flight environment includes hardware such as the IMU and CDU's as well as the vehicle (booster, CM, CSM, or LM) which the Guidance computer is controlling at the start of the run. Input variables for the environment include such parameters as position, velocity, attitude and angular velocity of the spacecraft, CDU angles, gyro bias errors, fuel loading, etc. The simulation of the booster, CM, CSM, or LM implies the use of a specific model for the spacecraft, but the user has the option of specifying certain off-nominal parameters related to spacecraft mass, engine performance, bending frequencies, thrust misalignments, etc., so as to probe the limits of performance of the guidance and control programs. On the other hand, he may wish to simplify the models and call for no bending, no fuel slosh, and no cross-coupled inertias so as to isolate a problem in the coding.

As an example of this modeling activity, consider the interaction of the digital autopilot and the LM: Since the digital autopilot samples the vehicle attitude and decides whether or not to fire reaction jets, one must investigate the stability of this controller in the presence of disturbances. The LM can have fuel sloshing and bending modes which perturb the read-out of attitude. Therefore, the strongest (in some sense) of these modes must be included in the LM simulation model. In other situations the sources of difficulty are not always obvious. The initial shift of fuel from one tank to another was assumed to be a small rapid transient. Later it was discovered that the initial imbalance could be large and the time-to-return to equilibrium was long. The adequacy of the center-of-gravity estimator in the computer had to be carefully re-examined with a new model added to the simulation.

A more complete discussion of the spacecraft models and input parameters is included in Section D of this chapter, as well as in Section 3. H.

Having set up his input deck, the user submits his job to the IBM 360. The simulation proceeds according to the requests in the input deck. The initial output from a simulation job is on-line printing of various quantities computed during the simulation. A larger accumulation of simulation output is written onto a rough output tape known as MARSROT. After the simulation run, the user may submit a job to edit the rough output tape and in this way obtain a selective print-out of key variables of interest. In addition, the user may compile plotting programs which more clearly display the evolution of variables with time.

Following diagnosis of the results of a run, the user may decide to reassemble the AGC program or to generate patches to an existing AGC program. Patches may be added to a simulation run deck as AGC special requests and the run deck may then be resubmitted for another test run.

2. A. 2 Structure of the Digital Simulator

The M. I. T. Apollo Digital Simulator consists of three major logically distinct sections:

Apollo Guidance Computer Instruction Simulator

Communicator

Environment

These elements of the Digital Simulator may, in turn, be utilized for three separate tasks: Bench Tests, Flight Tests, and Environment only runs. These three applications are shown in diagram form in Fig.2.A.2.

In the Bench Test mode, the AGC Instruction Simulator operates open loop without a Communicator or Environment. AGC instructions which refer to input-output counters or channels are not acted upon. The contents of the counter or channel remain unmodified, and the AGC simulation merely continues on to the next AGC instruction. If external stimulation of CDU, Optics, or PIPA counters or input-output channels is not required to debug an AGC program, then the Bench Test mode is useful; however this limited form of simulation is generally sufficient only in the initial phases of AGC program checkout.

The Flight Test mode is used if interaction between the AGC and its environment is desired. In this mode, the AGC Instruction Simulator operates with the Communicator and the Environment to produce a closed loop simulation. AGC output commands to the environment result in increments to the CDU, Optics, and PIPA counters, as well as possible input channel discretes. Control remains in the AGC Instruction Simulator until an AGC instruction refers to an input-output counter or channel. Every

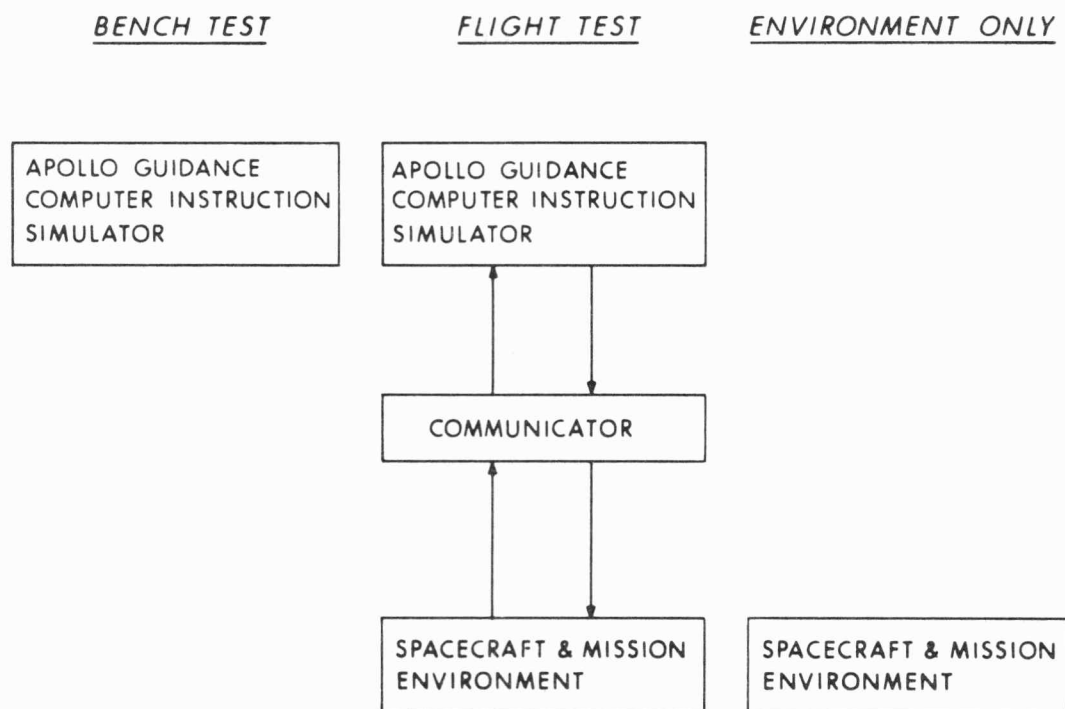


Figure 2. A. 2 Modes of Operation of Digital Simulation

such instruction triggers a transfer of control to the Communicator which determines whether an environment update is required to respond correctly to the AGC reference to the input-output register. If an environment update is required, control passes from the Communicator to the MAC environment, and the environment updates itself to the current AGC time. Following the environment update, the Communicator picks up AGC counter increments and input channel discretes from the environment and updates the AGC input registers. Control is then returned to the AGC Instruction Simulator which resumes the instruction by instruction simulation of the AGC program.

Environment only capability is available to users who wish to run the environment programs open loop, in the absence of an AGC or Communicator. This is of value primarily to members of the Digital Simulation Group for diagnostic purposes.

The AGC Instruction Simulator, the Communicator, and the Environment are described below in detail in Sections B, C, and D of this chapter. Input cards to the AGC Instruction Simulator and the Environment are fully described in Chapter 3.

2. B. AGC Instruction Simulator

The AGC Instruction Simulator is a large program which duplicates logically and arithmetically the action of the AGC computer in sequencing through a mission program. The AGC Instruction Simulator simulates only the AGC computer, not the hardware (such as the spacecraft) to which it is attached. The entire AGC memory, containing the mission program under test, together with its erasable memory and input-output channels is stored within the Simulator. The mission program is started at an instruction selected by the AGC programmer, and the Instruction Simulator executes the AGC program instruction by instruction, exactly duplicating the function of a real AGC. Execution of an AGC instruction causes the Instruction Simulator to call up a subroutine identified with the instruction, whose function is to duplicate bit for bit the internal operations performed by the real AGC in executing the same instruction. At the end of each instruction cycle, the state of the simulated AGC is virtually identical to the state of a real AGC executing the same instruction of the same mission program. As far as possible, the simulated contents of erasable memory, input-output channels and counters, interrupt activity, clock incrementation, etc., are the same as they would be in a real AGC. Truncation, round-off, overflow, arithmetic operations, timing, etc., exhibit the same behavior on the simulated AGC as they do in real life.

The simulated erasable memory differs somewhat from a real memory in that all registers which are not preloaded with specific values are set to a pseudo random number initially. If the AGC mission program is sensitive to the exact setting of non initialized erasable locations, then the behavior of the simulated AGC may differ from that of a real one.

In his use of the AGC Instruction Simulator, an AGC programmer has available a wide range of load, control, diagnostic and patch options which may be used to tailor a run to a particular objective.

First, the AGC programmer may load into the Instruction Simulator a specific revision of any AGC program which is currently available on disc storage of the IBM 360. Since this disc storage is periodically purged of AGC programs which are infrequently used, the AGC programmer should check the status of an AGC program he intends to run. An up to date list of available AGC programs will be maintained on the Digital Simulation Group bulletin board in the Assembly Control Room. The control cards used to load an AGC program into the Instruction Simulator are discussed in Section 3. B. 4.

After an AGC program has been loaded into the Simulator, an AGC programmer may control the running of his program with AGC special requests. These control cards are read by the Simulator Loader during initialization of the Instruction Simulator, and serve to implement certain diagnostic and run options which a user would like to apply to his run. Special requests are used primarily to start or stop the run at specified locations, to initialize the contents of erasable locations, to modify the contents of instructions in fixed memory, and to select a variety of diagnostic tools. A complete list of the currently available special requests is presented in Chapter 3, Section 3. C. 3 and 3. C. 4. An AGC programmer commonly uses these AGC special requests for the following purposes:

- Start simulation at a specified AGC instruction.

- Stop or abort the simulation at a specified AGC instruction.

- Exit when simulation exceeds a specified time.

- Modify AGC instructions or set contents of erasable locations.

- Request AGC erasable dumps or environment state variable dumps.

- Request AGC basic or Interpretive tracing.

- Request periodic snapshots of IBM 360 memory.

- Introduce time loss due to input counter jitter.

Advance the clock.

Initiate Go Sequences.

Initiate Environment Summary Printing.

Alter Environment vehicle model; fast-slow option.

Check contents of specific erasable registers to be sure they are unchanged during certain intervals.

Count the number of references to a specified location.

Initialize erasable to a specified background number.

Set initial contents of input-output channels.

Control telemetry rate.

Allow execution of instructions in erasable.

Introduce specified keystrokes or up telemetry words into AGC for diagnostic purposes.

Put AGC into standby mode.

Control AGC printout.

Print AGC clock.

Print specified message.

Some AGC special requests (such as start and stop options) exert overall control on a run, while other special requests (the majority) are activated only when a specified AGC instruction is executed by the Simulator, or when a specified AGC time has been reached. In addition it is possible to apply a condition clause to the special request which inhibits the execution of the request unless the specified condition is met. Chapter 3, Section C. 5 and C. 6 contains a complete description of the use of condition clauses attached to AGC special requests.

Through the use of AGC special requests to control, patch, and diagnose his run, an AGC programmer should be able to adapt the Instruction Simulator to provide any run data he might need for debugging purposes.

If the AGC program is being run in the Bench Test Mode, the simulation user need concern himself only with including the proper control cards in his simulation input deck (see Chapter 3, Section D) and with employing appropriate AGC special requests.

However, if the user is concerned with running his program in the Flight Test Mode, it is necessary to understand the functioning of the Communicator and the Environment (considered in Sections C and D of this Chapter) as well as the overall structure of the environment input deck (described in Chapter 3, Section F).

Regardless of the mode of operation, a knowledge of Rollbacks (Chapter 4) and Edits (Chapter 5) is essential.

2. C. Communicator

Interaction between the AGC and its environment is controlled by the Communicator, an assembly language program which acts as a buffer between the logically dissimilar AGC Instruction Simulator and the MAC coded Environment. The function of the Communicator is to decouple the AGC Simulator, which operates at a basic time step of one instruction cycle, from the Environment, which prefers to work with much longer time intervals. The Instruction Simulator executes by itself as many steps in the coding as possible. At some point, however, an instruction will call for a reading of some input variable, such as an accelerometer counter or a gimbal angle read-out. At this point control is transferred to the COMMUNICATOR, which attempts to provide the reading by a simple extrapolation of the previous environment state. If this can be done, control returns immediately to the Instruction Simulator, which proceeds to the next instruction in the AGC program. When the COMMUNICATOR does not have a valid extrapolation formula, a counter-read will cause a full environment update. The state of the environment is brought up to the AGC stopping time, and then control is returned to the Instruction Simulator. In this manner the microsecond time-stepping of the Instruction Simulator is decoupled from the larger natural time steps of the dynamic environment. In general, the Communicator updates the Environment over the longest possible time interval consistent with maintaining simulation accuracy.

2. D. Environment

2. D. 1 Structure of Environment

The Environment is a package of semi-autonomous MAC coded subroutines, controlled by an executive routine. Each subroutine simulates a relatively independent portion of the Environment, such as the IMU-CDU hardware, the active vehicle, (booster, CM, CSM, LM), the radar or optics system, the Astronaut, and so on up to the gravitational and atmospheric environment of the spacecraft. The Environment as a whole represents the spacecraft hardware and flight environment within which the AGC operates. As the AGC Simulator advances through its stored AGC Program, the AGC generates control stimuli, such as vehicle orientation commands, throttle commands, engine gimbal commands, etc., which are collected and filtered by the Communicator. The Environment programs close the control loop with the AGC by responding dynamically to these commands and feeding back data which the AGC may use in its subsequent control iteration.

Unmanned Mission Environment

The specific environment subprograms required in the simulation of a particular AGC mission program depend primarily on the spacecraft vehicles used in the flight and on whether the flight is manned or unmanned. For instance, the environment subroutines required to simulate an unmanned LM mission are shown in Fig. 2. A. 3. Three separate subroutines simulate the spacecraft during various mission phases: the Prelaunch subroutine (360 name: PRELUNCH) is used prior to lift-off and provides a simulation of the one-G environment encountered, for example, during alignment of the inertial measurement unit by gyro-compassing. Subroutine SATURN is

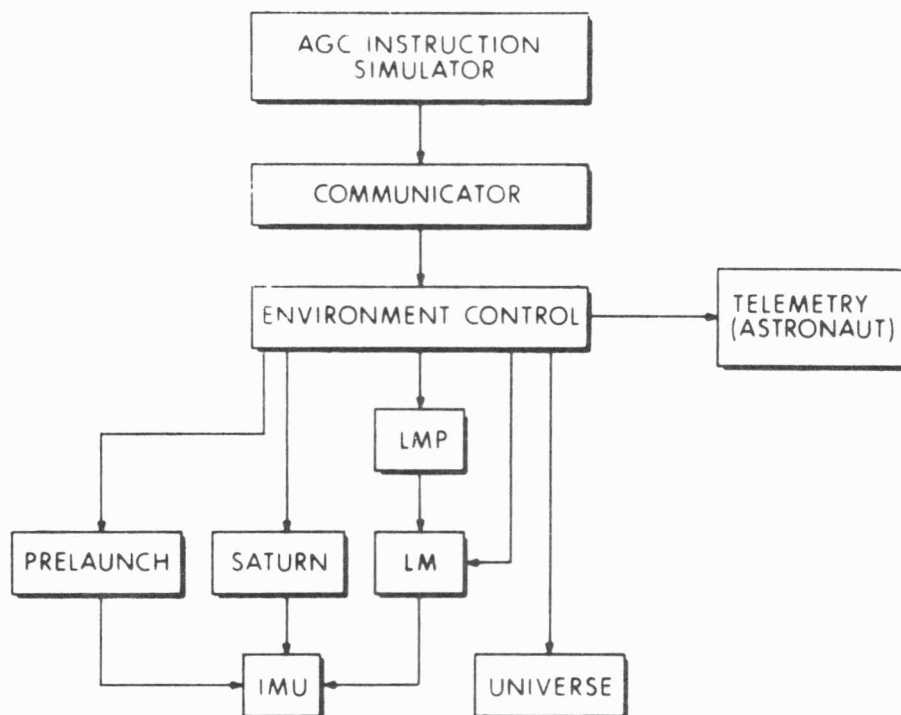


Figure 2. A.3 Subroutines for Simulating the Unmanned LM Mission

used after lift-off to provide the dynamic environment during boost into earth orbit. SATURN continues the simulation as long as the LM is attached to the booster. After separation from the booster, the simulation of the spacecraft dynamics is provided by the LM subroutine (360 name: LEM).

During any phase of flight, only one of the vehicle subroutines (PRELUNCH, SATURN, or LEM) is active. The primary function of the active vehicle subroutine is to simulate the attitude and acceleration of the navigation base to which the inertial measuring unit is attached. Throughout the simulation, attitude and integrated acceleration information from the active vehicle subroutine is transferred to subroutine IMU, which simulates the inertial measuring unit with its gyro stabilized platform and integrating accelerometers. The IMU subroutine resolves the integrated contact acceleration (non-gravitational acceleration) into platform coordinates and generates accelerometer counts which are sent to the AGC. The attitude of the navigation base relative to the attitude of the stable platform determines the gimbal angles, which are then encoded by the coupling data unit model and sent to the AGC.

The simulation of the vehicle trajectory is maintained by subroutine UNIVERSE after liftoff. The active dynamic subroutine (SATURN or LEM) accumulates increments to the position and velocity due to contact forces (thrust and drag). These increments are transferred to UNIVERSE which combines them with an integration of the free-fall trajectory.

In an unmanned LM flight, many switching functions normally controlled by an astronaut are automated by the LM mission programmer. The action of this device is simulated by subroutine LMP1. Coded words from the AGC, normally sent to a display, are received by LMP1 and are translated into switch actuations.

The simulation of telemetry updates to the AGC is set using special card inputs to the Astronaut Subroutine.

Program Environment Control coordinates the interaction of the various environment subroutines. In addition, Environment Control can print in a convenient format a set of variables which summarizes the state of the environment.

Manned Lunar Mission Environment

A somewhat more complex environment configuration, representing the subroutines required to simulate a manned lunar mission, is shown in Fig. 2. A. 4. In this configuration, five subroutines are required to simulate the spacecraft during the various phases of the mission. As before, the Prelaunch Subroutine (PRELUNCH) is the active dynamical subroutine prior to lift-off, and SATURN represents the vehicle from lift-off through boost to SIVB separation. After separation, the CSM program (360 name: CSMBLK2) maintains the attitude and acceleration of the combination of the CSM and LM. In later phases of flight, this same program may be used to simulate the CSM with LM detached. When the LM has been detached and it is desired to execute programs in the Lunar Module Guidance Computer (LGC), then simulation of spacecraft dynamics is maintained by subroutine LEM. The LEM routine may also be used in LGC simulations with the CM attached. The last phase of flight, reentry of the CM into the Earth's atmosphere, is simulated by the Entry program (360 name: LVCBLK2).

Selection of the active dynamic subroutine and the transition between active subroutines is supervised by the Environment Control program. It is the function of this program to ensure that all active environment subroutines are updated properly and that all programs receive the AGC commands relevant to them.

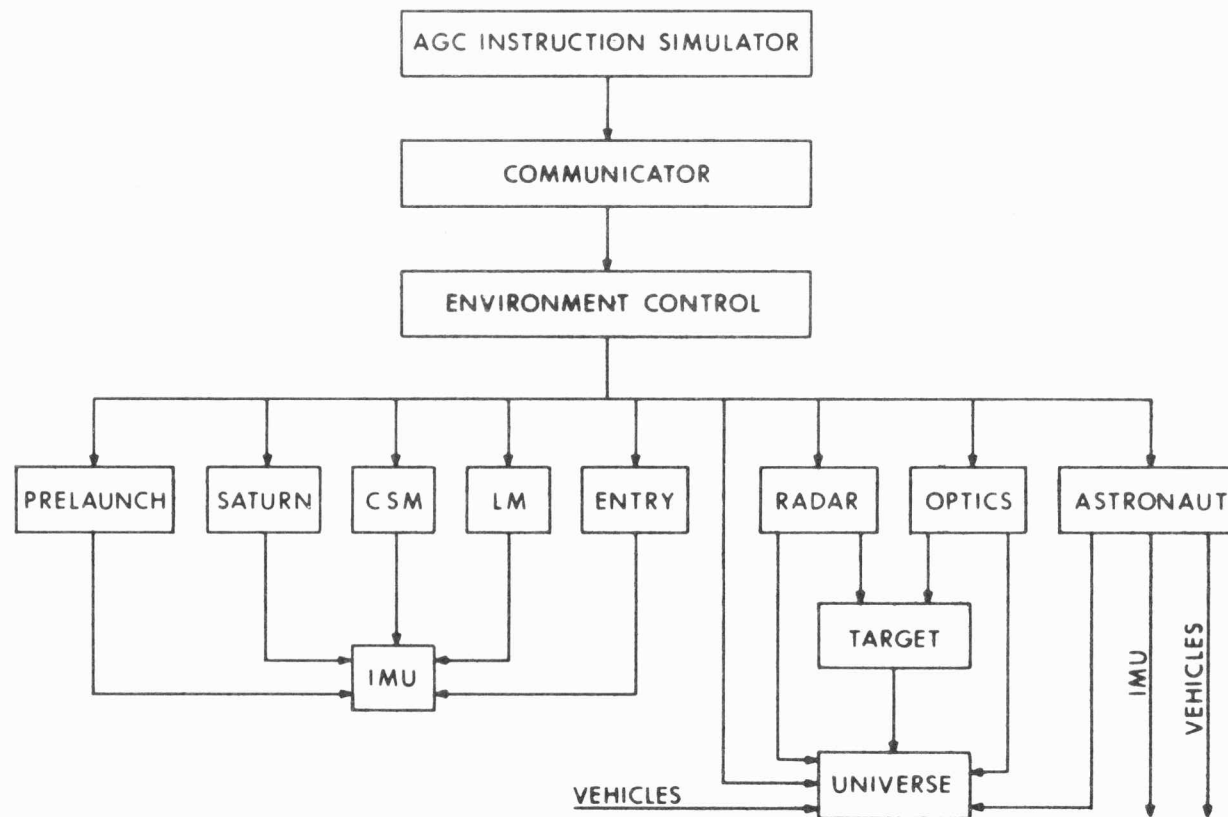


Figure 2. A. 4 Subroutines Used for Simulation of Manned Lunar Mission

The function of program IMU is the same for all phases of flight and for all active vehicle programs. IMU accepts attitude and integrated acceleration information from the active vehicle subroutine, and transforms these data into increments in accelerometer counts and gimbal angle counts which are sent to the AGC.

UNIVERSE is used by all active dynamical subroutines, except PRELUNCH, to maintain the vehicle trajectory.

In manned flights, the function of the Astronaut program (360 name: ASTRNAUT) is to simulate the active interface between the Astronaut and the flight hardware under his control, such as the Display and Keyboard (DSKY), the handcontroller, LM throttle, optics, and manual discretes. Under control of Astronaut input cards, (described in Section D. 3 of this Chapter, and in Chapter 3, Section H) the Astronaut activates individual programs in the AGC or LGC, through inputs to the DSKY, and controls the overall sequence of events in a mission. Typical simulated Astronaut actions include reacting to specific displays by typing inputs into the DSKY, performing star sightings, tracking landmarks, moving the LM handcontroller, commanding CM-SM separation, etc. A complete list of available Astronaut actions is tabulated in Chapter 3, Section H. Throughout a manned flight test, the AGC programmer determines the exact sequencing of events by including in his simulation input deck the appropriate sequence of Astronaut cards. In reacting to commanded actions, the Astronaut program interfaces directly with the active vehicle subroutine as well as with UNIVERSE and IMU.

Depending upon the requirements of a specific mission test flight, the programs RADAR, OPTICS, and TARGET may also be included as part of the Environment. Both rendezvous and landing radar tracking, optical star sighting, and tracking of a target vehicle are performed by the above programs, under command of the active AGC programs, or the direct control of the Astronaut program.

Diagram of Simulator

The full complement of programs used in the Digital Simulator is shown in Fig. 2. A. 5. The use of all the environment programs summarized in this figure is not normally required in any one mission simulation. The figure illustrates the use of a simulation initialization program, SIMSETUP, which is employed as a utility program to translate simplified, user oriented vehicle initial conditions into a detailed set of AGC special requests and environment input data cards. In every simulation run, whether initialized by SIMSETUP or not, these environment data cards are read by an environment input program, INPUTPRG, which processes the cards and controls the initialization of environment programs. During the initialization of the Environment, at the beginning of a simulation run, any Astronaut input cards present in the simulation input deck are read by the Astronaut input program and converted into digital code words for future processing by the Astronaut program.

The various environment programs which have been discussed above are listed in the following section, and then each routine is described in detail in Section D. 3.

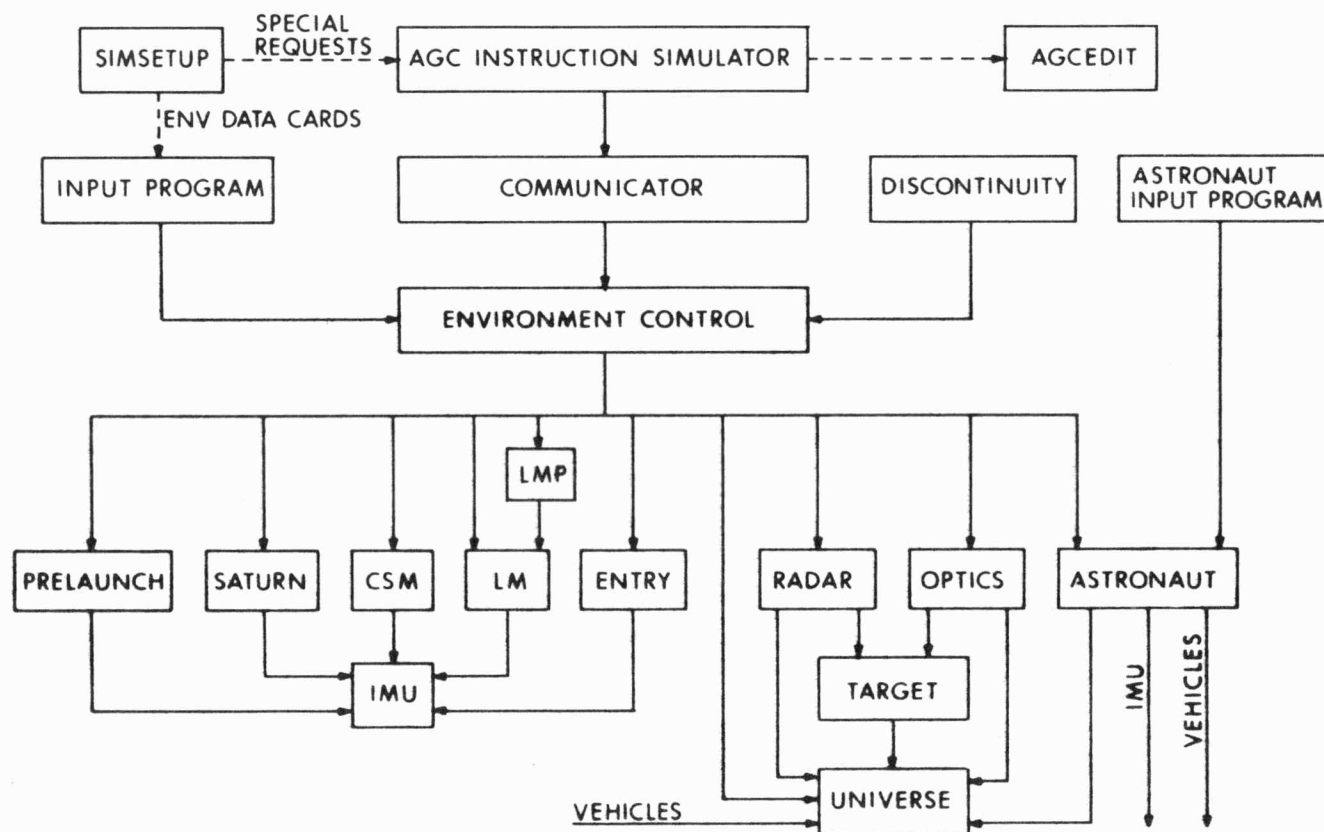


Figure 2. A. 5 Full Complement of Programs Used in Digital Simulation

2. D. 2 List of Environment Routines

The Block II MAC Programs which comprise the Environment Simulation are as follows:

ENVCNTRL	Environment Control Program Main control program for all the Environment Routines. 1) Controls the sequencing of subroutines within the Environment. 2) Generates Environment summary prints (ENVSUM Special Request).
INPUTPRG	Input Program A control program used only at the beginning of a run. 1) Reads all the Environment input cards. 2) Calls each Environment Routine for initialization.
UNIVERSE	Simulates the effects of the gravity and atmosphere of the Earth and the gravity of the Moon, as well as providing the positions of the Sun, Moon, and Stars. It also maintains the position and velocity of the spacecraft in flight at all times after liftoff.

PRELUNCH	<p>Prelaunch Program</p> <p>Simulates swaying vehicle on pad prior to liftoff. Maintains position and velocity of the spacecraft on the pad due to movement of the Earth or Moon.</p>
SATURN	<p>Simulates booster and spacecraft from liftoff to SIVB separation.</p> <p>SATURN uses Program ATTITUDE to provide booster orientation for use by SATURN autopilot.</p>
CSMBLK2	<p>Simulates the CSM from SIVB separation to entry, with or without attached LM.</p>
LEM	<p>Simulates the LM after SIVB separation, with or without attached CSM.</p>
LVCBLK2	<p>Simulates atmospheric entry of the command module from CM/SM separation to MACH 0.5.</p>
IMUBLK2	<p>Simulates the Block II IMU and CDU's.</p>
OPTICSB2	<p>Simulates the CSM's sextant and scanning telescope.</p>
RRADAR	<p>Simulates the LM Rendezvous Radar.</p>
LRADAR	<p>Simulates the LM Landing Radar.</p>
TARGET	<p>Simulates the position of the passive vehicle during rendezvous - to provide input for radar and optics.</p>

ASTRNAUT	<p>Astronaut Program</p> <p>Simulates astronaut actions such as keying in information through the DSKY, setting panel switches, and performing navigational sightings on stars or landmarks.</p> <p>It also simulates up-down telemetry.</p>
LMP1	<p>Simulates the LM Mission Programmer (Mechanical Man).</p>
DISCONT	<p>Discontinuity Program.</p> <p>Test routine used to change individual bits in AGC input channels.</p>
SIMSETUP	<p>Generates initial AGC erasable settings and Environment input data cards from simple, user oriented, initial conditions.</p>
ASTXMCI	<p>Astronaut Input Program.</p> <p>Reads Astronaut cards and translates them into decimal code words which are stored in the data file.</p>

List of Environment Data Files

The environment programs listed above are intended to be generally applicable to the simulation of any Block II mission. That is, the programs are mission independent. In order to simulate specific flights, the environment programs read files of data which contain those parameters that are mission dependent. These files, containing mission specific data required for the various flights, are read by the environment routines during the initialization of the Simulator. See Section 2. D. 4 of this chapter for a description of the initialization process.

The various environment data files and the mission flights with which they are identified are the following:

<u>Flight</u>	<u>File Name</u>	<u>Use</u>
SUNDANCE	LEM3. NOMINAL	LM initialization
COLOSSUS	FILE504. SATURN	Saturn initialization
	FILE504. CSM	CSM initialization
	FILE504. LVCBLK2	Entry initialization
	LEM3. NOMINAL	LM initialization
SUNDISK	SATURN. FILE205	Saturn initialization
	CSMBLK2. FILE205	CSM initialization
	LVCBLK2. FILE205	Entry initialization
SUNBURST	SATURN. DATA206	Saturn initialization
	206COEAR. LYBY40	Saturn cutoff early by 40 seconds
	206COEAR. LYBY100	Saturn cutoff early by 100 seconds
	LEM206. NOMINAL	LM initialization
All Missions	SIMEPHEM. F6667	Universe ephemeris from July 1, 1966 to July 31, 1967
	UNIV. EPHM6768	Universe ephemeris from July 1, 1967 to July 31, 1968
	UNIV. EPHM6869	Universe ephemeris from July 1, 1968 to July 31, 1969

There are other special purpose files with off nominal values, created to fulfill the particular needs of certain users. Creating new special files is a simple and routine matter; programmers who desire new files for off nominal performance should check with the simulation staff.

2. D. 3 Descriptions of Environment Routines

In this Section, the Environment Routines listed in Section 2. D. 2 are described in detail. Particular emphasis is placed on a tabulation of the models used to simulate dynamical characteristics of the vehicle hardware and flight environment. These models are described in sufficient detail to enable a simulation user to evaluate the adequacy of the model for his specific application. Model specification options which enable a user to change certain aspects of the model to suit his own requirements, are also described. More detailed documentation of the vehicle and flight environment math models may be found in the Chapter 6 section of the 205 and 206 GSOP's, as well as in MIT-IL Report E-2146, "The Digital Simulation for the Verification of Program SUNBURST (Unmanned LM, AS-206)." Programmers with a need for more specific documentation than is contained in the references cited should refer to the environment program listings.

Environment Control Program ENVCNTRL

Program ENVCNTRL acts as a supervisory routine for programs in the MAC environment. Its chief functions are: 1) to control the sequencing of calls to various programs in the environment, 2) to ensure that all programs receive AGC commands directed to them, 3) to control the transition between active vehicle subroutines (such as the transition between PRELUNCH and SATURN at liftoff, or the transition between SATURN and the spacecraft at SIVB separation), and 4) to generate on-line summary printout under the control of the AGC special request ENVSUM. The summary print is a three line summation of the current state of the spacecraft and of the flight environment. Minor functions of ENVCNTRL are to process requested returns from all Environment routines (returns requested at a specified future AGC time), and to compile statistics on the real time consumed by each element of the simulation.

ENVCNTRL also contains logic which enables the environment routines to run independently of the Instruction Simulator and the Communicator while operating in the Environment-only simulation mode. This mode of operation is useful if it is desired to observe the open loop response of the environment to control commands read in from cards.

Input Program - INPUTPRG

An environment initialization program by the name of INPUTPRG is used at the beginning of a simulation run to read all the environment input cards included in the simulation run deck, and to initialize all active environment programs. The environment input cards specify the initial state of individual environment programs. Each environment program requires a specific set of input cards, the contents of which are defined for each program in Chapter 3, Section H.

INPUTPRG "peeks" at each environment card before the card is read to determine whether the next card is a MAC data card, an Astronaut card, or an end of file card (*, /*, or //). MAC data cards are interpreted as environment initialization cards and their contents are filed away in specific blocks of the data file for later processing by the particular environment program which requires the initialization data. Astronaut cards, identified by an A in column 1, trigger a call to the Astronaut input program, ASTXMCI, which reads all Astronaut cards and translates their contents into digital code words which are written in the data file for subsequent processing by the Astronaut program. An end of file card terminates reading of input cards.

Following the reading and filing of environment input cards, INPUTPRG individually calls all environment programs which have initialization data filed away for their use. Thus, the inclusion of environment input cards for a specific program results in the initialization of that program. See Section 2. D. 4 for a discussion of environment initialization.

UNIVERSE Simulation Model

The subroutine UNIVERSE provides information pertaining to the natural environment of each Apollo mission. The motion of the Earth and Moon is tabulated in the ephemeris data file so that UNIVERSE can calculate the transformation between Earth fixed or Moon fixed coordinates and reference inertial coordinates. UNIVERSE also can calculate the position and velocity of the Sun, as well as the position of stars. For the Earth, UNIVERSE has a gravitational model which includes non-spherical Earth effects and the perturbations of the Moon and Sun. Similarly, the lunar gravitational model includes perturbations of the Earth and Sun. Atmospheric density information is available and is used to compute dynamic pressure for the booster simulation, and for Entry.

The reference inertial coordinate system used in the digital simulation is defined by the intersection of the mean equatorial plane and the mean orbit of the Earth (the ecliptic) at the nearest beginning of the Besselian year. The ecliptic is considered to be an inertial plane. Rectangular coordinates are defined so that the X axis is along the ascending node of the ecliptic on the equator (the equinox), the Z axis is along the mean North pole of the Earth or Moon, and the Y axis completes a right handed set. The origin of the coordinate system may be specified either at the center-of-mass of the Earth or the center-of-mass of the Moon.

Some advantages of using this reference inertial coordinate system, which is referred to the mean equinox at the nearest beginning of the year, are:

- 1) The precession of the pole must be considered in transforming between Earth fixed or Moon fixed coordinates and reference coordinates. If the reference equinox is near the mean equinox of date, then this transformation is small and many simplifications can be made. Precession and nutation can be described using small angle approximations.
- 2) In computing the gravitational perturbations due to non-sphericity of the primary body, one may neglect entirely the precession of the angular momentum vector. For the Earth, the maximum deviation of the pole

-4

due to precession is about $10''$, which gives an error of about 0.5×10^{-4} times the non-spherical perturbations, assuming that the reference equinox is within 6 months of the mean equinox of the given day. If the reference coordinate system were based on an epoch as much as 10 years away, then the error would be about 10^{-3} times the non-spherical perturbations. The error in the second order perturbation would be comparable to the magnitude of the third and fourth order perturbations.

3) The coordinate system referred to the beginning of the Besselian year is commonly used and is one in which much of the information from the Nautical Almanac Office is available. The disadvantage of using this coordinate system is that the standardized NASA ephemeris for the Moon and Sun is available on magnetic tape from the Jet Propulsion Laboratory. The information on these tapes is referenced to the 1950.0 coordinates.

An Earth fixed coordinate system is also used in program UNIVERSE. This system is defined with the Z axis along the Earth's angular rotation vector, the X axis in the meridian of Greenwich and the Y axis completing a right handed set.

Information given in terms of geodetic or selenographic latitude is assumed to be referenced to the Fischer ellipsoid of 1960 for the Earth, or to the 1735.4 Kilometer reference sphere for the Moon. The procedure for converting longitude, geodetic latitude, and altitude into the rectangular coordinates for a point is exactly correct only if altitude is given as the distance along the geocentric radius vector to the Fischer Ellipsoid, and latitude is the latitude of the projection of the point onto the ellipsoid along the geocentric radius vector. The procedure is only approximately correct if altitude is given as the distance along the normal to the ellipsoid and latitude is the latitude of the projection of the point onto the ellipsoid along the normal to the ellipsoid.

The transformation from Universe inertial to Moon fixed coordinates is assumed to be governed by the laws of Cassini, perturbed by a simplified linear time dependent physical libration. Cassini's laws (1721) for the motion of the Moon are the following:

1. The Moon rotates Eastward on an axis fixed in its body, with a constant angular velocity, and with a period of rotation exactly the same as that of the Moon's sidereal revolution about the Earth.
2. The inclination of the Moon's axis of rotation to the ecliptic (plane of Earth's rotation about the Sun) is constant. (assumed to be $1^{\circ} 32'.1$)
3. The lunar orbital plane, the ecliptic, and the lunar equator intersect in a single line. The ecliptic lies between the lunar orbital plane and the lunar equator.

Relative to a line between the centers of the Earth and Moon, a fixed point on the lunar surface will appear to move, due to optical and physical libration. Optical libration in latitude and longitude is inherent in Cassini's laws and is related to the geometry of the Earth - Moon system, not to physical effects. If the Moon's motion was regular, then there would be only optical libration.

Optical libration in longitude results from the fact that the Moon rotates at a constant rate about its own axis, but revolves in an elliptical orbit about the Earth, with a variable angular velocity. Optical libration in latitude is also present since the axis of rotation of the Moon is not perpendicular to the plane of the Moon's orbit about the Earth, and thus its axis of rotation is alternately tipped toward and away from the Earth. Optical librations result in maximum angular displacements of a point on the lunar surface of $\pm 7.9^{\circ}$ in longitude and $\pm 6.8^{\circ}$ in latitude.

Physical libration represents a deviation from the laws of Cassini, and is observable as small oscillations performed by the Moon about its mean position. This physical libration results from differences between the principal moments of inertia of the Moon. In the UNIVERSE program, the very complicated classical expressions describing physical libration are reduced to a linear, time dependent transformation which is appended to the inertial lunar orientation derived from Cassini's laws.

The free-fall trajectory of the vehicle is governed by the simulated gravitational field. The components of this field include the spherical Earth attraction, the fourth order non-spherical Earth perturbations, as well as the perturbations of the Sun and the Moon. The mass distribution of the Earth is assumed to be symmetric about the rotational axis, hence there is no dependence on longitude in the potential equation. In addition, there is no first-order perturbation because the origin of the coordinate system is at the center-of-mass of the Earth.

The simulation of a free-fall trajectory is maintained by UNIVERSE by means of a direct numerical integration of the gravitational acceleration. During periods of powered flight, the active dynamic subroutine (SATURN, CSMBLK2, LVCBLK2 or LEM) integrates the contact acceleration due to thrust or drag. Periodically these increments to position and velocity are transmitted to UNIVERSE for combination with a free-fall trajectory. In this manner the small natural time steps of the vehicle are decoupled from the longer natural time steps required for trajectory maintenance. An error is introduced by this technique, to the extent that the gravitation along the free-fall trajectory is different from the gravitation along the powered trajectory. By incorporating the thrusting increments every few seconds, this possible source of error is kept negligibly small.

During the boost and reentry phases of the simulation, UNIVERSE also maintains a model of the Earth's atmosphere, based on the 1962 U. S. Standard Atmosphere. Atmospheric perturbations due to winds are not included in the model. Values of molecular scale temperature and density are tabulated for 14 reference altitudes ranging from 0 to 160,000 meters, and the temperature gradient between reference altitudes is also tabulated. Using these reference data, atmospheric temperature and density at intermediate altitudes can be calculated and used in turn to generate dynamic pressure and Mach number which determine the aerodynamic forces acting on the vehicle.

Throughout phases of powered flight and Entry, the primary function of UNIVERSE is to maintain vehicle position and velocity by combining increments of position and velocity with a free-fall trajectory. However, the UNIVERSE program has the additional capability of functioning as a utility routine which responds to calls from other routines for information on star vectors and coordinate transformations. In this category of ancillary operations which UNIVERSE may be called upon to perform are the following:

- a) Compute position of Moon and Sun.
- b) Compute position and velocity of Moon and Sun.
- c) Compute principal gravitational attraction, non-spherical perturbations, and perturbation of the Sun for a given position vector.
- d) Transform vector from UNIVERSE reference frame to coordinates fixed in Earth or Moon and vice versa.
- e) Compute unit vector toward a given star.
- f) Compute the angle between a star and a horizon.
- g) Compute the angle between a star and a specified landmark.
- h) Find two stars closest to a specified unit vector.

The exact calling sequences required for the above UNIVERSE operations may be found in the preface to the UNIVERSE program listing.

Prelaunch Simulation Model

Subroutine PRELUNCH provides a simulation of the navigation base attitude and acceleration during the period of time before the Saturn booster lifts off the pad, or before the LM lifts off the surface of the Moon. The LM Prelaunch environment is discussed separately below. For Earth prelaunch, the simulated vehicle motion is the combination of the rotation of the Earth and the swaying of the booster on the launch pad. The swaying motion was included to test the ability of the gyrocompassing program to align the platform in the presence of disturbances.

Since the position and velocity during prelaunch operations is definitely known, it is not appropriate to simulate velocity and position as integrals of a simulated contact acceleration plus a gravitational acceleration. Any numerical errors would cause the vehicle to drift away from the launch pad. Instead, velocity, position, gravity, and the time-integral of gravity at the launch site are all computed as analytic functions of time.

The simulation user must specify the latitude, longitude, and altitude of the navigation base, and the time that lift-off will occur. The UNIVERSE program converts this information to a position vector \underline{r}_i for the nav-base relative to the center of the Earth at lift-off in reference inertial coordinates. UNIVERSE also provides the gravitation vector at the nav-base at lift-off, including non-spherical Earth effects and the attractions of the Moon and Sun. PRELUNCH computes the launch site lift-off gravity vector $\underline{\bar{g}}_i$ by adding to the gravitational vector the effect of centripetal acceleration due to the rotation of the Earth. The simulation model now assumes that this gravity vector is imbedded in the Earth and therefore can be simulated by an analytic rotation at Earth rate about the Earth's angular velocity vector. This assumption means that the Moon and Sun gravitational accelerations are also rotated fixed to the Earth - a small simulation error of the order of tidal effects. An additional small error is introduced by the assumption that the angular velocity vector of the Earth is exactly equal to its direction and magnitude at lift-off. This is in error by a small amount of Earth precession and nutation. Note that the angular velocity vector of the Earth is not coincident with the Universe Z coordinate direction.

Superimposed upon the deterministic model for gravity and Earth rotation is a stochastic model for the launch vehicle swaying in the wind on the launch pad. The swaying motion is assumed to have no preferred direction, so it is modeled by two independent random processes in the two horizontal directions. The swaying booster is modeled as a damped second-order linear oscillator having a natural frequency and damping ratio. The force of the wind on the side of the booster is expressed as an exponentially correlated Gaussian process. The actual implementation in the PRELUNCH program of the third order stochastic process relating the white noise driving function to the vehicle displacement uses state space techniques and a finite piecewise constant random forcing function.

A more sophisticated model would recognize that the swaying can be excited by vortex shedding. This causes a periodic lifting force at right angles to the direction of the wind. The frequency of this excitation is related to the wind velocity and the booster diameter. This effect should be modeled as a force having a preferred direction and having a strong frequency. However, we believe that our simpler model is adequate to test the interaction of the vehicle motion with the gyrocompassing program active in the AGC.

At the time of the user specified Guidance Reference Release, the PRELUNCH program sets bit 6, channel 30, of the CMC and at the specified lift-off time, PRELUNCH sets bit 5, channel 30 of the CMC as well as transferring control to SATURN. The above discretes are not sent to the LGC.

If it is desired to run a SIMFLIGHT checkout simulation on the Launch Pad, it is possible to inhibit the transfer of control to SATURN at lift-off by utilizing the SIMFLITE input option on the INPUTPRG control card. See listing of input cards to INPUTPRG in Chapter 3, Section H.

The PRELUNCH program is also used to simulate the prelaunch phase prior to LM lift-off from the lunar surface. There are, however, several significant differences between the use of PRELUNCH for an Earth environment and its use for a lunar environment. For lunar prelaunch, the Guidance Reference Release and Liftoff input parameters are ignored, and lift-off takes place when the LM Ascent Engine ignites. At present, the lunar gravity vector at the launch site is assumed to be imbedded in the Moon, and therefore may be simulated by an analytic rotation about the lunar angular velocity vector. Over time periods as short as a day this analytic rotation introduces errors in latitude and longitude due to short period lunar physical librations. (See UNIVERSE description). Methods of reducing this error are being investigated.

Saturn Simulation Model

The digital simulation program SATURN is a six degree of freedom rigid body dynamic model of the Saturn booster. Fuel slosh and bending are not simulated. The electronic signal shaping filters in the Flight Control Computer and models of the engine dynamics are added to the vehicle model at the user's option.

The Saturn flight trajectory is determined during each simulation by the Saturn steering subroutine ATTITUDE and by a preloaded data file. The file contains information about the different time fixed events as well as the different masses, mass flow rates, thrust levels, inertia and center of gravity versus mass curvefits, times of vehicle staging, guidance equation presettings, engine reference locations, time dependent gains for electronic filters, and ISP's necessary for the dynamic simulation.

SATURN receives from ATTITUDE the matrix describing the desired boost vehicle orientation. Error signals in boost vehicle pitch, yaw, and roll are extracted. After these error signals are limited and filtered, they determine the thrust direction of the gimballed engines and activate the Saturn RCS jet autopilot if appropriate.

Thrust is a polynomial function of time during the atmospheric phase of the flight, to simulate the change in thrust due to change in atmospheric pressure. Axial drag is computed by SATURN as a function of Mach number and dynamic pressure from the atmosphere model in subroutine UNIVERSE. Atmospheric effects are assumed to be negligible after first stage cutoff and thrust changes are assumed to be step functions. The effect of the normal component of drag on the trajectory is assumed to be negligible.

The standard equations of motion by Euler determine inertial angular acceleration from the resultant torques and mass dependent inertias. Cross coupled inertias are not simulated.

In SATURN's capacity as the active dynamic vehicle in a simulation, SATURN updates the time-integral of specific force and Apollo vehicle attitude in reference inertial coordinates as would be sensed at the location of the Apollo accelerometers and gyros simulated in IMUBLK2. In SATURN's capacity as a self steered dynamic vehicle, SATURN updates the same quantities as they would be sensed by the Saturn instrument unit (ten meters below on the Saturn) simulated by ATTITUDE. This independent calculation allows simulation of errors in the Saturn platform for Apollo take over studies.

Simulation of each individual jet firing in the SIVB RCS jet autopilot model consumes more computer time than is justified by the simulation requirements. The effect of jet firings is grouped so that thrust is a linear saturated function of vehicle attitude error with a one degree attitude error deadband. SIVB RCS jets control vehicle roll during parking orbit insertion burn, and control roll, pitch, and yaw during SIVB coast.

The ATTITUDE subroutine simulates in MAC language a generalized version of the Iterative Steering Mode (IGM) equations coded in the Launch Vehicle Digital Computer (LVDC) on board the SIVB. Subroutine ATTITUDE interfaces only with subroutine SATURN.

ATTITUDE operates in three modes to construct the desired attitude matrix for SATURN. During the atmospheric open loop burn, desired attitude is derived from pitch polynomials read from the data file. During closed loop guided burn, ATTITUDE solves the IGM equations to provide the desired attitude matrix. During orbital or trans-lunar coast, desired attitude is commanded according to the particular mission objectives (cold soak, wings level or whatever).

Initial state of the vehicle is taken from PRELUNCH (or the UNIVERSE input cards if the simulation starts in SATURN). Thereafter ATTITUDE estimates Saturn state for use in steering by integrating a free fall trajectory in short time steps and adding the effect of contact acceleration as calculated by SATURN. Except for periods of SIVB burn when the IGM equations are guiding, SATURN may be initialized at any point along its standard trajectory or in orbital coast.

The SATURN user specified input data card words CSTMODEL and THRMODEL respectively allow the user to alter some characteristics of the SIVB model used during orbital coast and SIVB thrusting. See SATURN input card listing, Chapter 3, Section H.

Through the use of CSTMODEL the user has the option of saving simulation computer time while simulating long SIVB coast periods by reducing the six degree of freedom model to a point mass, very high speed SIVB model.

The point mass model skips all dynamic equations in SATURN and holds inertial attitude constant; subroutine ATTITUDE is not called. Simulation of undamped tumbling is also available when SIVB is coasting.

When the AGC generates steering attitude error signals for an SIVB burn, a more detailed model of the SIVB may be desired. The user may specify via THRMODEL a detailed model of the electronic error signal filters in the Saturn Flight Control Computer and a model of the dynamics of the SIVB gimbaled engine.

Command and Service Module Simulation Model

The Command and Service Module is represented in the Simulation by the Environment subroutine CSMBLK2. Inputs to the CSMBLK2 program are generated by the AGC or the ASTRONAUT and include RCS jet firing commands, SPS Engine on-off commands, SPS engine gimbal commands, and various moding discretes. The CSMBLK2 program responds dynamically to these input stimuli and produces updated values of vehicle orientation, body rates, and acceleration which are fed to the IMU program to produce CDU and PIPA increments. Periodically, increments in position and velocity, as computed from integrals of contact acceleration are transmitted to UNIVERSE which maintains vehicle position and velocity.

Various aspects of the CSM model as programmed in the CSMBLK2 subroutine, such as vehicle mass properties, RCS jet representation, SPS thrust and actuator models, fuel slosh, and bending dynamics are discussed below. A more complete description of the CSM model employed in the Digital Simulation may be found in the 205 GSOP, Section 6. 4.

The CSM mass, center of gravity, and inertia are initialized for the specific vehicle configuration and propellant loading under test. Prior to the simulation, the masses, CG's and inertias of all vehicle components (including propellants) are appropriately combined. During the simulated mission, the vehicle mass is decremented by propellant usage, and vehicle CG and inertia are recalculated.

The SM Reaction Control System contains 16 thrust chambers in 4 separate and independent quads each containing 4 thrust chambers. Each of the 4 RCS jet quads are located at the same point on the X-axis. The quads are 90 degrees apart in the Y-Z plane and they are offset from alignment with the Y and Z axes by a positive rotation about the +X axis in

accordance with the right hand rule. The engine nozzles are positioned (canted) such that the exhaust gases are directed away from the spacecraft surface. The distance from the X axis to the RCS jet centerline intersection has been averaged and is modeled as a constant for each quad.

The individual RCS thruster is modeled as a constant thrust motor. The propellant usage of each RCS thruster is represented by a constant mass flow rate when the jet is fired, plus a propellant penalty for each jet firing.

The buildup and tailoff of thrust are modeled by instantaneous changes in thrust shifted in time from the electrical command. The shift is different for the application and removal of thrust. The amount of the shift is calculated to give a transient impulse typical of a real thruster.

During the simulated mission, the RCS propellant center of gravity is held as a constant. The propellant inertia is found by a linear interpolation between two tabulated points; one of which corresponds to zero mass, zero inertia, while the other corresponds to an approximate full load.

The SPS engine is considered to be a constant thrust, constant flow rate motor. As in the case of RCS thrusters, the buildup and tailoff of thrust are represented by instantaneous changes in thrust, shifted in time from the electrical command, with a different shift for the application and removal of thrust. The magnitude of the shift is calculated to approximate the transient impulse typical of a real engine.

The SM SPS propellant mass properties are stored for three different propellant loadings. During the simulated mission, the propellant center of gravity is found by a linear interpolation between two of the tabulated points; and the inertia is found by a cubic interpolation between three tabulated points.

The SPS engine is mounted on a gimbal system which permits angular motion in pitch and yaw. Engine angular position is controlled by closed loop servo-mechanisms in both axes. The servo loops include an amplifier (modeled as a constant gain, no lags), a clutch (modeled as a constant gain - torque/current ratio, no lags), position feedback and rate feed back. For small signal inputs, the operation of the servos is assumed to be linear. The total torque applied to the SPS engine includes clutch torque, angular velocity dependent torques caused by damping within the actuator and exhaust jet damping, angular position dependent torques caused by hose stiffness and snubber spring stiffness (if the angular deflection exceeds the position limit), and reaction torques caused by the acceleration of the vehicle. The angular acceleration of the engine is the ratio of total torque inertia and is integrated twice to obtain angular deflection. Total inertia includes both engine inertia and actuator inertia.

The operation of the SPS engine gimbal servomechanisms is non-linear in several respects. The commanded angular position is limited within the digital to analog converter which provides the signal. (The D/A converters are located within the optics CDU's). The torque applied by the actuators is limited by the actuator motor speed and is modeled as a fixed gimbal rate limit. The engine angular position is limited by snubbers (stiff springs) beyond a limited range of motion. Angular position is measured with respect to the engine mount. The mount is offset in angular position with respect to vehicle axes in both pitch and yaw. Therefore, the angle between the thrust vector direction and the vehicle axis will be the sum of the engine angular position and the engine mount angular offset. Angular positions are measured as positive rotations of the engine about the vehicle Y and Z axes (e. g., a positive engine angular position in pitch will result in a negative pitch acceleration of the vehicle).

The simulation of spacecraft bending is based on a three dimensional modal analysis of the CSM with LM attached. Bending motions at significant

vehicle locations are described by a displacement vector and an incremental rotation vector for each mode. The three lowest frequency modes are simulated. The first two modes are essentially "V" shaped, with bending occurring primarily at the docking tunnel. Since the bending simulation is fully three dimensional, pitch and yaw bending motions are coupled: i.e., excitation of bending motions in the pitch plane will give rise to subsequent yaw bending motions. Bending can be excited by RCS jet firings, SPS gimbal motions (with or without SPS thrust on), or by interaction with slosh motions. Initial bending amplitude in each of the three modes may be specified via CSMBLK2 input cards.

Slosh motions of the propellant in the SPS tanks is simulated as a linear spring mass system. Each tank in which sloshing occurs is characterized by a slosh mass, a slosh frequency and a slosh mass attachment point. Slosh mass motion is constrained to the plane defined by the vehicle Y and Z axes. Sloshing occurs either in the two storage tanks (no slosh in the full sump tanks) or in the two sump tanks (no slosh in the empty storage tanks). It is assumed, for each tank, that slosh mass is constant, slosh frequency is inversely proportional to the square root of vehicle mass, and the X coordinate of the slosh attachment point varies linearly with propellant mass. Slosh damping ratios are held constant.

In addition to the ability to specify vehicle position, velocity, orientation and body rates at run time, a considerable amount of flexibility has been introduced into the definition of the CSM model to enable a user to perform a wide variety of diagnostic runs and parameter studies. The options available to the user, and the CSMBLK2 input cards affected are tabulated in Chapter 3, Section H. Those options are also listed here for reference:

1. Vehicle configuration options.
 - a. Command and Service Module.
 - b. CSM with full docked LM.

2. Slosh and bending options.
 - a. Slosh and bending not simulated.
 - b. Slosh is simulated but not bending.
 - c. Slosh and bending are both simulated.
3. Cross Coupling Effects.
 - a. Engine gimbal offset in pitch and yaw, and Y and Z components of CG are zeroed.
 - b. Option a above, and in addition products of inertia are set to zero.
 - c. Options a and b plus Y and Z components of slosh moment arm vector are set to zero.
 - d. Option a above, and in addition engine gimbal angles are set to the commanded values after each time step.
 - e. Options b and d combined.
 - f. Options c and d combined.
4. Pitch plane motion.

If desired, pitch plane motion is prevented by inhibiting the simulation of pitch actuator dynamics, and ignoring inertia matrix elements other than I_{ZZ} .
5. Vehicle Weight options.
 - a. Specify total mass of vehicle or,
 - b. SPS propellant loading. (Percent)

6. Modifications to nominal vehicle.
 - a. Set diagonal inertia matrix elements.
 - b. Set vehicle CG vector.
 - c. Set thrust misalignment torque.
 - d. Set thrust angular misalignment.
 - e. Set slosh amplitude at SPS ignition.
 - f. Set initial bending amplitude.
 - g. Fail specified RCS jet at selected time.
 - h. Change nozzle trim command at a specified time.
 - i. Change SPS thrust at a specified time.

The primary method of selecting various options of the CSM model is via input cards, as above, but it is also possible to make direct changes to vehicle physical data by modifying the CSMBLK2 Parameter File. This Parameter File contains all mission dependant vehicle data and is read at the beginning of each run. (See Section 2.D. 4). Individual datum in the Parameter File may be modified at run time, on the entire File may be regenerated for particular off-nominal CSM vehicles.

In addition to the above modifications to the nominal vehicle, there also exists the capability within the CSMBLK2 program to bypass some of the detailed models in favor of an abbreviated vehicle simulation, for the purpose of conserving digital simulation running time. This ability is referred to as the "FAST VEHICLE OPTION". When this option is selected, the CSMBLK2 program is altered in the following manner:

- a. The dynamic simulation of the nozzle actuators are bypassed. After the SPS engine actuators have been enabled, their position is immediately set equal to subsequent AGC commands.

- b. Simulation of slosh, bending and the effect of nozzle motion on vehicle (tail wags dog effects) is eliminated.
- c. The maximum internal integration time step of the CSMBLK2 program is increased.
- d. The time validity of the CDU counter extrapolations is increased, and they are not invalidated by certain AGC outbits, namely pulse trains to the nozzle actuators.

The fast option can be selected by using the AGC special request ENVMODE. See Section 3. C for a description of this special request. Its use is recommended for mission programmers whose main interest is debugging code, checking erasable conflicts, testing program sequencing, or testing astronaut interaction. A considerable savings in computer time will result. Computer test runs which depend on the detailed dynamical interaction between the AGC and the Environment, such as Digital Autopilot analyses, should not use the Fast mode.

Lunar Module Simulation Model

The program LEM is responsible for simulating the Lunar Module in all of its modes of flight - both ascent and descent as well as CSM attached. For Lunar Module only test flights, the LEM program becomes active at separation between the Saturn SIVB stage and the LM, while for Lunar Mission flights, LEM becomes active at separation between the CSM and LM. During unmanned flights, separation sequencing is handed by the LM Mission Programmer environment subroutine (LMP1), and in manned flights this sequencing is controlled by the astronaut program (ASTRNAUT).

The LEM subroutine is mainly concerned with simulating the detailed dynamical response of the LM vehicle to orientation and thrusting commands provided by the LGC.

LEM accepts input commands generated by the LGC or the astronaut, such as RCS thrust commands, throttle pulse trains, DPS engine trim gimbal signals, or panel switch changes, and computes, as an output, vehicle attitude and thrust acceleration of the navigation base. LEM interacts with the UNIVERSE subroutine by accumulating perturbations in position and velocity due to contact accelerations, such as engine thrust, and periodically transmits the increments to UNIVERSE. These trajectory perturbations are added to the free fall trajectory by UNIVERSE to produce updated vehicle position and velocity. Final LM attitude, attitude rate, and integrated acceleration are fed to the IMUBLK2 program which resolves the integrated acceleration into platform axes and then computes the number of PIPA and CDU increments to be sent to the LGC.

The LM model contained in the LEM subroutine is discussed below with particular emphasis on the mass properties of the LM, the descent propulsion system, the ascent propulsion system, staging forces, fuel slosh,

fuel mass transfer, and model perturbation options available. A complete description of the LEM subroutine, as it was used in 206 verification runs, is included in MIT-IL Report E-2146 "The Digital Simulation for the Verification of Program SUNBURST (Unmanned LM, AS-206)".

Mass Properties

The LM mass, center of gravity, and inertia are initialized for the specific vehicle configuration and propellant loading under test. During the run, the mass, center of gravity, and inertia of the vehicle are recomputed at intervals of approximately every 1.7 seconds when the descent stage is present and thrusting, and approximately every 5 seconds when the ascent stage only remains. The center of gravity and the inertia are tabulated as functions of vehicle mass for the ascent - descent configuration, and the ascent only configuration. These properties are computed by a parabolic interpolation between the tabulated values.

Fuel and oxidizer can flow between the two sets of propellant tanks in the descent stage. Thus, an increase in pressure in either set, in the absence of a contact acceleration, can force propellant into the other set of tanks. A significant change in the center of gravity results. The largest excursion can occur when one half of the DPS propellant remains. In this case it is possible for virtually all the DSP propellant to shift into one set of tanks with the other set empty.

Under the influence of DPS thrust, the center of gravity shifts back to its nominal position. This effect is modeled as an exponential decay in the center of gravity offset. During the interval when this effect is included, the mass properties are recomputed approximately every .17 seconds to track the rapid motion of the center of gravity.

At the time mass properties are recomputed, all relative locations are reset and the torque about the new center of gravity produced by each RCS thruster is evaluated.

Descent Propulsion System

The thrust buildup of the DPS engine at ignition is modeled as a pure delay followed by a step change in thrust to the commanded thrust level. The delay is calculated so that the model matches the impulse provided by a typical engine on signal to a steady state thrust at the minimum thrust level.

The model for thrust decay after the engine off command is similar except that the step delay is calculated so that the impulse from receipt of the engine off command to zero thrust is the same in the model as in a typical engine.

In modeling the action of the DPS computer controlled throttle, the LEM subroutine distinguishes between thrust command and actual thrust. The thrust command varies linearly with the number of throttle incrementing pulses issued by the LGC.

The thrust command ranges from 12 to 94 percent of full rated thrust and is continuously throttleable in the region from 12 to 60 percent of full rated thrust.

If the LGC mission program should stop incrementing the automatic throttle counter when the thrust command is in the range from 60 to 94 percent of full rated thrust, the LEM subroutine will print a complaint on the on line printer.

In the range from 12 to 94 percent of full rated thrust, the actual thrust follows the thrust command with a first order lag.

When a full thrust command is issued and the actual thrust has, for all practical purposes, reached the thrust command, a different model is used. This model is intended to represent the effect of erosion on the engine operating at full commanded thrust. In this case the actual thrust increases linearly with time such that at the end of the mission duty cycle, the actual thrust will reach the full rated thrust.

When the engine is throttled down from the full commanded thrust, the LEM subroutine remembers the amount the thrust has increased due to erosion.

If the engine is subsequently given a full thrust command, the actual thrust will return to the thrust command plus the thrust increase due to erosion.

The number of bits in the automatic throttle counter which commands 94 percent of full rated thrust is less than the number required to fill the counter. Thus above a saturation count, additional pulses issued to the automatic throttle counter do not increase the thrust command.

These additional bits are counted in the throttle counter up to overflow in that counter. Any additional pulses are lost. When the throttle counter is decremented, the LGC must reduce the throttle counter contents to the saturation level before the thrust command is reduced.

The automatic throttle can be incremented only when the DPS is armed. Disarming of the DPS causes the automatic throttle to be reset to its ground state.

The LEM subroutine contains a table of ISP's at five thrust levels. For intermediate thrust levels, a linear interpolation is made to arrive at the value of ISP to be used at that thrust level.

The DPS trim gimbal actuators are modeled as constant speed drives with a pure delay in coming to speed and a pure delay in coming to a stop.

In addition to moving the DPS thrust vector by driving the trim gimbals, the LEM subroutine can, at the user's option, include a fixed misalignment of the thrust vector and a thrust dependent misalignment of the thrust vector.

The LEM subroutine limits trim gimbal excursion to ± 6 degrees about each axis. Thus, small angle approximations are used to compute the thrust vector.

Ascent Propulsion System

The thrust buildup to full thrust and the thrust decay from full thrust are modeled as pure delays followed by an instantaneous change in thrust, as in the DPS model. The delays are calculated so that the model produces transient impulses typical of a real engine.

The erosion characteristics of the APS are included at the option of the simulation user. When included, the thrust of the APS increases linearly with time.

The APS engine is modeled as a constant ISP engine.

At the user's option, the LEM subroutine includes a fixed error in the point of application of the APS thrust. The orientation of the thrust vector can also be misaligned from the nominal at the user's option.

Staging Forces

The LEM subroutine simulates the forces and torques on the ascent stage due to staging and the subsequent APS ignition. These forces and torques are initiated at the APS ignition following staging and are modeled by two sets of polynomial time series. The first of these represent force and torque during their buildup immediately following APS ignition. During this interval, the forces and torques increase linearly with time measured from APS ignition.

After the interval of linear buildup, the staging forces and torques are modeled by a cubic decay. During this interval the second set of time series is used. When the end of this interval is reached, the staging forces and torques are set to zero.

Fuel Slosh

The effect of sloshing propellant in the APS and DPS tanks is included during periods of APS or DPS thrusting. There is no attempt to model sloshing during intervals of free fall. RCS propellant sloshing is neglected. Slosh effects can be excluded at the option of the simulation user.

A slosh mode is modeled with a two degree of freedom mass-spring oscillator which is lightly damped. There is an oscillator for each APS and/or DPS propellant tank. The mass, frequency and slosh attachment point of each oscillator are all functions of the propellant remaining in the tank. The subroutine includes this functional relationship by storing tables of slosh masses, frequency parameters and slosh attachment points for each tank and linearly interpolating between table entries to obtain the slosh parameters.

The slosh mass in any tank varies from zero for a fully loaded or empty tank to a maximum at 50 percent loading. In the cylindrical portion of the DPS tanks the slosh mass is a constant.

The frequency of the oscillator is a function of fluid level, contact acceleration, and propellant tank radius.

When the CSM-attached configuration is used, the structural bending of the combination is simulated. The model includes three planar modes each for the LM X-Y and X-Z planes. These modes are all lightly damped oscillators which are driven by RCS forces, CSM and LM propellant slosh forces, and main engine thrust. This effect is optional and can be excluded by the user.

Command Module - Entry Simulation Model

The LVCBLK2 program provides a simulation of the Command Module from the time it separates from the Service Module, prior to entry, until its speed goes below one-half the speed of sound (Mach 0.5). The primary function of LVCBLK2 during a simulated entry is the calculation of forces and torques acting on the vehicle due to aerodynamic loads and RCS thruster firings, and the determination of the resulting linear and angular contact accelerations at the navigation base. To accomplish this, LVCBLK2 contains the following models: Command Module mass and inertia, reaction control system, and entry aerodynamics.

At present, the entry simulation does not account for changes in command module mass properties due to RCS propellant usage or heat shield ablator loss. Hence the mass, center of gravity, and inertia of the vehicle are assumed constant during the simulated entry at values corresponding to CM/SM separation.

The model of the reaction control system includes the logic required to turn individual RCS thrusters on or off upon appropriate commands from the AGC, and to enable either or both of the RCS systems (A or B) under astronaut control. The thrust VS. time profile employed to simulate the firing of an individual thruster is calculated to give the same total impulse during a transient firing as would be obtained from a real thruster. The thruster model also includes a minimum allowable pulse duration, and a propellant penalty for each firing.

The aerodynamic forces and moments acting on the CM during entry are calculated from a symmetrical aerodynamic model, which consists of three aerodynamic coefficients; C_A -the force coefficient along the command module axis, C_N -the force coefficient normal to the command module axis, and C_M -the pitching moment coefficient about the theoretical command module cone apex. Values of these aerodynamic coefficients are

presented at three angles of attack for each of eleven Mach numbers. Values at other angles of attack and Mach numbers are found by a parabolic interpolation between the three tabulated angles and by a linear interpolation between the two nearest reference Mach numbers. Using values of dynamic pressure obtained from the UNIVERSE program, the aerodynamic coefficients are converted to aerodynamic forces and moments acting on the command module.

IMU Simulation Model

Depending on the mission phase, subroutine PRELUNCH, SATURN, or LEM simulates the attitude and acceleration of the navigation base in the LM. The resulting attitude and integrated acceleration at the end of an update is transmitted to subroutine IMUBLK2. This simulation of the inertial measurement unit resolves the integrated acceleration into platform axes and determines the number of counts to be sent to the AGC from each simulated pulsed integrating pendulous accelerometer (PIPA). The orientation of the gyro stabilized platform is set initially by data cards in the simulation user's run deck. The subsequent motion of the platform is determined by the moding discretizes and pulse trains to the gyros received from the AGC, plus gyro drift effects.

The orientation of the navigation base compared with the orientation of the gyro stabilized platform determines the three gimbal angles. The gimbal angles can be read by the AGC through the coupling data unit (CDU) analogue-to-digital converter. In the IMU subroutine, the tracking errors of the CDU are simulated in great detail. This accurate model is of interest to simulation users who are testing in fine detail the performance of the digital autopilots. For users who are only interested in the gross mission performance, a much faster CDU simulation is available which is simply a quantization of the continuous gimbal angles.

Simulation users who wish to explore the effect of PIPA and gyro inaccuracies may specify in their input deck non-zero values for zero-acceleration gyro bias drifts, gyro drifts caused by accelerations along the gyro input axes, gyro drifts caused by accelerations along the spin reference axes, PIPA bias accelerations and PIPA scale factor errors.

OPTICS Simulation Model

OPTICSB2 simulates the sextant and scanning telescope shaft and trunnion angles, their associated CDU error counters (input counters), and their associated CDU read counters (output counters). OPTICSB2 reacts to the three position ZERO-MANUAL-CMC command module panel switch at all times. This panel switch is assumed to be in the MANUAL position until switched by the astronaut subroutine.

When the panel switch is in position ZERO, the shaft and trunnion angles move at maximum rate to their null positions. The CDU read counters follow the movement.

When the panel switch is in position MANUAL, OPTICSB2 receives from the astronaut subroutine the desired pointing vector of a star, landmark or vehicle. Shaft and trunnion angles move toward this pointing vector at one-half their maximum rate by the shortest route to simulate the astronaut moving the hand controller. When shaft and trunnion angles are less than one-half of a CDU read counter bit from the desired angles, movement stops and OPTICSB2 returns control to the Astronaut subroutine.

When the panel switch is in position CMC, OPTICSB2 responds to the following discretes: Command Module Computer Error Counter enable, zero optics CDU, TVC enable, zero optics, and disengage DAC. The CDU read counters respond to read counter pulses and to the outbits which start pulses coming from the CMC. Shaft and trunnion angles move at rates which are linear functions of the number of pulses in the CDU error counters. Optics CDU read counters are set to the shaft and trunnion angles, rounded to the nearest CDU bit. Incremental pulses are sent to the CMC and are subtracted from the error counters to complete the closed loop system.

An attempt by the CMC to bring OPTICSB2 past the physical shaft limitation of plus or minus 270 degrees, or past the trunnion line-of-sight limitation of 67 degrees causes an environment abort.

Rendezvous Radar Simulation Model

The program, RRADAR, simulates the LM rendezvous radar and its tracking of the CSM transponder. The program has two distinct modes of operation; a search-designate mode and a tracking mode.

In the search-designate mode the antenna stabilization system is simulated. In the hardware, the antenna is inertially stabilized in two axes by two servo loops which hold the antenna mechanical bore sight fixed in the presence of vehicle motion. These two stabilization loops are simulated by second order systems which are driven by the radar coupling data units digital-to-analog converters and by motion of the vehicle.

The program reacts to the radar CDU Error Counter Enable and CDU Zero discretes from the LGC. With the error counter enabled, the RRADAR program interprets error counter pulses as angular rate commands to the respective stabilization loops. The gimbal angle position is provided to the LGC through a simulation of the CDU analog-to-digital conversion. Since detailed simulation of this A to D conversion is relatively time consuming and of limited interest to most users, an optimal mode is available which bypasses it and simply sets the CDU read counters to the antenna gimbal angles, truncated to the nearest CDU bit.

In the search-designate mode, the angular excursion of the antenna is checked against specified limits at each update. When the mechanical limits are reached, the program holds the antenna at the limit until a combination of vehicle motion and LGC commands drive it back from the stops.

When the LGC issues the "Self-Track Enable" discrete, the RRADAR program attempts to switch to the tracking mode. The RRADAR program updates the CSM state vector through the TARGET program. It then determines; if the line of sight to the target is within a 2° cone of the antenna beam center; if the range is less than 400 nautical miles; if the

range rate is less than 1500 meters per second; and if a user specified delay has elapsed since the "Self-Track Enable" discrete was set. If all of the above conditions are met, the program sets the "Data Good" discrete and switches itself to its tracking mode.

In the tracking mode, which is distinguished by the presence of the LGC "Data Good" discrete, the RRADAR program no longer simulates the antenna stabilization loops. It is assumed that the radar can keep its beam center directed along the line of sight to the transponder. The beam center differs from the mechanical bore sight by a fixed angular bias whose magnitude can be specified by the user. In this mode the CDU encoding of the antenna gimbal angles is again just a truncation of the gimbal angle to the nearest CDU bit.

When in the tracking mode, the program responds to the setting of the "Radar Selection" and "Radar Activity" bits by the LGC requesting range and range rate information. This information is obtained by updating both the LM and the target state vectors to the "read" time and calculating the range or range rate. The quantity of interest is properly scaled and returned to the radar read counter in the LGC.

At each update the program checks to see that the various tracking and antenna angular limits are not violated, the program switches to its search-designate mode and removes the "Data Good" discrete.

The scaling of the range in the radar counter changes at a range of 52 nautical miles and a "Low Scale" inbit is set for the LGC when this occurs.

The program normally begins in the search-designate mode but if, at the start of the simulation, the program finds the "Self-Track Enable" discrete set, it will check the tracking conditions. If these are satisfied it will begin in the tracking mode, thus saving the user the effort of providing vehicle and antenna orientation which satisfy the tracking conditions.

Since there will be long periods in a mission when the radar is not active, the option to turn off the radar has been provided. The astronaut action,

LEMRADAR ON
OFF

will cause the radar to be turned on (off) and the "Radar Power On/Auto" to be reset. The RRADAR program will not be updated after having been turned off. This will save whatever simulation time RRADAR would have consumed.

TARGET - Passive Vehicle Simulation

The TARGET program maintains the free fall trajectory for the passive vehicle in a rendezvous simulation. It is used by the Astronaut and Rendezvous Radar programs for updating the state vector prior to optical or radar navigation marks. The program uses the same gravity models as the UNIVERSE program when integrating the state vector with the exception that TARGET always includes oblateness and other body perturbations.

There is a limited capability to introduce state vector discontinuities at user specified times. Aside from this, the TARGET program has no thrusting capability.

The TARGET program files its state vector in the CALLFILE prior to each CALLFILE dump, so it is available for editing purposes.

Astronaut Simulation Model - Program ASTRNAUT

The motivation for including an Astronaut program in the Digital Simulation is to provide the capability of simulating the interface between the real Astronaut and the AGC. The functions of the Astronaut are modeled in a logical fashion, not biological. That is, the Astronaut program is concerned with setting panel switches or typing keyboard inputs rather than representing the dynamics of a human transfer function.

The operations performed by the digital Astronaut may be broken down into 3 areas: Keystrokes, actions and subprograms. Keystrokes include the detailed typing of inputs on the Astronaut's keyboard, actions include the setting of panel switches, while subprograms include the more general macroscopic functions of the Astronaut, such as maneuvering to two stars and then marking them.

The input to the Astronaut program consists of Astronaut cards specifying in a free form, mnemonic style the sequence of actions to be performed by the Astronaut during a simulation run. Within the initialization pass of the simulation run, the Astronaut cards are read by an Astronaut Input Program, which translates the Astronaut cards into digital code words, and writes the code words into the Data File. The resulting Astronaut File is read sequentially by the Astronaut program at run time and determines its sequence of actions. A detailed description of the Astronaut input cards and the format of the Astronaut File is included in Chapter 3, Section H.

The sequence of digital code in the Astronaut File causes the Astronaut to perform specific functions. Each "sentence" in the File contains a conditional clause and a functional clause. The functional clause determines what action the Astronaut should perform, such as typing keyboard inputs, setting panel switches, or executing Subprograms, while the conditional clause informs the Astronaut under what conditions to perform the specified action.

Conditional type clauses contain the following terms:

1. TIME Specifies the exact time at which the Astronaut performs an action.
eg: TIME 11.7 VERB 33 ENTER
2. WAIT The Astronaut may be requested to wait a specified time interval before performing an action.
eg: WAIT 8.1 SCSMODE AUTO
3. VERIFY The Astronaut may use the VERIFY function to check the status of his Displays. For instance, a command to the Astronaut to verify that the Displays contain V 01 N 71 and then key in V 33 E would appear on the Astronaut Input Card as:
VERIFY V 01 N 71 THEN V 33 ENTER
4. IF The IF conditional clause is used to check the status of a flashing display. In general, the VERIFY function should be used to check flashing or non flashing displays whose sequencing is known, while the IF function is most applicable to the checking of flashing displays whose occurrence in time is not known. The IF statement will catch such flashing displays when they appear.
eg:
IF V 01 N 77 THEN V 34 ENTER

Action type clauses are tabulated in detail in Chapter 3, Section H. The actions programmed in the Astronaut routine include most astronaut functions that interface significantly with the AGC. These functions include:

Keystrokes	Verb, Noun, Clear, Enter, Digits 0-9, etc.
Mark buttons	Mark, Mark Reject, etc.
Panel Switches	IMU Power, SCS Mode, CM-SM Separation, Guidance Mode Primary, Abort Stage, Ascent Engine Arm, LM Throttle Setting, Thrust Switch, Deploy Landing Gear, etc.

Handcontroller and Minimum Impulse

+ Pitch, + Yaw, + Roll, etc.

Translation Controller

+ X, + Y, + Z, Engine off, etc.

LM Handcontroller

P, Q, R rates.

LM Radar

ON, OFF

LVC RCS Selection

A System, B System, Both

A major additional capability programmed into the Astronaut routine is the SUBROUTINE function, which may be called into use as an action clause by specifying a numbered SUBROUTINE on the Astronaut Input cards. An Astronaut SUBROUTINE constitutes a sequence of detailed astronaut actions carried out internally by the Astronaut program.

The SUBROUTINE capability was added to the Astronaut Program to enable a user to request the Astronaut to perform various macro operations, such as marking two stars, marking LM, LM coarse alignment, etc. Each such macro Astronaut function involves numerous detailed operations such as picking stars to be marked, maneuvering the vehicle to put the star in the field of view of the optical system, and marking the star repetitively. Since each numbered SUBROUTINE implies a sequence of fixed operations, an AGC user should discuss the implied sequence with a member of the Digital Simulation Group (preferably the Astronaut programmer) before employing the SUBROUTINE on an Astronaut Card.

The complete list of Astronaut SUBROUTINE's currently available is specified in Chapter 3, Section H. However, the various categories of SUBROUTINE's are summarized below:

- Maneuver to specified gimbal angle.
- CSM or LM coarse align maneuver and mark 2 stars.
- Mark one star from CSM.
- Mark LM from CSM.
- Null out terminal VG.
- Maneuver CSM or LM to see 2 stars.
- Key in landmark data after marking landmark.
- Track landmark.
- Mark star from lunar surface.
- Key in cursor or spiral angle.
- Maneuver LM to see specific star.
- Mark specific star from LM.

The categories of SUBROUTINEs will enlarge as future flight requirements are defined.

LM Mission Programmer

In unmanned flights with the LM spacecraft, the LM responds to control commands from up-telemetry (via the Astronaut program) and from the LM mission programmer (LMP). The functions of this controller are simulated by the MAC environment subroutine LMP1, some of whose responsibilities are summarized below.

The LEM subroutine will allow ignition of the main engine and incrementing of the DPS automatic throttle only if that engine is armed. Removal of the engine arm signal by the ASTRNAUT or LMP1 during a burn will cause engine shutdown and resetting of the automatic throttle counter to its ground state or minimum thrust setting.

Simulation of the staging of the ascent and descent stages is accomplished by the LEM subroutine upon receipt of the abort stage command from the LMP1 subroutine. It is necessary that the ENGINE ON signal be present for staging to occur upon receipt of the stage command.

Simulated ON and/or OFF failures of RCS thrusters are initiated in the LEM subroutine under the control of the simulation user. When an ON failure has occurred, the LEM subroutine returns a THRUSTER FAIL discrete to the LGC simulator. The LGC normally commands the LMP to isolate the thruster pair containing the failure. The LMP then commands the LEM subroutine to isolate the thruster pair. Once an isolation command is received, the LEM subroutine turns off the failed jet and its companion and ignores further commands to that pair. An OFF failure simply causes the LEM to turn off the jet and ignore all further commands to that jet. This sequence is unique to unmanned missions and unmanned portions of manned missions.

Since it is necessary to verify the transition from the Primary to Abort Guidance System, the LEM subroutine has this capability. Upon command from the LMP, the LEM ignores control commands originating in the LGC simulator. Since the LGC exercises a monitor function during AGS controlled phases, the LEM must have the capability to execute a burn without LGC commands. This is accomplished by the LMP which issues the necessary engine commands for an unguided burn. The LMP can return control to the LGC after the completion of the AGS controlled phase.

Numerous other LMP functions have been simulated; the interested user is referred to the LMP1 program listing.

Discontinuity Program

The MAC environment subroutine DISCONT has the capability of changing specified bits in the input channels under user control. This facility is useful if an AGC programmer wishes to test a portion of a mission program which reacts to changes in input channel bits. Through the use of DISCONT, a user may turn on or off any input channel bit at any specified time. Details on how to use DISCONT may be found in Chapter 3, Section H.

Simulation Setup Program - SIMSETUP

To aid in setting up initial conditions for digital simulations, an auxilliary program, SIMSETUP, may be run as a separate job.

The function of the SIMSETUP program is to resolve an input difficulty in the simulator. The difficulty arises because the digital Simulator will accept initial spacecraft position and velocity only in a universal inertial coordinate frame defined by the mean equinox and equator for the nearest beginning of the year. Since it is awkward to compute initial conditions in this coordinate frame, the SIMSETUP program was written to convert initial conditions from a variety of convenient, user oriented reference systems, such as latitude, longitude, altitude, into an R and V vector in the universal frame required for input to simulation. In order to set up non-trivial initial conditions for a simulation run, a user normally would run SIMSETUP first, using a convenient reference system to specify initial spacecraft position and velocity, and obtain from SIMSETUP a deck of cards which may be used as input cards in the simulation run deck. A more complete description of SIMSETUP is included in Chapter 3, Section H. Users who simply want their simulation runs to start with a spacecraft relatively far from the Earth, at an arbitrary attitude, may avoid the delights of SIMSETUP entirely by using a degenerate set-up.

Astronaut Input Program

An assembly language program, the Astronaut Input Program (360 name: ASTXMCI) is used to translate Astronaut input cards from free form mnemonic alphabetic characters to decimal code words. These code words are written into a block of the data file known as the Astronaut file. During the initialization of a simulation run, the detection of an Astronaut card (identified by having an "A" in column 1) in a simulation input deck causes control to be transferred to the Astronaut input program, which reads and translates Astronaut cards or prints remarks cards until the set of Astronaut input cards is exhausted. An assumption is made that all Astronaut cards are submitted in a single set, interspersed only by remarks cards. During the translation of Astronaut input cards, the Astronaut input program performs a variety of diagnostic and logical checks on the legality of Astronaut actions appearing on the cards and underlines any errors that may be detected.

The Astronaut File produced by the Astronaut Input Program at initialization time is processed sequentially by the Astronaut program at run time. Each word of the Astronaut file causes the Astronaut program to perform the specific function called for on the corresponding word of the Astronaut input card. The Astronaut program progresses through the Astronaut File throughout the simulation run, reading a sufficient number of words from the File to perform a desired action, before proceeding on to the next action.

The list of available Astronaut commands, as they appear on Astronaut input cards, are tabulated in Chapter 3, Section H. Included in this same Section are the digital code words corresponding to each legal Astronaut input command, as translated by the Astronaut Input Program.

2. D. 4 Initialization of Environment

In the preceding section of the Manual, the organization, structure, and modeling of the MAC environment has been presented in detail. The initialization of this environment, the interaction of subprograms within the environment, and the coordinate systems utilized by the environment are discussed below. The structure of a flight test simulation input deck, as well as a complete listing of all input parameters for environment subprograms are included in Chapter 3, Sections F and H.

The collection of the MAC environment routines is the first major operation performed in a flight test simulation run. MAC environment program collection is initiated and controlled by MAC run step control cards and subcontrol cards. See "Users Guide to MAC-360" and Chapter 3, Section F of the Manual. The MAC program collection process transfers all the MAC subroutines required to simulate the environment from general disc storage to high speed memory.

Certain of the environment programs (viz: UNIVERSE, SATURN, CSMBLK2, LEM, and LVCBLK2) require special parameter files which must be loaded into the data file during the collection process. These files contain mission specific data such as vehicle weights, inertias, RCS jet positions, boost trajectories etc., which the generalized vehicle simulation programs use to adapt themselves to specific missions. See Section 2. D. 2 for a list of available files. If the user does not load the proper files, the simulation will abort.

Following the MAC program collection process, the MAC environment is turned on for an initialization pass prior to any actual AGC simulation in order to allow the individual subroutines to:

1. Read input data for state variables.
2. Read mission specific vehicle design data from data file.
3. Enter return requests in the AGC.
4. Set initial AGC input channel discretizes.

The following sequence of operations is performed during an environment initialization:

- 1) The Environment Control program is turned on for initialization and immediately calls the INPUTPRG subroutine.
- 2) INPUTPRG reads the entire deck of MAC input data cards and transfers the information on the cards to blocks in the data file assigned to individual environment programs. An ID code on each input card identifies the environment program to which the input data is to be sent. Astronaut input cards are read separately from MAC data cards. If there are any astronaut cards to be read, a separate routine, the Astronaut Input Program (ASTXMCI), is called upon to read them. Finally, INPUTPRG individually calls each of the environment subroutines for which input data has been read.
- 3) As each environment subroutine is called, it reads the input data stored for it in the data file by INPUTPRG and uses this data to set initial values for state variables. Vehicle subroutines, in addition, read from the data file a block of design data which contains mission specific data -- such as jet positions, vehicle mass and inertias, engine performance data, fuel tank positions, etc.

- 4) Finally, each environment subroutine sets any relevant AGC input channel discretes that are normally on and, if there is any predicted activity that will occur at a specific time, sets return traps which force the AGC to stop and call the environment at the requested return time. The first keystroke in the Astronaut list, or the predicted time of liftoff are examples of functions that would use the return trap mechanism.

When the MAC environment is fully initialized, control is returned to the AGC simulator which starts the instruction by instruction simulation at the AGC location specified by the start special request. If there were no references to input-output registers, then control would not leave the AGC simulator and the environment would be left at its initial setting.

2. D. 5 Sequencing of AGC, Communicator, and Environment Updates

To allow the environment to set its own time steps, the environment routines are decoupled as much as possible from the AGC Instruction Simulator. This is accomplished by inserting a program called the Communicator between the AGC and the environment, whose function is to filter the high frequency AGC input-output activity, and produce low frequency environment updates. Instead of forcing separate environment updates to satisfy each AGC counter read request, the Communicator relies on extrapolations to provide estimated counter contents for most requests. When it becomes necessary to update the environment, the Communicator transmits batches of accumulated AGC generated commands to the environment. Following each environment update, the Communicator is provided with CDU, PIPA, and optics counter incrementation rates and accelerations, along with the time to which these quantities are presumed to be valid. Whenever possible, these rates are then used by the Communicator to satisfy AGC counter update requests. The Communicator continues to estimate counter contents in this manner until the time of validity of the counter rates has expired, or until a discontinuity (such as long pulse trains or an engine signal) invalidates the rates. Most environment updates are triggered by counter read requests for which previous extrapolation rates have become invalid.

The environment always lags the AGC until it is forced to do an update. Each update brings the environment up to the present AGC time. Internal sequencing within the environment is controlled by the Environment Control (ENVCNTRL) routine. In a typical environment update, ENVCNTRL calls one of the vehicle dynamics subroutines (PRELUNCH, SATURN, LEM, CSMBLK2, or LVCBLK2), which updates itself to AGC time, and then calls IMUBLK2 to provide the new counter contents. The vehicle subroutine may also call UNIVERSE to integrate position and velocity of the spacecraft. After the vehicle program and IMUBLK2 are updated to present time, the Environment Control program may make direct calls

to ASTRNAUT, OPTICSB2, LMP1, RADAR, or DISCONT (if any of these offline routines requested a return or is to receive a command from the AGC).

Automatic sequencing between vehicle subroutines occurs as follows:

PRELUNCH switches to SATURN at Liftoff

SATURN switches to LEM or CSMBLK2 at SIVB separation

CSMBLK2 switches to LVCBLK2 at CM/SM separation

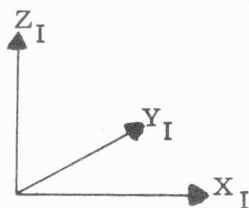
2. D. 6 Coordinate Systems and Transformation Matrices

Among the numerous coordinate systems used in the digital simulation, five are of special interest to users of the simulation. These coordinate systems are:

1. Universe inertial.
2. Local vertical.
3. Reference system for vehicle PITCH, YAW, and ROLL.
4. CSM reference system.
5. LM reference system.

All of these reference frames are used in the specification of input variables to environment programs, or in the interpretation of printed or filed output variables. Each of the reference frames is defined below, together with a description of its use.

Universe Inertial Frame (I frame)



X_I is a unit vector along the ascending node of the ecliptic on the equator (the equinox) and lies in the general direction of the first point of Aries.

$$\bar{Y}_I = \text{UNIT} (\bar{Z}_I * \bar{X}_I)$$

\bar{Z}_I is a unit vector along the mean north pole.

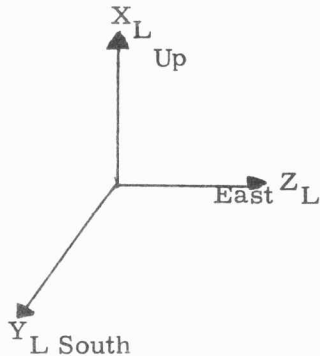
This axis system is defined by the intersection of the mean equatorial plane and the mean orbit of the Earth (the ecliptic) at the nearest beginning of the Besselian year. The origin of the coordinate system is at the center-of-mass of the Earth, or the center-of-mass of the moon, at the option of the user. Note that the Z axis of the Universe frame coincides with the mean north pole of the Earth, not the true north pole or the true angular momentum vector of the Earth. Thus, the Universe inertial Z axis does not reflect the precession or nutation of the Earth's angular momentum vector. Similarly, the moon's Z axis coincides with the mean north pole of the moon, which is only approximately aligned with the moon's angular momentum vector. In general, the moon's angular momentum vector lies within $1\ 1/2^\circ$ of the perpendicular to the ecliptic (plane of Earth's rotation about the Sun), while the Earth's angular momentum vector is misaligned by approximately $23\ 1/2^\circ$ from the perpendicular to the ecliptic.

Time in the Universe coordinate system is measured in days from the beginning of the Universe ephemeris data to a specified epoch, and then in seconds from this epoch to the final desired time. The beginning of the Universe ephemeris data is defined to be midnight at Greenwich preceding July 1 of the specified year. The intermediate epoch time is used as a convenient reference point, such as time of liftoff, midnight of a specified day etc. The computation of ephemeris days from July 1 proceeds as follows: for the first half of the year, add 184 to the number of days from the beginning of the year; and for the last half of the year, subtract 181 for normal years, and 182 for leap years.

The Universe frame is the basic reference for all other frames. Orientation of the stable member, the vehicle attitude, and the local vertical are specified in terms of the Universe inertial coordinate system (often referred to as the I frame).

Refer to the description of the UNIVERSE model, Section 2. D. 3, or to the UNIVERSE listing for more information on the Universe inertial frame.

Local Vertical Frame (L frame)



X_L is a unit vector opposite in direction to the gravity vector, computed by adding to the gravitational vector the effect of centripetal acceleration due to the rotation of the Earth or Moon.

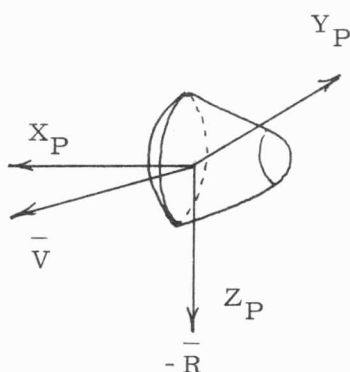
Z_L is a unit vector in the East direction.

$\bar{Y}_L = \text{UNIT}(\bar{Z}_L * \bar{X}_L)$ and is a unit vector in the South direction.

The gravity vector is computed once at initialization time for a specific mission time, such as liftoff. The simulation assumes that this gravity vector is imbedded in the body and rotates at constant rate. This implies that gravitational accelerations due to the perturbation of other celestial bodies are also rotated fixed to the Earth or Moon, and results in a small error. The assumption is also made that the Earth or Moon rotates exactly about the angular momentum vector at Liftoff which ignores the effects of the precession and nutation of the angular momentum vector.

The local vertical system is used in Prelaunch to define the orientation of the vehicle axes and the stable member axes. During Prelaunch, it is convenient to refer the vehicle and stable member to the East, South, Up system (local vertical) rather than to the Universe inertial frame.

Preferred Reference System for Vehicle PITCH, YAW, and ROLL (P frame)



$$\bar{X}_P = \text{UNIT} (\bar{Y} * \bar{Z})$$

$$\bar{Y}_P = \text{UNIT} (\bar{V} * \bar{R})$$

Unit vector perpendicular to the momentum plane.

$$\bar{Z}_P = \text{UNIT} (- \bar{R})$$

Unit vector directed toward the center of the Earth, or Moon.

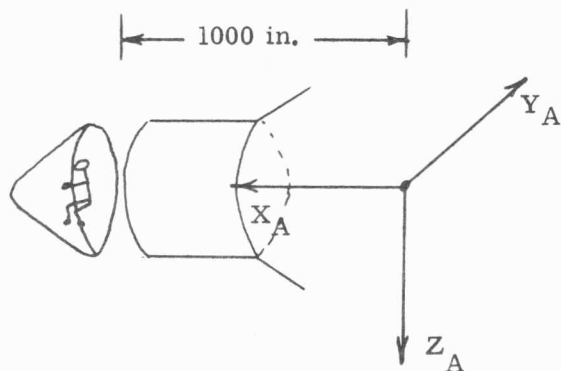
\bar{V} = velocity vector in I frame.

\bar{R} = position vector from center of Earth or Moon in I Frame.

Vehicle pitch, yaw, and roll is measured with respect to a reference frame defined by the vehicle's momentum plane. The Z axis of this reference plane always points to the center of the Earth, or Moon and the Y axis is always perpendicular to the plane containing \bar{R} and \bar{V} (the momentum plane). Therefore, zero pitch, zero yaw, zero roll would mean that the vehicle's Z axis was down and the vehicle's Y axis was perpendicular to \bar{V} and \bar{R} . The vehicle's X axis will be parallel to the velocity vector only if the velocity vector is perpendicular to \bar{R} .

This preferred coordinate system, or P frame, is used as the reference system to define PITCH, YAW, and ROLL on the CSMBLK2, LEM, and LVCBLK2 simulation input cards. Vehicle pitch, yaw, and roll is also measured with respect to the P frame in the summary print triggered by the ENVSUM special request, as well as in the call file dumps, and in vehicle on-line print statements. PITCH, YAW, and ROLL are defined as Euler angle rotations about the axes Y_P , Z_P , and X_P in that order.

CSM Reference System (Vehicle Body Axes - V Frame)



X_A is a unit vector along the vehicle's axis of symmetry and directed forward.

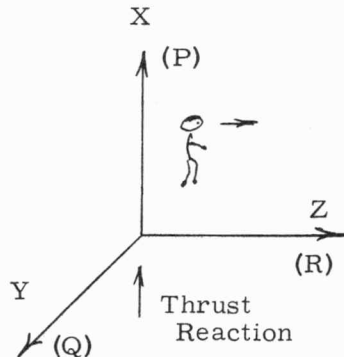
$\bar{Y}_A = \text{UNIT}(\bar{Z}_A * \bar{X}_A)$
Unit vector directed to the astronaut's right.

Z_A is a unit vector down, relative to the Astronaut.

The CSM reference axes, X_A , Y_A , Z_A are assumed to be fixed to the vehicle, and coincide with the standard Apollo Vehicle Reference System, referred to interchangeably as the A frame or V frame. The origin of this frame is defined to be a point 1000 in. aft of the mold line of the heat shield main structure-ablation interface.

The attitude angles PITCH, YAW, and ROLL are measured from the P frame to the vehicle body axes, and constitute Euler angles about Y_A , Z_A , and X_A in that order.

LM Reference System (Vehicle Body Axes - V Frame)



X is a unit vector up, relative to the Astronaut.

$$\bar{Y} = \text{UNIT} (\bar{Z} * \bar{X})$$

Unit vector directed to the Astronaut's right.

Z is a unit vector forward relative to the Astronaut.

The direction of the thrust reaction vector is generally along X.

The LM body axes X, Y, Z are imbedded in the vehicle. On the above sketch, P, Q, and R represent vehicle angular rates about the X, Y, and Z axes respectively. Vehicle PITCH, YAW, and ROLL are measured from the P frame to the LM body axes, and represent Euler angle rotation about axes Y, Z, and X respectively. Relative to the Astronaut, the X and Z axes are oriented differently in the CSM and LM, however this does not change the definitions of PITCH, YAW, and ROLL.

The origin of the LM reference system is fixed at a point 254 inches below the RCS reference plane (roughly in the plane of the landing pads).

Transformation Matrices

A number of transformation matrices relating inertial space to nav base, stable member to inertial, etc., are made available to simulation users through the call file. The matrices and their definitions are as follows:

*	
T_{IN}	Transformation from navigation base to Universe inertial frame.
*	
T_{IL}	Transformation from local vertical frame to inertial frame.
*	
T_{SN}	Transformation from the navigation base to the stable member axes.
*	
T_{SI}	Transformation from the Universe inertial frame to the stable member frame.

In addition, the matrix T_{IP}^* , relating the preferred system for PITCH, YAW, ROLL may be derived from the position and velocity vectors \bar{R} and \bar{V} which are also in the call file.

The notation used in these matrices is:

*
T
(TO) (FROM)

Thus, a vector expressed in navigation base components may be transformed into a vector in inertial components through:

$$\bar{V}_I = T_{IN}^* \bar{V}_N$$

Multiple transformations may be expressed as follows:

$${}^*T_{IN} = {}^*T_{IS} {}^*T_{SN} = {}^*T_{SI} {}^*T_{SN}$$

In the above example, the transformation from nav base to inertial is represented as the result of the double transformation from nav base to stable member, and finally from stable member to inertial.

All angles used in the simulation that either are extracted from transformation matrices or are used to generate a transformation matrix are Euler angles, and are handled in the order Y, Z, X. Vehicle orientation angles PITCH, YAW, and ROLL are identified with Y, Z, X respectively.

Although a transformation matrix ${}^*T_{AB}$ transforms a vector from B to A, the Euler angles used to form the matrix are assumed by convention within the Simulator to be measured from A to B.

Thus, if the Euler angles

PITCH, YAW, ROLL measured from P (Preferred) to
V (Vehicle)

are used to form a transformation matrix, the resulting matrix will be:

${}^*T_{PV}$ which transforms from V to P

Similarly, if Euler angles are extracted from the matrix ${}^*T_{IS}$ (transforms from S to I), using the standard Euler angle extractor in the Simulator, the resulting Euler angles will be measured from I to S.

2. D. 7 Specification of Vehicle and Stable Member Orientation

Three matrices are of general interest in the specification of vehicle and stable member orientation:

- *
 T_{IS} Transformation from stable member to Universe Reference Inertial (I Frame). The T_{IS}^* matrix is constructed from the stable member orientation angles, \bar{EULER} , which define the initial stable member orientation with respect to the Universe Inertial Frame. (See IMUBLK2 input card description in Section 3. H).
- *
 T_{VP} Transformation from Preferred Frame (P Frame) to vehicle body axes. This matrix is constructed from the vehicle initial orientation angles PITCH, YAW, and ROLL (defined in the input card description for SATURN, CSMBLK2, LEM, or LVCBLK2 in Section 3. H).
- *
 T_{SV} Transformation from vehicle body axes to stable member axes. The Euler angles used to construct this matrix are the gimbal angles, which relate the stable member orientation to the navigation base (assumed to be coincident with the vehicle axes).

Any two of these matrices or the sets of angles used to generate the matrices may be used to define the third. For instance, using the notation of the preceding Section, the gimbal angles may be derived from a knowledge of \bar{EULER} and PITCH, YAW, and ROLL as follows:

1. \bar{EULER} is used to construct T_{IS}^* .
2. PITCH, YAW, ROLL are used to construct T_{VP}^* .

3. Spacecraft position and velocity (\bar{R} and \bar{V}) are used to construct \bar{T}_{PI}^* (transformation matrix defining the orientation of the Preferred Frame with respect to the Universe Inertial Frame).

These matrices are then used to define T_{SV}^* :

$$T_{SV}^* = T_{SI}^* T_{IP}^* T_{PV}^*$$

and since the matrices are orthogonal and normalized:

$$T_{SV}^* = T_{IS}^T T_{PI}^* T_{VP}^T$$

Gimbal angles may then be extracted as Euler angles of the transformation matrix \bar{T}_{SV}^* .

The specification of initial conditions for the environment program IMUBLK2 and the Vehicle programs SATURN, CSMBLK2, LEM, or LVCBLK2 illustrate an application of the above manipulations. Initial vehicle attitude is specified by the input parameters PITCH, YAW, and ROLL as described in Section 3. H. Stable member orientation is initialized by the three angles in \bar{EULER} which are input parameters to IMUBLK2. The IMUBLK2 program also requires specification of the initial CDU angles, which will be equal to the initial gimbal angles in most cases. Therefore, the general procedure is to specify PITCH, YAW, ROLL and \bar{EULER} as independent parameters and compute the CDU angles (gimbal angles) as dependant parameters.

An extensive set of routines designed to perform the above operations to generate initialization parameters for the environment input cards and for AGC erasable loads is contained in the utility program SIMSETUP which is described in Section 3. H.

CHAPTER 3

User's Guide to Simulation Input Cards

3. A Preface

Chapters 3, 4, and 5 of this manual constitute an updated revision of ADSUG, the Apollo Digital Simulation User's Guide. Major changes in operating procedures, run step control cards, AGC special requests, Rollbacks, and Edits have been incorporated in this revision. These changes reflect not only the transition from the H1800 to the IBM 360 computer, but they also reflect the fact that the IBM 360 Digital Simulator has been considerably expanded in capability relative to the H1800 version. Several new AGC special requests have been added: BACKGRND, DELETE, ENVMODE, EQUALS, GOSEQOK, MSG, NOGOSEQ and TLOSS, as well as the new condition codes: BEFORE, AFTER, and CYCLE. Class 3 special requests may now be triggered by time, AGC location, or input-output Channel; and AGC traces (both TRACE and ITRACE) may specify the number of lines to be traced, in lieu of a trace stop location. For rollback runs, the capability now exists to change AGC special requests as well as Astronaut input cards at the beginning of the run.

The nature of these changes is extensive and thus AGC users are urged to study carefully the sections of this manual dealing with AGC special requests (Sections B and C of this chapter), Rollbacks, (Chapter 4), and Edits (Chapter 5). A review of the entire manual will enable an AGC programmer to apply the complex Simulation more precisely and efficiently to his specific requirements. As demand for IBM 360 computer time rises, it will be increasingly important to utilize the Simulation efficiently.

3. B. Simulator Control Cards

3. B. 1 360 Job Control Cards

The basic independent entity submitted to the IBM 360 computer is known as a JOB. The first card(s) of any job are known as the Job Control Cards and specify information peculiar to the entire job, such as job libraries, job priority, job name, user name, user's address, print estimate, time estimate, contract numbers, etc. Job Control Cards are generally defined in the IBMSystem/360 operating system manual "Job Control Language". Job control cards are specifically defined in memos that will appear from time to time as the IBM operating procedures at our installation change.

3. B. 2 360 Job Step

After the Job Control Card(s), the remainder of the entire job deck is composed of one or more partitions each known to the IBM operating system as a JOB STEP. For the purposes of simulation users, an IBM Job step consists of two parts:

3. B. 2. A - Job Step Control Cards

This group starts with an "EXEC" card and ends with a "SYSIN DD *" card. These cards give directions to the IBM operating system about the job step, and are defined generally in the IBMSystem/360 operating manual, "Job Control Language". Specific Job Step Control Cards required for simulation runs are defined in memos that will appear from time to time as the IBM operating procedures at our installation change. Listings of example run decks will be maintained on the Digital Simulation Group bulletin board in the Assembly Control Room.

3. B. 2. B - Job Step Data Cards

This group contains all the cards of the Job Step to be read and acted upon by programs other than the IBM operating system. This group contains all MAC control and input data cards, Simulator control and input cards, edit control and input cards, etc.

The one or more Job Steps of an IBM Job are executed in sequential order. Any reasonable number of IBM Job steps may be placed with one job.

3. B. 3 MAC Run Step

When any MAC programs are to be executed in an IBM Job Step, the IBM Job Step is considered to consist of one or more partitions, each known as a MAC Run Step. Each MAC Run Step is divided into two sections:

3. B. 3. A - MAC Run Step Control Cards

Defined in "User's Guide to MAC-360", Section 8.2.3. It is assumed the reader has thorough knowledge of that section.

3. B. 3. B - Input Cards to MAC Programs and/or to Non-MAC Programs executed under MAC Control

1. AGC simulator control and special request cards are input to the non-MAC simulator program.
2. AGCEDIT control and request cards are input to non-MAC AGCEDIT program.
3. Astronaut cards are input to non-MAC astronaut input program.

4. Inputs to MAC programs (environment initialization, etc.) are defined in general format in "User's Guide to MAC-360", Section 6. 2.

Flight test simulations, astronaut only simulations, environment only runs, MAC Edits, MAC plots of simulation results are all considered MAC Run Steps. Bench test simulations or basic AGC edits are not MAC steps.

For flight test or astronaut only simulations desiring snapshot/rollback capability, the MAC Run Step Control Cards for that simulation must start with the control card

* SNAPSHOT

This card is not specifically defined in the MAC manual. Further, the HOLD MARSROT simulator director card must be present in the simulation deck.

Any reasonable number of MAC Run Steps may be placed into one IBM Job Step. However, MAC Run Steps and non-MAC executions may not be intermixed within the same IBM Job Step.

3. B. 4 AGC Instruction Simulator Director Cards

The AGC Instruction Simulator Program is used in bench test, flight test, and astronaut only simulations. Three director cards, (1) the SIMULATE card, (2) the Program Source card and (3) the HOLD MARSROT card, provide the simulator with basic control

information with which to process the special request cards, to execute the specified AGC program as modified by the special requests, and to retain for future use an output file specified by the special requests. The fields of these, unless otherwise specified, must be separated by at least one blank. The three cards are:

3. B. 4. A - Simulate Card

This card is of the form:

* SIMULATE C

where the * is in Column 1 and SIMULATE is in Column 9.

C = AGC for all current Block II simulations.

C = BLK2 for old style Block II programs that do not allow
 11 bit store code interpretive addressing.

C = AGC4 for Block I simulations on the 360.

For compatibility between the IBM 360 and the H 1800, the H1800 monitor cards and H1800 MAC control cards may be retained in the IBM 360 input deck, thus maintaining the integrity of the full H1800 input deck. These H1800 cards follow the last IBM step control card for Bench test simulations, or the last MAC Run Step Control Card for MAC controlled runs (Flight simulations, Astronaut only runs, Environment only, MAC edits, and MAC plots). All of these H1800 control cards are printed and by-passed. In MAC controlled simulations, if the user has no cards he wants printed and by-passed, the SIMULATE card can replace the last preceding MAC Run Step Control Card (the " * SIMULATE" MAC Run Step Control Card).

3. B. 4. B - Program Source Card

The program source card directs the simulator to the correct YUL program to be simulated. This card is of the form

M ASSYTAPE PROGNAME REVISION NUMBER

where M is in Column 1 and the rest of the card starts in Column 9, each field separated by at least one blank. ASSYTAPE is the name of the assembly tape on which the YUL program resides. PROGNAME is the name of the YUL program. REVISION is written as the next field. NUMBER is the YUL program's 3 digit revision number. If it is desired to run a bad assembly, the word BAD should be supplied after the revision number. This simulator will abort the simulation whenever a bad or unused word is referenced. The user may change the status of any bad or unused words by suitable class 2 requests.

3. B. 4. C - Hold Marsrot Card

MARSROT is the MARS rough output tape. This tape is used for two purposes: (1) to store simulation output generated by DUMP, ITRACE and CFDUMP special requests for use by subsequent edit runs, and (2) to store all the simulation programs and the snapshots taken during simulation for re-starting via rollback. Requests for a Marsrot Tape are of the form

M HOLD MARSROT NAME TITLE

where M is in Column 1, HOLD MARSROT in Column 9, NAME is a 16 character field starting in Column 41 containing the simulation user's name, and TITLE is a 20 character field starting in Column 61 containing a title the user is giving to this Marsrot. If the user does not wish to generate a Marsrot, this card must be excluded from the deck.

3. C. AGC Instruction Simulator Special Request Cards

Special request cards may be included in the simulator input card deck immediately after the AGC instruction simulator director cards or in the rollback deck immediately after the " * ROLLBACK MARSROT FILENUMB TIME" card. They are used to define special conditions for a simulation run. They may, for example, call for diagnostic information for debugging purposes, or they may introduce temporary modifications (octal corrections) into the YUL program to be run. More than one special request may specify a particular location, if desired.

3. C. 1 - Formats

Special request cards have a fixed field format. The format used differs somewhat from the YUL system assembly language because of the need for a 16 column location field.

Column 1 through 8	Card No. and Printer line spacing control
Column 9 through 24	Location Field
Column 25 through 32	Op Code Field
Column 33 through 48	Data Address 1
Column 49 through 64	Data Address 2
Column 65 through 80	Remarks or Condition-Indicator

In general, a word may be positioned anywhere within a field (although none of the characters of the word should spill over into the next field). In the following, whenever information is described as being in the "data address field", the field implied is data address field 1 unless the context indicates a different interpretation.

3. C. 2 - Addressing and Bank Notation

The Block II AGC has a number of banks of switched fixed-memory. Each of the banks may be addressed in the range 2000-3777, provided that the appropriate bank switching register contains the corresponding bank address. In addition, the Block II computer has switched erasable-memory, addressable in the range 1400-1777.

In order to provide maximum user's convenience in addressing locations in AGC memory, the Mars AGC simulator recognizes six different methods of representing addresses:

- (a) Unsigned numeric address. From one to six octal or decimal digits (imbedded blanks allowed) may be used. The digits should be followed by a D if they are to be interpreted as decimal. If 8's or 9's are present but the D is not, the field is treated as decimal.
- (b) A two-octal-digit bank indicator, comma, and four-octal-digit subaddress from 2000 to 3777 for BLK2 (imbedded blanks not allowed) may be used to refer to locations in switched fixed-memory banks only. Block II superbanks 30-33 are addressed as 40-43.
- (c) Erasable-bank notation. This consists of the character E, followed by a digit from 0-7, a comma, and a subaddress in the range 1400-1777.
- (d) A YUL symbol defined in the program being run.
- (e) A YUL symbol as above, followed by at least one blank followed by a signed numeric modifier. The rules for the numeric modifier are the same as in (a) above.

- (f) A channel number may be indicated instead of an AGC location by CHANNELXX where XX is the two digit octal channel number.

In addition, some requests use the YUL assembler's definition of blank and signed numeric fields. See BL and SA definitions below.

3. C. 3 - Glossary of Special Requests

The following codes may appear in the OP code fields of special requests. The classifications are defined following this OP code listing.

<u>OPCODE</u>	<u>Classification</u>	<u>OPCODE</u>	<u>Classification</u>
ABORT	C3	GOSEQ	C3
ADRES	C2, BL	GOSEQOK	C13
ADVAN	C3	IGOSEQ	C3
BACKGRND	C1, NR	ICOUNT	C3
BKGRND	C1, NR	ITRACE	C3
BBCON	C2, BL, SA	ITRSTOP	C3
2BCADR	C2, BL, SA	KEYSTRK	C3
2CADR	C2, BL, SA	MAXTIME	C1
CADR	C2, BL, SA	MSG	C3
CFDUMP	C3	NOGOSEQ	C13
CHANGE	C2, BL	NOPRINT	C13
CHANL	C2, NR	OCTAL	C2, BL
CLOCK	C3	ONPRINT	C13
CONST	C2	(Op-codes)	C2, BL, SA
DEC	C2, BL	REPLACE	C3
DEC*	C2, BL	SNAP	C13
2DEC	C2, BL	STANDBY	C3
2DEC*	C2, BL	START	C1, NR
DELETE	C13	STOP	C3
DUMP	C3	TCSGO	C1
ECADR	C2, BL, SA	TELEM	C1
ENVMODE	C3	TEMPCHK	C1
ENVSUM	C3	TLOSS	C13
EQUALS	C2, SA	TRACE	C3
EXERASE	C1	TRSTOP	C3
2FCADR	C2, BL, SA	UNUSED	C2
		UPTL	C3

The special request classification definitions are as follows:

C1

The special request is Class 1. Class 1 requests exercise their specified control on the simulation from the beginning of the simulation (or if newly inserted during rollback, from the re-start) until superceded by an overriding Class 1 or Class 3 request, if any. Conditional clauses cannot be appended to Class 1 requests.

C2

The special request is Class 2. Class 2 requests override either the contents or the assembler appended flags or both of one or more AGC locations. The change takes place before the start of simulation. For Class 2 requests submitted at rollback, the request overrides the contents or flags or both of the AGC locations as they existed at the snapshot to which the rollback is to, excepting that these rollback requests may not apply to the channels or locations 0-60. Class 2 requests are mainly used to patch AGC coding or to initialize erasable. Since these patches are applied at the beginning of a run, conditional clauses cannot be attached to Class 2 requests.

C3

The special request is Class 3. The execution of a Class 3 request is triggered during the simulation by one of 2 methods.

- (a) If the location field specifies an AGC location or channel, the request is triggered whenever that AGC location or channel is accessed, whether for the next AGC instruction or for an operand.
- (b) If the location field specifies a time, the request is triggered when the simulator's clock has incremented to or beyond that

time. The delay, if any, will usually not exceed 50 microseconds but may be as long as 1 millisecond. The time in the location field should be a decimal real number and of such length (9 characters or more is safe) so that no possible AGC location value conflict results. The simulator's clock is set by the START special request and is not alterable by program or special requests during the run. Though a 10 millisecond increment of time on the simulators clock, the channel 3/4 clock, and the TIME2/TIME1 clock are identical, the absolute value of the 3 clocks is not the same unless the programmer sets, via Class 2 requests, the channel 3/4 clock and the TIME2/TIME1 clock to agree with the simulator's clock. The simulator's clock is the master clock used by MAC programs in flight and astronaut only simulations.

Initial environment time is not set equal to the initial simulator time, as specified on the START card, however. To synchronize time between the AGC Instruction Simulator and the environment, the user must set the UNIVERSE input parameter TZERO to the initial simulator clock time, truncated to 10 milliseconds. See Section 3. F. 2.

Class 3 requests, once triggered, are actually executed only if attached conditions, if any, are satisfied. To make a Class 3 request conditional, two requirements must be met. First, the word CONDITIONAL must appear in the condition-indicator field (column 65-80) of the last or only card of the request (exception, see MSG). Second, the last or only card of the request must be immediately followed by a card or cards containing condition opcodes in their opcode fields. Descriptions of condition codes and multiple conditions are given in Sections 3. C. 5 and 3. C. 6.

Each Class 3 request is assigned an ID number for use by the DELETE special request and for identification on the MARSROT.

C13

The special request is Class 1 if the location field is blank and is Class 3 if the location field is non-blank. See definitions for Class 1 and Class 3 requests.

NR

This special request is not to be used in rollbacks.

BL

For this request the location specified in the location field (channels excepted) is set into a location counter much the same as the YUL assembler does. Subsequent BL type requests may have the location field blank and the simulator will assign the location field value from the location counter. Each BL type request causes the location counter to be incremented (after the request is fully processed) by the number of AGC locations used by the request. Intervening non-BL requests have no effect upon the location counter setting.

SA

Unless otherwise prohibited, these requests may have required address fields blank or signed numeric. The value of the field is determined as in the YUL assembler.

3. C. 4 - Description of Special Requests

ABORT

The simulation aborts just before accessing the AGC word specified in the location field but after execution of any other special requests addressed to this location. The data address field is ignored. See MAXTIME request for note about rollbacks.

ADRES

This request causes the location specified in the location field to be loaded with a constant which is the low 12 bits of the address of the location given in the data address field. (2000-3777 for locations in switched banks and 1400-1777 for locations in erasable banks E3 and up). The remarks field is ignored.

ADVAN

This request may be used to speed up the simulation at points where the AGC is idling while waiting for an interrupt caused by a time counter. It is virtually mandatory to make the ADVAN request conditional; further, the condition should be one whose outcome is altered by the program executed in the interrupt mode. The reason for this is that the instruction to which the ADVAN request is attached is not executed when the condition on the ADVAN is satisfied. As a result, the first instruction executed after the RESUME takes place is again the one to which the ADVAN is attached. This would result in another clock-advance immediately, unless the condition were no longer satisfied.

The data address field is ignored.

BACKGRND

At the beginning of a simulation but not at rollback, any erasable locations that have not been initialized by either the assembler or the special requests are loaded with a constant background value which is randomly generated once for the entire run. This request allows the user to specify the background value by a 5 octal digit integer in the address field. All other fields are ignored.

BKGRND

This request generates background from the six decimal digit H1800 log number in the address field.

BBCON

This request generates a constant word suitable for placement into the BBANK register. Address field 1 must either specify a location in a fixed bank, or be equivalent to a bank number (0-37). Bits 15-11 of the word generated are the fixed bank number, bits 10-4 are zeros, and bits 3-1 are the EBANK number for the location specified in address field 2, which must not be blank. The word generated is flagged as a constant. The remarks field is ignored.

2BCADR

2BCADR is a synonym of 2CADR.

2CADR

This request generates two constant words (in L and L + 1) suitable for use with a DXCH Z instruction. The first word is identical to that which would be generated by the ADRES special request (using the contents of address field 1). If field 1 specifies

a location in fixed memory, field 2 must specify an erasable location. In this case, fields 1 and 2 are used to generate the second word under the rules for BBCON. If field 1 specifies an erasable location, field 2 must be blank; the second word generated in this case is 0000X, where X is the EBANK number (0-7) for the field 1 location. Both words are flagged as constants. The remarks field of this request is ignored.

CADR

This is like ADRES except that the constant loaded is the complete address of the location specified in the data address. The remarks field is ignored.

CFDUMP

In flight mode simulations, certain key quantities are maintained in the MAC call file by the environment programs. To correlate these environment variables (e. g., position, velocity, attitude, etc.) with AGC versions of similar quantities, call file dumps may be used. Execution of a CFDUMP request causes the environment to be updated to the time of execution of the flagged instruction, followed by writing of locations 100 through 249 of the call file into MARSROT. (The AGCEDIT program is capable of retrieving this information; see the section below.) The data address field of this request is ignored.

CHANGE

This special request may be used as an octal patch card (see also OCTAL). The data address must contain one to five octal digits, optionally preceded by a sign; the location field contains the relevant address. This card is used where the status, as well as the content, of the YUL program word in that location is to change. From the point of view of simulation, the YUL program consists

of four kinds of information: instructions, constants, unused locations, and bad words. The classification is made at YUL assembly time (e. g. DEC, OCT, etc. become constants, and TC, CCS, etc. become instructions; misspelled opcodes or tags cause words to be marked "bad"). The simulator will not execute constants, bad, or unused locations, but will come to an abort stop. The CHANGE card directs the simulator to remove any constant, unused or bad-location flags associated with the location in question. The remarks field is ignored.

CHANL

The CHANL request may be used to specify the initial contents of any existent Block II channel register. The card is interpreted as OCTAL is, except that the location field is treated as a channel number rather than a memory address.

CLOCK

This card directs the simulator to display the running time in seconds whenever the instruction specified in the location field is executed, unless suppressed by means of a NOPRINT or ONPRINT request. The data address field is ignored.

CONST

This card directs the simulator to plant a constant flag in the location specified in the location field. This will cause the simulator to abort should the instruction be executed. The data address may be used to specify a range of addresses that may not be executed. Here, constant flags will be planted from the location in the location field through the location in the data address. This option is used when a given simulation run is intended to execute only a portion of the program that is actually loaded. Should the program go astray, the simulator will abort the run and dump at the instant an unwanted section of program is executed. In this way, one may check out large programs a little at a time. The remarks field of this request is ignored.

DEC, DEC*, 2DEC, 2DEC*

These four requests load constants whose values are entirely equivalent to such constants generated by the YUL system. The arguments for the unstarred versions may lie anywhere in data address field 1; imbedded blanks are allowed. The arguments for the starred versions may begin in the first column of data address field 1, and end at the asterisk supplied by the user. DEC and DEC* generate one-word constants; 2DEC and 2DEC* generate two-word constants, which are placed into the location specified in the location field, and the word following.

DELETE

This request causes the simulator to disable the Class 3 request whose ID is given in the address field. All AGC instruction simulator requests have an ID assigned when first encountered in any initialization phase. A disabled request uses up some execution time but otherwise has no effect upon the simulation.

DUMP

A DUMP special request consists of one or more dump cards and causes selected portions of AGC erasable memory and/or of the channels to be copied onto MARSROT for post-run editing.

For each DUMP card, data address fields 1 and 2 determine the range to be dumped. If data address field 2 is blank, only the single location specified in data address 1 is dumped. If data address field 2 is non-blank, then its value must be greater than or equal to the value of data address field 1, and both fields must refer to the same type of memory, i. e., both to AGC erasable locations or both to channels.

The DUMP request can cause more than one area of memory to be dumped by concatenation of DUMP cards. The concatenation

is indicated by the single word MORE in the remarks field. The next card must be a DUMP card as defined in the preceding paragraph excepting that the location field must be blank. Any reasonable number of DUMP cards may be concatenated, and if the entire DUMP request is to be conditional, the word CONDITIONAL must appear in the remarks field of the last DUMP card of the concatenation. Channels and E-memory may be dumped by the same DUMP request but not by the same DUMP card.

ECADR

This request is like CADR, except that an 11-bit erasable memory address (0-3777) is generated. The location is flagged as a constant. The remarks field is ignored.

ENVMODE

During flight test simulations, this request causes the characteristics of the environment model, and therefore the fidelity of the simulation to be modified, according to the specification indicated by the integer in the address field, as follows:

ENVMODE 0 Accurate IMU, Accurate Vehicle

ENVMODE 1 Fast IMU, Accurate Vehicle

ENVMODE 2 Fast IMU, Fast Vehicle

A full description of the use of this special request is contained in Digital Simulation Memo #14-67 dated 22 September, 1967.

ENVSUM

This request causes an environment summary to be printed during flight mode simulations. The data address field of the request must contain a decimal integer corresponding to a valid display format (i. e., one defined in the relevant environment programs). This format may include arbitrary combinations of printing, punching, or data-filing as desired. Appendix C contains a description of the current format and an example of its use. At present, only one display format is implemented and any decimal integer in the data address field will produce the same display format, with two exceptions.

The exceptions are used to trigger special functions within the Environment, as follows:

- | | |
|-----------|--|
| ENVSUM 10 | Forces an internally generated summary print every DT seconds where $DT = 4$ unless overridden by an Environment input card. See INPUTPRG data card format in Chapter 3, Section F. 2. |
| ENVSUM 50 | Initiates a trap to catch and record the time of occurrence of the next 3 PIPA pulses. (Used in Prelaunch Earth rate tests). |

EQUALS

The symbol in the location field is defined, for the simulation only, to be the value specified in the address field. If the symbol has been previously defined, a complaint is made but the EQUALS definition is accepted. The EQUALS card must appear in the simulation deck before the special requests wherein the new definition of the symbol is to be used. Special requests appearing before the EQUALS card and using this symbol will use the most recent previous definition of the symbol or, if none, a fatal loader error results.

Use the EQUALS special request with extreme caution, doubly so in AGCEDITs.

EXERASE

This card allows execution of instructions other than TC's or TCF out of erasable. Normally, because of indexing capabilities, programs will not be executed in erasable memory. The

simulator will abort at the first non-TC or TCF instruction that is executed in erasable, unless an EXERASE card is included among the special requests. The location, data address, and remarks fields are ignored.

Caution: EXERASE has no effect upon the special/central/counter locations. The simulator always allows execution from A and L. Only TC's may be executed from Q. No instructions may be executed from locations 3 to the end of the special/counter region.

2FCADR

This request generates two constant words (in L and L + 1) suitable for use with a DXCH FBANK instruction. The address field value must be a fixed bank AGC location. The first word is generated under the rules for CADR; the second word is generated under the rules ADRES.

GOSEQ

This causes the simulator to execute a GO sequence (restart) instead of the next AGC instruction. The data address field is ignored.

GOSEQOK

This request sets the simulator's general inhibit/allow switch controlling GOSEQ and 1GOSEQ requests to the allow state. Subsequent GOSEQ and 1GOSEQ requests will be allowed if their own attached conditions, if any, are also satisfied.

1GOSEQ

This is like GOSEQ, except that it effectively disappears immediately after causing the GO sequence.

ICOUNT

This request counts the number of times the specified location is executed. Two options are available: If the data address field is non-blank, its contents are treated as a tag or numeric address of a location in erasable memory in which the count is to be kept. This option permits the count to be examined by conditional requests and to be included in dumps onto MARSROT. It is potentially hazardous, however, since no checks are made by the simulator to insure that the erasable location is not otherwise referred to during the run. The simulator zeroes the location at the time it encounters the ICOUNT request. The second option is that of blank data address fields. In this option, the count is kept in a non-AGC location, making it inaccessible to the AGC program. Upon run termination a list of the count tallies for blank data address ICOUNT requests is printed.

ITRACE

The simulation user can, by appropriate use of ITRACE and ITRSTOP requests, cause information to be written on MARSROT so that a post-simulation AGCEDIT run can generate a pseudo-trace printout of about one line per interpretive operation code, showing relevant "before" and "after" contents of addressed locations.

To allow for the wide range of Class 3 request triggering, the ITRACE request causes the simulator to turn on I-tracing when it next encounters an opcode pair or store word in the AGC's A register ready for interpretive opcode dispatching and if I-tracing is off. (I- tracing being off includes having just gone off at the completion of interpretive execution of the previous opcode pair or store word). The interpreter does not recognize an opcode pair or store word until after the word has been accessed from memory. (i. e., when it is in the A register). Requests attached to those AGC words are executed before the word is accessed.

If the first address field of the ITRACE card is non-blank, the simulator uses this value as the location field for a simulator-created (but not printed) ITRSTOP request. This ITRSTOP request will turn off I-tracing only if the I-tracing was turned on by the associated ITRACE request.

If the first address field is blank and the second address field contains a decimal integer, and if the ITRACE request succeeds in turning on the I-trace, then I-tracing will be turned off after this many opcode pair/store words have been I-traced.

ITRSTOP

This request causes the simulator to turn off I-tracing after completing interpretive execution of the opcodes associated with the opcode pair or store word next encountered in the AGC's A register.

KEYSTRK

This request provides the means to introduce keystroke and mark bits and the associated interrupt into the AGC without the use of the environment. The data address field contains the address (less than 10000) of the first word of a list, which may be patched or assembled into the program. The last word of the list must have an address less than 10000. If the contents of the word in the list exceeds 37 (octal), the word is placed into channel 16, triggering interrupt 6; otherwise, it goes into channel 15, and triggers interrupt 5. If the word to be delivered is flagged as unused (see UNUSED), the equivalent of an ABORT special request is executed instead, providing a simple means of ending the process.

MAXTIME

To abort a simulation run at a specified value of AGC time, the MAXTIME request is used. The location and remarks fields are ignored; the data address field may contain up to 8 decimal digits. This decimal integer is compared with the content of the AGC clock once per AGC second. The simulation is aborted when the specified time in seconds has been reached.

The 360 operator can request a type out of simulator time and maxtime whenever he wishes. This helps him to determine the progress of your run and to decide, if your job time estimate is exceeded, whether to force an abort or to allow normal completion of the job.

When the simulation ends as a result of a MAXTIME, ABORT or STOP special request, and when a terminal snapshot has been taken, the user may rollback the simulation to that terminal snapshot by deleting the ABORT or STOP request or by inserting a new MAXTIME request, whichever one of the requests ended the simulation.

MSG

The address field contains a decimal integer indicating the number of subsequent cards, columns 9-80, that are to be printed, one line per card, whenever the special request is executed. The request consists of the MSG card plus all the cards containing information to be printed. The second address field of the MSG card is ignored. If the request is conditional the word CONDITIONAL appears in the remarks field of the MSG card.

NOGOSEQ

This request sets the simulator's general inhibit/allow switch controlling GOSEQ and 1GOSEQ requests to the inhibit state. Subsequent GOSEQ and 1GOSEQ requests will be ignored (i. e., will not be executed).

NOPRINT

Normally, significant events of the run are logged on the printer along with the total elapsed running time in seconds. Events so recorded, and their code names are:

Interrupts	RUPT
Resumes	RESUME
Inhibits	INHIBIT
Allows	ALLOW
Fixed-bank switching	BANKSW
Erasable-bank switching	EBANKSW
Storage in Output Registers	OUTX
Erasable Dumps on MARSROT	EDUMP
Clock Printout	CLOCK
Call File Dumps	CFDUMP

(The last three occur under special request control).

To enable the customer to exercise some control over the listing of such information, the two special requests NOPRINT and ONPRINT are provided. The simple request NOPRINT turns off all printing of the above messages. Optionally, beginning in data address field 1 and continuing through column 80 if necessary, one or more of the code names above, separated by blanks or commas, may be listed. For example, the card NOPRINT OUTX, BANKSW will suppress printing of references to output registers and bank-switching messages, but does allow printing of all other events.

The word ALLRUPT may optionally be used: it is equivalent to the listing of all four of the words RUPT, RESUME, INHIBIT, and ALLOW.

The ONPRINT request is similar in interpretation, but opposite in effect. For example, the card ONPRINT ALLRUPT, EDUMP turns off printing of all messages except the four interrupt types, and the erasable-dump message. The use of the ONPRINT card with no list is legal; it turns on all printing and prints a fantastic amount. In the absence of an ONPRINT or NOPRINT card, the simulator considers all event printing off.

OCTAL (optional spelling OCT)

The OCTAL card is like the CHANGE card, except that constant or bad-word flags, if any, are preserved. The remarks field is ignored. (See also EPUNCH).

ONPRINT

See NOPRINT.

Op-Codes

The following op code words are recognized: TC, CCS, TCF, DAS, LXCH, INCR, ADS, CA, CAF, CS, INDEX, DXCH, TS, XCH, AD, MASK, READ, WRITE, RAND, WAND, ROR, WOR, RXOR, DV, BZF, MSU, QXCH, AUG, DIM, DCA, DCS, SU, BZMF, and MP. It is the customer's responsibility to use these legitimately; the simulator will not, for example, complain about a TCF addressed to erasable memory: a CCS will result. Further, no checking of extended - nonextended status takes place. Also recognized are the implied-address codes EXTEND, INHINT, RELINT, and RESUME. The remarks field is ignored.

REPLACE

When executed, this request causes the contents of the register specified in field 1 to be replaced by the contents of the register specified in field 2 (both addresses must be smaller than 10000 octal).

SNAP

This request is used to provide restart capability for long simulation runs. A " * SNAPSHOT" MAC Run Step Control Card and a HOLD MARSROT director card must be used in conjunction with the SNAP request; otherwise the SNAP request has no effect on the run. Records are written on MARSROT during the initialization and at specified intervals during the simulation run. This permits restarting the simulation by executing a ROLLBACK at an arbitrary snapshot (see Chapter 4). The data address field of the SNAP request may be blank, or contain up to 8 decimal digits to specify the number of AGC seconds between snapshots.

If no SNAP request with a blank location field is provided or if such a card also has its address field blank, a snapshot interval of 600 seconds is assumed.

If the SNAP request has a non-blank location field (i. e., is a Class 3 request), a snapshot is taken when the request is triggered during the run. Further, if such a request has a non-blank address field of up to 8 decimal digits, the snapshot interval is reset to that value; its cycle period is set to begin with the time of the snapshot caused by this Class 3 request.

Any number of snapshots may be taken in any single AGC second, but a rollback may be made only to the first snapshot in any AGC second.

STANDBY

This request may be used when it is desired to simulate a period of low-power computer operation. The data address field contains a decimal constant (digits, with decimal point if desired) which specifies the time interval of standby operation. This interval

begins at the time at which the instruction specified in the location field would have been executed, given that the condition attached (if any) is satisfied. During the interval, only Channels 3 and 4 are changed; at the end of the interval, a GO sequence is executed to simulate the power turn-on operation. When the STANDBY is executed, the request disappears, so that at most one standby occurs per request.

START

This card specifies the location from which the simulation will select the first instruction to be executed. Any location may be specified, except that starting in switched erasable-memory is illegal. If the starting address is in switched-fixed, the contents of the appropriate bank registers are adjusted accordingly. If the data address field is non-blank, its contents are interpreted as a decimal number (which may include a decimal point) whose value is to be used to preset the simulator clock at the start of the run. The initial simulator clock time is expressed in seconds; it is truncated to a multiple of 10 milliseconds within the simulator.

To synchronize time between the AGC Instruction simulator and the environment, the UNIVERSE input parameter TZERO (see Section 3. F. 2) must agree with the initial simulator time specified on the START card, and both times must be truncated to 10 milliseconds.

STOP

This card specifies the location at which to end the simulation. Other special requests attached to the location are processed before the run is stopped, but the instruction itself is not. The data address field is unused. See MAXTIME for note concerning rollbacks.

A STOP request is the only way to stop a MAC controlled simulation without causing a MAC Abort.

TCSGO

If this request is used, any TC to self (except from A) or TCF to self, triggers execution of a GO sequence instead of the normal abort. The location, data address, and remarks fields are ignored.

TELEM

The data address field of this request specifies in octal the number of 10 millisecond steps (i. e. , TIME 1 increments) to elapse between generated down-telemetry interrupts. The location and remarks fields are ignored.

TEMPCHK

This request checks a number of erasable registers to be sure they are unchanged during certain intervals. The registers MPAC through MPAC + 11 are checked at the execution of a RESUME to verify that they have not been modified while the program was in the interrupt state. The registers ITEMP through ITEMP + 18 are checked to ensure that they have not been modified except where the interrupt state is inhibited or when the computer is in the interrupt mode.

TLOSS

The purpose of this request is to allow the user to control the number of AGC instructions executed in any time period, and thus test the AGC's ability to withstand execution time loss due to counter jitters attributable to noise. The address field contains a value between 0 and 1.0 expressing the ratio: time to be lost over total available time. This request affects time accounting only, not counter values, and does not delay interrupts or input settings. The simulator assumes a loss ratio of zero in the absence of this request.

For example, the request:

TLOSS .2

results in a 20% loss of instruction execution time, so a given task will take approximately 25% longer to execute.

TRACE

This card instructs the simulator to print an instruction-by-instruction trace. If the simulator is already tracing, this request is totally ignored. The location field specifies the trace

starting location or time (see Class 3 special request definition, Section 3. C. 3). The data address field may, but need not, specify a trace stopping location or time. CHANNELXX placed in the location field initiates a trace upon a reference to CHANNELXX (see Section 3. C. 2, part f). If the first address field is blank and the second address field contains a decimal integer, and if the TRACE request initiated a trace, then tracing will be turned off after the number of instructions specified in the second address field have been traced. The printing of the trace is unaffected by the NOPRINT or ONPRINT requests.

TRSTOP

This is an unconditional trace stop, no matter how the trace was started, excepting tracing initiated from computer console. The data address field is ignored.

UNUSED

This request is like CONST, except that unused-location indicators replace the words involved. This will cause the program to abort should an attempt be made to reference such a location, either as data or as the next instruction.

UPTEL

This request is similar to KEYSTRK in its list-delivery character. In UPTEL, however, the 15-bit contents of words in the list are placed into counter 45 and interrupt 7 is set. Like KEYSTRK, the attempted delivery of a word flagged as unused causes the run to be aborted. This request is quite useful in checking the response of the AGC program to bad frames of telemetry information.

3. C. 5 Glossary of Special Request Conditions

Condition codes associated with Class 3 special requests:

L	LEQ
LC	LEQC
G	GEQ
GC	GEQC
EQ	UNEQ
EQC	UNEQC
EVERY	OVERFLOW
BEFORE	CYCLE
AFTER	

3. C. 6 Description of Special Request Condition Cards

The simulator expects condition cards following Class 3 request cards if the last card (except MSG) contains the word **CONDITIONAL** in the remarks field (Column 65-80). Each special request condition card contains a condition opcode in the opcode field. Condition cards may be concatenated by the word **OR** or the word **AND** in the remarks field of the preceeding condition card.

Conditions are processed serially. If a condition is true and the remarks field contains **AND**, or if a condition is false and the remarks field contains **OR**, the next condition is executed. If a condition is true and the remarks field does not contain **AND**, condition processing stops and the special request to which the conditions were attached is executed. If a condition is false and the remarks field contains **AND**, the next condition is declared false without its execution, except to take action on the contents of its remarks field. If a condition is false and the remarks field does not contain **AND** or **OR**, condition processing stops and the special request to which the conditions were attached is not executed.

The following conditions compare the contents of the erasable or fixed fixed register addressed in the location field with the 15 bit pseudo address or integer in the first address field under a 5 octal digit mask specified in the second address field (blanks implies 77777).

L	Less than	+0 = -0
G	Greater than	+0 = -0
EQ	Equal to	+0 ≠ -0
LEQ	Less than or equal to	+0 = -0
GEQ	Greater than or equal to	+0 = -0
UNEQ	Not equal to	+0 ≠ -0

Similarly, the above opcodes with a suffix C cause the contents, rather than the address, of the erasable or fixed-fixed location specified in the first address field to be used in the comparison.

The condition EVERY requires that an integer N (decimal with D, or octal) of octal value 7777 or less be specified in the data address field. The condition is true every N times. For example, EVERY 3 is satisfied the third time, the sixth time, etc. The location and second address fields are blank. If EVERY is used in a condition concatenation, its counter is incremented only if the condition is actually tested.

The condition OVERFLOW is true if the AGC accumulator contains positive or negative overflow. The location and address fields are ignored.

The condition AFTER is true if the decimal real number time value in the first address field is less than the current AGC time in the simulator's clock. The location and second address fields are ignored.

The condition BEFORE is true if the decimal real number time value in the first address field is greater than the current AGC time in the simulator's clock.

CYCLE causes a time triggered special request to be scheduled for repeated occurrence. CYCLE is not an actual condition because it is always executed regardless of the true/false results of previous conditions in the concatenation. CYCLE must only be attached to a time-triggered special request. The address field contains a decimal real number time interval that causes the special request to be triggered in cyclic fashion throughout the simulation.

3. D Structure of Input Deck for Benchtest Runs

The Benchtest mode of operation is the simplest method of utilizing the digital simulation. As shown in Fig. 2. A. 2, the Benchtest mode employs only the AGC Instruction Simulator. The simulation proceeds open loop without a Communicator or Environment. References by the AGC program to input-output counters or channels are ignored and external interface interactions, such as astronaut activity or input counter incrementation are not simulated.

The structure of a typical Benchtest input deck is shown in Fig. 3. D. 1, and consists of the following types of input cards:

1. IBM-360 Job Step Control Cards.
2. " * SIMULATE AGC" card.
3. AGC program source card.
4. " HOLD MARSROT " card (optional)
5. AGC Instruction Simulator Special Request cards.
6. Terminator card (/ *).

See Section 3. B and 3. C for a description of these input cards. Simulation input decks are normally followed by an edit deck, the structure and format of which is described in Chapter 5. Up to date example decks, including a Benchtest run, followed by an edit, may be found on the Digital Simulation Group Bulletin Board in the Assembly Control Room.

In a Benchtest simulation, the sequence of operations performed to set up, initialize and run the simulation include the following:

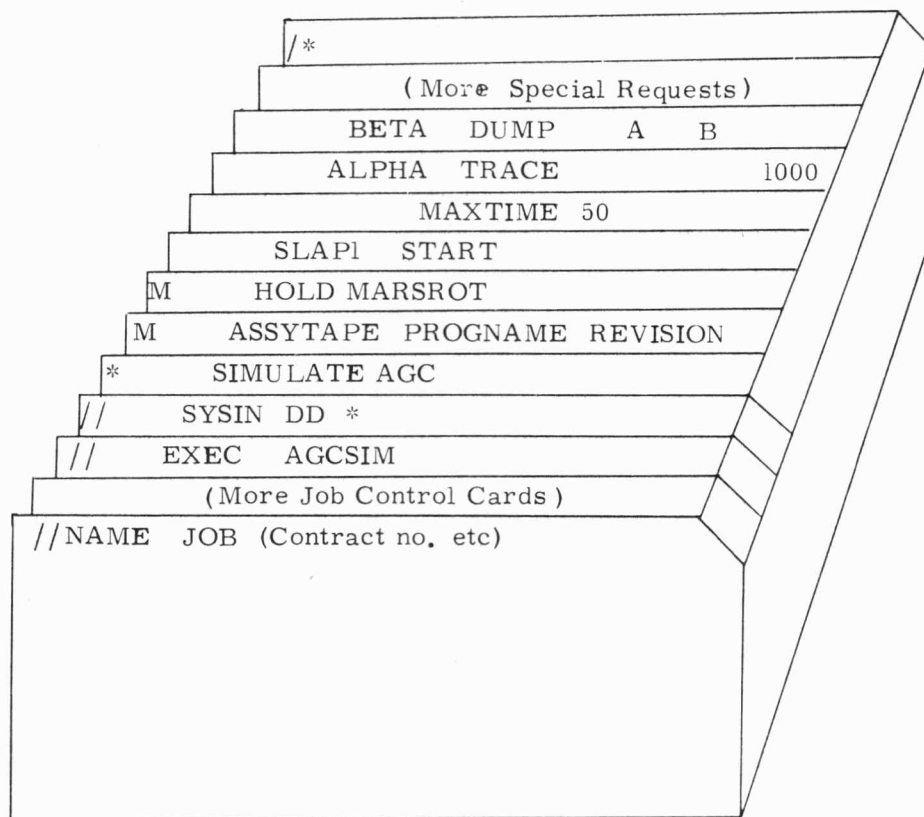


Figure 3.D.1 Structure of Bench Test Simulation Input Deck

1. Processing of IBM 360 Job Step Control Cards.
2. Loading of AGC Instruction Simulator.
3. Loading of AGC program from disc.
4. Processing of AGC special requests.
5. Initiation of instruction by instruction AGC simulation.
6. Printing, dumping and tracing of AGC variables.
7. Termination of run by AGC special request, abort, or exit condition.

The initiation of the above sequence of operations for a Benchtest run is controlled by the EXEC card included in the IBM 360 Job Step Control Cards. Execute procedures, specified on the EXEC card determine the type of run to be initiated, and govern the structure of the simulation input deck. Specifically, an EXEC card of the form:

```
// EXEC AGCSIM
```

is identified with a Benchtest simulation. As a result of the recognition of the "EXEC AGCSIM" card, by the 360 operating system, the AGC Instruction Simulator is loaded and called, and the above sequence of operations for a Benchtest run is executed.

Various other execute procedures are available, each of which is associated with a specific type of simulation run. For instance, Flight tests are currently controlled by the procedure EXEC MACSIM, AGC Edit runs by EXEC AGCEDIT, and MAC Edit runs by EXEC MACEDIT. These execute procedures are subject to change as the IBM-360 facility evolves, and thus users are referred to the Digital Simulation Bulletin Board for the latest control card formats and run deck configurations.

3. E Structure of Input Deck for Astronaut Only Runs

The Astronaut only mode of operation of the Simulator may be considered to be a subset of the Flight test mode. As shown in Fig. 2. A. 2, the Flight test mode incorporates the AGC Instruction Simulator, the Communicator, and the Environment. In the case of Astronaut only runs, the Environment consists only of the Environment Control Program (ENVCNTRL) and the Astronaut Program (ASTRNAUT). The function of the astronaut only mode is to enable AGC programmers to simulate the interaction of an Astronaut with the Apollo Guidance Computer, without introducing a flight environment. The Astronaut only mode is useful for checkout of AGC programs such as the display and keyboard routines, or whenever dynamical interaction of the Astronaut with a flight environment is not necessary.

As Astronaut only run deck is shown in Fig. 3. E. 1 and consists of the following categories of input cards:

1. IBM-360 Job Step Control Cards.
2. " * COLLECT ENVCNTRL" card.
3. "NOPROG" cards.
4. " * SIMULATE AGC" card.
5. AGC Program Source card.
6. "HOLD MARSROT" card (optional).
7. AGC Instruction Simulator Special Request cards.
8. MAC input card to INPUTPRG.
9. Astronaut input cards.
10. Terminator card (/ *).

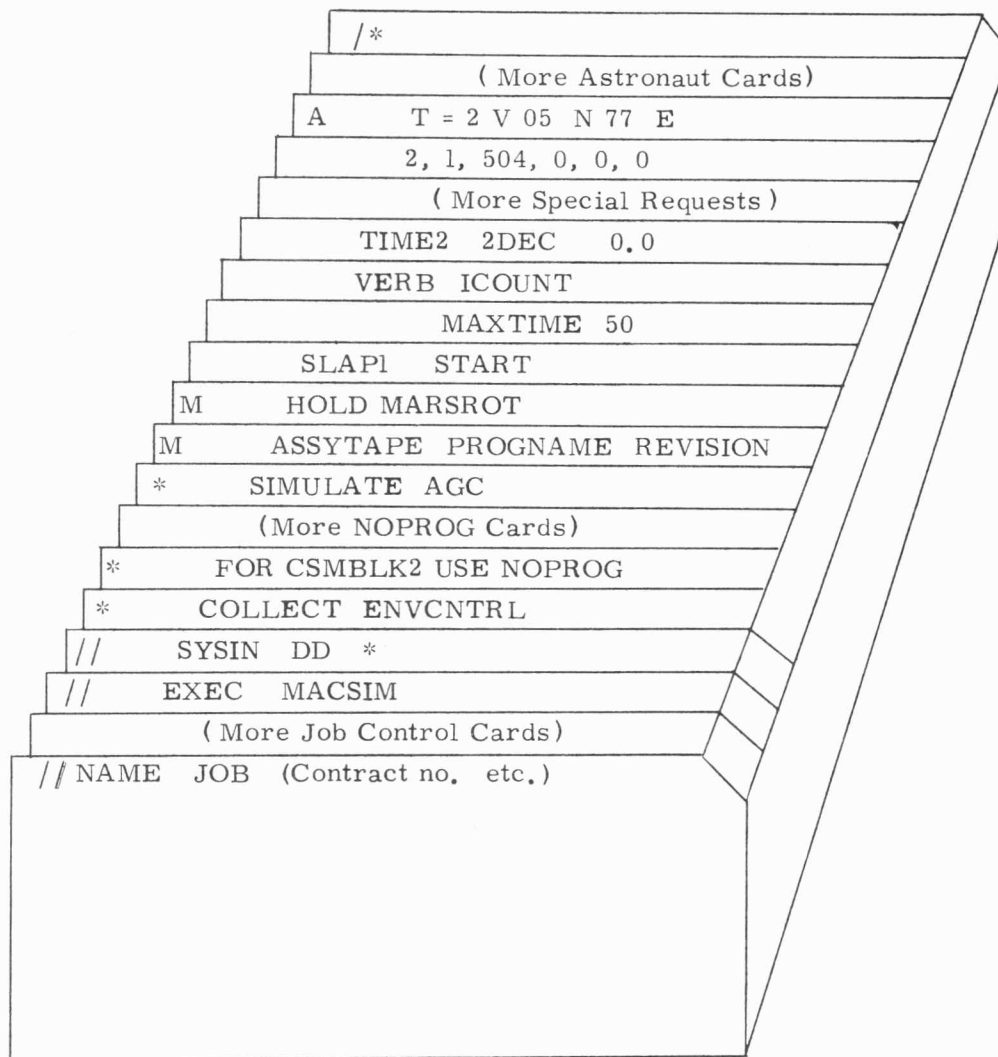


Figure 3.E.1 Structure of Astronaut Only Simulation Input Deck

The initiation of an Astronaut Only run is controlled by specifying the execute procedure:

```
// EXEC MACSIM
```

in the IBM 360 Job Step Control cards, and by providing Astronaut cards but no vehicle cards in the Environment input deck. The " * COLLECT ENVCNTRL " card loads the entire complement of the MAC Environment programs identified in Fig. 2. A. 5. Programs not required in Astronaut only runs (such as the vehicle programs PRELUNCH, SATURN, CSMBLK2, LEM, and LVCBLK2, as well as RADAR and OPTICSB2) may be bypassed in the MAC program collection process by the inclusion of a "NOPROG" card for each program to be bypassed.

A description of the " * SIMULATE AGC" card as well as the Program Source Card and the AGC Instruction Simulator Special Request Cards may be found in Sections 3. B and 3. C, while the format of the INPUTPRG input card and the Astronaut input cards are included in Section 3. H.

In an Astronaut only simulation, the sequence of operations performed to set up, initialize, and run the simulation include the following:

1. IBM 360 Job Step Control Cards are processed.
2. MAC environment programs are loaded and the main program (ENVCNTRL) is called.
3. The AGC Instruction Simulator is loaded and called by ENVCNTRL.
4. The AGC program is loaded from disc.

5. AGC special requests are processed.
6. The Environment is called for an initialization pass. All Astronaut cards are read and the Astronaut program (ASTRNAUT) is initialized.
7. The instruction by instruction AGC simulation is initiated, and interacts with the Astronaut program.
8. AGC variables and MAC environment variables are printed, dumped and traced under user control.
9. The run is terminated by AGC special request, abort, or exit condition (including MAC abort).

3. F Structure of Input Deck for Flight Test Runs

The flight test mode incorporates the AGC Instruction Simulator, the Communicator, and the Environment, as shown in Fig. 2. A. 2. Unlike the Astronaut only mode which utilizes only a subset of the Environment, the Flight test mode employs all of the programs depicted in Fig. 2. A. 5. The Flight test simulation mode should be used if dynamical interaction between the flight environment (vehicle hardware, gravity, stars etc.) and the Apollo Guidance Computer is significant.

A description of the AGC Instruction Simulator, the Communicator, and the Environment, as well as a discussion of the interaction between elements of a Flight test simulation is included in Chapter 2. This chapter also contains documentation on the various models used in the environment programs to simulate vehicle hardware, in addition to a tabulation of the sequence of operations performed during an environment initialization.

A representative Flight test run deck is presented in Fig. 3. F. 1 and consists of the following categories of input cards:

1. IBM-360 Job Step Control Cards.
2. " * COLLECT ENVCNTRL" card.
3. " NOPROG" Cards.
4. " LOAD FILE" Cards.
5. " * SIMULATE AGC" Card.
6. AGC Program Source card.
7. "HOLD MARSROT" card (optional).
8. AGC Instruction Simulator Special Request Cards.

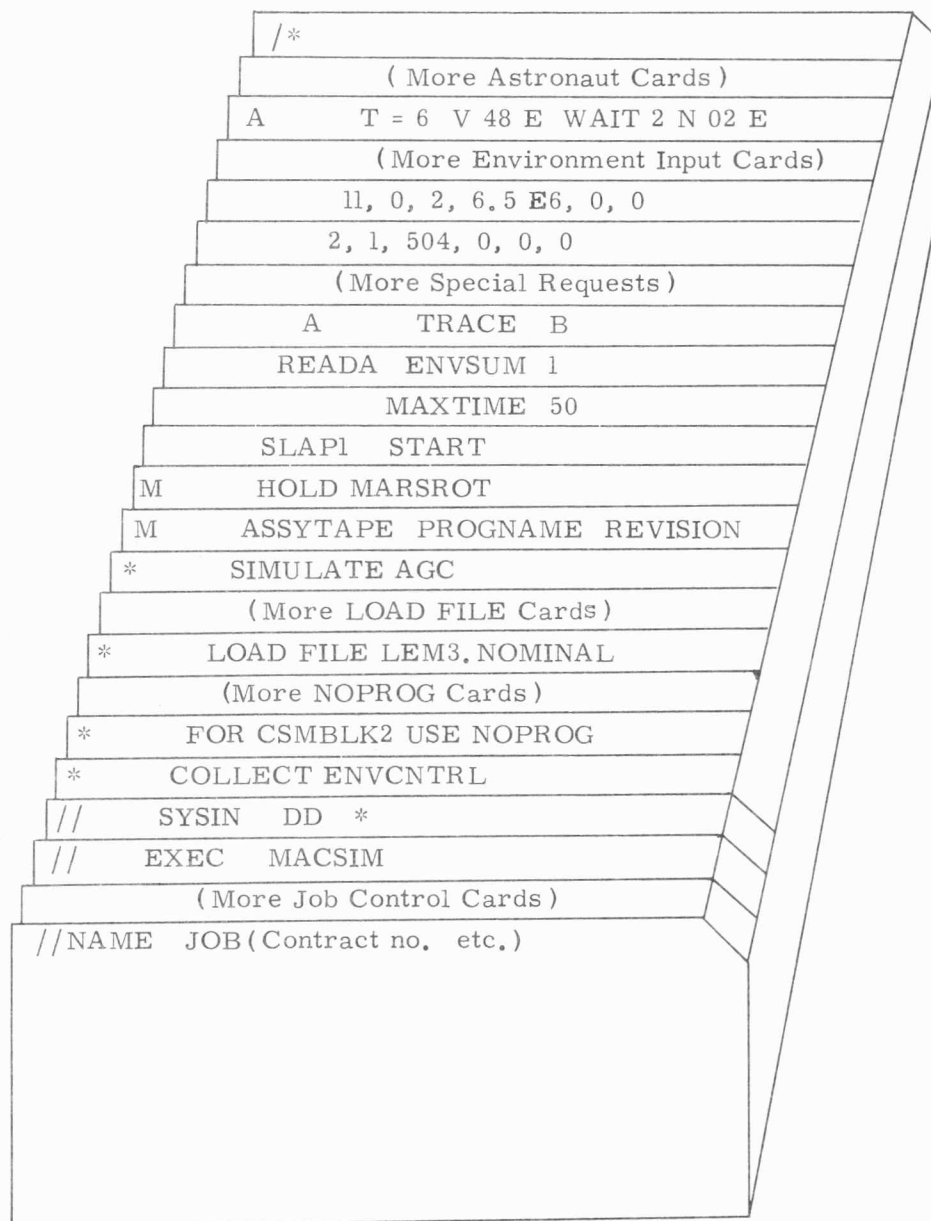


Figure 3.F.1 Structure of Flight Test Simulation Input Deck

9. MAC input cards for all active environment programs.
10. Astronaut input cards.
11. Terminator card.

A Flight test run is identified and initiated by specifying the execute procedure:

```
// EXEC MACSIM
```

in the IBM 360 Job Step Control Cards, and by including input cards to one or more environment vehicle programs (PRELUNCH, SATURN, CSMBLK2, LEM, or LVCBLK2). The "*" COLLECT ENVCNTRL" card loads the entire complement of MAC environment programs identified in Fig. 2. A. 5. Programs not required in a Flight test, such as inactive vehicle programs, or unused OPTICSB2 or RADAR, may be bypassed in the MAC Program Collection process by the inclusion of a "NOPROG" card for each program to be bypassed. The elimination of unnecessary environment programs through the use of "NOPROG" cards results in a considerable reduction in collection time and an increase in simulation efficiency.

The flight vehicle environment programs which are collected for a particular Flight test run are written in such a form that they are generally applicable to any Block II mission. In order to simulate a specific mission, each vehicle program (SATURN, CSMBLK2, LEM, LVCBLK2) that is or will be active, must read a file of mission specific data. In addition, the UNIVERSE program must read an ephemeris file containing celestial parameters keyed to the year in which the simulated flight is performed. This mission specific data is available to the environment programs in the form of data files which must be explicitly loaded by means of "LOAD FILE" cards included in the simulation input deck. The available files are listed by mission in Section 2. D. 2.

A description of the " * SIMULATE AGC" card as well as the Program Source card and the AGC Instruction Simulator Special Request cards may be found in Sections 3. B and 3. C, while the format of input cards to all environment programs is included in Section 3. H.

The particular set of environment input data cards which must be included in a Flight Test input deck depends on the requirements of the run to be performed. In general, all Flight Test run decks must contain input data cards for the following environment programs:

1. INPUTPRG
2. UNIVERSE
3. IMUBLK2
4. Vehicle program which is active at the beginning of the run (PRELUNCH, SATURN , CSMBLK2, LEM, or LVCBLK2)
5. All vehicle programs that will be active in later phases of the simulation run (SATURN, CSMBLK2, LEM, or LVCBLK2)
6. All other programs that may be called during any phase of the simulation run (ASTRNAUT, TARGET, LMP1, OPTICSB2, RADAR, or DISCONT)

The format of the above data cards is presented in Section 3. H, and up to date example decks demonstrating the structure of Flight Test input decks for various applications are maintained on the Digital Simulation Group Bulletin Board in the Assembly Control Room.

In a Flight test simulation, the sequence of operations performed to set up, initialize, and run the simulation include the following:

1. IBM 360 Job Step Control Cards are processed.
2. MAC environment programs and mission specific data files are loaded from disc, and the main program (ENVCNTRL) is called.
3. The AGC Instruction Simulator is loaded and called by ENVCNTRL.
4. The AGC program is loaded from disc.
5. AGC special requests are processed.
6. The Environment is called for an initialization pass. All environment input cards and Astronaut input cards are read and all active environment programs are initialized. (See Section 2. D. 4).
7. The instruction by instruction AGC simulation is initiated, and interacts with the simulated vehicle hardware and flight environment.
8. AGC variables are printed, dumped, and traced, and MAC variables are printed, dumped and summarized, under user control.
9. The run is terminated by AGC special request, abort, or exit condition (including MAC abort).

3. G Structure of Input Deck for Environment Only Runs

The Environment only mode of operation of the Simulator is a diagnostic mode useful primarily to members of the Digital Simulation Group. In this mode, the environment portion of the Simulator operates open loop, without using the AGC Instruction Simulator or the Communicator, as shown in Fig. 2. A. 2. All of the MAC environment programs shown in Fig. 2. A. 5 may be used in an environment only simulation under control of the main MAC program ENVCTRL. AGC output channel control commands, such as CDU or GYRO pulse trains, engine throttle or trim commands, radar read requests etc. may be presented to the Environment via input cards, with one AGC command per card. See Section 3. G. 20 for the format of these input cards. The input commands may be considered to be open loop discontinuities fed to the environment since the sequence of input commands is not affected by any feedback from the environment.

The Environment Only mode is used principally to study the dynamical reaction of the environment to specific AGC command discontinuities.

An Environment only run deck is shown in Fig. 3. G. 1 and consists of the following categories of input cards:

1. IBM-360 Job Step Control Cards.
2. " * COLLECT ENVCTRL" Card.
3. "NOPROG" cards.
4. "LOAD FILE" cards.
5. " * GO" card.
6. MAC input cards for all active environment programs.
7. Terminator card, (/ *)

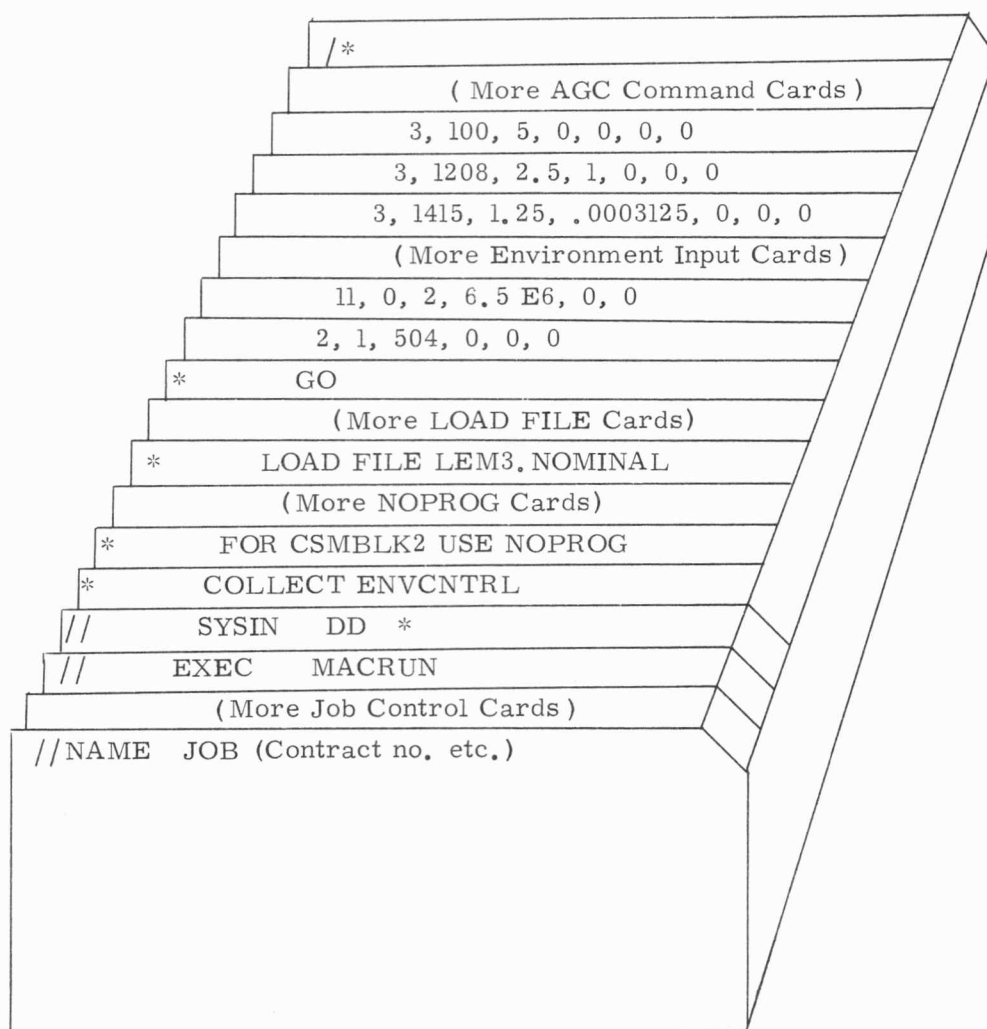


Figure 3.G.1 Structure of Environment Only Input Deck

Environment only runs are normal MAC runs, and run under the MAC execute procedure:

```
// EXEC  MACRUN
```

This card is part of the IBM-360 Job Step Control Card Deck. The " * COLLECT ENVCNTRL" card loads the entire complement of MAC environment programs identified in Fig. 2. A. 5. Those programs which are not required for a specific environment only run should be bypassed in the MAC collection process by the inclusion of a "NOPROG" card for each program to be bypassed.

The "LOAD FILE" and environment input data cards are identical to the corresponding cards in a Flight test deck. Available data files are listed by mission in Section 2. D. 2 and input card formats are included in Section 3. H. The use of a "GO" card, rather than a "SIMULATE AGC" card reflects the fact that Environment Only runs are MAC runs, not AGC simulation runs.

Since the AGC Instruction Simulator is not present in Environment Only runs, no AGC program is collected and no AGC special requests are included in an Environment Only input deck.

3. H Environment Input Cards

3. H. 1 Ground Rules for Input Data Cards

The format and description of all environment input cards is documented in this section. Environment input cards are used to define the initial state of the simulated vehicle, to enter Astronaut inputs, or to enter user selected input options. The number and type of environment input cards in a simulation deck depends on the requirements of the simulation to be performed, as follows:

1. Bench Test runs - no environment input cards are required.
2. Astronaut only runs - one or more input cards to INPUTPRG as well as Astronaut input cards are required - see Fig. 3. E. 1 for structure of input deck.
3. Flight test runs - input cards must be provided for INPUTPRG, UNIVERSE, IMUBLK2, and for all active vehicle programs (PRELUNCH, SATURN, CSMBLK2, LEM, or LVCBLK2). Active vehicle programs are defined as any vehicle program that will be utilized at any time throughout the run. In addition, ASTRNAUT, TARGET, LMP1, OPTICSB2, RADAR, or DISCONT cards may be required if any of these programs are called at any time during the run. See Fig. 3. F. 1 for a description of the structure of a representative Flight test input deck.
4. Environment only runs - same environment input cards as Flight test runs.

There are three general methods of specifying the value of variables used in environment programs; some of the parameters are compiled into a program, others are read from a mission specific data file (see Section 2. D. 4), and a third set of parameters is initialized from input data cards. The value of parameters compiled into environment programs may be changed by recompiling the program, or by using the MAC PATCH facility soon to be available for Rollback runs. Parameters whose values are derived from mission specific data files may be changed by regenerating the data file. Finally, the set of environment variables that are initialized from input data cards may be modified by repunching the input cards. The variables included in this third set and their format on input data cards are described below.

The inclusion of data cards for a particular environment program in a simulation input deck will result in the initialization of that program. It is not important that the input cards for the various routines appear in any particular order. The cards for each routine must be correctly arranged but it does not matter whether they appear before or after the cards for another routine. For instance, the first SATURN input card must be in front of the second SATURN input card, but SATURN Cards may be either before or after the LEM cards. Astronaut input cards may appear anywhere within the environment input card portion of a simulation deck, but there may be only one group of such cards.

3. H. 2 Format of Environment Data Cards

Environment data cards may be punched in either the standard MAC format (Column 1 containing an F, and Columns 9-80 containing 6 data fields of 12 characters each), or in the comma - delimited format (described in the Users Guide to MAC - 360, Section 6.2.2). The use of comma - delimited input cards allows AGC programmers to punch data in free form format, with up to 17 digits per data word. For compatibility with the standard MAC format, it is assumed that comma - delimited cards contain the same six fields of data that the standard MAC format input data cards contain. Comma - delimited format contains neither F, R, A, or * in Column 1, nor / * in Columns 1 and 2. Six data words, in free form format, are punched in Columns 9-80.

In the following description of data card formats for individual environment programs, the data card fields are presented in free form format, although it is understood that the cards may also be punched in standard MAC format. Astronaut cards are punched in a special free form alphabetic format, the structure of which is documented in Section 3. G. 15.

3. H. 3 Summary of Input Card Requirements

The Block 2 environment routines are listed below, together with the number of input cards each requires and the identification code associated with each program. This ID code must appear in the first data field of each environment input card to identify the recipient of the data.

<u>Environment Program</u>	<u>Input Cards Required</u>	<u>ID Code</u>
ENVCNTRL	NONE	ID = 3
INPUTPRG	1 or more cards	ID = 2
UNIVERSE	3 input cards	ID = 11
PRELUNCH	2 input cards	ID = 208
SATURN	3 input cards	ID = 260
CSMBLK2	3 or more cards	ID = 290
LEM	5 or more cards	ID = 280
LVCBLK2	2 input cards	ID = 293
IMUBLK2	3 or more cards	ID = 270
OPTICSB2	1 optional card	ID = 213
RADAR	4 input cards	ID = 13
TARGET	2 input cards	ID = 14
ASTRNAUT	any number of cards	Special format
LMP1	any number of cards	ID = 10
DISCONT	any number of cards	ID = 4
SIMSETUP	4 or more cards	-

See Section 2.D for a discussion of the function of each of these environment programs, and a summary of the dynamical models incorporated in each routine. A description of the input card data fields for each environment program follows.

3.G.4

INPUT CARDS FOR INPUTPRG

MAINTAINED BY PIETER MIMNO

INPUTPRG REQUIRES ONE TO THREE INPUT CARDS WITH THE FOLLOWING FORMAT:

	FIELD 1	FIELD 2	FIELD 3	FIELD 4	FIELD 5	FIELD 6
CARD 1:	ID	PRNTCTRL	FLIGHTNO	UNUSED	LATT	LONGT
CARD 2: (OPTIONAL)	ID	TIMETEST	ENVSUMDT	TTPRINT	TINHIB1	TINHIB2
CARD 3: (OPTIONAL)	ID	SIMFLITE				

MEANING OF SYMBOLS

ID	IDENTIFICATION NUMBER OF INPUTPRG. MUST HAVE THE VALUE "2".
PRNTCTRL	PRINT SUPPRESSION CONTROL. 0 = INHIBIT PRINTING FOR ALL ENVIRONMENT ROUTINES 1 = ENABLE PRINTING FOR ALL ENVIRONMENT ROUTINES (PRINTING FROM EACH ROUTINE DEPENDS ON THE SETTING OF ITS INDIVIDUAL PRINT CONTROL)
FLIGHTNO	FLIGHT NUMBER LEGAL VALUES = 205, 206, 207, 208, OR 504
UNUSED	THIS FIELD WAS USED IN THE PAST TO SELECT THE ASTRONAUT ONLY RUN OPTION. NOW HOWEVER, INPUTPRG ACTIVATES THE ASTRONAUT ONLY RUN OPTION IF IT DETECTS ASTRONAUT CARDS IN THE ENVIRONMENT INPUT DECK, BUT NO VEHICLE INPUT CARDS.
LATT	TARGET LATITUDE FOR SPLASHDOWN. USED IN THE COMPUTATION OF RANGE TO GO AND CROSS RANGE IN THE ENVIRONMENT SUMMARY PRINT (ENVSUM)
LONGT	TARGET LONGITUDE FOR SPLASHDOWN. LATT AND LONGT MAY BE SET TO ZERO FOR NON REENTRY RUNS.

TIMETEST	0	NORMAL SETTING
	NZ	TRIGGERS DIAGNOSTIC LOGIC WHICH READS THE REAL TIME CLOCK FREQUENTLY AND COMPILES STATISTICS ON THE AMOUNT OF REAL TIME USED BY ALL MAJOR COMPONENTS OF THE SIMULATOR. THE MAGNITUDE OF TIMETEST REPRESENTS AN INCREMENT OF TIME OVER WHICH RAW TIMING DATA ARE SMOOTHED TO GENERATE ONE DATA POINT OF TIMING INFORMATION.
ENVSUMDT	0	NORMAL SETTING.
	NZ	SETS A PARAMETER WHICH IS USED IN CONJUNCTION WITH AN ENVSUM 10 SPECIAL REQUEST. IF AN ENVSUM 10 IS ATTACHED TO AN AGC INSTRUCTION, THE EXECUTION OF THAT INSTRUCTION CAUSES AN INTERNALLY GENERATED ENVIRONMENT SUMMARY PRINT EVERY 4 SECONDS IF ENVSUMDT = 0, AND EVERY ENVSUMDT SECONDS IF ENVSUMDT IS NON ZERO.
TTPRINT	0	NORMAL SETTING
	NZ	URNS ON PRINT OF SETTING OF REAL TIME CLOCK DURING TIMING RUNS. (TIMETEST NZ)
TINHIB1	NOT USED AT PRESENT. BUT IT MUST CONTAIN A REAL NUMBER, USUALLY 0. IT IS READ AND PRINTED BUT OTHERWISE IGNORED.	
TINHIB2	SAME AS TINHIB1.	
SIMFLITE	0	NORMAL SETTING
	NZ	MODIFIES ENVIRONMENT TO RUN SIMFLIGHT TESTS ON THE LAUNCH PAD.

NOTE: FOR MOST RUNS, THE CONTENTS OF CARDS 2 AND 3 ARE ALL ZERO, AND THESE CARDS NEED NOT BE INCLUDED IN THE INPUT DECK.

3.G.5

INPUT CARDS FOR UNIVERSE

MAINTAINED BY MELVIN KALISH

UNIVERSE REQUIRES 3 INPUT CARDS WITH THE FOLLOWING FORMAT:

	FIELD 1	FIELD 2	FIELD 3	FIELD 4	FIELD 5	FIELD 6
	-----	-----	-----	-----	-----	-----
CARD 1:	ID	EPOCH	ORIGIN	R(X)	R(Y)	R(Z)
CARD 2:	ID	TZERO	OUTCTRL	V(X)	V(Y)	V(Z)
CARD 3:	ID	CONFIG	PERT	UNUSED	UNUSED	UNUSED

THE SYMBOLS APPEARING ON THE INPUT CARDS ARE DEFINED BELOW AND ARE IDENTIFIED BOTH BY THE NAME WHICH APPEARS IN THE HEADINGS ABOVE AND BY CARD AND WORD NUMBER. FOR EXAMPLE, (1,2) MEANS CARD 1, FIELD 2.

ID (N,1) N = 1 TO 3. IDENTIFICATION NUMBER OF UNIVERSE. MUST CONTAIN THE VALUE "11" .

EPOCH (1,2) TIME, MEASURED IN EPHEMERIS DAYS, FROM JULY 1.0 (MIDNIGHT AT GREENWICH JUST PRECEDING JULY 1) TO THE POINT AT WHICH IT IS DESIRED TO HAVE T (ARGUMENT USED IN CALLS TO UNIVERSE) EQUAL TO ZERO.

ORIGIN (1,3) ORIGIN OF THE COORDINATE SYSTEM IN WHICH INITIAL POSITION AND VELOCITY ARE GIVEN. 1 = EARTH, 2 = MOON. (SEE CONFIG.)

R(X) (1,4) INITIAL POSITION VECTOR, IN METERS (I. E., THE POSITION AT T = TZERO). (SEE CONFIG.)

R(Y) (1,5)

R(Z) (1,6)

TZERO (2,2) INITIAL VALUE OF TIME, MEASURED IN SECONDS FROM EPOCH.

OUTCTRL (2,3) THIS VARIABLE, WHICH MAY BE EFFECTIVELY SET TO ZERO BY THE ENVCONTROL PROGRAM, CONTROLS THE TYPE OF INFORMATION PRINTED BY UNIVERSE. 0 TURNS OFF VIRTUALLY ALL PRINTING. 1 CAUSES PRINTING OF OCCASIONAL COMMENTS WHEN EVENTS OCCUR, PLUS TIME, POSITION, AND VELOCITY EVERY MODE 5 CALL. 2 CAUSES ADDITIONAL PRINTING OF TIME AND MODE AT EVERY UNIVERSE CALL.

V(X) (2,4) INITIAL VELOCITY VECTOR, IN METERS / SEC . (I. E., THE VELOCITY AT T = TZERO). (SEE CONFIG)

V(Y) (2,5)

V(Z) (2,6)

CONFIG (3,2) THIS WORD CONTROLS THE INITIAL INTERNAL CONFIGURATION OF UNIVERSE. 2 INDICATES THAT TRAJECTORY MAINTENANCE IS NOT TO BE UNIVERSE'S RESPONSIBILITY (USED WHEN STARTING A SIMULATION RUN IN PRELAUNCH OR BOOST MODES). 8 INDICATES THAT THE RUN IS STARTING IN ENTRY MODE. 0 IS FOR ALL OTHER CASES. IF CONFIG = 2, THE DATA IN ORIGIN, R(X), R(Y), R(Z), V(X), V(Y), AND V(Z) IS IGNORED. VALID DATA (LIKE ALL ZEROS) MUST BE SUPPLIED ANYWAY.

PERT (3,3) IF PERT = 0, SPHERICAL GRAVITATIONAL FIELD MODELS WILL BE USED IN MODE 5 TRAJECTORY-UPDATE CALLS. OTHERWISE, PERTURBATIONS WILL BE INCLUDED.

(3,4) - (3,6) ARE NOT USED.

3.G.6

INPUT CARDS FOR PRELUNCH

MAINTAINED BY KEITH GLICK

PRELUNCH REQUIRES TWO INPUT CARDS WITH THE FOLLOWING FORMAT:

	FIELD 1	FIELD 2	FIELD 3	FIELD 4	FIELD 5	FIELD 6
	-----	-----	-----	-----	-----	-----
CARD 1:	ID	LAT	LONG	ALT	TGUIDREL	TLIFTOFF
CARD 2:	ID	WEST TIP	SOUTH TIP	AZIMUTH	UNUSED	SWAY

MEANING OF SYMBOLS

ID	IDENTIFICATION NUMBER OF PRELUNCH. MUST HAVE THE VALUE "208".
LAT	GEODETTIC OR SELENOGRAPHIC LATITUDE OF THE LAUNCH SITE, IN DEGREES.
LONG	GEOGRAPHIC OR SELENOGRAPHIC EAST LONGITUDE OF THE LAUNCH SITE, IN DEGREES.
ALT	HEIGHT IN METERS OF THE NAV-BASE ABOVE THE FISCHER REFERENCE ELLIPSOID, OR LUNAR REFERENCE SPHERE OF RADIUS 1735.4 KM. THE MEAN LUNAR SURFACE HAS AN ALTITUDE OF 2690 METERS.
TGUIDREL	TIME OF GUIDANCE REFERENCE RELEASE, IN SECONDS FROM EPOCH.
TLIFTOFF	TIME OF LIFTOFF, IN SECONDS FROM EPOCH. DURING LUNAR PRELAUNCH SIMULATIONS, THIS TIME DOES NOT DETERMINE THE TIME OF LIFTOFF.
WEST TIP	AVERAGE WESTERLY TIP OF THE VEHICLE, IN DEGREES.
SOUTH TIP	AVERAGE SOUTHERLY TIP OF THE VEHICLE, IN DEGREES.
AZIMUTH	AZIMUTH OF THE VEHICLE'S Z AXIS, COUNTER-CLOCKWISE FROM EAST, IN DEGREES.
UNUSED	THIS FIELD IS NO LONGER USED. BUT IT MUST CONTAIN A REAL NUMBER, NORMALLY 0.
SWAY	ROOT-MEAN-SQUARE DISPLACEMENT OF THE NAV-BASE DUE TO SWAYING, IN METERS. THIS APPLIES TO EARTH PRELAUNCH SIMULATIONS ONLY.

3.G.7

INPUT CARDS FOR SATURN

MAINTAINED BY LANCE DRANE

SATURN REQUIRES THREE DATA CARDS WITH THE FOLLOWING FORMAT:

	FIELD 1 -----	FIELD 2 -----	FIELD 3 -----	FIELD 4 -----	FIELD 5 -----	FIELD 6 -----
CARD 1:	ID	TFILE1	TFILE2	PRTSTART	ATTPRINT	TFROMLO
CARD 2:	ID	THRMODEL	CSTMODEL	WB(X)	WB(Y)	WB(Z)
CARD 3:	ID	UNUSED	UNUSED	CHI PITCH	CHI YAW	CHI ROLL

MEANING OF SYMBOLS

ID SATURN SURROUTINE IDENTIFICATION NUMBER. ALL THREE MUST BE 260.

TFILE1 START AND STOP TIMES (IN SECONDS) FOR WRITING PLOTTING DATA INTO THE DATA FILE.
TFILE2 NORMALLY BOTH 0.

PRTSTART TIME (IN SECONDS) AT WHICH TO BEGIN SATURN (AND ATTITUDE) PRINTING.
NORMALLY 0.

ATTPRINT A NON-ZERO INTEGER VALUE WILL DECREASE PRINTING BY THE ATTITUDE PROGRAM. ATTPRINT IS USED AS A COUNTER TO DETERMINE WHETHER OR NOT ATTITUDE SHOULD PRINT EACH PASS. NORMALLY 1.

TFROMLO START TIME (IN SECONDS FROM LIFTOFF) FOR BEGINNING A SATURN SIMULATION IN MIDFLIGHT.
WARNING: SATURN MAY NOT BE INITIALIZED DURING ITERATIVE STEERING PERIODS. NORMALLY 0.

THRMODEL THRUST MODEL DETERMINES THE PRECISION (AND CONVERSELY THE RUNNING SPEED) OF THE SIMULATION WHILE THE SATURN IS THRUSTING.
0 = SIMULATE IN THE FLIGHT CONTROL COMPUTER, THE ANALOG SIGNAL SHAPING FILTERS IN THE ERROR CHANNELS AND RATE GYRO FEEDBACK CHANNELS.
SIMULATE THE TRANSFER FUNCTION OF BETA COMMAND TO ENGINE GIMBAL ANGLE. (TO BE USED WHEN APOLLO IS GUIDING THE SIVB).
1 = IGNORE THE ABOVE. SPEEDS SIMULATION BY A FACTOR OF TWENTY.

CSTMODEL COAST MODEL DETERMINES THE PRECISION (AND CONVERSELY THE RUNNING SPEED) OF THE SIMULATION WHILE THE SIVB IS COASTING.
 0 = FLY 'BELLYDOWN' IN COAST PERIODS. BELLYDOWN ADJUSTS SATURN'S ATTITUDE SO THAT:
 ROLL AXIS (X) FORMS THE TRIAD. APPROXIMATELY IN DIRECTION OF VELOCITY VECTOR
 PITCH AXIS (Y) IS PERPENDICULAR TO THE RV PLANE ($V \times R$).
 YAW AXIS (Z) IS DOWN THE R VECTOR.

THE FOLLOWING MODELS SPEED THE SIMULATION CONSIDERABLY BY SKIPPING ALL DYNAMIC EQUATIONS.
 1 = POINT MASS SIMULATION DURING PARKING ORBIT COAST. ATTITUDE HELD INERTIALLY CONSTANT.
 2 = POINT MASS SIMULATION DURING TRANS-LUNAR COAST. ATTITUDE HELD INERTIALLY CONSTANT.
 3 = BOTH 1 AND 2 ABOVE.
 4 = HOLDS INERTIALLY CONSTANT WB DURING ENTIRE RUN.
 THIS MODEL CAN BE USED TO SIMULATE UNDAMPED TUMBLING.

WB THE INITIAL ANGULAR RATE OF ROTATION IN INERTIAL SPACE OF THE SATURN BOOSTER MEASURED IN THE SATURN BOOST (B) COORDINATE SYSTEM IN DEGREES PER SECOND. IGNORED IF SATURN FOLLOWS PRELAUNCH (WHEN WV IS ROLLED 180 DEGREES AND USED) AND IGNORED IN RUNS FROM LIFTOFF (WHEN EARTHRAE IS USED).

CHI PITCH EULER ANGLES DESCRIBING TRANSFORMATION FROM SATURN CALCULATION SYSTEM (C) TO SATURN VEHICLE SYSTEM (B). TO START SIMULATION AFTER LIFTOFF, THEY MUST BE READ FROM THE NOMINAL TRAJECTORY RUN BY THE DIGITAL SIMULATION GROUP ' ROOM 235.
 CHI YAW
 CHI ROLL

UNUSED WILL BE READ, PRINTED, AND THEN IGNORED. MUST BE A REAL NUMBER HOWEVER. NORMALLY 0.

MAINTAINED BY ROBERT WADSWORTH AND SID DAVID

CSMBLK2 REQUIRES THREE INPUT CARDS WITH THE FOLLOWING FORMAT:

	FIELD 1	FIELD 2	FIELD 3	FIELD 4	FIELD 5	FIELD 6
CARD 1:	ID	PRNTCTRL	DTPRINT	CONFIG	SBFLAG	DEBUG
CARD 2:	ID	PLOTCODE	TFILE	YPFLAG	PITCH	YAW
CARD 3:	ID	ROLL	WVX	WVY	WVZ	FUELLOAD OR MASS

THE SYMBOLS APPEARING ON THE INPUT CARDS ARE DEFINED BELOW AND ARE IDENTIFIED BOTH BY THE NAME WHICH APPEARS IN THE HEADINGS ABOVE, AND BY THE CARD AND FIELD NUMBER. FOR EXAMPLE, (1,2) MEANS CARD 1, FIELD 2. IF THE SIMULATION IS STARTED IN PRELUNCH OR SATURN, THE DATA FOLLOWING YPFLAG ARE IGNORED AND CSMBLK2 INITIALIZES FROM THE CALLFILE AND THE CSM MISSION FILE.

ID (N,1) N = 1 TO 3. IDENTIFICATION NUMBER OF CSMBLK2. MUST CONTAIN THE VALUE "290" .

PRNTCTRL (1,2) THIS VARIABLE, WHICH MAY BE EFFECTIVELY SET TO ZERO BY THE ENVCONTROLBLK2 PROGRAM, CONTROLS THE TYPE OF INFORMATION PRINTED BY CSMBLK2 AS FOLLOWS:

- 0 VIRTUALLY ALL PRINTING IS INHIBITED.
- 1 PRINTS TIME, TIME STEP, VEHICLE MASS, AND VARIABLES DESCRIBING VEHICLE MOTION.
- 2 ADDS PRINTING OF ENGINE GIMBAL ANGLES AND OTHER VARIABLES OF THE ACTUATOR LOOP, PROVIDED TVC ENABLE AND OPTICS ERR CTR ENABLE ARE BOTH ON.
- 4 PRINTS MASS, CG, INERTIAS, AND SLOSH FREQ, MASS, SPRING CONST AND MOMENT ARM VECTOR (IF SIMULATED) AT EACH RECOMPUTATION. DATA CALLED BY 2 ARE ALSO PRINTED.
- 5 DATA CALLED BY 2 ARE PRINTED EACH TIME AN RCS JET GOES ON OR OFF.

DTPRINT (1,3) PRINT PERIOD CONTROL FOR *STOP-TIME* PRINTING. PRINTING IS INHIBITED UNTIL THE NEXT MULTIPLE OF DTPRINT. A VALUE OF ZERO PERMITS PRINTING EVERY DT. UNITS ARE SECONDS.

CONFIG (1,4) THIS VARIABLE DEFINES THE VEHICLE CONFIGURATION TO BE SIMULATED AS FOLLOWS:

- 1 COMMAND SERVICE MODULE
- 2 COMMAND SERVICE MODULE WITH FULL DOCKED LEM.

SBFLAG (1,5) THIS VARIABLE CONTROLS THE SIMULATION OF SLOSH AND BENDING AS FOLLOWS:

- 0 SLOSH AND BENDING NOT SIMULATED
- 1 SLOSH IS SIMULATED BUT NOT BENDING
- 2 SLOSH AND BENDING ARE BOTH SIMULATED

DEBUG (1,6) THIS VARIABLE PERMITS THE USER TO CHANGE SOME OF THE VEHICLE PARAMETERS, ELIMINATING CROSS COUPLING AMONG VEHICLE MOTIONS AS FOLLOWS:

- 0 NO CHANGE
- 1 ENGINE GIMBAL OFFSET IN PITCH AND YAW, AND Y AND Z COMPONENTS OF CG ARE ZEROED
- 2 PRODUCTS OF INERTIA ARE SET TO ZERO, ALSO 1
- 3 Y AND Z COMPONENTS OF SLOSH MOMENT ARM VECTOR ARE SET TO ZERO, ALSO 1 AND 2
- 5 ENGINE GIMBAL ANGLES ARE SET TO THE COMMANDED VALUES AFTER EACH TIME STEP, ALSO 1
- 6 SAME AS 2 AND 5
- 7 SAME AS 3 AND 5

PLOTCODE (2,2) THIS VARIABLE CONTROLS THE TYPE OF INFORMATION FILED BY CSMBLK2 FOR SUBSEQUENT PLOTTING. IF THE INTEGER PART OF PLOTCODE IS NON-ZERO, TIME AND SEVERAL OTHER SIMULATION VARIABLES ARE STORED IN THE DATAFILE AT INTERVALS APPROXIMATELY EQUAL TO THE DECIMAL PART OF PLOTCODE. FILING AFTER EACH INTERNAL TIME STEP WILL RESULT IF THE DECIMAL PART IS ZERO. THE INITIAL DATAFILE LOCATION IS 100000. THE PERIOD OF THE FILE (NUMBER OF SCALAR VARIABLES FILED AFTER EACH TIME STEP) IS STORED AT LOCATION 99999.

THE LISTED VARIABLES ARE STORED IN THE DATA FILE IF THE INTEGER PART OF PLOTCODE ASSUMES THE FOLLOWING VALUES:

PLOTCODE	VARIABLES	DESCRIPTION
0	NO DATA ARE FILED	
1	JPYC,JPZC,JPY,JPZ,A,FINT I	SPS NOZZLE COMMAND,POSITION,CONTACT ACCELERATION, THE INTEGRAL OF A FROM TSTART TO PRESENT TIME I
2	FINT,W,A,DW/DT R I R	FINT,ANGULAR VELOCITY,CONTACT ACCELERATION,ANGULAR ACCELERATION
3	FINT,W,BY,BZ,PR,YR R	FINT,ANGULAR VELOCITY,LINEAR AND ANGULAR BENDING AT THE NAV BASE
4	JRY,JRZ,JAY,JAZ,JTY,JTZ,JCY,JCZ	SPS NOZZLE RATE,ACCELERATION,CLUTCH TORQUE, CLUTCH TORQUE COMMAND
5	QOY,QOZ,QFY,QFZ,W,A,MASS R I	Y AND Z SLOSH MOTION IN FUEL AND OXIDIZER TANKS, ANGULAR VELOCITY,CONTACT ACCEL,VEHICLE MASS
6	JPYC,JPZC,JPY,JPZ,A,FINT, QOY, QOZ, QFY, QFZ I	SPS NOZZLECOMMAND,POSITION,CONTACT ACCELERATION, THE INTEGRAL OF A FROM TSTART TO PRESENT TIME, I Y AND Z SLOSH MOTION IN FUEL AND OXIDIZER TANKS.
7	JPYC,JPZC,JPY,JPZ,A,FINT, QOY,QOZ,QFY,QFZ,RY,RZ, PR,YR I	SPS NOZZLE COMMAND,POSITION,ACCELERATION, THE INTEGRAL OF A FROM TSTART TO PRESENT TIME, I Y AND Z SLOSH MOTION IN FUEL AND OXIDIZER TANKS, LINEAR AND ANGULAR BENDING AT THE NAV BASE

TFILE	(2,3)	THIS VARIABLE DEFINES THE TIME AT WHICH CSMBLK2 BEGINS TO FILE SIMULATION DATA
YPFLAG	(2,4)	THIS VARIABLE, IF NON-ZERO, IS USED AS A FLAG TO PREVENT PITCH PLANE VEHICLE MOTION. PITCH ACTUATOR DYNAMICS ARE NOT SIMULATED AND INERTIA MATRIX ELEMENTS OTHER THAN IZZ ARE IGNORED.
PITCH	(2,5)	SUCCESSIVE EULER ROTATIONS, IN DEGREES, FOR INITIAL VEHICLE AXIS ORIENTATION WITH RESPECT
YAW	(2,6)	TO THE PREFERRED COORDINATE SYSTEM (Z = UNIT(-R), Y = UNIT(V*R), X = Y*Z)
ROLL	(3,2)	
WVX	(3,3)	INITIAL ANGULAR RATE, IN DEG/SEC, ABOUT X-VEHICLE AXIS.
WVY	(3,4)	INITIAL ANGULAR RATE, IN DEG/SEC, ABOUT Y-VEHICLE AXIS.
WVZ	(3,5)	INITIAL ANGULAR RATE, IN DEG/SEC, ABOUT Z-VEHICLE AXIS.
MASS	(3,6)	INITIAL VEHICLE MASS, IN POUNDS
OR		
FUELLOAD	(3,6)	INITIAL SPS PROPELLANT LOADING EXPRESSED AS A FRACTION OF MAXIMUM LOADING (ZERO THRU ONE)

ADDITIONAL CARDS MAY BE INCLUDED IN THE DECK IF THE USER WISHES TO INTRODUCE CERTAIN MODIFICATIONS TO THE NOMINAL VEHICLE. EACH CARD MUST CONTAIN THE ID NUMBER (290) IN ITS FIRST WORD AND A PERTURBATION CODE (PERTCODE) IN ITS SECOND WORD. A MAXIMUM OF 16 ADDITIONAL CARDS CAN BE USED. THE INTERPRETATION OF THE DATA IN WORDS 3 - 6 DEPENDS UPON THE VALUE OF WORD 2 (PERTCODE) AS FOLLOWS:

WORD 2 (PERTCODE)	WORD 3	WORD 4	WORD 5	WORD 6	COMMENTS
1	IXX	IYY	IZZ	0.0	DIAGONAL INERTIA MATRIX ELEMENTS (SLUG FT SQ.)
2	CGX	CGY	CGZ	0.0	VEHICLE CENTER OF GRAVITY VECTOR (IN. APOLLO)
3	MISTY	MISTZ	0.0	0.0	THRUST MISALIGNMENT TORQUE - PITCH AND YAW (FT. LBS.)
4	EPY	EPZ	0.0	0.0	THRUST ANGULAR MISALIGNMENT - PITCH AND YAW (DEGREES)
5	QOY	QOZ	QFY	QFZ	SLOSH AMPLITUDE AT SPS IGNITION - OXIDIZER TANK Y AND Z COMPONENTS AND FUEL TANK Y AND Z COMPONENTS (IN.)
6	QY1	QY2	QY3	0.0	INITIAL BENDING AMPLITUDE IN YAW PLANE - FIRST 3 MODES (IN.)
7	QZ1	QZ2	QZ3	0.0	INITIAL BENDING AMPLITUDE IN PITCH PLANE - FIRST 3 MODES (IN.)
8	RCSJETNO	TIME	0.0	0.0	FAILURE (ZERO THRUST) OF SPECIFIED SM RCS JET WILL OCCUR AT THE SPECIFIED TIME (TIME IN SECONDS)
9	JPYTRIM	TIME	0.0	0.0	PITCH NOZZLE TRIM COMMAND WILL BE CHANGED TO THE SPECIFIED VALUE (DEGREES) AT THE SPECIFIED TIME (SEC.)
10	JPZTRIM	TIME	0.0	0.0	YAW NOZZLE TRIM COMMAND WILL BE CHANGED TO THE SPECIFIED VALUE (DEGREES) AT THE SPECIFIED TIME (SEC.)
11	FRACTTHR	TIME	0.0	0.0	SPS THRUST WILL BE CHANGED TO THE SPECIFIED FRACTION OF THE NOMINAL VALUE AT THE SPECIFIED TIME.

MAINTAINED BY KEITH GLICK AND HERBERT CHASAN

LEM REQUIRES FIVE INPUT CARDS WITH THE FOLLOWING FORMAT:

	FIELD 1	FIELD 2	FIELD 3	FIELD 4	FIELD 5	FIELD 6
	-----	-----	-----	-----	-----	-----
CARD 1:	ID	PRNTCTRL	DTPRINT	STAGE	THRUST	THBITS
CARD 2:	ID	PITCH	YAW	ROLL	WVX	WVY
CARD 3:	ID	WVZ	MODEL	DEBITS	AEBITS	RCSBITS1
CARD 4:	ID	RCSBITS2	RCSBITS3	RCSBITS4	JPY	JPZ
CARD 5:	ID	DFUELL	AFUELL	CFUELL	CGOFFSET	UNUSED

THE SYMBOLS APPEARING ON THE INPUT CARDS ARE DEFINED BELOW AND ARE IDENTIFIED BOTH BY THE NAME WHICH APPEARS IN THE HEADINGS ABOVE, AND BY CARD AND FIELD NUMBER. FOR EXAMPLE, (1,2) MEANS CARD 1, FIELD 2.

ID (N,1) N = 1 TO 3. IDENTIFICATION NUMBER OF LEM. MUST CONTAIN THE VALUE "280".

PRNTCTRL (1,2) THIS VARIABLE, WHICH MAY BE EFFECTIVELY SET TO ZERO BY THE ENVCONTROLBLK2 PROGRAM, CONTROLS THE TYPE OF INFORMATION PRINTED BY LEM.
 0: VIRTUALLY ALL PRINTING TURNED OFF
 1: REGULAR PRINT EVERY DTPRINT
 2: SAME AS 1 WITH ADDITIONAL PRINTOUT OF INDIVIDUAL FORCES AND MOMENTS.
 3: REGULAR PRINT EVERY REACTION TIME
 4: SAME AS 3 WITH ADDITIONAL PRINTOUT OF INDIVIDUAL FORCES AND MOMENTS.
 5: REGULAR PRINT EVERY DTPRINT AND AT REACTION TIMES
 6: SAME AS 5 WITH ADDITIONAL PRINTOUT OF INDIVIDUAL FORCES AND MOMENTS.
 7: SAME AS 5 WITH ADDITIONAL PRINTOUT OF ANGULAR ACCEL. ABOUT CG DUE TO ENGINE TORQUE.

DTPRINT (1,3) PRINT INTERVAL (SEC.) A VALUE OF ZERO GIVES PRINTING EVERY DT.

STAGE (1,4) THIS VARIABLE IS 0 TO START WITH THE ASCENT STAGE, 1 TO START WITH THE DESCENT STAGE AND 2 TO START WITH CSM ATTACHED.

THRUST (1,5) THIS VARIABLE IS A NON-DIMENSIONAL STATEMENT OF THE INITIAL THRUST DESIRED. USE 0 OR 1 TO START WITH THE ASCENT STAGE. USE 0 OR .1 TO 1. TO START WITH THE DESCENT STAGE.

THBITS (1,6) THIS IS A PACKED WORD. EACH DIGIT CONTAINS 0,1, OR 2 CORRESPONDING TO 0, PLUS OR MINUS 3 SIGMA ERRORS RESPECTIVELY. THE DIGITS ARE ORDERED FROM RIGHT TO LEFT, STARTING AT THE DECIMAL POINT. UNPACKED, THEY ARE:
 DIGIT 1: THRUST ERROR OF DESCENT ENGINE AT START OF MISSION DUTY CYCLE.
 DIGIT 2: THRUST ERROR OF DESCENT ENGINE 370 SEC. AFTER START OF MISSION DUTY CYCLE.
 DIGIT 3: THRUST ERROR OF ASCENT ENGINE AT START OF MISSION DUTY CYCLE.
 DIGIT 4: THRUST ERROR OF ASCENT ENGINE 460 SEC. AFTER START OF MISSION DUTY CYCLE.
 DIGIT 5-10: NOT USED.

PITCH (2,2) SUCCESSIVE EULER ROTATIONS, IN DEGREES, FROM PREFERRED OR LOCAL VERTICAL AXES TO VEHICLE
 YAW (2,3) AXES. THE LOCAL VERTICAL SYSTEM IS DEFINED BY:

ROLL (2,4)
$$\begin{matrix} & & - & - & - \\ & & I & I & I \\ \text{UNIT(X)} & = & \text{UNIT(V * R)} * \text{UNIT(-R)} \\ \text{LV} & & & & \end{matrix}$$

$$\begin{matrix} & & - & - & - \\ & & I & I & I \\ \text{UNIT(Y)} & = & \text{UNIT(V * R)} \\ \text{LV} & & & & \end{matrix}$$

$$\begin{matrix} & & - & - & - \\ & & I & I & I \\ \text{UNIT(Z)} & = & \text{UNIT(-R)} \\ \text{LV} & & & & \end{matrix}$$

WHERE R AND V ARE POSITION AND VELOCITY IN REFERENCE INERTIAL COORDINATES.

WVX (2,5) INITIAL ANGULAR RATE, IN DEG/SEC, ABOUT X-VEHICLE AXIS. ALSO CALLED P.
 WVY (2,6) INITIAL ANGULAR RATE, IN DEG/SEC, ABOUT Y-VEHICLE AXIS. ALSO CALLED Q.
 WVZ (3,2) INITIAL ANGULAR RATE, IN DEG/SEC, ABOUT Z-VEHICLE AXIS. ALSO CALLED R.

MODEL (3,3) PACKED WORD SPECIFYING LEM MODEL AND PROGRAM FUNCTIONS DESIRED. UNPACKED, THEY ARE:

DIGIT 1: 0 PLOT PRINTING SUPPRESSED.
 1 PRINT VARIABLES TO BE PLOTTED.

DIGIT 2: 0 NO PLOT.
 1 PLOT ROLL, PITCH, YAW, ANGULAR RATES AND ANGULAR ACCELERATION.
 2 PLOT SLOSH FORCE, ANGULAR RATES AND ANGULAR ACCELERATION.
 3 PLOT TRIM GIMBALS

DIGIT 3: 0 INERTIA MATRIX IS NOT DIAGONAL.
 1 INERTIA MATRIX IS DIAGONAL WITH EQUAL ELEMENTS, THE CG IS PUT ON THE X-AXIS, THE JET IMPINGEMENT AND STAGING FORCES ARE ALWAYS ZERO, AND THE INERTIA AND CG REMAIN UNCHANGED UNLESS STAGING OCCURS.
 2 INERTIA MATRIX IS SET TO THE INERTIA OF THE EMPTY ASCENT STAGE AND HELD FIXED. CG AND MASS ARE RESET.

DIGIT 4: 0 NO RCS JET FAILURES OCCUR.
 1 RCS JET FAILURES OCCUR AS SPECIFIED BY INPUT CARDS AFTER THE FIFTH INPUT CARD.

DIGIT 5: 0 NO SLOSH FORCES ARE INCLUDED.
 1 SLOSH FORCES ARE INCLUDED.

DIGIT 6: 0 ENGINE ARM OFF. NOTE ENGINE ARM MUST BE TURNED ON BEFORE EITHER ENGINE IS FIRED.
 1 ENGINE ARM ON.

DIGIT 7: 0 VEHICLE DOES NOT BEND.
 1 VEHICLE BENDS IF CSM IS ATTACHED.

THE DECIMAL POINT IS TO THE RIGHT OF THE SEVENTH DIGIT.

DEBITS (3,4) PACKED ERROR BITS FOR DESCENT ENGINE (SEE THRITS, ABOVE). UNPACKED, THEY ARE:
 DIGIT 1: X POSITION ERROR
 DIGIT 2: Y POSITION ERROR
 DIGIT 3: Z POSITION ERROR
 DIGIT 4: MISALIGNMENT OF THRUST AXIS WITH ENGINE CENTER LINE ABOUT Y-AXIS.
 DIGIT 5: MISALIGNMENT OF THRUST AXIS WITH ENGINE CENTER LINE ABOUT Z-AXIS.
 DIGIT 6: MISALIGNMENT OF ENGINE CENTER LINE WITH X-VEHICLE AXIS ABOUT Y-AXIS.
 DIGIT 7: MISALIGNMENT OF ENGINE CENTER LINE WITH X-VEHICLE AXIS ABOUT Z-AXIS.
 DIGIT 8: THRUST DEPENDENT MISALIGNMENT (DUE TO BENDING), ABOUT Y-AXIS.
 DIGIT 9: THRUST DEPENDENT MISALIGNMENT (DUE TO BENDING), ABOUT Z-AXIS.
 DIGIT 10: NOT USED.

AEBITS (3,5) PACKED ERROR BITS FOR ASCENT ENGINE (SEE THBITS, ABOVE). UNPACKED, THEY ARE:
 DIGIT 1: IF THIS BIT IS ZERO, SKIP DEBITS, AEBITS, AND RCSBITS (ALL ARE SET TO ZERO).
 DIGIT 2: X POSITION ERROR
 DIGIT 3: Y POSITION ERROR
 DIGIT 4: Z POSITION ERROR
 DIGIT 5: MISALIGNMENT ABOUT Y-AXIS
 DIGIT 6: MISALIGNMENT ABOUT Z-AXIS
 DIGIT 7-10: NOT USED

RCSBITS1 (3,6) PACKED ERROR BITS FOR RCS QUAD 1 (JETS 1-4). UNPACKED, THEY ARE:
 DIGIT 1: X POSITION ERROR
 DIGIT 2: Y POSITION ERROR
 DIGIT 3: Z POSITION ERROR
 DIGIT 4: MISALIGNMENT ABOUT X-AXIS
 DIGIT 5: MISALIGNMENT ABOUT Y-AXIS
 DIGIT 6: MISALIGNMENT ABOUT Z-AXIS
 DIGIT 7-10: NOT USED

RCSBITS2 (4,2) PACKED ERROR BITS FOR RCS QUAD 2 (JETS 5-8). SEE RCSBITS1 ABOVE FOR UNPACKING.

RCSBITS3 (4,3) PACKED ERROR BITS FOR RCS QUAD 3 (JETS 9-12). SEE RCSBITS1 ABOVE FOR UNPACKING.

RCSBITS4 (4,4) PACKED ERROR BITS FOR RCS QUAD 4 (JETS 13-16).

JPY (4,5) INITIAL ORIENTATION, IN DEGREES, OF TRIM GIMBAL ABOUT THE - Y VEHICLE AXIS.

JPZ (4,6) INITIAL ORIENTATION, IN DEGREES, OF TRIM GIMBAL ABOUT THE - Z VEHICLE AXIS.

DFUELL (5,2) INITIAL LOADING OF DESCENT TANKS. USE 1 FOR FULL TANKS AND 0 TO 1 FOR PARTIAL LOADING.

AFUELL (5,3) INITIAL LOADING OF ASCENT TANKS. USE 1 FOR FULL TANKS AND 0 TO 1 FOR PARTIAL LOADING.

CFUELL (5,4) INITIAL LOADING OF SM TANKS. USE 1 FOR FULL TANKS AND 0 TO 1 FOR PARTIAL LOADING.

CGOFFSET (5,5) THIS VARIABLE IS A FRACTION WHICH RANGES FROM 0 TO 1. IF IT IS NON ZERO, THE CENTER OF GRAVITY WILL BE OFFSET AT DPS IGNITION. THE CG WILL RETURN TO THE NOMINAL POSITION UNDER THE INFLUENCE OF THRUST ACCELERATION. THE VALUE ONE CAUSES THE WORST CASE OF OFFSET, THE VALUE ZERO CAUSES THIS EFFECT TO BE IGNORED.

5 DATA CARDS ARE REQUIRED BY THE LEM PROGRAM. THE FOLLOWING CARDS CAN BE ADDED ACCORDING TO THE NEEDS OF THE USER. X IS AN INTEGER FROM 6 UP TO AND INCLUDING 18.

JET FAILURES

TFAIL1 (X,2) TIME (SEC) OF THE FIRST DESIRED RCS JET FAILURE.

JET1 (X,3) NUMBER (1-16) OF JET WHICH FAILED AT TIME TFAIL1. THIS WORD IS POSITIVE IF JET FAILS ON AND NEGATIVE IF JET FAILS OFF. IF MORE THAN ONE JET FAILS AT ANY ONE TIME, AN INPUT CARD IS REQUIRED FOR EACH FAILURE.

TDETECT1 (X,4) TIME (SEC) AT WHICH APPROPRIATE JET PAIR IS TURNED OFF. TDETECT1 MUST BE GREATER THAN TFAIL1.

TFAILN2 (X,2) TIME (SEC) OF THE SECOND JET FAILURE.

JETN2 (X,3) NUMBER OF JET WHICH FAILED AT TFAILN2.

TDETECTN2 (X,4) TIME (SEC) APPROPRIATE JET PAIR IS TURNED OFF.

• • • •

• • • •

• • • •

• • • •

INITIAL SLOSH MASS DISPLACEMENT

Y COMP	(X,2)	Y COMPONENT (INCHES) OF INITIAL SLOSH MASS DISPLACEMENT AT APS OR DPS ENGINE IGNITION. THE CORRESPONDING TANK IS SPECIFIED BY A CODE IN (X,6). ONE INPUT CARD IS REQUIRED FOR EACH CODE USED.	CODE	DESCRIPTION
			130	ALL TANKS
			120	CSM OXIDIZER
			111	LEM DPS FUEL LOCATED ON THE + Y AXIS
			110	LEM DPS OXIDIZER LOCATED ON THE + Z AXIS
			100	LEM APS OXIDIZER

Z COMP	(X,3)	Z COMPONENT (INCHES) OF INITIAL SLOSH MASS DISPLACEMENT AT APS OR DPS ENGINE IGNITION. THE CORRESPONDING TANK IS SPECIFIED BY A CODE IN (X,6).	CODE	DESCRIPTION
			130	ALL TANKS
			120	CSM OXIDIZER
			111	LEM DPS FUEL LOCATED ON THE + Y AXIS
			110	LEM DPS OXIDIZER LOCATED ON THE + Z AXIS
			100	LEM APS OXIDIZER
Y COMP	(X,4)	Y COMPONENT (INCHES) OF INITIAL SLOSH MASS DISPLACEMENT AT APS OR DPS ENGINE IGNITION. THE CORRESPONDING TANK IS SPECIFIED BY A CODE IN (X,6).	CODE	DESCRIPTION
			120	CSM FUEL
			111	LEM DPS FUEL LOCATED ON THE - Y AXIS
			110	LEM DPS OXIDIZER LOCATED ON THE - Z AXIS
			100	LEM APS FUEL
Z COMP	(X,5)	Z COMPONENT (INCHES) OF INITIAL SLOSH MASS DISPLACEMENT AT APS OR DPS ENGINE IGNITION. THE CORRESPONDING TANK IS SPECIFIED BY A CODE IN (X,6).	CODE	DESCRIPTION
			120	CSM FUEL
			111	LEM DPS FUEL LOCATED ON THE - Y AXIS
			110	LEM DPS OXIDIZER LOCATED ON THE - Z AXIS
			100	LEM APS FUEL
CODE	(X,6)	130 ALL TANKS IF CODE 130 IS USED, DO NOT USE CODES 120, 111, 110, OR 100.		
		120 CSM OXIDIZER AND FUEL		
		111 LEM DPS FUEL		
		110 LEM DPS OXIDIZER		
		100 LEM APS FUEL AND OXIDIZER		

3.G.10

INPUT CARDS FOR LVCBLK2

MAINTAINED BY LEONARD SILVER

LVCBLK2 REQUIRES TWO INPUT CARDS WITH THE FOLLOWING FORMAT:

	FIELD 1	FIELD 2	FIELD 3	FIELD 4	FIELD 5	FIELD 6
CARD 1:	ID	PITCH	YAW	ROLL	WVX	WVY
CARD 2:	ID	WVZ	PRNTCTRL	CHNL	NOCL0	DTPRINT

THE SYMBOLS APPEARING ON THE INPUT CARDS ARE DEFINED BELOW AND ARE IDENTIFIED BOTH BY THE NAME WHICH APPEARS IN THE HEADINGS ABOVE, AND BY CARD AND FIELD NUMBER. THE NOTATION (N,M) MEANS CARD N, FIELD M.

ID (N,1) N = 1 TO 3. IDENTIFICATION NUMBER OF LVCBLK2. MUST CONTAIN THE VALUE "293" .

PITCH (1,2) SEE BELOW.

YAW (1,3) SEE BELOW.

ROLL (1,4) PITCH, YAW, AND ROLL FORM AN EULER SET. THE INITIAL VEHICLE AXIS ORIENTATION W.R.T. THE INITIAL PREFERRED AXIS ORIENTATION IS OBTAINED BY THE APPLICATION OF PITCH, YAW, AND ROLL, IN THAT SEQUENCE. ROTATIONS BEGIN AT THE PREFERRED SYSTEM AND TERMINATE PRODUCING THE VEHICLE SYSTEM. THE INITIAL ATTITUDE OF THE PREFERRED SYSTEM IS GIVEN BY:

$$\text{UNIT-Z PREF. RESOLVED ONTO REF. SYSTEM} = \text{UNIT} \begin{pmatrix} - \\ -R \\ I \end{pmatrix}$$

$$\text{UNIT-Y PREF. RESOLVED ONTO REF. SYSTEM} = \text{UNIT} \begin{pmatrix} - & - \\ V & * R \\ I & I \end{pmatrix}$$

$$\text{UNIT-X PREF. RESOLVED ONTO REF. SYSTEM} = (\text{UNIT-Y}) * (\text{UNIT-Z})$$

ANGLES ARE INPUT IN DEGREES.

WVX (1,5) SEE BELOW.

WVY (1,6) SEE BELOW.

WVZ (2,2) WVX, WVY, AND WVZ ARE THE X, Y, AND Z COMPONENTS OF THE INITIAL VEHICLE ANGULAR VELOCITY VECTOR EXPRESSED IN DEG/SEC IN THE VEHICLE SYSTEM.

PRNTCTRL (2,3) THIS VARIABLE, WHICH MAY BE EFFECTIVELY SET TO ZERO BY THE ENVCONTROLBLK2 PROGRAM, CONTROLS THE TYPE OF INFORMATION PRINTED BY LVCBLK2. 0 TURNS OFF VIRTUALLY ALL PRINTING. 1 PERMITS PRINTING OF VEHICLE RCS FUEL CONSUMPTION AND VARIABLES DESCRIBING VEHICLE MOTION, AS WELL AS RCS DISCRETES. 2 ADDS PRINTING OF VARIABLES SPECIFICALLY PERTAINING TO ATMOSPHERIC ENTRY, I.E. VEHICLE VELOCITY WITH RESPECT TO THE ATMOSPHERE, DYNAMIC PRESSURE AND DYNAMIC PRESSURE RATE, MACH NUMBER, ANGLE OF ATTACK, ETC.

CHNL (2,4) CHNL CONTROLS THE UTILIZATION OF THE CM RCS SYSTEMS AS FOLLOWS:

IF CHNL = ZERO: BOTH A AND B RCS SYSTEMS WILL BE EMPLOYED.

IF CHNL = 1: ONLY THE A RCS SYSTEM WILL BE ENABLED.

IF CHNL NEQ 0 OR 1: ONLY THE B RCS SYSTEM WILL BE ENABLED.

NOCLO (2,5) NOCLO IS A MULTIPLICATIVE FACTOR WHICH, IF EQUAL TO 1, WILL ADD A TERM TO THE ROLL AXIS AERODYNAMIC TORQUE EQUATION TO ACCOUNT FOR A LARGE ROLL PROTUBERANCE. OTHERWISE, NOCLO SHOULD BE SET TO ZERO.

DTPRINT (2,6) DTPRINT IS A TIME IN SECONDS WHICH CONTROLS FREQUENCY OF LVCBLK2 PRINTING, SUBJECT TO ENABLING BY PRNTCTRL (2,3).

IF DTPRINT = 0: EACH LVCBLK2 UPDATE WILL BE PRINTED.

IF DTPRINT NZ: LVCBLK2 PRINTS AT TIME INCREMENTS NEAREST DTPRINT.

IF DTPRINT = INF: ONLY RCS DISCONTINUITIES WILL BE PRINTED.

3.G.11

INPUT CARDS FOR IMUBLK2

MAINTAINED BY ROBERT WADSWORTH

IMUBLK2 REQUIRES THREE INPUT CARDS WITH THE FOLLOWING FORMAT:

	FIELD 1 -----	FIELD 2 -----	FIELD 3 -----	FIELD 4 -----	FIELD 5 -----	FIELD 6 -----
CARD 1:	ID	PACKEDBITS	DTPRINT	DTFIL	UNUSED	XCDU
CARD 2:	ID	YCDU	ZCDU	XERCTR	YERCTR	ZERCTR
CARD 3:	ID	YEULER	ZEULER	XEULER	OUTCTRL	TPLTFIL

AN ADDITIONAL THREE CARDS, DEFINING GYRO DRIFTS, AND PIPA BIAS AND SCALE ERRORS, MAY BE INCLUDED IN THE INPUT DECK IMMEDIATELY AFTER THE FIRST THREE CARDS. IF THE CARDS ARE OMITTED, ZEROES ARE ASSUMED. THE FORMAT IS:

	FIELD 1 -----	FIELD 2 -----	FIELD 3 -----	FIELD 4 -----	FIELD 5 -----	FIELD 6 -----
CARD 4:	ID	BD(0)	BD(1)	BD(2)	ASDIA(0)	ASDIA(1)
CARD 5:	ID	ASDIA(2)	ASDSA(0)	ASDSA(1)	ASDSA(2)	BA(0)
CARD 6:	ID	BA(1)	BA(2)	SF(0)	SF(1)	SF(2)

THE SYMBOLS APPEARING ON THE INPUT CARDS ARE DEFINED BELOW AND ARE IDENTIFIED BY THE NAME WHICH APPEARS IN THE HEADINGS ABOVE, AND BY CARD AND FIELD NUMBER. FOR EXAMPLE (1,2) MEANS CARD 1, FIELD 2.

ID (N,1) N = 1 TO 6. IDENTIFICATION NUMBER OF IMUBLK2. MUST CONTAIN THE VALUE "270" .

PACKEDBITS (1,2) THIS WORD CONTAINS 0 OR NZ IN INDIVIDUAL DIGIT POSITIONS (COUNTING FROM RIGHT TO LEFT, STARTING AT THE DECIMAL POINT), SIGNIFYING:

- DIGIT 1: COARSE ALIGN ENABLE BIT (CH 12, BIT 4)
- DIGIT 2: ZERO IMU CDUS BIT (CH 12, BIT 5)
- DIGIT 3: ENABLE IMU ERROR COUNTER BIT (CH 12, BIT 6)
- DIGIT 4: ISS TURN ON DELAY COMPLETE BIT (CH 12, BIT 15)
- DIGIT 5: IMU POWER (0 = ON, NZ = OFF)
- DIGIT 6: CONTROLS THE SPEED AND ACCURACY OF THE IMU MODEL BY INCLUDING OR ELIMINATING THE CDU SWITCHING DYNAMICS.
 - 0 = SLOW DETAILED MODEL
 - 1 = FAST SIMPLIFIED MODEL
- DIGITS 7 THROUGH 10 ARE NOT USED.

DTPRINT	(1,3)	PRINT PERIOD CONTROL FOR "STOP-TIME" PRINTING. WHEN STOP-TIME PRINTING IS DONE, FURTHER PRINTING OF THIS TYPE IS INHIBITED UNTIL THE NEXT MULTIPLE OF DTPRINT. UNITS ARE SECONDS.
DTFILE	(1,4)	THIS VARIABLE MAY BE USED TO LIMIT THE AMOUNT OF DATA FILED FOR SUBSEQUENT PLOTTING. FILING, IF ACTIVE (SEE FIELD 3,6), WILL OCCUR ONLY AT INTERVALS GREATER THAN OR EQUAL TO DTFILE.
	(1,5)	IS UNUSED.
XCDU	(1,6)	THESE SPECIFY, IN DEGREES, THE CONTENTS OF THE GIMBAL-ANGLE COUNTERS IN THE CDU'S. (THE AGC
YCDU	(2,2)	CDU COUNTERS ARE SET BY SIMULATOR SPECIAL REQUESTS.) PLEASE NOTE THAT IMU GIMBAL ANGLES ARE
ZCDU	(2,3)	NOT DEPENDANT ON XCDU, ETC. INITIAL VALUES OF GIMBAL ANGLES DEPEND UPON:
		1) R AND V INPUTS TO UNIVERSE
		2) PITCH, YAW, AND ROLL INPUT TO VEHICLE, AND
		3) EULER ANGLE INPUT TO IMUBLK2.
		IF ALL THREE COUNTERS ARE SPECIFIED AS ZERO, IMUBLK2 WILL INITIALIZE THE GA COUNTERS EQUAL TO THE GIMBAL ANGLES. (IF THE USER WANTS 0, 0, 0 HE SHOULD SPECIFY 0, 0, 0.000001)
XERCTR	(2,4)	THESE SPECIFY, IN COUNTS, THE INITIAL CONTENTS OF THE ISS ERROR COUNTERS.
YERCTR	(2,5)	
ZERCTR	(2,6)	
YEULER	(3,2)	SUCCESSIVE EULER ROTATIONS, IN DEGREES, FOR INITIAL PLATFORM ORIENTATION. THE 0, 0, 0
ZEULER	(3,3)	ORIENTATION IS THE UNIVERSE COORDINATE FRAME, UNLESS THE RUN STARTS IN THE PRELAUNCH MODE.
XEULER	(3,4)	IN THE LATTER CASE, THE REFERENCE FRAME IS A GEOGRAPHIC FRAME, WITH AXES UP (ALONG GRAVITY), SOUTH, AND EAST. (SEE THE PRELAUNCH PROGRAM FOR DETAILS.)
OUTCTRL	(3,5)	THIS VARIABLE, WHICH MAY BE EFFECTIVELY SET TO ZERO BY THE ENVCNTRL PROGRAM, CONTROLS THE TYPE OF INFORMATION PRINTED BY IMUBLK2. 0 TURNS OFF PRINTING. 2 PERMITS PRINTING OF IMU PULSE TRAIN COMMANDS. 3 ADDS PRINTING OF A NUMBER OF QUANTITIES AT EACH IMU STOP-TIME. 4 ADDS PRINTING WITHIN THE GIMBAL ANGLE ENCODING LOOP. 5 ADDS PRINTING WITHIN THE COARSE ALIGN LOOP. TO PRINT A SINGLE OPTION ABOVE, INDEPENDENT OF THE OTHER OPTIONS, SET OUTCTRL TO THE VALUE OF THE DESIRED OPTION PLUS 0.1
TPLOTFIL	(3,6)	AT VALUES OF T ABOVE TPLOTFIL, IMU WILL WRITE INTO THE DATA FILE ENTRIES WHICH MAY BE RECOVERED AND PLOTTED IN A SURSEQUENT RUN. THIS VARIABLE SHOULD NORMALLY BE SET TO A LARGE NUMBER, TO INHIBIT PLOTTING. PLOT DATA ARE FILED IN THE DATAFILE STARTING AT LOCATION 50000. SUBJECT TO THE LIMITATION OF (1,4), TIME, GIMBAL ANGLES, CDU COUNTS, AND GIMBAL ANGLE RATES ARE FILED AFTER EACH UPDATE.

BD(0)	(4,2)	GYRO BIAS DRIFTS, IN MERU.
BD(1)	(4,3)	
BD(2)	(4,4)	
ASDIA(0)	(4,5)	GYRO DRIFTS CAUSED BY ACCELERATIONS ALONG GYRO INPUT AXES, IN MERU / G .
ASDIA(1)	(4,6)	
ASDIA(2)	(5,2)	
ASDSA(0)	(5,3)	GYRO DRIFTS CAUSED BY ACCELERATIONS ALONG GYRO SPIN AXES, IN MERU / G .
ASDSA(1)	(5,4)	
ASDSA(2)	(5,5)	
BA(0)	(5,6)	PIPA BIAS ACCELERATIONS, IN CM / SEC SQ .
BA(1)	(6,2)	
BA(2)	(6,3)	
SF(0)	(6,4)	PIPA SCALE FACTOR ERRORS, IN PARTS PER MILLION. (IF SF = 2.0 PPM, THIS MEANS THAT FOR EVERY
SF(1)	(6,5)	PULSE'S WORTH OF DELTA-V INTO THE PIPA, ONLY 1/1.000002 PULSES COME OUT.)
SF(2)	(6,6)	

3.G.12

INPUT CARDS FOR OPTICS82

MAINTAINED BY LANCE DRANE

OPTICS82 MAY HAVE ONE OPTIONAL INPUT CARD WITH THE FOLLOWING FORMAT. IF NO INPUT CARD IS SUPPLIED, THE PROGRAM ASSUMES NOMINAL VALUES FOR THESE PARAMETERS, AS DEFINED BELOW.

	FIELD 1	FIELD 2	FIELD 3	FIELD 4	FIELD 5	FIELD 6
	-----	-----	-----	-----	-----	-----
CARD 1:	ID	OUTCTRL	SFT ANGLE	TRUN ANGLE	UNUSED	UNUSED

MEANING OF SYMBOLS

ID IDENTIFICATION NUMBER OF OPTICSBLK2. MUST CONTAIN THE VALUE "213".

OUTCTRL CONTROLS THE AMOUNT AND TYPE OF PRINTING FROM THE OPTICSBLK2 PROGRAM. (MAY BE OVERRIDDEN BY A ZERO IN THE PRNTCTRL FIELD OF THE INPUTPROG CARD.)
 0 = TURN OFF VIRTUALLY ALL PRINTING.
 1 = PRINT AGC COMMANDS TO OPTICS AND CHANGES IN PANEL SWITCH SETTINGS.
 2 = PRINT WHEN AGC PULSE TRAINS ARE TURNED ON OR OFF.
 3 = PRINT GENERAL OPTICS INFORMATION EACH TIME OPTICSBLK2 IS CALLED.
 NORMAL VALUE IS 3.

SFT ANGLE INITIAL SHAFT ANGLE, IN DEGREES. NORMALLY 0.0

TRUN ANGLE INITIAL TRUNNION ANGLE, IN DEGREES. NORMALLY 0.0

UNUSED WILL BE READ AND THEN IGNORED. MUST BE A REAL NUMBER HOWEVER. NORMALLY 0.

IF AN INPUT CARD IS NOT SUPPLIED, THE PROGRAM WILL ASSUME THE FOLLOWING VALUES:

ID = 213
 OUTCTRL = 3
 SFT ANGLE = 0
 TRUN ANGLE = 0

INPUT CARDS FOR RRADAR

MAINTAINED BY KEITH GLICK

RADAR REQUIRES FOUR DATA CARDS WITH THE FOLLOWING FORMAT:

	FIELD 1 -----	FIELD 2 -----	FIELD 3 -----	FIELD 4 -----	FIELD 5 -----	FIELD 6 -----
CARD 1:	ID	SHAFT	TRUNNION	S ERCTR	T ERCTR	UNUSED
CARD 2:	ID	S OUTCTR	T OUTCTR	OUTCTRL	DTPRINT	UNUSED
CARD 3:	ID	TRKDEL	TFILE1	TFILE2	POWEROFF	UNUSED
CARD 4:	ID	S ERROR	T ERROR	E RANGE	ERRATE	FAST

MEANING OF SYMBOLS:

ID IDENTIFICATION NUMBER OF RRADAR. MUST HAVE THE VALUE "13".

SHAFT INITIAL ANGLE OF ANTENNA SHAFT, IN DEGREES.

TRUNNION INITIAL ANGLE OF ANTENNA TRUNNION, IN DEGREES.

S ERCTR INITIAL SHAFT RATE, IN DEGREES PER SECOND. (NORMALLY 0)

T ERCTR INITIAL TRUNNION RATE, IN DEGREES PER SECOND. (NORMALLY 0)

S OUTCTR INITIAL VALUE OF SHAFT CDU COUNTER, IN DEGREES.

T OUTCTR INITIAL VALUE OF TRUNNION CDU COUNTER, IN DEGREES.

OUTCTRL CONTROLS THE AMOUNT AND TYPE OF PRINTING FROM THE RADAR PROGRAM. (MAY BE OVERRIDDEN BY A ZERO IN THE PRNTCTRL FIELD OF THE INPUTPRG CARD.)
 0 = TURN OFF VIRTUALLY ALL PRINTING.
 1 = PRINT ANTENNA ANGLES, CDU COUNTERS, RANGE, AND RANGE RATE AT REGULAR INTERVALS OF APPROXIMATELY DTPRINT SECONDS.
 2 = SAME AS 1, BUT PRINT ADDITIONAL INFORMATION SUCH AS ERROR COUNTER VALUES AND RATES.

DTPRINT APPROXIMATE TIME INTERVAL BETWEEN EACH ON-LINE PRINTOUT, IN SECONDS.

TRKDEL DELAY, IN SECONDS, FROM THE TIME TRACKING CONDITIONS ARE SATISFIED UNTIL THE RADAR ACQUIRES THE TARGET AND RETURNS THE "DATA GOOD" DISCRETE. THE RADAR MAY BE PREVENTED FROM ACQUIRING THE TARGET BY SETTING THIS VARIABLE TO A LARGE NUMBER.

TFILE1	TIME, IN SECONDS, AT WHICH THE RADAR PROGRAM BEGINS FILING DATA FOR PLOTTING.
TFILE2	TIME, IN SECONDS, AT WHICH THE RADAR PROGRAM STOPS FILING DATA FOR PLOTTING.
POWEROFF	IF NON-ZERO, THIS VARIABLE CAUSES THE RADAR TO BE BYPASSED UNTIL AN ASTRONAUT ACTION TURNS ON THE POWER.
S ERROR	SHAFT ANGLE ERROR, IN RADIANS.
T ERROR	TRUNNION ANGLE ERROR, IN RADIANS.
E RANGE	RANGE ERROR, IN METERS.
ERRATE	RANGE RATE ERROR, IN METERS PER SECOND.
FAST	CONTROLS THE SPEED AND ACCURACY OF THE RADAR MODEL BY INCLUDING OR ELIMINATING THE CDU SWITCHING DYNAMICS. 0 = SLOW DETAILED MODEL. 1 = FAST SIMPLIFIED MODEL.

3.G.14

INPUT CARDS FOR LRADAR

MAINTAINED BY KEITH GLICK

LRADAR REQUIRES THREE INPUT CARDS WITH THE FOLLOWING FORMAT:

	FIELD 1 -----	FIELD 2 -----	FIELD 3 -----	FIELD 4 -----	FIELD 5 -----	FIELD 6 -----
CARD 1:	ID	K	MEMBIT	VDG1	VDG2	VDG3
CARD 2:	ID	VDG4	VDG5	RDG1	RDG2	RDG3
CARD 3:	ID	RDG4	RDG5	OUTCTRL	UNUSED	UNUSED

MEANING OF SYMBOLS

ID IDENTIFICATION NUMBER FOR LRADAR. MUST HAVE THE VALUE "13".

K INITIAL LANDING RADAR ANTENNA POSITION. (1 OR 2)

MEMBIT A VARIABLE WHICH INDICATES ALTITUDE HISTORY OF LM. THIS VARIABLE IS USED ONLY WHEN THE SLANT RANGE TO THE SURFACE IS LESS THAN 3000 FEET BUT GREATER THAN 2500 FEET.

 1 IF SLANT RANGE HAS BEEN LESS THAN 2500 FEET.

 2 IF SLANT RANGE HAS NOT BEEN LESS THAN 2500 FEET.

VDG1-5 A SET OF FIVE TIMES AT WHICH THE VELOCITY DATA GOOD DISCRETE WILL BE REMOVED FOR A THREE SECOND PERIOD AND THEN RESTORED. THE TIME IS MEASURED IN SECONDS FROM EPOCH.

RDG1-5 THE ANALOGUE OF VDG FOR RANGE DATA GOOD DISCRETE.

OUTCTRL THIS VARIABLE, WHICH MAY BE EFFECTIVELY SET TO ZERO BY THE ENVCONTROLBLK2 PROGRAM, CONTROLS THE PRINTING BY LRADAR.

3.G.15

INPUT CARDS FOR TARGET

MAINTAINED BY KEITH GLICK

TARGET REQUIRES TWO INPUT CARDS WITH THE FOLLOWING FORMAT:

	FIELD 1 -----	FIELD 2 -----	FIELD 3 -----	FIELD 4 -----	FIELD 5 -----	FIELD 6 -----
CARD 1:	ID	T	DTPRINT	RT(X)	RT(Y)	RT(Z)
CARD 2:	ID	UNUSED	UNUSED	VT(X)	VT(Y)	VT(Z)

MEANING OF SYMBOLS

ID IDENTIFICATION NUMBER OF TARGET. MUST HAVE THE VALUE "14".

T TIME AT WHICH INITIAL VALUES OF THE TARGET STATE VECTORS RT AND VT ARE SPECIFIED.
 MEASURED IN SECONDS FROM EPOCH.

DTPRINT APPROXIMATE TIME INTERVAL BETWEEN EACH ON-LINE PRINTOUT, IN SECONDS.

RT(X) INITIAL POSITION VECTOR OF TARGET AT TIME T, IN METERS.
RT(Y) (EXPRESSED IN REFERENCE INERTIAL COORDINATES)
RT(Z)

VT(X) INITIAL VELOCITY VECTOR OF TARGET AT TIME T, IN METERS/SECOND.
VT(Y) (EXPRESSED IN REFERENCE INERTIAL COORDINATES)
VT(Z)

MAINTAINED BY STEVE SAVATSKY

A CHARACTER OTHER THAN A OR R IN COLUMN 1 SIGNALS THE END OF THE ASTRONAUT INPUT DECK AND CAUSES A RETURN TO ENVCNTRL. CARDS WHICH HAVE R IN COLUMN 1 ARE PRINTED, SO THAT REMARKS MAY BE USED AS DESIRED.

THERE ARE FOUR TYPES OF INPUT STATEMENTS FOR THE ASTRONAUT ROUTINE, WHICH MAY BEGIN ON A CARD ANYWHERE IN OR AFTER COLUMN 9. EACH INPUT STATEMENT CONSISTS OF TWO CLAUSES -- A CONDITION CLAUSE, AND AN ACTION CLAUSE.

	CONDITION CLAUSE	ACTION CLAUSE
	-----	-----
1)	TIME XXXX.XXX	(ASTRONAUT ACTIONS)
2)	WAIT XXXX.XXX	(ASTRONAUT ACTIONS)
3)	IF (DISPLAY CONDITIONS)	THEN (ASTRONAUT ACTIONS)
4)	VERIFY (DISPLAY CONDITIONS)	THEN (ASTRONAUT ACTIONS)

THESE INPUT STATEMENTS REQUEST CERTAIN ASTRONAUT ACTIONS TO BE PERFORMED WHEN

TIME STATEMENT

- 1) A CERTAIN TIME IS REACHED. THE DESIRED TIME IS SPECIFIED IN THE CONDITION CLAUSE AFTER THE WORD "TIME", AND THE ASTRONAUT ACTIONS TO BE PERFORMED ARE SPECIFIED IN THE ACTION CLAUSE. (TIMES ARE IN SECONDS, AND ARE SPECIFIED IN THE FORM 123.456 . UP TO TEN SIGNIFICANT DIGITS WILL BE PROCESSED. LEADING AND TRAILING ZEROS ARE OPTIONAL. A POSITIVE SIGN MAY BE GIVEN IF DESIRED, BUT A NEGATIVE SIGN IS ILLEGAL, SINCE NEGATIVE TIMES MAY NOT BE USED. NO ADDITIONAL WORD SUCH AS "THEN" IS NEEDED TO INTRODUCE THE ACTION CLAUSE.)

WAIT STATEMENT

- 2) A CERTAIN AMOUNT OF TIME HAS ELAPSED SINCE THE LAST ASTRONAUT ACTION. THE DESIRED TIME INTERVAL IS SPECIFIED IN THE CONDITION CLAUSE AFTER THE WORD "WAIT", AND THE ASTRONAUT ACTIONS TO BE PERFORMED ARE SPECIFIED IN THE ACTION CLAUSE. (INTERVALS ARE SPECIFIED THE SAME AS TIMES. NO ADDITIONAL WORD SUCH AS "THEN" IS NEEDED TO INTRODUCE THE ACTION CLAUSE. THE WORDS "IF" AND "THEN" MAY NOT BE USED IN EITHER THE TIME OR THE WAIT STATEMENTS.)

IF-THEN STATEMENT

- 3) THE VERB-NOUN DISPLAYS START FLASHING, AND A CERTAIN DISPLAY PATTERN EXISTS. (THE DESIRED DISPLAY PATTERNS ARE SPECIFIED IN THE CONDITION CLAUSE AFTER THE WORD "IF", AND THE ASTRONAUT ACTIONS TO BE PERFORMED ARE SPECIFIED IN THE ACTION CLAUSE AFTER THE WORD "THEN".)

VERIFY STATEMENT

- 4) A CERTAIN DISPLAY PATTERN EXISTS. UNLIKE THE IF-THEN STATEMENT, VERIFY CHECKS THE PRESENT DISPLAY WHETHER OR NOT THE VERB-NOUN DISPLAYS START FLASHING. (THE DESIRED DISPLAY PATTERNS ARE SPECIFIED IN THE CONDITION CLAUSE AFTER THE WORD "VERIFY", AND THE ASTRONAUT ACTIONS TO BE PERFORMED ARE SPECIFIED IN THE ACTION CLAUSE AFTER THE WORD "THEN".)

RELATIVELY LONG STATEMENTS WHICH WILL NOT FIT ON A SINGLE CARD MAY BE CONTINUED FROM CARD TO CARD.

CONDITION CLAUSES

THE CONDITION CLAUSE OF A "TIME" OR "WAIT" STATEMENT HAS ONE SIMPLE FORMAT WHICH WAS EXPLAINED ABOVE. "IF-THEN" AND "VERIFY" CONDITION CLAUSES ARE MORE FLEXIBLE AND REQUIRE FURTHER EXPLANATION.

IF-THEN STATEMENTS

AN IF-THEN STATEMENT MAY HAVE ANY OF FOUR BASIC FORMATS.

- 1) IF (DISPLAY CONDITIONS) THEN (ASTRONAUT ACTIONS)
- 2) IF WITHIN X (DISPLAY CONDITIONS) THEN (ASTRONAUT ACTIONS)
- 3) IF NOABORT (DISPLAY CONDITIONS) THEN (ASTRONAUT ACTIONS)
OTHERWISE (ASTRONAUT ACTIONS) END
- 4) IF WITHIN X NOABORT (DISPLAY CONDITIONS) THEN (ASTRONAUT ACTIONS) OTHERWISE (ASTRONAUT ACTIONS) END

"WITHIN X" IS THE TIME LIMIT IN SECONDS FOR THE VERB-NOUN FLASH TO COME ON AND THE DISPLAY CONDITIONS TO BE MET. IF THE CONDITIONS ARE NOT MET WITHIN X SECONDS, THE SIMULATION ABORTS -- UNLESS THE NOABORT OPTION IS SPECIFIED, IN WHICH CASE THE OTHERWISE CLAUSE IS EXECUTED.

"NOABORT" IS DESCRIBED IN ANOTHER SECTION BELOW.

THE STATEMENT "IF THEN (ASTRONAUT ACTIONS)", WITH NO DISPLAY CONDITIONS SPECIFIED, MAY BE USED TO CAUSE AN UNCONDITIONAL ASTRONAUT ACTION THE NEXT TIME THE VERB-NOUN DISPLAYS START FLASHING.

DISPLAY CONDITIONS

ANY NUMBER FROM ZERO TO TEN DISPLAY CONDITIONS MAY BE SPECIFIED IN THE "IF" CLAUSE. THE DESIRED DISPLAY CONDITIONS MAY BE LISTED IN ANY ORDER. COMMAS MAY BE USED AS DESIRED.

EACH DISPLAY CONDITION IS SPECIFIED BY A THREE WORD PHRASE COMPRISING

- 1) THE NAME OF THE DISPLAY,
- 2) THE RELATION THE DISPLAY SHOULD HAVE TO SOME SPECIFIED VALUE, AND
- 3) THE VALUE WITH WHICH THE DISPLAY IS TO BE COMPARED, IN THAT ORDER.

	DISPLAY	RELATION	VALUE
E.G.	VERB	=	11
	R1	GEQ	+01200
	CHECK FAIL	=	ON

NUMERIC DISPLAYS

THE SIX NUMERIC DISPLAYS ARE REFERRED TO BY THE FOLLOWING NAMES.

MODE OR M
VERR OR V
NOUN OR N
R1
R2
R3

THE CONTENT OF EACH NUMERIC DISPLAY MAY BE COMPARED TO SEE IF IT IS "EQUAL TO", "UNEQUAL TO", "LESS THAN", "LESS THAN OR EQUAL TO", "GREATER THAN", OR "GREATER THAN OR EQUAL TO" SOME SPECIFIED VALUE. THESE COMPARISONS ARE REFERRED TO AS FOLLOWS.

BLANK, = OR EQ	(EQUAL TO)
UNEQ	(NOT EQUAL TO)
L	(LESS THAN)
LEQ	(LESS THAN OR EQUAL TO)
G	(GREATER THAN)
GEQ	(GREATER THAN OR EQUAL TO)

VALUES FOR THE MODE, VERR, AND NOUN DISPLAYS ARE SPECIFIED AS TWO DIGIT OCTAL NUMBERS. VALUES FOR R1, R2, AND R3 MAY BE ANY FIVE DIGIT NUMBER EITHER OCTAL OR DECIMAL. LEADING ZEROS MAY BE OMITTED. A SIGNED NUMBER IS ASSUMED TO BE DECIMAL, UNSIGNED OCTAL. OCTAL VALUES ARE ALWAYS CONSIDERED POSITIVE AND RANGE FROM 00000 TO 77777.

FOR EXAMPLE

IF MODE UNEQ 16, VERR = 16, NOUN EQ 44,
R1 G +17003, R2 L +42, R3 LEQ -164,
THEN (ASTRONAUT ACTIONS).

OFF-ON DISPLAYS

ALARM LIGHTS AND THE VERR-NOUN FLASH MAY BE CHECKED FOR ON OR OFF CONDITION. THEY ARE REFERRED TO BY THE FOLLOWING NAMES.

CHECK FAIL (OR CF)
OPERATOR ERROR (OR OE)
PROGRAM ALARM (OR PA)
PROGRAM CAUTION (OR PC)
RELEASE DISPLAY (OR RD)
FLASH

FOR EXAMPLE

IF CHECK FAIL = OFF, THEN (ASTRONAUT ACTIONS)
IF RELEASE DISPLAY ON, THEN (ASTRONAUT ACTIONS)

NOABORT CONDITIONALS

THE APPEARANCE OF THE WORD "NOABORT" (CORRECTLY POSITIONED) IN AN "IF" STATEMENT MEANS THAT, DEPENDING ON WHETHER THE DISPLAY CONDITIONS ARE EVERYWHERE TRUE OR ANYWHERE FALSE, ONE OR THE OTHER OF TWO DIFFERENT ACTION CLAUSES WILL BE EXECUTED.

1) IF ALL CONDITIONS ARE TRUE, THE ASTRONAUT ACTIONS LISTED IN THE "THEN" CLAUSE (FROM THE WORD "THEN" TO THE WORD "OTHERWISE") ARE EXECUTED AND THE ACTIONS LISTED IN THE "OTHERWISE" CLAUSE (FROM THE WORD "OTHERWISE" TO THE WORD "END") ARE SKIPPED.

2) IF ANY OF THE CONDITIONS ARE FALSE, THE "THEN" CLAUSE IS SKIPPED AND THE "OTHERWISE" CLAUSE IS EXECUTED.

THE WORDS "OTHERWISE" AND "END" MAY NOT APPEAR IN THE CONDITIONAL PART OF ANY STATEMENT (I.E., FROM "IF" TO "THEN" OR FROM "VERIFY" TO "THEN"). HOWEVER, A FULL "IF NOABORT-THEN-OTHERWISE-END" SEQUENCE MAY SPAN AS MANY CARDS AS DESIRED.

A NESTING RULE APPLIES FOR ALL "NOABORT-THEN-OTHERWISE-END" SEQUENCES IN THAT IF A SECOND "NOABORT" APPEARS BETWEEN THE "THEN" AND THE "OTHERWISE" FOR THE FIRST "NOABORT", THEN THE "THEN-OTHERWISE-END" OF THAT SECOND "NOABORT" MUST APPEAR BEFORE THE "OTHERWISE" OF THE FIRST "NOABORT". SIMILARLY, IF A SECOND "NOABORT" APPEARS BETWEEN THE "OTHERWISE" AND THE "END" FOR THE FIRST "NOABORT", THEN THE "THEN-OTHERWISE-END" OF THE SECOND "NOABORT" MUST APPEAR BEFORE THE "END" FOR THE FIRST "NOABORT". UNDER THIS RULE, NESTING MAY BE INFINITE.

VERIFY STATEMENTS

A "VERIFY" STATEMENT MAY HAVE ANY OF SIX BASIC FORMATS.

- 1) VERIFY (DISPLAY CONDITIONS) THEN (ASTRONAUT ACTIONS)
- 2) VERIFY WITHIN X (DISPLAY CONDITIONS) THEN (ASTRONAUT ACTIONS)
- 3) VERIFY EVERY Y WITHIN X (DISPLAY CONDITIONS) THEN (ASTRONAUT ACTIONS)
- 4) VERIFY NOABORT (DISPLAY CONDITIONS) THEN (ASTRONAUT ACTIONS) OTHERWISE (ASTRONAUT ACTIONS) END
- 5) VERIFY WITHIN X NOABORT (DISPLAY CONDITIONS) THEN (ASTRONAUT ACTIONS) OTHERWISE (ASTRONAUT ACTIONS) END
- 6) VERIFY EVERY Y WITHIN X NOABORT (DISPLAY CONDITIONS) THEN (ASTRONAUT ACTIONS) OTHERWISE (ASTRONAUT ACTIONS) END

VERIFY IS LIKE THE IF-THEN STATEMENT, EXCEPT THAT IT CHECKS THE PRESENT DISPLAY PATTERN WHETHER OR NOT THE VERB-NOUN DISPLAYS START FLASHING. DISPLAY CONDITIONS ARE THE SAME AS FOR THE IF-THEN STATEMENT.

THE VERIFICATION TAKES PLACE ONLY ONCE, UNLESS "WITHIN X" IS SPECIFIED. THEN VERIFICATION TAKES PLACE EVERY 1 SECOND FOR X SECONDS. IF, INSTEAD, "EVERY Y WITHIN X" IS SPECIFIED, THE VERIFICATION INTERVAL BECOMES EVERY Y SECONDS RATHER THAN EVERY 1 SECOND.

NOABORT BEHAVES THE SAME AS IN THE IF-THEN STATEMENT. WHEN "NOABORT" IS NOT SPECIFIED, IF VERIFICATION OF THE DISPLAY FAILS, THE SIMULATION ABORTS.

ACTION CLAUSES

ACTION CLAUSES ARE THE SAME FOR ALL TYPES OF INPUT STATEMENT. THE ACTION CLAUSE CAN SPECIFY VIRTUALLY ANY SEQUENCE OF KEYSTROKES, PAUSES, OR MARKS, AND CONTROL THE SETTINGS OF PANEL SWITCHES AND HAND-CONTROLLERS.

KEYSTROKES

ANY OF THE FOLLOWING KEYSTROKES MAY BE SPECIFIED IN THE ACTION CLAUSE.

DIGITS 0-9

+

-

VERB (OR V)

NOUN (OR N)

ENTER (OR E)

CLEAR (OR C)

ERROR RESET (OR ER)

KEY RELEASE (OR KR)

CONSECUTATIVE DIGIT KEYSTROKES SHOULD BE GROUPED INTO SINGLE WORDS.

FOR EXAMPLE

TIME 35 V 21 N 16 E +07525 E

"KEYSTROKES FOLLOW" OR "UPTTEL FOLLOWS" MAY BE USED TO CAUSE KEYSTROKES FROM THE ASTRONAUT PROGRAM TO BE TRANSMITTED VIA DSKY OR UP-TELEMETRY (INITIAL STATE OF ASTRONAUT IS DSKY-MODE). "KEYSTROKES FOLLOW" MAY BE PRECEDED BY ONE OF THE WORDS "MAIN" OR "NAV" ("MAIN" IS ASSUMED IF NEITHER IS SPECIFIED).

SWITCHES, PUSHBUTTONS, AND HAND-CONTROLLERS

CERTAIN SWITCHES, PUSHBUTTONS, AND HAND-CONTROLLERS MAY BE CONTROLLED VIA THE ASTRONAUT PROGRAM. THE FUNCTIONS RECOGNIZED ARE

KEY WORD	MODIFIERS
-----	-----
MARK	
MARKREJ	
MARKX	
MARKY	
UPTELSW	ACCEPT BLOCK
IMUPOWER	OFF ON
OPTMODE	MANUAL AGC ZERO ENCODERS
SCSMODE	OFF FREE ATTITUDE HOLD AUTO
SCS05GSW	ON OFF
CMSMSEP	
S4BSEP	
GUIDEMODE	BACKUP PRIMARY
HANDCTRL	NEUTRAL NO PITCH + PITCH - PITCH NO ROLL + ROLL - ROLL NO YAW + YAW - YAW

BLOCK I ONLY

KEY WORD	MODIFIERS
-----	-----
TRNSW	AGC MANUAL
IMUMODE	ZERO ENCODERS FINE ALIGN COARSE ALIGN LOCK CDUS ATTITUDE CONTROL ENTRY
G&NSYNCH	ON OFF
SCSMODE	G&N ATTITUDE CONTROL G&N DV G&N ENTRY SCS ATTITUDE CONTROL SCS DV SCS ENTRY

MULTIPLE MODIFIERS MAY BE USED TO CAUSE CONTROL OF SEVERAL AXES AT ONCE. IF A PARTICULAR AXIS IS NOT EXPLICITLY SPECIFIED, A COMMAND IS GENERATED TO PLACE THAT AXIS INTO THE CONDITION LAST SPECIFIED VIA THIS PROGRAM (INITIAL CONDITION = "NEUTRAL").

MINIMPLS (SAME AS HANDCTRL)

TRANSCTL NEUTRAL
 NO X
 + X
 - X
 NO Y
 + Y
 - Y
 NO Z
 + Z
 - Z
 ENGINE OFF
 ABORT

LEMHCTRL P
 Q
 R

25

LRHCRATE XX

ABORTSTG

ASENGARM ON
 OFF

DCENGARM ON
 OFF

LEMTHROT XX

MULTIPLE MODIFIERS MAY BE USED WITH TRANSCTL, AS WITH HANDCTRL.

THE WORD FOLLOWING THE P, Q, OR R MUST BE A SIGNED OR UNSIGNED DECIMAL CONSTANT, SIGNIFYING HAND CONTROLLER DEFLECTION IN UNITS OF DEGREES/SEC OF LEM ATTITUDE RATE.

THIS WORD MUST BE FOLLOWED BY THE WORD "INFINITE" OR BY AN UNSIGNED DECIMAL CONSTANT, SIGNIFYING HAND CONTROLLER DEFLECTION RATE IN UNITS OF DEGREES/SEC/ SEC OF LEM ATTITUDE ACCELERATION.

CAUSE LEM ASCENT-DESCENT STAGING.

ENABLES OR DISABLES FIRING OF THE ASCENT ENGINE.

ENABLES OR DISABLES FIRING OF THE DESCENT ENGINE, AND ENABLES OR DISABLES "AUTO" THROTTLE. WHEN "AUTO" THROTTLE IS DISABLED, THE THROTTLE IS RESET TO THE MANUAL THROTTLE SETTING.

SET THE DESCENT MANUAL THROTTLE TO XX PERCENT. (E.G. IF XX IS 94, THE MANUAL THROTTLE WILL BE SET TO 94 PERCENT. THE AUTO THROTTLE SETTING IS THEN ADDED TO THE MANUAL THROTTLE FOR THE TOTAL THROTTLE SETTING.)

LEMRA DAR	OFF ON	TURN ON OR OFF THE LEM RENDEZVOUS RADAR POWER.
LVC RCS	BOTH A B	DETERMINES WHICH RCS SYSTEM A, B, OR BOTH IS ACTIVATED FOR THE COMMAND MODULE IN ENTRY.
THRUST SW	MANUAL AUTO	URNS CONTROL OF THE DESCENT ENGINE THROTTLE OVER TO THE ASTRONAUT OR THE LGC.
IU CMC		BOOSTER DOES OWN STEERING. COMMANDS THE CMC TO STEER THE SIVB.
PROCEED		CAUSES THE STANDBY BUTTON TO BE DEPRESSED FOR 150 MILLISECONDS.
CSMRATE	0 1 2 3	WHEN DOING ANY ASTRONAUT SUBPROGRAM USING THE CSM "ASTALINE" CLOSED LOOP RATE MANEUVER, PUT THIS ACTION IN THE ASTRONAUT FILE IMMEDIATELY BEFORE THE DESIRED SUBPROGRAM. THIS WILL GIVE THE ASTRONAUT SUBPROGRAM THE SAME RATE AS THE DAP: <div style="margin-left: 40px;"> CSMRATE 0 MEANS 0.05 DEG / SEC 1 MEANS 0.2 DEG / SEC 2 MEANS 0.5 DEG / SEC 3 MEANS 4.0 DEG / SEC </div>

DEPLOY LANDING GEAR

NESTED STATEMENTS IN ACTION CLAUSES

"WAIT" STATEMENTS MAY BE IMBEDDED IN ANY ACTION CLAUSE.
AN IMBEDDED WAIT STATEMENT BEHAVES LIKE THE NORMAL WAIT STATEMENT,
EXCEPT THAT KEYSTROKES, INSTEAD OF BEING DELAYED, ARE SENT IMMEDIATELY
TO THE COMMUNICATOR PROGRAM WITH THE DELAY TIME APPENDED.

ASTRONAUT SUBPROGRAMS

CERTAIN ROUTINES SUCH AS STAR SIGHTING AND MARKING ,
SPACECRAFT MANEUVERING , LANDMARK TRACKING , ETC CAN
BE ACCOMPLISHED BY USING INTERNAL ASTRONAUT
SUBPROGRAMS VIA THE ASTRONAUT CARD STATEMENT

SUBPROGRAM XXXXX

WHERE XXXXX IS A 1 - 5 DIGIT UNSIGNED INTEGER
SPECIFYING WHICH SUBPROGRAM TO USE. ONLY ONE
SUBPROGRAM CAN BE ACTIVE AT ONE TIME, BUT ASTRONAUT WORD
PROCESSING WILL CONTINUE UNLESS THE STATEMENT

HOLD CARDS

IS GIVEN. THEN, NO FURTHER ASTRONAUT WORDS WILL BE
PROCESSED UNTIL THE CURRENT SUBPROGRAM CONCLUDES ITS
ACTION AND RELEASES THE HOLD.

DO NOT USE ANY OF THESE SUBPROGRAMS UNLESS YOU HAVE
CAREFULLY DISCUSSED THIS WITH THE ASTRONAUT PROGRAMMER.

THE CURRENT SUBPROGRAM NUMBER ASSIGNMENTS ARE

- 1 MANEUVER THE CSM TO GIMBLE ANGLES OGA, IGA, MGA SPECIFIED IN
R1, R2, R3 RESPECTIVELY.
- 2 MANEUVER THE CSM TO 2 GOOD STARS AND THEN MARK THEM.
THE ASTRONAUT ROUTINE FIRST MAKES NUMEROUS STAR CALLS TO
UNIVERSE TO FIND THE OPTIMUM SET OF 2 STARS, NEXT THE
MANEUVER IS MADE, AND FINALLY THE MARK IS PERFORMED.
- 3 CSM MARKING OF A SINGLE STAR WHOSE NUMBER IS IN R1, USING
AUTO OPTICS AND MANUAL OPTICS.
- 4 CSM MARKING THE LEM.
- 5 NULL OUT TERMINAL VG.
VG-X, VG-Y, VG-Z ARE IN R1, R2, R3 RESPECTIVELY.
- 6 HOLD CARDS.
- 8 CSM MARKING OF A STAR USING MANUAL OPTICS ONLY.
- 10 CSM MANEUVER TO BRING 2 STARS INTO THE FIELD OF VIEW
(THE MANEUVER SECTION OF SUBPROGRAM 2).
- 20 KEY IN LANDMARK DATA AFTER MARKING.
- 30 TRACK LANDMARK - USE SEXTANT, MANUAL OPTICS, NO MANEUVER.
- 31 TRACK LANDMARK - USE SEXTANT, MANUAL OPTICS.
ASTRONAUT MANEUVERS AT START SO OPTICS POINTS AT EARTH.
- 32 TRACK LANDMARK - USE SEXTANT, AUTO OPTICS, NO MANEUVER.

- 33 TRACK LANDMARK - USE SEXTANT, AUTO OPTICS, MANEUVER.
- 33 TRACK LANDMARK - USE SCANNING TELESCOPE, MANUAL OPTICS,
NO MANEUVER.
- 35 TRACK LANDMARK - USE SCANNING TELESCOPE, MANUAL OPTICS,
MANEUVER.
- 36 TRACK LANDMARK - USE SCANNING TELESCOPE, AUTO OPTICS,
NO MANEUVER.
- 37 TRACK LANDMARK - USE SCANNING TELESCOPE, AUTO OPTICS,
MANEUVER.
- 40 MANEUVER LEM TO SEE 2 STARS.
THE ASTRONAUT FIRST MAKES CALLS TO UNIVERSE TO FIND THE
CLOSEST TWO STARS, THEN MANEUVERS LEM TO SEE THE STARS.
- 50-57 MARK PREVIOUS LANDMARK (SAME OPTINS CORRESPONDING TO 30-37).
- 60 LUNAR SURFACE STAR SIGHTING IN FRONT DETENT.
- 61 LUNAR SURFACE STAR SIGHTING IN REAR DETENT.
- 62 SAME AS 60 BUT MARK THE SAME STAR AGAIN.
- 63 LUNAR SURFACE - CALIBRATE FOR REAR DETENTS.
- 64 KEY IN CURSOR ANGLE.
- 65 KEY IN SPIRAL ANGLE.
- 67 LEM ORBITAL BACKUP MANEUVER TO SEE 2 STARS WITH THE
FOLLOWING FORMAT:
- SUBPROGRAM 67 STARNO = A
 PLANETV = B, C, D
 LOSAS = E, F HOLD CARDS
- WHERE A IS THE STAR NUMBER.
B, C, D ARE THE X, Y, Z COMPONENTS OF
THE PLANET VECTOR.
E, F ARE THE AZIMUTH AND ELEVATION OF THE
VIEW VECTOR.
- 68 LEM MARK STAR, BACKUP,
WITH THE FOLLOWING FORMAT:
- SUBPROGRAM 68 NOMARKS = G
 STARNO = A
 PLANETV = B, C, D
 LOSAS = E, F HOLD CARDS.
- WHERE G IS THE NUMRER OF MARKS TO BE TAKEN AND
A THROUGH F ARE THE SAME AS 67.

9X LEM STAR MARKING (X = NUMBER OF MARKS).
1XY LEM COARSE ALIGNMENT - MANEUVER AND TWO STAR MARKS.
X = NUMBER OF MARKS FOR THE FIRST STAR.
Y = NUMBER OF MARKS FOR THE SECOND STAR.

APPENDIX A

ASTRONAUT INPUT CARDS ARE NOT READ DIRECTLY BY THE ASTRONAUT PROGRAM. AN INPUT ROUTINE, ASTROINPUTPROG, IS USED TO READ, TRANSLATE, AND STORE THE CODED STATEMENTS IN THE DATA FILE. THE ASTRONAUT PROGRAM THEN TAKES ITS INPUT FROM THESE CODED STATEMENTS IN THE DATA FILE.

THE ORDINARY USER NEED ONLY CONCERN HIMSELF WITH THE INPUT CARDS THAT ARE READ BY ASTROINPUTPROG. FOR OTHERS WHO FIND IT USEFUL TO MAKE CHANGES DIRECTLY TO THE DATA FILE, THE FILED CODING FOR INPUT STATEMENTS ARE DOCUMENTED BELOW.

SINCE IT IS NOW POSSIBLE TO REGENERATE THE ASTRONAUT FILE ON ROLLBACKS, THERE SHOULD BE LESS NEED FOR USERS TO CHANGE CODES IN THE DATA FILE.

CONDITION CLAUSES

BEGIN NEW STATEMENT	8 888 888 888
TIME	9 900 000 000
WAIT (NEW STATEMENT)	8 800 000 000
IF	9 910 000 000
IF WITHIN X	9 910 000 002
IF NOABORT	9 910 000 004
IF WITHIN X NOABORT	9 910 000 006

DISPLAY CONDITIONS

EACH THREE WORD CONDITION IN THE INPUT STATEMENT IS CODED AND STORED IN THE DATA FILE AS ONE WORD. EACH SUCH WORD HAS THE DECIMAL POINT AT THE RIGHT END. DIGIT 1 (HIGH ORDER) SPECIFIES THE DISPLAY (0 = FLASH, 1 = MODE, 2 = VERB, 3 = NOUN, 4 = R1, 5 = R2, 6 = R3, 7 = CHECK FAIL, 8 = PROGRAM ALARM, 9 = RELEASE DISPLAY). DIGIT 4 SPECIFIES THE TYPE OF COMPARE (0 FOR EQ, 1 FOR UNEQ, 2 FOR L, 3 FOR LEQ, 4 FOR G, AND 5 FOR LEQ). DIGIT 5 SPECIFIES THE SIGN FOR THE R REGISTERS (0 = BLANK, 1 = +, 2 = -). THE R REGISTER VALUES ARE 5 DIGITS IN POSITIONS 6 THROUGH 10. THE MODE, VERB, AND NOUN VALUES ARE TWO DIGITS IN POSITIONS 9 AND 10. THE FLASH, CHECK FAIL, PROGRAM ALARM, AND RELEASE DISPLAY VALUES ARE 0 OR 1 IN DIGIT 10 IF THEY ARE TO BE OFF OR ON, RESPECTIVELY.

ACTION CLAUSES

BEGIN ACTION CLAUSE	7 777 777 777
WAIT (WITHIN ACTION CLAUSE)	7 000 000 000
VERIFY	9 920 000 000
VERIFY WITHIN X	9 920 000 002
VERIFY EVERY Y WITHIN X	9 920 000 003
VERIFY NOABORT	9 920 000 004
VERIFY WITHIN X NOABORT	9 920 000 006
VERIFY EVERY Y WITHIN X NOABORT	9 920 000 007
OTHERWISE	9 000 1XX XXX
END	9 000 0XX XXX
(WHERE XXXXX SPECIFIES THE NUMBER OF END WORDS YET TO FOLLOW.)	
SUBPROGRAM	9 000 2YY YYY
(WHERE YYYYY IS THE SUBPROGRAM NUMBER.)	

KEYSTROKES

MAIN KEYSTROKES FOLLOW	6 000 500 000
NAV KEYSTROKES FOLLOW	6 000 500 001
UPTTEL FOLLOWS	6 000 500 002

KEYSTROKE WORDS ARE FILED IN THE FOLLOWING FORMAT.

K B A 0 S D D D D D

K SPECIFIES THE NUMBER OF DIGIT KEYS TO BE STRUCK (0 TO 5).
 B SPECIFIES AN ALPHA KEY TO BE STRUCK BEFORE THE DIGIT KEYS. A SPECIFIES AN ALPHA KEY TO BE STRUCK AFTER THE DIGIT KEYS. (CODING FOR B AND A IS -- 0 = NO KEY, 1 = VERB, 2 = ERROR RESET, 3 = KEY RELEASE, 6 = ENTER, 7 = CLEAR, AND 8 = NOUN.) S SPECIFIES THE SIGN KEY, IF ANY, TO PRECEDE THE DIGITS (0 = NO SIGN, 1 = +, AND 2 = -). DDDDD SPECIFIES FROM 0 TO 5 SUCCESSIVE DIGIT KEYSTROKES, STARTING WITH THE LEFTMOST.

THUS THE WORD

3 0 6 0 2 0 5 7 0 0

DENOTES THE 5 KEYS -, 0, 5, 7, ENTER

AND

0 0 0 0 1 0 0 0 0 0

THE SINGLE KEY +.

SWITCHES, PUSHBUTTONS, AND CONTROLLERS

KEY WORD -----	MODIFIERS -----	FILED CODE -----
MARK		6 000 000 000
MARKREJ		6 000 100 000
MARKX		6 000 200 000
MARKY		6 000 300 000
TRNSW	AGC	6 000 600 001
	MANUAL	6 000 600 002
UPTELSW	ACCEPT	6 000 600 011
	BLOCK	6 000 600 012
IMUMODE	ZERO ENCODERS	6 000 600 021
	FINE ALIGN	6 000 600 022
	COARSE ALIGN	6 000 600 023
	LOCK CDUS	6 000 600 024
	ATTITUDE CONTROL	6 000 600 025
	ENTRY	6 000 600 026
IMUPOWER	OFF	6 000 600 027
	ON	6 000 600 028
OPTMODE	MANUAL	6 000 600 031
	AGC	6 000 600 032
	ZERO ENCODERS	6 000 600 033
G&NSYNCH	ON	6 000 600 041
	OFF	6 000 600 042
SCSMODE	G&N ATTITUDE CONTROL	6 000 600 051
	G&N DV	6 000 600 052
	G&N ENTRY	6 000 600 053
	SCS ATTITUDE CONTROL	6 000 600 054
	SCS DV	6 000 600 055
	SCS ENTRY	6 000 600 056
	OFF	6 000 600 057
	FREE	(SAME AS OFF)
	ATTITUDE HOLD	6 000 600 058
	AUTO	6 000 600 059
SCS05GSW	ON	6 000 600 061
	OFF	6 000 600 062
CMSMSEP		6 000 600 063
S4BSEP		6 000 600 064
GUIDEMODE	BACKUP	6 000 600 070
	PRIMARY	6 000 600 071

HANDCTRL	NEUTRAL	6 000 610 000
	NO PITCH	6 000 610 XX0
	+ PITCH	6 000 610 XX1
	- PITCH	6 000 610 XX2
	NO ROLL	6 000 610 X0X
	+ ROLL	6 000 610 X1X
	- ROLL	6 000 610 X2X
	NO YAW	6 000 610 0XX
	+ YAW	6 000 610 1XX
	- YAW	6 000 610 2XX
MINIMPLS	(SAME AS HANDCTRL)	6 000 670 XXX
TRANSCTL	NEUTRAL	6 000 620 000
	NO X	6 000 62X XX0
	+ X	6 000 62X XX1
	- X	6 000 62X XX2
	NO Y	6 000 62X X0X
	+ Y	6 000 62X X1X
	- Y	6 000 62X X2X
	NO Z	6 000 62X 0XX
	+ Z	6 000 62X 1XX
	- Z	6 000 62X 2XX
	ENGINE OFF	6 000 621 XXX
	ABORT	6 000 622 XXX
LEMHCTRL	P	6 000 650 001
	Q	6 000 650 002
	R	6 000 650 003
LRHCRATE		6 000 650 004
ABORTSTG		6 000 660 020
ASENGARM	ON	6 000 660 011
	OFF	6 000 660 010
DCENGARM	ON	6 000 660 001
	OFF	6 000 660 000
LEMTHROT	XX	6 000 660 030
LEMRA DAR	OFF	6 000 660 040
	ON	6 000 660 041
LVC RCS	BOTH	6 000 660 050
	A	6 000 660 051
	B	6 000 660 052
THRUSTSW	MANUAL	6 000 660 060
	AUTO	6 000 660 061
IIJ		6 000 660 100
CMC		6 000 660 101

PROCEED		6 000 660 136
CSMRATE	0	6 000 660 140
	1	6 000 660 141
	2	6 000 660 142
	3	6 000 660 143
DEPLOY LANDING GEAR		6 000 660 152

3.G.17

INPUT CARDS FOR LMP1

MAINTAINED BY KEITH GLICK

A VARIABLE NUMBER OF CARDS ARE READ. EACH MUST CONTAIN THE LMP PROGRAM IDENTIFICATION NUMBER (10) IN ITS FIRST WORD. THE REMAINING WORDS ARE DESCRIBED BELOW BY CARD AND WORD NUMBER. FOR EXAMPLE (1,2) MEANS CARD 1, FIELD 2.

WORD (1,2) THIS IS THE NUMBER OF LMP INPUT CARDS FOLLOWING THIS CARD (TOTAL NUMBER OF LMP CARDS - 1).

EACH ADDITIONAL CARD REPRESENTS A COMMAND TO THE LMP. THIS ALLOWS THE LMP TO REACT TO COMMANDS WHICH ARE NORMALLY RECEIVED VIA UPLINK OR THE ABORT GUIDANCE SYSTEM. HOWEVER ANY LEGITIMATE LMP COMMAND CAN BE ENTERED IN THIS WAY.

WORD (J,2) THIS WORD IS THE TIME IN SECONDS WHEN THE COMMAND IS RECEIVED BY THE LMP.

WORD (J,3) THIS IS A WORD IDENTIFYING THE COMMAND SPECIFIED BY THIS CARD. THE WORD IS FORMED ACCORDING TO THE FOLLOWING ALGORITHM.

$$\text{COMMAND WORD} = 10 + .001(\text{xxx})$$

WHERE xxx IS THE DECIMAL COMMAND. IN ADDITION TO THE NORMAL AGC INITIATED COMMANDS THE FOLLOWING COMMANDS MAY BE USED.

298	SWITCH GUIDANCE MODE FROM PRIMARY TO ABORT GUIDANCE SYSTEM.
299	SWITCH GUIDANCE MODE FROM ABORT GUIDANCE SYSTEM TO PRIMARY GUIDANCE SYSTEM.
300	ENGINE ON DISCRETE ON.
301	ENGINE ON DISCRETE OFF.
302	ENGINE OFF DISCRETE ON.
303	ENGINE OFF DISCRETE OFF.

3.G.18

INPUT CARDS FOR DISCONT

MAINTAINED BY PIETER MIMNO

THE DISCONTINUITY ROUTINE IS AN ENVIRONMENT PROGRAM WHICH ENABLES SIMULATION USERS TO CHANGE THE SETTING OF INDIVIDUAL BITS IN INPUT CHANNELS 15, 16, 30, 31, 32, AND 33. BIT CHANGES ARE SPECIFIED, ONE PER CARD, ON MAC FORMAT DATA CARDS. A SET OF THESE CARDS CONSTITUTE THE SEQUENCE OF INPUT CHANNEL DISCONTINUITIES TO BE PROCESSED DURING THE SIMULATION.

THE MAXIMUM NUMBER OF DISCONTINUITY CARDS WHICH MAY BE SUBMITTED IN ONE RUN IS 49.

THE FORMAT OF THE DISCONT INPUT CARDS IS AS FOLLOWS:

FIELD 1	FIELD 2	FIELD 3	FIELD 4	FIELD 5	FIELD 6
-----	-----	-----	-----	-----	-----
ID	CHANNEL	BIT	TON	TOFF	UNUSED

WHERE:

ID IDENTIFICATION NUMBER OF DISCONT. MUST HAVE THE VALUE "4".

CHANNEL CHANNEL NUMBER TO BE MODIFIED (LEGAL SETTING 15, 16, 30, 31, 32, 33).

BIT BIT NUMBER WITHIN CHANNEL TO BE MODIFIED (LEGAL SETTING 1 - 15).
THE BIT NUMBER IS MEASURED FROM RIGHT TO LEFT WITHIN A CHANNEL.

TON TIME IN SECONDS AT WHICH THE SPECIFIED BIT IS SET TO 1.

TOFF TIME IN SECONDS AT WHICH THE SPECIFIED BIT IS SET TO 0.

UNUSED SET TO 0.

NOTE: THE INVERSION OF CHANNELS 30 - 33 IS ASSUMED TO TAKE PLACE WITHIN THE AGC, WITH THE FOLLOWING RESULT

CHANNEL 15-16 TURNING A BIT ON WITH THE DISCONT ROUTINE CAUSES A 1 TO BE SET INTO THE AGC
INPUT REGISTER.

CHANNEL 30-33 TURNING A BIT ON WITH THE DISCONT ROUTINE CAUSES A 0 TO BE SET INTO THE AGC
INPUT REGISTER.

IT IS NOT NECESSARY TO FILL IN BOTH A TON AND A TOFF. FOR A SINGLE BIT CHANGE, THE UNUTILIZED FIELD (TON OR TOFF) SHOULD BE SET TO 0.

3.G.19

INPUT CARDS FOR SIMSETUP

MAINTAINED BY DAN LICKLY OR JAY SAMPSON

THIS ROUTINE IS DESIGNED TO AID IN GENERATING INPUT CARDS FOR BOTH THE AGC AND THE ENVIRONMENT, FOR SIMULATION RUNS. THE PROGRAM CONSISTS OF TWO MAJOR SECTIONS. THE FIRST IS AN ANGULAR ORIENTATION ROUTINE, WHICH IS EXECUTED ONCE PER CASE. THIS SECTION READS THREE DATA CARDS, WHICH DESCRIBE BOTH THE ORIENTATION OF THE STABLE MEMBER FRAME RELATIVE TO THE FRAME USED BY THE UNIVERSE PROGRAM, AND THE VEHICLE ORIENTATION RELATIVE TO THE LOCAL HORIZONTAL FRAME. THE FIRST CARD CONTAINS AN ATTITUDE CODE, AMODE, WHICH DETERMINES THE FORMAT OF THE NEXT TWO CARDS, AS FOLLOWS:

AMODE=0. STABLE MEMBER FRAME SPECIFIED BY EULER ANGLES, VEHICLE ORIENTATION SPECIFIED BY PITCH, YAW, AND ROLL FROM LOCAL HORIZONTAL.

CARD 2. THIS CARD CONTAINS TIME, IN SECONDS, FROM UNIVERSE'S EPOCH, TWO BLANKS, AND THE EULER ANGLES FOR STABLE MEMBER ORIENTATION, IN DEGREES. THESE ARE Y-, Z-, AND X ROTATIONS, IN THAT ORDER.

CARD 3. THIS CARD, FOR VEHICLE ORIENTATION, SPECIFIES PITCH, YAW, AND ROLL ANGLES WITH RESPECT TO LOCAL HORIZONTAL.

AMODE=1. STABLE MEMBER FRAME SPECIFIED BY EULER ANGLES, VEHICLE ORIENTATION SPECIFIED BY GIMBAL ANGLES.

CARD 2. THIS CARD CONTAINS TIME, IN SECONDS, FROM UNIVERSE'S EPOCH, TWO BLANKS, AND THE EULER ANGLES FOR STABLE MEMBER ORIENTATION, IN DEGREES. THESE ARE Y-, Z-, AND X ROTATIONS, IN THAT ORDER.

CARD 3. THIS CARD, FOR VEHICLE ORIENTATION, SPECIFIES OUTER, INNER, AND MIDDLE GIMBAL ANGLES.

AMODE=2. STABLE MEMBER FRAME SPECIFIED BY LAUNCH POINT DATA, VEHICLE ORIENTATION SPECIFIED BY PITCH, YAW, AND ROLL FROM LOCAL HORIZONTAL.

CARD 2. THIS CARD CONTAINS LATITUDE AND LONGITUDE OF THE LAUNCH POINT, IN DEGREES, THE ALTITUDE, IN METERS, THE AZIMUTH FROM NORTH TO THE X-AXIS, AND THE TIME FROM EPOCH OF GUIDANCE REFERENCE RELEASE.

CARD 3. THIS CARD, FOR VEHICLE ORIENTATION, SPECIFIES PITCH, YAW, AND ROLL ANGLES WITH RESPECT TO LOCAL HORIZONTAL.

AMODE=3. STABLE MEMBER FRAME SPECIFIED BY LAUNCH POINT DATA, VEHICLE ORIENTATION SPECIFIED BY GIMBAL ANGLES.

CARD 2. THIS CARD CONTAINS LATITUDE AND LONGITUDE OF THE LAUNCH POINT, IN DEGREES, THE ALTITUDE, IN METERS, THE AZIMUTH FROM NORTH TO THE X-AXIS, AND THE TIME FROM EPOCH OF GUIDANCE REFERENCE RELEASE.

CARD 3. THIS CARD, FOR VEHICLE ORIENTATION, SPECIFIES OUTER, INNER, AND MIDDLE GIMBAL ANGLES.

AMODE=4. STABLE MEMBER FRAME SPECIFIED BY AZIMUTH AND ELEVATION, VEHICLE ORIENTATION SPECIFIED BY PITCH, YAW,
 ----- AND ROLL FROM LOCAL HORIZONTAL.

CARD 2. THIS CARD CONTAINS LATITUDE AND LONGITUDE, IN DEGREES, THE ALTITUDE, IN METERS, THE AZIMUTH FROM
 NORTH TO THE X-AXIS, THE ELEVATION IN DEGREES FROM THE X-AXIS IN THE XZ-PLANE, AND THE TIME FROM
 EPOCH, IN SECONDS.

CARD 3. THIS CARD, FOR VEHICLE ORIENTATION, SPECIFIES PITCH, YAW, AND ROLL ANGLES WITH RESPECT TO LOCAL
 HORIZONTAL.

AMODE=5. STABLE MEMBER FRAME SPECIFIED BY AZIMUTH AND ELEVATION, VEHICLE ORIENTATION SPECIFIED BY GIMBAL
 ----- ANGLES.

CARD 2. THIS CARD CONTAINS LATITUDE AND LONGITUDE, IN DEGREES, THE ALTITUDE, IN METERS, THE AZIMUTH FROM
 NORTH TO THE X-AXIS, THE ELEVATION IN DEGREES FROM THE X-AXIS IN THE XZ-PLANE, AND THE TIME FROM
 EPOCH, IN SECONDS.

CARD 3. THIS CARD, FOR VEHICLE ORIENTATION, SPECIFIES OUTER, INNER, AND MIDDLE GIMBAL ANGLES.

THE AMODE CARD MUST BE PRECEDED BY A CARD WHOSE ONLY ENTRY IS THE FLIGHT NUMBER. VALID FLIGHT NUMBERS ARE:

205, 206, 501, 503, 504.

THE LOCAL HORIZONTAL SYSTEM TO WHICH PITCH, YAW, AND ROLL ARE REFERRED IS DEFINED AS FOLLOWS:

$$\begin{aligned} \text{UNITZ} &= - \text{UNIT (R)} \\ \text{UNITY} &= \text{UNIT (V * R)} \\ \text{UNITX} &= \text{UNIT (UNITY * UNITZ)} \end{aligned}$$

THUS X LIES IN THE LOCAL VERTICAL PLANE.
 Y IS PERPENDICULAR TO THE R, V PLANE (MOMENTUM PLANE)
 Z POINTS DOWN TO THE CENTER OF THE EARTH

THE FIRST SECTION OF THE PROGRAM DOES NOTHING BUT READ THE INPUT CARDS AND PRINT THEM.

THE SECOND MAJOR PROGRAM SECTION MAY BE USED TO GENERATE ENVIRONMENT & AGC INPUT CARDS FROM A VARIETY OF POSITION AND VELOCITY INFORMATION. THIS SECTION OF THE PROGRAM MAY BE EXECUTED AS MANY TIMES AS DESIRED PER ORIENTATION OF THE PREFERRED COORDINATE SYSTEM. THE FIRST EXECUTION GENERATES CARDS TAGGED RN (& VN) FOR AGC INPUT. SUBSEQUENT EXECUTIONS GENERATE UNTAGGED CARDS. A CARD CONTAINING 9999.999999 IN ITS FIRST WORD (WORDS 2 - 6 MUST BE VALID DATA, LIKE ZEROS) CAUSES CONTROL TO BE PASSED BACK TO SECTION 1 TO READ ANOTHER SET OF ORIENTATION-DATA CARDS.

EACH POSITION-CONVERSION REQUIRES AT LEAST ONE CARD. THE FIRST WORD OF THIS CARD IS THE MODE-CONTROL WORD, SPECIFYING POSITION & VELOCITY MODES IN THE FORM RMODE.VMODE (SEE BELOW FOR DESCRIPTIONS OF THE MODES). THE SECOND WORD IS NORMALLY TIME, MEASURED IN SECONDS, FROM THE EPOCH SPECIFIED ON THE UNIVERSE INPUT CARDS (SEE BELOW). THE THIRD WORD IS THE BINARY SCALE FACTOR (SIGN IGNORED). WORDS 4, 5, & 6 ARE INTERPRETED AS DESCRIBED BELOW. THE USE OF VELOCITY-INPUT AND SUBSIDIARY CARDS IS ALSO EXPLAINED BELOW. IN ALL CASES WHERE VELOCITY-INPUT CARDS ARE READ BY THIS PROGRAM, WORDS 1 & 2 ARE IGNORED, AND WORD 3 IS THE BINARY SCALE FACTOR. THE UNSCALED VELOCITY OUTPUT UNITS ARE METERS PER CENTISECOND.

AFTER THE FIRST SET OF R AND V CARDS HAVE BEEN READ, THE UNIVERSE R AND V (UR, UV,) ARE AVAILABLE. THE R AND V ARE THEN USED TO GENERATE THE LOCAL HORIZONTAL REFERENCE SYSTEM TO WHICH THE VEHICLE PITCH, YAW, AND ROLL ANGLES WILL BE REFERRED. FINALLY THE EULER ANGLES, GIMBAL ANGLES (ASSUMED TO BE EQUAL TO CDU ANGLES), AND PITCH, YAW, ROLL ARE CALCULATED AND PRINTED OUT.

FOLLOWING THIS ANGLE PRINTOUT, REFSMMAT AND CDU ANGLES ARE PRINTED IN AGC SPECIAL REQUEST FORMAT.

RMODE = 1. WORDS 4, 5, & 6 ARE INTERPRETED AS A POSITION VECTOR (MKS) IN UNIVERSE COORDINATES.

VMODE = 0. NO VELOCITY REQUESTED.

VMODE = 1. WORDS 4, 5, & 6 OF THE NEXT CARD ARE INTERPRETED AS MKS VELOCITY IN UNIVERSE COORDINATES.

RMODE = 2. WORDS 4, 5, & 6 ARE GEOGRAPHIC LATITUDE (FISCHER ELLIPSOID), LONGITUDE (BOTH IN DEGREES), AND ALTITUDE (METERS, MEASURED ALONG THE RADIUS VECTOR).

VMODE = 0. NO VELOCITY REQUESTED.

VMODE = 1. VELOCITY IS THAT DUE TO EARTH RATE. SCALE FACTOR = 7.

VMODE = 2. WORDS 4, 5, & 6 OF THE NEXT CARD ARE INTERPRETED AS FLIGHT-PATH ANGLE, AZIMUTH (BOTH IN DEGREES), AND VELOCITY MAGNITUDE (METERS/SECOND).

RMODE = 3. WORDS 4, 5, & 6 ARE INTERPRETED AS A POSITION VECTOR (MKS) IN THE COORDINATE FRAME SPECIFIED BY THE NEXT CARD. WORDS 1 & 2 OF THE NEXT CARD ARE IGNORED. THE REMAINING WORDS DETERMINE THE (GEOCENTRIC) COORDINATE FRAME OF THE INPUT AS FOLLOWS: X IS UPWARD, PARALLEL TO THE LOCAL VERTICAL AT THE GEOGRAPHIC LATITUDE & LONGITUDE (DEGREES) SPECIFIED BY WORDS 4 & 5, AT THE TIME (SECONDS FROM UNIVERSE EPOCH) OF WORD 3. Z IS ROTATED ABOUT -X FROM NORTH BY THE AMOUNT SPECIFIED (DEGREES) BY WORD 6.

VMODE = 0. NO VELOCITY REQUESTED.

VMODE = 1. VELOCITY IS THAT DUE TO EARTH RATE. SCALE FACTOR = 7.

VMODE = 2. WORDS 4, 5, & 6 OF THE THIRD CARD ARE INTERPRETED AS A VELOCITY VECTOR (MKS) IN THE LOCAL VERTICAL FRAME ABOVE.

THE DECK REQUIRED TO RUN THIS PROGRAM SHOULD CONTAIN THE NORMAL IBM JOB CONTROL CARDS FOR A MAC RUN, FOLLOWED BY * COLLECT SIMSETUP , * LOAD FILE UNIV.EPHM6869 (OR APPROPRIATE YEAR), AND * GO . THE FIRST THREE CARDS FOLLOWING THE * GO CARD ARE READ BY THE UNIVERSE PROGRAM. ALL THREE UNIVERSE CARDS MUST HAVE "11" IN THEIR FIRST MAC DATA CARD FIELD (ID NUMBER OF UNIVERSE IN SIMULATIONS). FIELD 2 OF CARD 1 IS EPOCH --- THIS IS MEASURED IN DAYS FROM JULY 1 OF THE EPHEMERIS YEAR. WORDS 3 - 6 OF CARD 1 MUST BE FILLED IN, BUT ARE IGNORED. THE SAME IS TRUE OF WORDS 2 - 6 OF CARD 2 (REGRETTABLE, BUT THAT'S THE WAY IT IS). WORD 2 OF CARD 3 MUST CONTAIN 2 (CONFIGURATION). WORD 3 MUST BE FILLED IN. WORDS 4 - 6 ARE NOT READ.

THE NEXT THREE CARDS ARE THE THREE ORIENTATION DATA CARDS FOR SECTION 1. THESE ARE FOLLOWED BY CARDS AS NECESSARY FOR POSITION & VELOCITY, TERMINATED BY A 9999.999999 CARD (FOLLOWED BY THREE MORE ORIENTATION CARDS AND POSITION AND VELOCITY CARDS, ETC., IF A SUBSEQUENT CASE IS TO BE RUN OR /* (TERMINATOR CARD) IF NOT.

3.G.20

ENVIRONMENT ONLY INPUT CARDS

MAINTAINED BY PIETER MIMNO

ENVIRONMENT ONLY RUNS MAY BE USED TO TEST THE DYNAMICAL RESPONSE OF THE ENVIRONMENT PROGRAMS TO SPECIFIC AGC COMMANDS. SEE SECTION 3.G FOR A DISCUSSION OF THE FUNCTION OF ENVIRONMENT ONLY RUNS AND A DESCRIPTION OF THE INPUT DECK. AGC COMMANDS ARE READ BY THE ENVIRONMENT FROM INPUT CARDS, ONE AGC COMMAND PER CARD, WITH THE FOLLOWING FORMAT:

FIELD 1	FIELD 2	FIELD 3	FIELD 4	FIELD 5	FIELD 6
-----	-----	-----	-----	-----	-----
ID	IDWD	TIME	MISC	DTCYC	TENDCYC

WHERE:

ID IDENTIFICATION NUMBER FOR ENVIRONMENT ONLY CARDS. MUST HAVE THE VALUE "3".

IDWD CODE DEFINING THE AGC CHANNEL AND BIT NUMBER WHICH HAS CHANGED OR AN ENVIRONMENT UPDATE CMMD.

TIME AGC TIME AT WHICH BIT HAS CHANGED STATE.

MISC DATA DESCRIBING NEW STATE OF BIT.
0 OR 1 FOR DISCRETE CHANGES.
+ OR - LENGTH OR 0 FOR PULSE TRAINS.

DTCYC REPETITION RATE FOR FORCED STOP TIME UPDATES (ENVIRONMENT UPDATES FORCED INTERNALLY BY ENVCNTRL.
IF DTCYC = 0, ENVIRONMENT UPDATES WILL NOT BE FORCED.

TENDCYC TIME AT WHICH FORCED INTERNAL UPDATES ARE TERMINATED.
NOT MEANINGFUL UNLESS DTCYC IS NON ZERO.

THE MAXIMUM NUMBER OF ENVIRONMENT INPUT CARDS WHICH MAY BE SUBMITTED IN ONE RUN IS 129.

LEGAL VALUES OF IDWD AND MISC SPECIFYING CHANGES IN A PARTICULAR AGC OUTPUT CHANNEL (DIGITS 3 AND 4 OF IDWD) AND BIT (DIGITS 1 AND 2 OF IDWD) ARE THE FOLLOWING:

CHANNEL -----	BIT ---	IDWD CODE -----	MISC CODE -----	DESCRIPTION -----
5	1-8	501- 508	1 = ON 0 = OFF	RCS JETS.
6	1-8	601- 608	1 = ON 0 = OFF	RCS JETS.
7				NOT USED BY ENVIRONMENT.
10				NOT USED BY ENVIRONMENT.
11	1-15	1101-1115	1 = ON 0 = OFF	FLASH AND ENGINE ON-OFF DISCRETES.
12	1-15	1201-1215	1 = ON 0 = OFF	MODING DISCRETES.
13	1-4	1304	0 - 7	RADAR ACTIVITY. MISC CONTAINS THE SETTING OF CHANNEL 13, BITS 1-3.
13	5-15	1305-1315	1 = ON 0 = OFF	DISCRETES.
14	1-3	1401-1403	1 = ON 0 = OFF	DISCRETES.
14	4	1404	+,- LENGTH 0 = OFF	LM THROTTLE PULSE TRAIN.
14	5	1405	+,- LENGTH 0 = OFF	ENTRY DRIVE (SPARE).
14	6-10	1471-1473	+,- LENGTH 0 = OFF	GYRO PULSE TRAINS TO X, Y, Z.
14	11-12	1411-1412	+,- LENGTH 0 = OFF	OPTICS PULSE TRAINS TO SHAFT AND TRUNNION.
14	13-15	1413-1415	+,- LENGTH 0 = OFF	CDU PULSE TRAINS TO Z, Y, X.

MANY OF THE ABOVE DISCRETES ARE READ BY THE ENVIRONMENT, BUT ARE NOT ACTED UPON. CHECK WITH MEMBERS OF THE DIGITAL SIMULATION GROUP TO DETERMINE WHICH CHANNEL AND BIT SETTINGS ARE REACTED TO BY THE ENVIRONMENT.

IN ADDITION TO THE ABOVE CHANNEL AND BIT CHANGE COMMANDS, THE ENVIRONMENT MAY BE UPDATED TO A SPECIFIC TIME BY SETTING:

IDWD = 100
TIME = ENVIRONMENT UPDATE TIME.
MISC = REASON FOR UPDATE.

WHERE:

MISC = 0	REASON FOR UPDATE UNSPECIFIED.
1-3	READ CDUX, Y, Z.
4-5	READ OPTICS T, S.
6-8	READ PIPA X, Y, Z.
9-11	READ HANDCONTROLLER Q, P, R.
12	OUTPUT UPDATE (ENVSUM, CFDUMP)
13	SET COUNTER.
14	CHANNEL 12 DELAYED DISCRETE.
15	PULSE TRAIN SEGMENTATION.
16	OUTPUT CHANNEL DISCRETE.
17	ASTRONAUT CALL.
18	FINAL CALL.
19	RADAR UPDATE.
20	EXIT UPDATE.
21	ABORT UPDATE.

THE USE OF A CARD WITH FORMAT:

3, 100, TIME, 0

WILL CAUSE THE ENVIRONMENT TO UPDATE ITSELF TO T = TIME. THE LAST CARD IN A SET OF ENVIRONMENT ONLY INPUT CARDS IS NORMALLY OF THIS FORM, IN ORDER TO BRING THE ENVIRONMENT UP TO THE DESIRED TERMINATION TIME.

IF IT IS DESIRED TO FORCE THE ENVIRONMENT TO UPDATE ITSELF AT FIXED INCREMENTS OF TIME, THIS MAY BE ACCOMPLISHED BY THE USE OF A SINGLE CARD OF THE FORM:

3, 100, TIME, 0, DTCYC, TENDCYC

THIS COMMAND WILL CAUSE ENVCTRL TO FORCE ENVIRONMENT UPDATES EVERY DTCYC SECONDS, STARTING AT T = TIME AND STOPPING AT T = TENDCYC. FORCED UPDATES OF THE ENVIRONMENT MAY BE USED TO SIMULATE THE CYCLICAL UPDATES THAT WOULD NORMALLY BE CAUSED IN A FLIGHT TEST SIMULATION BY AGC COUNTER READS.

CHAPTER 4

User's Guide to Rollbacks

4. A. Introduction to Rollback Concept

Experience in AGC simulation has shown that some simulations run from 10 minutes to 5 hours on the IBM 360. If the simulation stops for any reason more than 5 minutes after simulation actually began it is increasingly economical to restart (rollback) the simulation at some point prior to where the trouble, if any, developed, or if there is no trouble, to restart the simulation where it ended. Sometimes a user may desire to restart the simulation at the exact end point and cause it to die immediately, thus having restored the entire data file for use by a subsequent MAC run step within the same IBM Job Step.

To give a simulation this rollback (restart) capability, the initial simulation must be a MAC controlled simulation, have " * SNAPSHOT " as the first MAC run step control card and must include the " HOLD MARSROT " AGC instruction simulator director card. During initialization, the MAC monitor, as well as all MAC programs collected, the initial content of the data file, and most non-MAC programs to be used in the simulation are written on the MARSROT.

A snapshot consists of writing onto the MARSROT all of the data file changes since the previous snapshot plus all the variable areas of MAC and non-MAC programs. A snapshot is taken at the beginning of each simulation so enabled, and is then repeated either at intervals or at chosen times or at AGC locations specified by SNAP special requests. Finally a snapshot is taken at the very end of the simulation (before AGC initiated stop/aborts but after environment initiated stop/aborts).

4. B. Restarting a Simulation via a Rollback

A rollback is a MAC Run Step. The MAC Run Step control cards in this case consist of one card

```
*   ROLLBACK   MARSROT   NNNNNNNN   TIME
```

where the * is in column 1, each field is separated by at least one blank, NNNNNNNN is the 8 digit MARSROT number and TIME is an integer expressed in seconds which is associated with a particular snapshot taken during a previous simulation using that MARSROT. Most pages printed in each simulation give, on the second line, the MARSROT number and the integer value of the latest snapshot.

The simulator announces a snapshot by printing

```
SNAPSHOT   X   SECONDS   RUNNING   X.Y
```

where X is the truncated integer simulation time and X.Y is the actual simulation time. A rollback to time X actually becomes a rollback to simulation time X.Y of the first or only snapshot taken in that one second interval.

The MAC Run Step data input cards for rollback consist of (1) special requests to the simulator followed by (2) input cards to environment programs. These are described in Sections 4. C. and 4. D. below.

A rollback MAC Run Step may be followed by more MAC Run Steps within the same IBM Job Step or the rollback may be an IBM Job Step with other IBM Job Steps following, if desired.

A simulation stopped by the AGC may be restarted from the last snapshot. If the run was terminated by MAXTIME or STOP or ABORT, these requests must be overridden (MAXTIME) or deleted (STOP and ABORT). A simulation stopped by the environment (i. e., one that has the statement "ENVIRONMENT STOPPING THIS RUN" printed before the last snapshot message is printed) must not be rolled back at that snapshot except when the purpose is only to reestablish the data file for a subsequent MAC Run Step.

4. C. AGC Instruction Simulator Special Requests for Rollback

Most AGC instruction simulator special requests (see Sections 3.C.3 and 3.C.4) are legal at rollback (illegal requests are BACKGRND, BKGRND, CHANL, START). These requests are added to the initial special requests and to the special requests for rollbacks made to snapshots of earlier time value than the current rollback. Class 1 requests override their previous settings. Class 2 requests are applied as at initialization. Class 3 requests can be cancelled by using the DELETE request. If two rollbacks are made to the same snapshot, the special requests submitted with the first rollback are considered to have never existed.

4. D. Environment Input Cards for Rollback

Currently, the only environment input that can be made at rollback is to replace the entire astronaut deck. The user must rollback to a snapshot at which the astronaut has not yet processed to the point where the new astronaut deck first differs from the old deck. To help in the decision, remember that the astronaut processes his input stream sequentially, that the astronaut does not process beyond WAIT X or TIME X until the time arrives, and that the astronaut does not process beyond the THEN word until an IF or VERIFY have been either satisfied or finally rejected.

For example, if the Astronaut is rolled back to a time at which it is actively waiting for an AGC return that was initiated by a specific TIME X or WAIT X in the input stream, then that TIME X or WAIT X is by definition processed, and a change in X may not be made.

CHAPTER 5

User's Guide to Edits

5. A. Introduction to Edit Capability

The edit program provides a capability to extract information selectively from MARSROT tapes generated during a digital simulation run. The results are presented to the user as printed output or as input for his MAC language programs which are a part of the edit run.

Requests can be made to the edit program to process information placed on the MARSROT tape as a result of three simulator special requests: DUMP, CFDUMP, and ITRACE (see Chapter 3, Section C: Special request cards).

Three options are available to process the relevant information:

- (1) dumps can be edited selecting certain registers, the contents of which are printed in any of several formats with user supplied scale factors;
- (2) information pertaining to DUMP and CFDUMP may be edited and placed in MAC's data file for processing by user supplied MAC language programs; and
- (3) ITRACE information can be edited and printed.

The editing process may be constrained within limits supplied as data by the user; in particular editing may be limited to one interval of time, or by means of conditionals applied to AGC data or addresses.

There are two distinct modes of operation for the edit program. The first is called AGCEDIT; the second MACEDIT. The AGCEDIT mode should be selected when no MAC program input is to be generated. MACEDIT mode must be used when user supplied MAC language programs are used. The MAC-EDIT mode may be used in place of AGCEDIT if it is desired to do the edit before or after another MAC step within the same IBM job step.

Requests to the edit program are made with a deck of request cards. The input card formats are generally identical to those used by the AGC simulator, except for the Editor Director card, which has an E in column 1, and is sentence read in columns 9-80. Edit Directors are chiefly used to introduce cards appropriate to one of the three edit options. Detail cards, identifying the dump to be processed, have simulator format; i. e.,

Cols. 1-7	Not used
Col. 8	Right print remarks card identifier
Cols. 9-24	Location Field
Cols. 25-32	Operation Code Field
Cols. 33-48	Data Address 1
Cols. 49-64	Data Address 2
Cols. 65-80	Remarks or MAC subroutine designator

The two address fields and the location field may generally contain unsigned octal or decimal integers, octal bank notation (XX, XXXX or EX, XXXX for E-Banks) or symbols with or without octal or decimal modifiers.

5. B. 1 Tape Labeling

```

1      9      80
↓      ↓      ↓
*      AGCEDIT  MARSROT  XXXXXXXXXX
      ←----- free form -----→

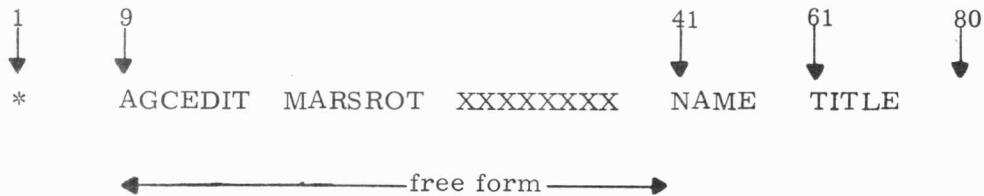
```

```

1      9      41      61      80
↓      ↓      ↓      ↓      ↓
*      AGCEDIT      NAME      TITLE
      ←free form→

```

5-3



The name-title method is the only way to identify a marsrot tape if the simulation and edit are run as part of the same IBM job. The simulator output contains the MARSROT file number on each page. This file number can then be used in subsequent edit runs. The file number is the only way to identify a particular tape if the same name and title were used in more than one simulation run, and the tapes were not released. If more than one tape exists with the same name-title identifier, and the edit program input specifies this name-title without a log number, the most recently generated tape will be selected.

5. B. 2 Tape Disposition

Often, it is desirable to edit only a small portion of data on a MARSROT immediately following a simulation, with the option of saving the tape so that more detailed information may be selected later. Also, the user may wish to revise a MAC subroutine to meet the special requirements of a certain run. For these and other reasons, the ability to save a MARSROT tape for repeated use has been provided.

Every edit deck should contain explicit tape disposition information by including one of the following edit directors:

- a) E SAVE MARSROT
- b) E HOLD MARSROT or
- c) E RELEASE MARSROT

SAVE MARSROT and HOLD MARSROT produce the same result. If no tape disposition director is included, the tape is saved.

5. B. 3 Facilities for Editing AGC Dumps

The edit directors

E EDIT AGC DUMPS

E EDIT LGC DUMPS

allow the user to select and edit any number of AGC dumps in the order in which they were generated. In general, simulator decks contain several DUMP requests. To select certain dumps, the user should supply a copy of the first or only DUMP card of each simulator DUMP request he wishes edited. With each DUMP card, he may supply detail cards which specify conditions under which the dump is to be edited, and formats and scale factors for selected registers in the area dumped. These are known as condition and subset cards, respectively. Cards for EDIT AGC DUMPS are arranged as follows:

E EDIT AGC DUMPS

Time card (optional)

A DUMP B C

Conditions for the above (optional)

Subset cards for the above (optional)

Repeat the above 3 items as necessary.

To select a particular dump, then, a copy of the first or only DUMP card of the DUMP request submitted to the simulator is required. The contents of columns 2-8 and 65-80 are irrelevant in this edit option, even if the simulator dump was conditional, or MORE was specified. The condition cards of the simulation deck must not be included in the input to the edit program.

Simulator conditionals affect the edit run only in determining which dumps were recorded on MARSROT tape. However, given a particular MARSROT, dumps may be (further) selected by condition cards inserted in the edit deck, immediately following the DUMP card to which they apply. These conditions may be attached only to channels or erasable registers dumped by the DUMP request.

The following comparisons may be used:

G	Greater than	+0 = - 0
GEQ	Greater than or equal to	+0 = - 0
L	Less than	+0 = - 0
LEQ	Less than or equal to	+0 = - 0
EQ	Equal to	+0 ≠ - 0
UNEQ	Unequal to	+0 ≠ - 0

These operation codes are satisfied only if the contents of the register whose address is in the location field and the quantity in the A address field satisfy the indicated inequality. Thus,

VA UNEQ 35D

is satisfied only if $C(VA) \neq 35$. As another example,

JDDR EQ SUBRO5

is satisfied only if the contents of JDDR are equal to the AGC 15 bit pseudo address of the symbol SUBRO5, as assigned during assembly. The right-hand quantity may be an unsigned octal or decimal integer or a symbol with or without an octal or decimal modifier. Similarly,

GC	> C()	+0 = -0
GEQC	\geq C()	+0 = -0
LC	< C()	+0 = -0
LEQC	\leq C()	+0 = -0
EQC	= C()	+0 \neq -0
UNEQC	\neq C()	+0 \neq -0

compare the contents of two locations. Then, for example

$$VA \quad EQC \quad VC + 4$$

is satisfied only if $C(VA) = C(VC + 4)$.

In general a user may wish to specify several simultaneous inequalities. This is facilitated by including the words AND and OR in Data Address 2 of the condition card. It is understood that the resulting logical expression is in its sum of products form; i. e., parenthesis free. The entire expression is satisfied if all members of one subexpression (sub-expressions are separated by OR's) is satisfied. Thus,

A	GC	B	AND
C	EQ	D	AND
E	LEQ	F	OR
G	EQC	H	AND
L	UNEQ	M	OR
J	EQ	K	

is satisfied if either

$$(1) \ C(A) > C(B) \text{ and } C(C) = D \text{ and } C(E) \leq F$$

or

$$(2) \ C(G) = C(H) \text{ and } C(L) \neq M$$

or

$$(3) \ C(J) = K$$

The dump will be edited only if, in this example, one or more of the above conditions (1), (2), or (3) is satisfied. All condition cards must appear before any subset cards (see below).

While the simulator dumps all erasable in the ranges specified by the simulator DUMP request, it is desirable to be able to select only certain registers to be edited. At the same time, provisions should be made for formats, scale factors, and headings (in the form of right print cards). The following options are available:

OCTAL	Edit registers in octal.
OCT	Edit registers in octal.
SINGLE	Edit registers are converted to signed decimal fractions and multiplied by an optionally supplied scale factor. The octal value is also printed.
DOUBLE	Edit registers interpreted as double precision pairs, converting each pair to a signed decimal fraction and multiplying by an optionally supplied scale factor.

The subsets may be supplied in any order. The registers specified in OCTAL, OCT, SINGLE, and DOUBLE must be within general erasable included in the DUMP to which they refer. If no subset cards are supplied, all registers dumped during simulation are edited.

In detail, the above options are used as follows:

(a) A OCTAL B or
A OCT B

This card causes all registers, from A to B inclusive, to be edited in octal only. If Data Address 1 is blank, only A is printed.

(b) A SINGLE B XXX.XXXXXXXXXBXX
(2 Dec - Style Constant)

All registers from A to B inclusive are converted to signed decimal fractions and multiplied by an optionally supplied scale factor. The octal value is also printed. If Data Address 1 is blank, only A is printed. The optional scale factor is described in the description of DOUBLE.

(c) A DOUBLE B XXX.XXXXXXXXXBXX
(2DEC - Style Constant)

The registers from A to B are converted to a decimal fraction, having forced sign agreement (if Data Address 1 is blank only (A, A + 1) is printed). They are then multiplied by the scale factor beginning in Data Address 2 and continuing to the end of the card (with blanks equivalent to 1.0). The result is printed as a signed fixed-point number if it is between 10^{-1} and 10^{10} , or in floating-point format otherwise. The supplied scale-factor may be of the same general form as DEC and 2DEC: a signed fixed-point number, followed by a possible signed decimal exponent, followed by a possible signed binary exponent. The result must be between 10^{-64} and 10^{63} , but need not be entirely fractional or whole as is demanded for assembly; e. g., 4.6325 E 1 B-2 is a legitimate scale factor.

Finally, another application of condition statements is to select different subsets and scale factors for a dump, conditional upon contents of registers in the dump. That is, a dump card may be submitted in the input deck several times, each copy having its own conditions and formats.

5. B. 4 MAC Input Facilities

Recognizing the need for user programmed post-simulation processing as opposed to the "canned" AGCEDIT processing described above, facilities have been provided which allow one to edit data from the MARSROT tape using MAC subroutines of his own specification. In this mode of operation, the edit program retrieves information from MARSROT, places the edited information in the MAC data file, and passes control to a specified user supplied MAC routine. Upon return, the edit program resumes its search for relevant dumps on MARSROT. AGC dumps and call file dumps may be supplied to MAC subroutines in this fashion.

One MAC subroutine will in general process several files (i.e., dumps and call file dumps at several locations) and it is desirable to allow several MAC subroutines to process simulation results. This capability is provided by means of an identification, called PID.FID. This identifier is composed of two integers separated by a decimal point (for example 3.7). PID is the program identification of the MAC program, which will process this file, and FID is the identification for the resulting edited file which is placed in the MAC data file. Thus, for example, files identified by PID.FIDs of 3.1 to 3.7 represent seven distinct files (separately identified by FID) all to be edited by MAC program 3, as specified by PID. PID.FID is entered in the data file as a floating point number.

Up to 10 MAC programs may be supplied; they are referred to during editing as EDIT1 through EDIT10 (see note on deck composition). The program ID for EDITn is n, so for example EDIT3 may use a dump known as 3.1 and a call file dump labelled 3.2. Cards for generating MAC input data are arranged as follows:

E GENERATE MAC INPUT DATA

Time Card (optional)

A DUMP B C PID. FID

File Cards for the above (optional)

X CFDUMP PID. FID

The DUMP and CFDUMP cards are copies of cards submitted with a corresponding simulation. The identifier PID. FID is punched starting in column 65. At least one DUMP card or one CFDUMP card must be present. If no file cards are included for a DUMP card, only PID. FID, AGC time, and a zero are filed in the first 3 data file locations. The DUMP cards together with file cards, and CFDUMP cards can be arranged in any order. The order in which the editing is done and files produced is determined by the order in which DUMPS and CFDUMPS appear on the MARSROT tape.

Filing and format conversion specifications for AGC dumps are supplied by means of combinations of the following detail cards, referred to in general as file cards.

1. X DPFIL Y Interprets AGC registers from X through Y as double precision pairs and files a signed floating decimal fraction for each pair. If Data Address 1 is blank, only the pair (X, X + 1) is filed.
2. X SPFIL Y Files registers X through Y as single precision floating decimal signed integers. If Data Address 1 is blank, only X is filed.

- | | | | |
|------|---------|---|--|
| 3. X | 2SCFILE | Y | Files registers X through Y like SPFILE except that the registers are interpreted as two's complement numbers (CDU angles, etc.) If Data Address 1 is blank, only X is filed. |
| 4. X | OCTFILE | Y | Files registers X through Y as follows:
Each of the five octal digits is transformed into a decimal digit; e.g., 76543 ₈ is filed as +76,543.0. |
| 5. X | CHFILE | Y | Files channels X through Y as in OCTFILE. -1 is filed as the contents of a non-existent channel. If Data Address 1 is blank, only channel X is filed. |
| 6. | VACFILE | | Files a fixed length item of 23 words which includes the work area and vector accumulator in DPFIL format, and the index and step registers in SPFILE format. These words are taken from the VAC area in use when the dump was taken. It is required that AGC registers FIXLOC thru the last of the VAC areas be included in the dump. |

Combinations of the above file cards write a series of consecutive locations in the data file starting with location 0, just as a series of FILE WRITE statements in a MAC program would do. All locations in the data file which are not used by the edit program are available to MAC programs. In the same fashion, editing of CFDUMP s causes the last 150 words of the call file to be written into data file locations 100 through 249 for processing by MAC programs.

During execution of MACEDIT, three types of calls are made to each MAC subroutine.

A) Initialization call. Before the tape search begins, each program whose ID appears on at least one DUMP or CFDUMP card is called once, with the following information in the data file:

0	PID.000
1	-1.0
2	0.0
3	9999.999999

During this call, all initialization of the MAC edit program should be done and all input cards read. It is recommended that each MAC data card intended for EDITn have n.0 punched in the first field for input checking. At the end of this call, as always, return should be via EXIT. Subroutines are called in ascending order.

B) File calls. Each time a requested file has been found on MARSROT the requesting MAC subroutine is called. The file is made up as follows:

AGC Erasable Dumps

<u>File Location</u>	<u>Contents</u>
0	PID. FID
1	AGC time at which dump was taken.
2	0
3 through $2 + j_1$	j_1 words generated by file card 1.
$3 + j_1$ through $2 + j_1 + j_2$	j_2 words generated by file card 2.
$3 + \sum_{i=1}^{k-1} j_i$ through $2 + \sum_{i=1}^k j_i$	j_k words generated by file card k.
$3 + \sum_{i=1}^k j_i$	9999.999999
$4 + \sum_{i=1}^k j_i$ through end of data file	available for use by MAC program.

Call File Dumps

<u>File Location</u>	<u>Contents</u>
0	PID. FID
1	AGC time at which dump was taken
2	0.0
3-99	unaltered
100-249	MAC call file (last 150 words)
250-end of data file	available for use by MAC programs

C) Summary call. Each subroutine whose ID appears on a dump or call file dump card is called at the end of the MARSROT search. The file contents here are:

0	PID.000
1	Time at end of run
2	0.0
3	9999.999999

Again, MAC subroutines are called in ascending order. Subroutines should do any summary editing and return via EXIT.

All user-supplied MAC subroutines are run under the control of a short MAC program named EDIT.CONTROL, which may be thought of as part of the edit program. Its action is of no concern to the user unless plots are being generated via FILEPLOT. In this case, the user must not do the final CALL FILEPLOT, (-1) but rather let EDIT.CONTROL do this. To implement this feature, EDIT.CONTROL always reads one MAC data

card, which has to be the first of the MAC data cards submitted (see note on input deck composition). The first word must be zero. If the second word (PLOTSW) is non-zero the final FILEPLOT call is made after all subroutines have returned from their summary call. This plot parameter is available to all subroutines, as EDITn is called by the statement CALL EDITN, PLOTSW each time (see EDIT.CONTROL logic).

5. B. 5 Interpretive Tracing Options

Recognizing that the majority of computation in the AGC is done within the interpreter, the interpretive trace options provide a powerful tool for fault isolation and numerical analysis of such programs. One line is printed for each interpretive instruction, containing the following information:

Location	:	The location of the operation code is printed in FBANK notation.
Op Code	:	The symbolic operation code is edited, with a suffix asterisk if the instruction has an indexed operand address. In this case, a second line is printed displaying the contents of the indicated index register.
Operand Address	:	The operand address is printed in one of four forms: Work area or shift count: 00D-37D. X1 through QPRET : Symbolic. General erasable : E-Bank notation. Fixed Memory : F-Bank notation.
Results	:	Results in MPAC are edited as signed, fixed-point, nine digit decimal fractions, enclosed in parenthesis if vector. If the results are left in an index register, they are edited in octal, as well as in bank notation and complemented bank notation if possibly meaningful.

Operand Contents : The contents of the operand location prior to instruction execution are edited in a format appropriate to the instruction.

To edit all trace generated during simulation, simply submit the edit director:

E EDIT INTERPRETIVE TRACE

or

E EDIT ITRACE

To select only certain portions of trace for editing, submit a copy of each ITRACE card which generated a desired portion after the edit director.

5. B. 6 Time Selection Options

One way the user can select a portion of MARSROT data of interest is by using time selection options. This enables one to focus his attention on a given period of time during the simulation. Time selection is requested by submitting TIME cards of the following format:

XX.XXX TIME YYY.Y

where the X's and Y's are "2DEC" style constants. The card is coded in simulator format. The above card selects only those data generated between XX.XXX and YYY.Y seconds. Time cards may appear in any of four places in the edit input deck. If a TIME card immediately follows * AGCEDIT, all data outside the selected interval is completely ignored. If one immediately follows the edit directors

E EDIT AGC DUMPS or E EDIT LGC DUMPS

E GENERATE MAC INPUT DATA

E EDIT INTERPRETIVE TRACE or E EDIT ITRACE

no data is supplied to that edit option unless it is within the specified interval. Obviously, all times on the above TIME cards must fall within the times on a TIME card following * AGCEDIT (provided one is present).

Since a TIME card following * AGCEDIT effectively shortens the MARSROT tape for that run, considerable savings may be obtained when editing a portion of a multi-reel MARSROT; the TIME card prevents the edit program from searching down reels on which there is no data of interest. An additional capability is that ITRACE can now be turned on or off at any AGC time. In other words, the use of a TIME card allows the user to select a series of ITRACE records to be edited, even though the first and/or last record to be edited is in the middle of an ITRACE group on the MARSROT tape.

5. B. 7 Composition of Edit Input Decks

The edit input deck has two general forms depending on whether MAC subroutines are used. If no MAC processing is to be done, the edit deck is of the following form:

```
*      AGCEDIT (MARSROT XXXXXX)
      (Possible TIME card)

E      (Edit director)
      (Possible TIME card)
      (Possible detail cards)
      (Other edit directors, time cards, and detail cards)
```

If MAC subroutines are to be used, the input deck takes on a different form:

```
*      COLLECT EDIT. CONTROL

*      FOR EDIT1 USE YOURPRG1

*      FOR EDIT2 USE YOURPRG2
      ...
*      FOR EDITm USE YOURPRGm

*      AGCEDIT (optional)

*      AGCEDIT (MARSROT XXXXXX)
      (Possible TIME card)

E      (edit director)
      (Possible TIME card)
      (Possible detail cards)
```

E (Possible edit directors, TIME cards, detail cards)

F 0, PLOTSW
 (EDIT. CONTROL input card; PLOTSW non-zero if plotting)

F (MAC data cards for each MAC subroutine. Initialization
 (card reading) calls are made in ascending order).

* ENDJOB

No IBM job and job step control cards are shown in the above deck set-ups.
Refer to Chapter 3, Section B. for discussion of these control cards.

Since all BCD codes are translated to EBCDIC, input cards can
use both codes, even in the same input deck.

5. C. Edit Logic

5. C. 1 AGCEDIT

1. The edit programs are entered via IBM O/S bypassing EDIT. CONTROL.
2. After finding and editing a dump, the results are printed.
3. One line of print-out is produced for each interpretive instruction edited.
4. Exit after all requests are completed.

5. C. 2 MACEDIT

1. MAC executes collection phase and passes control to EDIT. CONTROL.
2. EDIT. CONTROL reads plotter control card.
3. EDIT. CONTROL issues an initial call to each MAC program used with the call file set as follows:

location	0	PID.000
	1	-1.0
	2	0.0
	3	9999.999999

4. After finding and editing a requested dump, control is given to the indicated user supplied MAC program with the data file loaded as follows:

location	0	PID. FID
	1	AGC time
	2	0.0
	3-n	Edited dumps

5. After finding and editing a call file dump, control is given to the indicated user supplied MAC program with the data file loaded as follows:

location	0	PID. FID
	1	AGC time
	2	0.0
	3-99	unused
	100-249	CALLFILE ₁₀₀ to CALLFILE ₂₄₉

6. Summary calls are given to each subroutine after all requests have been processed with the data file loaded as follows:

location	0	PID. 0
	1	AGC time
	2	0.0
	3	9999.999999

7. If the second field of the plotter card is non-zero, control is transferred to program FILEPLOT.
8. EXIT back to MAC supervisor.

5. D Summary of Edit Control Cards

5. D. 1 Summary of edit directors

E	EDIT AGC DUMPS
E	EDIT LGC DUMPS
E	EDIT ITRACE
E	EDIT INTERPRETIVE TRACE
E	GENERATE MAC INPUT DATA
E	HOLD MARSROT
E	SAVE MARSROT
E	RELEASE MARSROT

5. D. 2 Summary of simulator special request cards, copies of which can be included in the edit input deck.

DUMP

CFDUMP

ITRACE

5. D. 3 Summary of subset cards which can be used in editing dumps.

OCTAL

OCT

SINGLE

DOUBLE

5. D. 4 Summary of subset cards which can be used when generating
MAC input data.

DPFILE

SPFILE

2SCFILE

OCTFILE

CHFILE

VACFILE

5. D. 5 Summary of conditionals

G

GEQ

L

LEQ

EQ

UNEQ

GC

GEQC

LEQC

EQC

UNEQC

5. D. 6 Summary of connectives

AND

OR

5.E LISTING OF EDIT.CONTROL

FIRST PASS THROUGH, READ CONTROL CARDS.

CALL (NONMAC, RESIDENT) MARSROT

NEVER EXECUTE. EDIT PROG WILL LOAD

START CALL (NONMAC, RESIDENT) AGCEDITM

READ TEMPI, PLOTSW

IF TEMPI= 0, GO TO CARDOK

EDIT.CONTROL CARD INPUT ID ERROR.

EXIT

CARDOK IF PLOTSW = 0, GO TO DOEDIT

PRINT MSG, SP2

EDITCONTROL CARD INPUT REQUESTS PLOTTING.

DOEDIT SET FILE READ 0

FILE READ PIDFID

AFTER ALL SUMMARY CALLS, AGCEDIT CALLS MAC WITH PIDFID OF 9999.999999 TO TERMINATE THE RUN.
 IF PIDFID = 9999.999999, GO TO PENULT

FOR ALL OTHER CALLS, JUST CALL THE INDICATED SUBROUTINE AND RETURN TO AGCEDIT IMMEDIATELY.

DO (INTEGER (PIDFID)), BADCALL, CALL1, CALL2, CALL3, CALL4,

CALL5, CALL6, CALL7, CALL8, CALL9, CALL10, BADCALL

CALL (NONMAC, RESIDENT) AGCEDITM

GO TO DOEDIT

DO FINAL FILEPLOT CALL IF REQUESTED.

PENULT IF PLOTSW = 0, EXIT

CALL FILEPLOT, (-1)

EXIT

SUBROUTINE CALL STATEMENTS.

CALL1 CALL EDIT1, PLOTSW
CALL2 CALL EDIT2, PLOTSW
CALL3 CALL EDIT3, PLOTSW
CALL4 CALL EDIT4, PLOTSW
CALL5 CALL EDIT5, PLOTSW
CALL6 CALL EDIT6, PLOTSW
CALL7 CALL EDIT7, PLOTSW
CALL8 CALL EDIT8, PLOTSW
CALL9 CALL EDIT9, PLOTSW
CALL10 CALL EDIT10, PLOTSW

FOR CALLS WHICH ARE OUT OF RANGE:

BADCALL DO TO BADCALLX
PRINT MSG, SP0
THIS PROGRAM ID WAS FOUND IN THE DATA FILE:
PRINT BLANK, BLANK, BLANK, PIDFID, SP2
PRINT MSG, SP2
BAD PROGRAM ID IN DATA FILE
BADCALLX EXIT
START AT START

Appendix A

Call File Map

The following table describes the assignment of Call File locations for Block II digital simulations.

The call file is used to transmit variables between MAC environment programs, and to store variables which are of use for summary printing (ENVSUM Special Request) or for environment parameter dumps (CFDUMP Special Request). Many of the locations outlined are not normally updated without special provision. Therefore, anyone writing edit programs for Call File dumps should check with a member of the Digital Simulation Group for more detailed information on Call File updates.

Location	Mnemonic	Units	Maintained By	Remarks
2000080 to 2000099			ENVCNTRL	Call File area used to accumulated raw timing data prior to processing.
2000100 to 2000103	T - STARVI	Sec Unit vector	ASTRNAUT	Star vector of planet vector to be marked by ASTRNAUT.

Table A.1 Call File Storage Map

Location	Mnemonic	Units	Maintained by	Remarks
2000104	T	sec	UNIVERSE	Time at which the following six locations were updated.
2000105 to 2000107	\bar{R}_M	m	UNIVERSE	The three components of moon position in the Reference Inertial System whose origin is the center of the earth.
2000108 to 2000110	\bar{R}_S	m	UNIVERSE	The three components of sun position in the Reference Inertial System whose origin is the center of the earth
2000111 to 2000126	TRCON ₁ to TRCON ₁₆	sec	SATURN CSMBLK2 LEM LVCBK 2	Accumulated firing time for each RCS jet. These locations were updated at the time filed in Call File location 2000220.
2000127	T	sec	TARGET	Time at which the following seven locations were updated.
2000128	ORIGIN	N/A	TARGET	Origin of the Reference Inertial System in which following two vectors are given.
2000129 to 2000131	\bar{R}_{T_I}	m	TARGET	Position of target vehicle in the Reference Inertial System whose origin is specified by ORIGIN in Call File location 2000128.

Table A.1 (Cont.) Call File Storage Map

Location	Mnemonic	Units	Maintained by	Remarks
2000132 to 2000134	$\bar{V}T_I$	m/sec	TARGET	Velocity of the target vehicle in the Reference Inertial System whose position is specified by ORIGIN in location 2000128.
2000135	T	sec	OPTICSB2 RADAR	Time at which the following two locations were updated.
2000136	SHAFT	rad	OPTICSB2 RADAR	Shaft angle of the optics if the CSM is the active vehicle or the rendezvous radar if the LEM is the active vehicle.
2000137	TRUNNION	rad	OPTICS B2	Trunnion angle of the optics if the CSM is the active vehicle or the rendezvous radar if the LEM is the active vehicle.
2000138 2000139				Unused
2000140 to 2000148	* T_{IL}	N/A	PRELUNCH SATURN	The transformation from the Local Vertical System to the Reference Inertial System. The PRELUNCH subroutine updates this transformation prior to liftoff. The last update was made at the time filed in Call File location 2000220. At liftoff the SATURN subroutine files this transformation and thereafter it remains constant.

Table A.1 (Cont) Call File Storage Map

Location	Mnemonic	Units	Maintained by	Remarks
2000149	IMUTSTOP	sec	IMUBLK2	Time beyond which the vehicle program should not go without updating the IMUBL K2 subroutine.
2000150	T	\sec	IMUBLK2	Time at which preceding location and following twenty-four locations were updated.
2000151	AOG	rad	IMUBLK2	IMU outer gimbal angle.
2000152	AIG	rad	IMUBLK2	IMU inner gimbal angle.
2000153	AMG	rad	IMUBLK2	IMU middle gimbal angle
2000154 to 2000162	* T _{SN}	N/A	IMUBLK2	The transformation from the Navigation Base System to the Stable Member System.
2000163 to 2000171	* T _{SI}	N/A	IMUBLK2	The transformation from the Reference Inertial System to the Stable Member System.
2000172	CDU ₀	rad	IMUBLK2	IMU X CDU angle.
2000173	CDU ₁	rad	IMUBLK2	IMU Y CDU angle.
2000174	CDU ₂	rad	IMUBLK2	IMU Z CDU angle
2000175	TRIMALPH	rad	LVCBLK2	Initial trim angle of attack

Table A.1 (Cont) Call File Storage Map

Location	Mnemonics	Units	Maintained by	Remarks
2000176	T	sec	SATURN CSM BLK2 LEM LVCBLK2	Time at which following six locations were updated.
2000177 to 2000179	$\bar{D}R$	m	SATURN CSM BLK2 LEM LVCBLK2	The second integral of contract acceleration, \bar{A}_I , over the interval from the last UNIVERSE update to the time filed in Call File location 2000176. (See Call File location 2000189).
2000180 to 2000182	$\bar{D}V$	m/sec	SATURN CSM BLK2 LEM LVCBLK2	The first integral of contact acceleration, \bar{A}_I , over the interval from the last UNIVERSE update to the time filed in Call File location 2000176. (See Call File location 2000189).
2000183	GEOMALT	m	UNIVERSE	Altitude above the Fischer Ellipsoid for the Earth.
2000184 to 2000188	T ERCTRX ERCTRY ERCTRZ	bits	IMUBLK2	Error counter enable in bits.
2000189	\bar{A}_I	m/sec ²	PRELUNCH SATURN LEM CSM BLK2 LVCBLK2	The vector contact acceleration of the navigation base in the Reference Inertial System. This includes all non-gravitational accelerations.

Table A.1 (Cont) Call File Storage Map

Location	Mnemonic	Units	Maintained by	Remarks
2000192 to 2000194	$\bar{A}DOT_I$	m/sec ³	PRELUNCH SATURN LEM CSMBLK2 LVCBLK2	The time rate of change of A_I .
2000195	PIPTSTOP	sec	PRELUNCH SATURN CSMBLK2 LVCBLK2 LEM	The time beyond which the acceleration and acceleration rate (See Call File location 2000189-194) Taylor series would yield an error of more than one PIPA pulse. This time is used by the IMUBLK2 subroutine to limit PIPA recipes.
2000196	T	sec	PRELUNCH UNIVERSE	Time at which the following thirteen locations were updated.
2000197	ORIGIN	N/A	UNIVERSE	The origin of the Reference Inertial System. Earth = 1 Moon = 2
2000198 to 2000200	\bar{R}_I	m	PRELUNCH UNIVERSE	The position of the spacecraft in the Reference Inertial System

Table A.1 (Cont) Call File Storage Map

Location	Mnemonic	Units	Maintained by	Remarks
2000201 to 2000203	\bar{V}_I	m/sec	PRELUNCH UNIVERSE	The velocity of the spacecraft in the Reference Inertial System.
2000204 to 2000206	$\bar{V}ATM_I$	m/sec	UNIVERSE	The velocity of the spacecraft with respect to the atmosphere in the Reference Inertial System.
2000207	Q	kg/m- sec ²	UNIVERSE	Dynamic pressure acting on the spacecraft. Filed only during atmospheric flight.
2000208	QDOT	kg/m- sec ²	UNIVERSE	Time rate of change of dynamic pressure. Filed only during atmospheric flight.
2000209	MACH	N/A	UNIVERSE	Mach number. Filed only during atmospheric flight.
2000210				Unused
2000211	T	sec	PRELUNCH SATURN LEM CSMBLK2 LVCBLK2	Time at which the following three locations were updated.

Table A.1 (Cont) Call File Storage Map

Location	Mnemonic	Units	Maintained by	Remarks
2000212 to 2000214	\bar{F} INT	m/sec	PRELUNCH SATURN LEM CSMBLK2 LVCBLK2	The integral of contact acceleration, \bar{A}_I , over the interval from the beginning of the simulation to the time filed in Call File location 2000211. (See Call File location 2000189).
2000215 to 2000217				Unused
2000218	ANGTSTOP	sec	PRELUNCH SATURN LEM CSMBLK2 LVCBLK2	The time beyond which the angular velocity and angular acceleration (See Call File location 2000230-235) Taylor series would yield an error of more than one CDU pulse. This time is used by the IMUBLK2 subroutine to limit CDU recipes.
2000219	METHOD	N/A	INPUTPRG PRELUNCH SATURN CSMBLK2	A flag indicating which subroutine is integrating the equations of motion for the spacecraft.

Table A.1 (Cont) Call File Storage Map

Location	Mnemonic	Units	Maintained by	Remarks
2000220	T	sec	PRELUNCH SATURN LEM CSMBLK2 LVCBLK2	Time at which the following sixteen locations were updated.
2000221 to 2000229	* T_{IN}	N/A	PRELUNCH SATURN LEM CSMBLK2 LVCBLK2	Transformation from the Navigation Base System to the Reference Inertial System.
2000230 to 2000232	\bar{W}_N	rad/sec	PRELUNCH SATURN LEM CSMBLK2 LVCBLK2	The angular velocity of the navigation base expressed in the Navigation Base System.
2000233 to 2000235	$\bar{\alpha}_N$	rad/sec ²	PRELUNCH SATURN LEM CSMBLK2 LVCBLK2	The angular acceleration of the navigation base expressed in the Navigation Base System.

Table A.1 (Cont) Call File Storage Map

Location	Mnemonic	Units	Maintained by	Remarks
2000236	VIRGIN	N/A	PRELUNCH SATURN LEM CSMBLK2 LVCBLK2	A flag to indicate whether or not the Call File has been initialized
2000237 to 2000239	PITCH YAW ROLL	rad	ENVCTRL	The Euler rotations from the Local Horizontal System to the Navigation Base System.
2000240	MASSVEH	kg	PRELUNCH SATURN LEM CSMBLK2 LVCBLK2	Mass of the vehicle updated at the time filed in Call File location 2000220.
2000241	RCSFUEL	kg	LEM CSMBLK2 LVCBLK2	Total RCS fuel remaining, updated at the time filed in Call File location 2000220.
2000242	SPSFUEL	kg	LEM CSMBLK2	Mass of remaining SPS fuel, updated at the time filed in Call File location 2000220.

Table A.1 (Cont) Call File Storage Map

NOTE: CSMBLK2 and LEM share the following locations. See next page for CSMBLK2 assignments.

Location	Mnemonic	Units	Maintained by	Remarks
2000243	DPSFUEL	kg	LEM	Mass of remaining DPS fuel updated at the time filed in Call File location 2000220.
2000244	APSFUEL	kg	LEM	Mass of remaining APS fuel updated at the time filed in Call File location 2000220.
2000245	FC	Bits	LEM	Thrust command in bits, updated at the time filed in Call File location 2000220.
2000246	THRUST/ FMAX	Non dimen.	LEM	Thrust as fraction of full thrust.
2000247 to 2000249	$\bar{A}NGAE$	rad/sec ²	LEM	Angular acceleration of vehicle due to main engine and eg. offset, ie, $\bar{A}NGAE = INVINERT (\bar{R} * \bar{THRUST})$

Table A.1 (Cont) Call File Storage Map

CSM assignment of locations 2000243 - 2000248

Location	Mnemonic	Units	Maintained by	Remarks
2000243 to 2000245	CGX CGY CGZ	Meters	CSMBLK2	Center of gravity of vehicle updated at the time filed in Call File location 2000220.
2000246 to 2000248	IXX IYY IZZ	Newton M sec ²	CSMBLK2	Moments of inertia of vehicle updated at the time filed in Call File location 2000220.

Appendix B

Data File Map

The following four tables contain the current layout of assigned locations in the data file. The data file is a 300,000 word file accessible to MAC programs which contains input data to the MAC environment programs and output data generated by the environment programs. The last block of 250 words in the data file (locations 299750 - 299999) is kept in core memory at all times and is accessible to both the AGC Simulator and the Environment. This block, which is known as the COMMBLK, is the primary means of communication between the assembly language coded AGC Instruction Simulator and the MAC coded Environment.

The Data File is used during simulations for four distinct purposes:

1. Storage of variables to be plotted in a subsequent edit run.
2. Storage of environment input card images by INPUTPRG.
3. Storage of environment vehicle mission specific parameter files (see Section 2. D. 4).
4. Communication between Instruction Simulator and Environment.

The data file locations utilized for these four functions are documented in Tables B. 1 to B. 4 below.

Table B. 1
Data File Storage Layout

<u>Data File Location</u>	<u>Use</u>
000 } to 999 }	Unassigned
1000 } to 49998 }	Storage area used to accumulate timing for plotting by SIDPLOT.
49999	IMU PLOT FLAG If location 49999 is zero, timing data may overflow into the IMU PLOT storage area. If location 49999 is non zero, the IMU PLOT storage area is reserved.
50000 } to 99998 }	IMU PLOT storage area used to accumulate IMU data for plotting.
99999	VEHICLE PLOT FLAG If location 99999 is zero, IMU PLOT data may overflow into the VEHICLE PLOT storage area. If 99999 is non zero, the VEHICLE PLOT storage area is reserved.
100000 } to 283999 }	VEHICLE PLOT storage area used to accumulate vehicle dynamical data for plotting.
284000 } to 285999 }	Storage area for card images of environment input cards. (See Table B. 2).
286000 } to 299749 }	Storage area for environment vehicle program mission specific parameter files. (See Table B. 3).
299750 } to 299999 }	Communication block used to transmit data between the AGC Instruction Simulator and the Environment. (See Table B. 4).

Table B. 2
Environment Input Card Image Storage Layout

<u>Starting Data File Location</u>	<u>Use</u>
284000	Input File for DISCONT (Input Channel Discontinuity Routine)
284250	Environment only command list.
284900	TARGET card image file.
285000	LMP1 card image file.
285100	UNIVERSE card image file.
285200	RADAR card image file.
285300	PRELUNCH card image file.
285400	OPTICSB2 card image file.
285500	SATURN card image file.
285600	IMUBLK2 card image file.
285700	LEM card image file.
285800	CSMBLK2 card image file.
285900	LVCBLK2 card image file.

Table B. 3
Mission Specific Vehicle Parameter Files

<u>Starting Data File Location</u>	<u>Use</u>
286000	LVCBLK2 Parameter File.
287000	SATURN Parameter File.
288000	CSMBLK2 Parameter File.
289000	LEM Parameter File.
290000	SATURN Trajectory File.
292000	ASTRONAUT Input File.
293750 } to 299749 }	UNIVERSE Ephemeris File.

Table B. 4
Communications Block

<u>Data File Location</u>	<u>Contents</u>	<u>Use</u>
299750 to 299839	Commands to Environment	
299840	ASTRNAUT Return Request	} Environment Return Requests
299841	IMUBLK2 Return Request	
299842	Vehicle Return Request	
299843	OPTICSB2 Return Request	
299844	RADAR Return Request	
299845	DISCONT Return Request	
299846	LMP1 Return Request	
299847	Unused	
299848	Unused	
299849	Latest Return Request to ENVCNTRL	

Block 2 Data File
Communication Block

<u>Data File Location</u>	<u>Contents</u>	<u>Use</u>
	<u>CHAN 15</u>	
299850	5 4 3 2 1	} Input Bits and Channel Assignments
	<u>CHAN 16</u>	
299851	7 6 5 4 3 2 1	
	<u>CHAN 30</u>	
299852	10 9 8 7 6 5 4 3 2 1	
	<u>CHAN 31</u> <u>CHAN 30</u>	
299853	5 4 3 2 1 15 14 13 12 11	
	<u>CHAN 31</u>	
299854	15 14 13 12 11 10 9 8 7 6	
	<u>CHAN 32</u>	
299855	10 9 8 7 6 5 4 3 2 1	
	<u>CHAN 33</u> <u>CHAN 32</u>	
299856	5 4 3 2 1 15 14 13 12 11	
	<u>CHAN 33</u>	
299857	15 14 13 12 11 10 9 8 7 6	
299858 } to 299859 }	Unused	

Block 2 Data File
Communication Block

<u>Data File Location</u>	<u>Contents</u>	<u>Use</u>
	<u>CHAN 5</u>	
299860	8 7 6 5 4 3 2 1	
	<u>CHAN 6</u>	
299861	8 7 6 5 4 3 2 1	
	<u>CHAN 10</u>	
299862	10 9 8 7 6 5 4 3 2 1	
	<u>CHAN 11</u> <u>CHAN 10</u>	
299863	5 4 3 2 1 15 14 13 12 11	
	<u>CHAN 11</u>	
299864	15 14 13 12 11 10 9 8 7 6	
	<u>CHAN 12</u>	
299865	10 9 8 7 6 5 4 3 2 1	
	<u>CHAN 13</u> <u>CHAN 12</u>	
299866	5 4 3 2 1 15 14 13 12 11	
	<u>CHAN 13</u>	
299867	15 14 13 12 11 10 9 8 7 6	
	<u>CHAN 14</u>	
299868	10 9 8 7 6 5 4 3 2 1	
	<u>CHAN 14</u>	
299869	15 14 13 12 11	

Output Bits
and Channel
Assignments

Block 2 Data File (cont.)

<u>Data File Location</u>	<u>Contents</u>	<u>Use</u>	
299870	CDU Recipe Origin	} Base Time for Recipes	
299871	OPTICS Recipe Origin		
299872	PIPA Recipe Origin		
299873	BMAG Recipe Origin		
299874	Unused		
299875	CDU SIGMA	} Noise SIGMAS	
299876	OPTICS SIGMA		
299877	PIPA SIGMA		
299878	BMAG SIGMA		
299879	Unused		
299880	T ZERO	} General Variables	
299881	ASTRNAUT Card Flag		
299882	ENVSUM Format Number		
299883	VEHICLE, HALTFLAG		CSM = 0
			Descent = 1
			Ascent = 2
			EXIT = 9999.999999
299884	TINHIB1		
299885	TINHIB2		
299886	FLIGHT NUMBER		
299887	TLIFTOFF		
299888	ENVMODE (Fast-slow option)		
299889	Unused		
299890	Unused		
299891	Unused		

Block 2 Data File (cont.)

<u>Data File Location</u>	<u>Contents</u>	<u>Use</u>
299892	UPLINK 0 = OFF, 1 = ON	Astronaut Keyboard Displays
299893	FLASH 0 = OFF, 1 = ON	
299894	MODE	
299895	VERB	
299896	NOUN	
299897	R1	
299898	R2	
299899	R3	
299900	CHECK FAIL 0 = OFF, 1 = ON	
299901	PROG ALARM 0 = OFF, 1 = ON	
299902	RELEASE DISPLAYS 0 = OFF, 1 = ON	KEYSTROKE BUFFER
299903	Unused	
299904	Location in Buffer next to be emptied by COMMUNICATOR (negative when buffer is empty)	
299905	Location in Buffer next to be filled by ASTRNAUT (positive when buffer is empty)	
299906	TIME OF KEYSTROKE	
299907	CODE	
299908	TIME	
299909	CODE	
:		
299949		

Block 2 Data File (cont.)

<u>Data File Location</u>	<u>Contents</u>	<u>Use</u>
299950	CDUX	<div> <div>Counters and Rates</div> <div> <u>Format</u> INCREMENT RATE RATE DOT TEXPIRE </div> </div>
299954	CDUY	
299958	CDUZ	
299962	OPTT	
299966	OPTS	
299970	PIPAX	
299974	PIPAY	
299978	PIPAZ	
299982	BMAGX, Q-HL	
299986	BMAGY, P-HL	
299990	BMAGZ, R-HL	
299994	RADAR	
299995		Unused
:		
299998		
299999		Permanently Unused

Appendix C

Environment Summary Print

The AGC Instruction Simulator Special Request ENVSUM (see Section 3. C. 4) is used to force the MAC environment program ENVCNTRL to generate lines of print summarizing the state of the environment. This summary print represents, in a compact format, the response of the environment to control stimuli from the AGC. In long production runs, this summary print is often used as the primary output from the environment programs, while AGC edits are utilized as the primary output from the AGC Instruction Simulator. In analyzing the results of long simulation runs, it is convenient to scan through the environment summary print to detect off nominal behavior of the AGC or environment, or to verify nominal functioning.

An example of the ENVSUM format is included in Fig. C. 1, and a description of each term is presented below:

<u>MISSION TIME</u>	Units-seconds
Defined as	ABS (T - TLIFTOFF).
where:	
T	is the AGC Simulator time (initialized by the Address Field of the AGC Special Request START - see Section 3. C. 4).
TLIFTOFF	is the time of liftoff specified on the PRELUNCH input card. If the run does not start in PRELUNCH, TLIFTOFF is assumed to be 0.

MISSION TIME, as printed out by ENVSUM, runs backward to liftoff time (which is defined as zero time), and then forward from liftoff. For lunar prelaunch, the TLIFTOFF parameter does not cause liftoff, but the parameter is used in the computation of MISSION TIME, prior to liftoff. After lunar liftoff, the actual liftoff time is used in the computation of MISSION TIME, thus introducing a possible transient in the time print-out at lift-off.

LAT-GD

Units -degrees.

Geodetic latitude for the Earth or selenocentric latitude for the Moon. LAT-GD is computed by UNIVERSE from a specification of the position vector \mathbf{R}_I in reference inertial coordinates, and a code indicating earth fixed or moon fixed coordinates.

LONG

Units -degrees.

Geodetic longitude for the Earth or selenocentric longitude for the Moon.

ALT

Units-nautical miles.

Geometric altitude above the Fischer Ellipsoid for the Earth, or altitude above the lunar reference sphere (radius 1735.4 km) for the Moon.

INER. VEL

Units -ft/sec.

Absolute value of inertial velocity of navigation base.

GAMMA

Units-degrees.

Flight path angle. Magnitude of the angle between the local horizontal and the velocity vector.

AZIMUTH

Units-degrees.

Azimuth angle of velocity vector in horizontal plane, measured from North.

CDUX, Y, Z

Units-degrees.

CDU angles as simulated by the environment program IMUBLK2.

AOG, AIG, AMG

Units-degrees.

Gimbal angles in the order Outer, Inner, Middle, as simulated by the IMUBLK2 program.

Gimbal angles are measured from the stable member to the navigation base and represent three Euler rotations about the stable member Y, Z, and X axes respectively.

EULERX, Y, Z

Units-degrees.

Orientation angles of the stable member with respect to the reference inertial coordinate system (I frame). These are Euler rotations about the reference inertial Y, Z, and X axes respectively which define the orientation of the stable member in space. Throughout a run, these angles will remain fixed unless the stable member is rotated by a coarse alignment, torqued by gyros in a fine alignment, or allowed to drift through the action of gyro bias errors.

XRATE, YRATE, ZRATE Units-degrees/sec.

Vehicle body rates resolved into navigation base axes. For a rigid vehicle (no bending), the navigation base rates are identical to the vehicle body rates. However, for a flexible vehicle, the rates about the navigation base axes may differ significantly from the rates about the vehicle body axes.

ROLL, PITCH, YAW Units-degrees.

Vehicle orientation angles with respect to a local horizontal coordinate system (the P frame) described in Section 2.D. 6. The angles PITCH, YAW, and ROLL are measured from the P frame to the vehicle body axes, and represent Euler rotations about the Y, Z, and X axes respectively of the P frame.

DCDUX, DCDUY, DCDUZ Units-counts.

Integrated CDU pulses generated by the AGC since the last ENVSUM printout. A running total of all CDU pulses generated by the AGC is maintained by the ENVCNTRL program, which then prints this running total as part of the ENVSUM format, and finally zeros the total after each ENVSUM printout. ENVSUM thus prints the total number of CDU pulses generated by the AGC between ENVSUM printouts.

This running summation of CDU command activity is a useful measure of the stability of the AGC steering laws, since unexplained transients in the ENVSUM printout of accumulated CDU commands may indicate spurious AGC behavior.

DPIPX, DPIPZ, DPIPZ

Units-counts.

Integrated PIPA pulses generated by the IMUBLK2 program since the last ENVSUM printout. As in the case of integrated CDU pulses, the DPIP printout represents a running summation of PIPA pulses which have been sent to the AGC by the IMUBLK2 program between ENVSUM printouts. DPIPX, Y, Z are zeroed after each such printout.

DGYROX, DGYROY, DGYROZ Units-counts.

Integrated gyro torquing pulses generated by the AGC since the last ENVSUM printout. A running total of all gyro torquing pulses generated by the AGC is maintained by the ENVCNTRL program, which then prints this running total as part of the ENVSUM format, and finally zeroes the total after each ENVSUM printout. ENVSUM thus prints the total number of gyro torquing pulses generated by the AGC between ENVSUM printouts.

MASS

Units-lb.

Total mass of vehicle including loss of mass due to SPS and RCS fuel usage. Vehicle mass is held constant during entry, neglecting mass loss due to RCS fuel expenditure and ablation.

DELV

Units-ft/sec.

Velocity increment imparted by contact accelerations, (such as the SPS engine, or atmospheric drag), since the SPS engine last came on. DELV is reset to zero each time the DPS engine is ignited, and thus does not reflect the scalar magnitude of the actual velocity vector.

FUEL

Units-lb.

This represents the total RCS fuel used including fuel expended in +X translation.

ACC

Units-ft/sec.²

Magnitude of contact acceleration at the navigation base. This includes all non-gravitational accelerations.

APOGEE, PERIGEE

Units-nautical miles.

Apogee and perigee of the current orbit of the spacecraft, in nautical miles, referred to the radius of the Earth at the launch site ($6.373328 \cdot 10^6$ meters), or to the 1735.4 km reference sphere for the Moon.

TFF

Units-sec.

Time of free fall to a reference altitude. For the boost phase from Earth, the reference altitude is 280,000 ft. and for all other mission phases near Earth, the reference is 400,000 ft. Near the Moon, the reference altitude is 100,000 ft. If the orbit of the spacecraft will not intersect the reference altitude (circular orbits, some hyperbolic orbits, and portions of orbits below the reference altitude), TFF is set to infinity.

RI-TOGO, CROSS R Units-nautical miles.

Range and cross range from the present position of the space-craft, projected on the Earth's surface, to the desired target splash-down point. The desired target point is read in on the input card to the environment program INPUTPRG (See listing of INPUTPRG parameters in Section 3. H. 4) Range and cross range are measured as great circles on the surface of the Earth.

ALL UNITS IN LB, FT, NMI, SEC, DEG.

MISSION TIME	LAT-GD LONG. ALT-FE	INER.VEL GAMMA AZIMUTH	CDUX CDUY CDUZ	AOG AIG AMG	EULERX EULERY EULERZ	XRATE YRATE ZRATE	ROLL PITCH YAW	DCDUX DCDUY DCDUZ	DPIPX DPIPY DPIPZ	DGYROX DGYROY DGYROZ	MASS DELV FUEL	ACC APOGEE PERIGEE	TFF RI-TOGO CROSS R
12526.0	12.588	25269.357	-179.06	-179.09	84.51	-0.02	179.96	0	0	0	51672	0.0	16216.3
	339.687	27.956	-167.11	-167.29	-143.63	-0.00	59.06	0	0	0	2	9017.5	7763.9
	1533.428	117.604	-2.43	-2.60	36.15	0.01	4.55	0	0	0	794.4	52.5	1337.8
12528.0	12.549	25254.930	-179.06	-179.14	84.51	-0.02	179.90	0	0	0	51672	0.0	16216.3
	339.756	27.978	-167.11	-167.28	-143.63	-0.00	59.15	0	0	0	2	9017.5	7748.1
	1537.322	117.620	-2.43	-2.59	36.15	0.01	4.53	0	0	0	794.4	52.5	1337.5
12530.0	12.510	25240.515	-179.06	-179.19	84.51	-0.02	179.85	0	0	0	51672	0.0	16212.3
	339.824	28.001	-167.11	-167.27	-143.63	-0.00	59.24	0	0	0	2	9017.5	7752.2
	1541.216	117.637	-2.43	-2.56	36.15	-0.00	4.53	0	0	0	794.4	52.5	1337.3
12532.0	12.470	25226.112	-179.06	-179.22	84.51	-0.02	179.80	0	0	0	51672	0.0	16210.3
	339.892	28.024	-167.11	-167.26	-143.63	-0.00	59.34	0	0	0	2	9017.5	7756.4
	1545.111	117.653	-2.43	-2.52	36.15	-0.00	4.54	0	0	0	794.4	52.5	1337.1

T = 12532.2915 SCSMODE CHANGED TO G+N MONITOR (NO MODE).

T = 12532.5289 SCSMODE CHANGED TO G+N DV.

T = 12532.7411 ENGINE SIGNAL CHANGED TO 1.

C-8

3001	DISPLAY:	T1=20027	SECONDS RUNNING:	12534.000	0001
	41 34 XX	XXXXX XXXXX XXXXX			

MISSION TIME	LAT-GD LONG. ALT-FE	INER.VEL GAMMA AZIMUTH	CDUX CDUY CDUZ	AOG AIG AMG	EULERX EULERY EULERZ	XRATE YRATE ZRATE	ROLL PITCH YAW	DCDUX DCDUY DCDUZ	DPIPX DPIPY DPIPZ	DGYROX DGYROY DGYROZ	MASS DELV FUEL	ACC APOGEE PERIGEE	TFF RI-TOGO CROSS R
12534.0	12.431	25221.316	-179.72	-178.31	84.51	-0.42	179.57	-59	-41	0	51613	13.3	16253.8
	339.960	28.060	-167.11	-166.50	-143.63	-1.21	60.23	0	-3	0	11	9043.4	7760.5
	1549.008	117.671	-1.93	-0.84	36.15	-2.85	6.42	46	41	0	793.5	52.1	1336.8
12536.0	12.392	25229.874	179.43	-179.06	84.51	-0.54	178.35	-77	-98	0	51476	13.3	16361.3
	340.027	28.113	-167.11	-166.16	-143.63	0.26	60.69	0	-9	0	37	9105.9	7764.6
	1552.911	117.690	-1.46	0.14	36.15	0.86	6.82	42	99	0	793.4	51.4	1336.4

4346	DISPLAY:	T1=20647	SECONDS RUNNING:	12538.000	0013
	41 06 40	-00668 -00109 0XXXX			

12538.0	12.353	25238.414	179.10	-179.54	84.51	-0.09	177.81	-30	-98	0	51338	13.4	16449.8
	340.095	28.168	-167.26	-166.14	-143.63	0.14	60.80	-13	-8	0	64	9168.8	7762.8
	1556.823	117.710	-1.73	0.40	36.15	0.34	6.78	-24	99	0	793.3	50.6	1336.0

4374	DISPLAY:	T1=21013	SECONDS RUNNING:	12539.000	0007
	41 06 40	-00668 -00109 +00645			

FIGURE C.1

INDEX

ABORT, 3-10, 3-14
 ABORTSTG, 3-89
 ACC, C-6
 action clauses for astronaut, 2-65
 addressing, 3-8
 ADOT, A-7
 ADRES, 3-10, 3-14
 ADVAN, 3-10, 3-14
 A frame, 2-82
 AFTER condition, 3-33
 AGC dump, 3-18, 5-1, 5-15
 AGCEDIT, 3-36, 5-1 to 5-11, 5-20,
 5-21, 5-23
 AGC programs, 2-10
 AGCSIM, 3-36
 AGC Simulator Director cards, 3-4
 AGC special requests, 2-10, 2-11, 3-7
 to 3-30
 AGC time in simulation, 3-29
 A_I , A-6
 ALLOW, 3-25
 ALLRUPT, 3-25
 ALT, C-2
 AMODE, 3-101
 AND condition, 3-32
 ANGAE, A-12
 ANGTSTOP, A-9
 angular velocity and acceleration, A-10
 AOG, AIG, AMG, A 5, C-3
 APOGEE, C-6
 apollo vehicle reference system, 2-82
 APSFUEL, A-12
 ascent engine, 2-52
 ASENGARM, 3-89
 ASSYTAPE, 3-6
 astronaut, 2-18, 2-19, 2-24, 2-63, to
 2-66, 3-52, 3-82 to 3-98, B-4
 ASTRNAUT, (see astronaut)
 astronaut card flag, B-8
 astronaut cards for rollback, 4-5
 astronaut file, 2-71, 3-94, to 3-98
 astronaut input cards, 2-29, 3-37, 3-42
 astronaut input program, 2-21, 2-24,
 2-29, 2-63, 2-71, 2-73
 astronaut keystrokes, 3-87
 astronaut keyboard display, B-9
 astronaut only mode, 3-37, 3-38, 3-49
 astronaut subprograms, 3-91, to 3-93
 astronaut switches, 3-88
 ASTXMCI program (astro input prog.)
 2-21, 2-24, 2-29, 2-63, 2-71, 2-73
 atmosphere, 2-33, 2-55
 ATTITUDE program, 2-38, 2-39
 AZIMUTH, C-3
 BACKGRND, 3-10, 3-15, 4-4
 bank notation, 3-8
 BANKSW, 3-25
 BBON, 3-10, 3-15
 BCD codes, 5-22
 BEFORE condition, 3-33
 bench test mode, 2-6, 2-7, 3-34 to
 3-36, 3-49
 bending, 2-44
 bias errors in gyro, pipas, 3-76
 BKGRND, 3-10, 3-15, 4-4
 body axes, 2-82
 boost phase, 2-38, 2-39
 CADR, 3-10, 3-16

INDEX

- call file dump, 3-16, 5-1, 5-16
- call file map, A-1 to A-13
- card image storage, B-3
- Cassini's laws, 2-32
- CDU angles, 2-87, 3-75, A-5, C-3
- center of gravity, A-13
- CFDUMP, 3-10, 3-16, 3-25, 5-1, 5-13, 5-14, A-1
- CGX, Y, Z, A-13
- CHANGE, 3-10, 3-16
- CHANL, 3-10, 3-17, 4-4
- CHANNELXX, 3-9, 3-30
- CHECK FAIL, 3-85
- CHFILE, 5-14
- class 1 special requests, 3-10, 3-11, 4-4
- class 2 special requests, 3-10, 3-11, 4-4
- class 3 special requests, 3-10, 3-11, 3-12, 3-22, 3-27, 4-4
- class 13 special requests, 3-10, 3-12
- CLOCK, 3-10, 3-17, 3-25
- CMC, 3-90
- CMSMSEP, 3-88
- COLLECT card, 2-72, 3-27, 3-37, 3-38, 3-42, 3-43
- comma-delimited cards, 3-51
- command module, 2-41 to 2-47
- COMMBLK, B-5 to B-10
- communications block, B-5 to B-10
- communicator, 1-4, 2-8, 2-13, 2-75
- CONDITIONAL, 3-12, 3-19, 3-31 to 3-33
- condition cards for edits, 5-7, 5-8
- conditional clauses for astronaut 2-64
- condition field, 3-7
- CONST, 3-10, 3-17
- coordinate systems, 2-77 to 2-81
- counter extrapolation, 2-13
- counter increments and rates, B-10
- CROSS R, C-7
- CSMBLK2.FILE205, 2-25
- CSMBLK2 program, 2-17, 2-18, 2-23 2-41 to 2-47, 2-82, 3-52, 3-62, B-3, B-4
- CSM model, 3-63, 3-65
- CSMRATE, 3-90
- CSTMODEL, 2-40
- CYCLE condition, 3-33
- data address field, 3-7
- data files, 2-24, 2-25, 2-72, 3-43, 5-14, 5-15, 5-16, B-4
- data file map, B-1 to B-10
- DCDUX, Y, Z, C-4
- DCENGARM, 3-89
- DEC, DEC *, 3-10, 3-18
- DELETE, 3-10, 3-12, 3-18
- DELV, C-5
- DEPLOY LANDING GEAR, 3-90
- descent engine, 2-49 to 2-51
- detail cards, 5-13, 5-14, 5-21
- DGYROX, Y, Z, C-5
- digital autopilot, 2-4, 2-47
- director cards, 3-4
- DISCONT program (discontinuity) 2-21, 2-24, 2-69, 3-52, 3-100, B-3
- DOUBLE edit format 5-10
- DPFILE, 5-13

INDEX

- DPIPX, Y, Z C-5
- DUMP, 3-10, 3-18, 3-19, 5-1, 5-6,
5-9, 5-10, 5-13
- DPSFUEL, A-12
- DR, A-6
- DV, A-6
- Earth fixed frame, 2-31
- EBANKSW, 3-25
- EBCDIC code , 5-22
- ECADR, 3-10, 3-19
- EDIT.CONTROL, 5-16, 5-17, 5-21,
5-23, 5-28
- edit control cards, 5-25 to 5-27
- edits, 5-1 to 5-27
- EDIT AGC DUMPS, 5-6
- edit conditionals, 5-7, 5-8
- edit director cards, 5-2, 5-21
- edit formats, 5-9
- edit initialization call, 5-15
- edit input decks, 5-6, 5-21, 5-22
- EDIT INTERPRETIVE TRACE, 5-19
- EDIT ITRACE, 5-19
- EDIT LGC DUMPS, 5-6
- edit logic, 5-23
- edit MAC programs, 5-12
- EDUMP, 3-10, 3-25
- entry, 2-55
- ENVCNTRL program, 2-15, 2-17,
2-18, 2-22, 2-28, 2-75, 2-76,
3-52, 3-105, B-3
- environment, 1-3, 1-4, 2-14 to 2-76
- environment input cards, 2-4, 3-44,
3-49 to 3-107, 4-5, B-3
- environment only mode, 2-7, 3-46 to
3-48, 3-49, 3-105
- environment summary prints, 3-20
- ENVMODE, 2-47, 3-10, 3-19, B-8
- ENVSUMDT, 3-56
- ENVSUM, 3-10, 3-19, 3-20, B-8
- ephemeris data file, 2-25, 2-30
- EPOCH, 3-57
- EQ conditional, 3-32, 5-7
- EQC conditional, 3-33, 5-8
- EQUALS, 3-10, 3-20
- erasable memory, 2-9
- ERCTRX, Y, Z A-6
- EULER angles for IMU, 2-86, 3-75
- EULER X, Y, Z C-3
- EVERY conditional, 3-33
- EXEC card, 3-2, 3-36, 3-38
- execute procedures see EXEC
- EXERASE, 3-10, 3-20, 3-21
- failures in RCS jets, 3-65, 3-67, 3-70
- fast options, 2-40, 2-46, 2-47, 2-57, 3-19,
B-8, 3-60, 3-62, 3-67, 3-74, 3-79
- FC, A-12
- F cards, 3-51
- file calls for edits, 5-15
- FILEPLOT, 5-16, 5-17, 5-24
- FILE504.CSM, 2-25
- FILE504.LVCBLK2, 2-25
- FILE504.SATURN, 2-25
- final call for edits, 5-16
- FINT, A-9
- FIXLOC, 5-14
- FLASH, 3-85
- FLIGHTNO, 3-55, B-8
- flight number, 3-55, B-8
- flight test mode, 2-6, 2-7, 3-41 to 3-45,
3-49
- forced ENVSUM, 3-20

INDEX

forced updates(env only mode), 3-107
 formats for edits, 5-9
 format of MAC data cards, 3-51
 fuel loading, 3-69
 fuel usage, A-11, A-12
 GAMMA, C-3
 G conditional, 3-32, 5-7
 GC conditional, 3-33, 5-8
 generate MAC input data, 5-13
 GEOMALT, A-6
 GEQ conditional, 3-32, 5-7
 GEQC conditional 3-33, 5-8
 gimbal angles, 2-87, 3-75
 GO card, 3-48
 GOSEQ, 3-10, 3-21
 GOSEQOK, 3-10, 3-21
 gravity, 2-30, 2-33, 2-35, 2-80
 GUIDEMODE, 3-88
 gyro errors, 3-75
 hand controller, 3-88, 3-89
 HANDCTRL, 3-88, 3-89
 HOLD MARSROT, 3-4, 3-6, 3-34,
 3-35, 3-38, 3-42, 4-1, 5-3, 5-5, 5-25
 ICOUNT, 3-10, 3-22
 ID codes for input cards, 3-52
 I frame, 2-78
 IF, THEN, 2-64, 3-83
 IMUBLK2 program, 2-15, 2-16, 2-18,
 2-23, 2-57, 3-52, 3-74, B-3
 IMU plot flag, B-2
 IMU POWER, 3-88
 IMUTSTOP, A-5
 INERTIA, A-13
 INNER. VEL, C-2
 INHIBIT, 3-25
 initialization of environment, 2-29,
 2-72 to 2-74, 3-45
 initialization of simulator, 1-5, 3-36,
 3-39, 3-45
 input bit assignments in COMMBLK,
 B-6
 input card image storage, B-3
 input cards for environment, 3-55 to 3-107
 input channel discontinuities, 3-100
 INPUTPRG program, 2-20, 2-21, 2-22,
 2-29, 2-73, 3-49, 3-52, 3-55,
 B-1, B-3
 instruction patches, 3-10, 3-26
 instruction simulator, 1-3, 1-4, 2-6,
 2-9 to 2-12, 2-75
 interpretive trace, 3-22, 3-23, 5-1, 5-19
 interpretive trace edits, 5-18 to 5-19
 ITRACE, 3-22, 3-23, 5-1, 5-19
 ITRSTOP, 3-10, 3-23
 IU, 3-90
 IXX, IYY, IZZ, A-13
 job control cards, 3-2
 job step, 3-2
 job step control card, 3-34, 3-35, 3-38,
 3-42
 KEYSTRK, 3-10, 3-23
 keystroke buffer, B-9
 keystrokes, 3-87
 KEYSTROKES FOLLOW, 3-87
 labels for MARSROT tapes, 5-3, 5-4
 landing radar, 3-80
 LAT-GD, LONG, C-2

INDEX

- latitude, longitude computation, 2-31
- LATT, LONGT, 3-55
- L conditional, 3-32, 5-7
- LC conditional, 3-33, 5-8
- LEMHCTRL, 3-89
- LEM program 2-15, 2-18, 2-23, 2-48
to 2-55, 3-52, 3-66, B-3, B-4
- LEMRADAR, 2-61, 3-90
- LEMTHROT, 3-89
- LEM3.NOMINAL, 2-25
- LEQ conditional, 3-32, 5-7
- LEQC conditional, 3-33, 5-8
- L frame, 2-80
- libration of moon, 2-31, 2-32
- liftoff, 3-59
- LM handcontrol, 3-89
- LM mission programmer, 3-99
- LM model, 3-67
- LMP1 program, 2-15, 2-16, 2-24,
2-67 to 2-68, 3-52, 3-99, B-3
- LM reference system, 2-83
- LOAD FILE cards, 3-41, 3-42, 3-43
- local horizontal frame, 2-81
- local vertical frame, 2-80
- location field, 3-7
- LRADAR program, 2-23, 3-80, B-3
- LRHC RATE, 3-89
- lunar module, 2-48 to 2-55
- lunar motion, 2-31, 2-32
- LVCBLK2.FILE 205, 2-25
- LVCBLK2 program, 2-18, 2-23, 2-55,
3-52, 3-72, B-3
- LVCRCs, 3-90
- MAC data cards, 2-29
- MACEDIT, 3-36, 5-1, 5-12 to 5-17, 5-23
- MACH, A-8
- MAC input cards, 3-37, 3-38, 3-42
- MAC input facilities, 5-12 to 5-17
- MACPATCH, 3-50
- MACRUN, 3-47, 3-48
- MAC run step, 3-3
- MACSIM, 3-36, 3-38, 3-39, 3-42, 3-43
- MAIN, 3-87
- manned missions, 2-17 to 2-19
- MARK, 3-88
- MARSROT, 3-6, 3-18, 3-22, 3-27, 5-1
5-3, 5-4, 5-7, 5-12, 5-20
- MASS, C-5
- MASSVEH, A-11
- matrices, 2-84, 2-85, 2-86, 2-87
- MAXTIME, 3-10, 3-24
- METHOD, A-9
- MINIMPLS, 3-89
- minimum impulse, 3-89
- mission specific parameter files, B-4
- mission time, C-1
- MODE, 3-85
- momentum plane, 2-81
- MORE, 3-19, 5-6
- MPAC, 5-18
- MSG, 3-10, 3-24
- NAV, 3-87
- NOABORT, 3-86
- NOGOSEQ, 3-10, 3-24
- noise sigmas, B-8
- NOPRINT, 3-10, 3-25
- NOPROG cards, 3-37, 3-38, 3-42
- NOUN, 3-85, 3-87

INDEX

OCT, 3-10, 3-26
 OCTAL, 3-10, 3-26
 OCTAL edit format, 5-10
 OCTFILE, 5-14
 ONPRINT, 3-10, 3-25, 3-26
 OP codes, 3-10, 3-26
 OP code field, 3-7
 optical libration of moon, 2-32
 optics, 2-58
 OPTICSB2 program, 2-18, 2-23, 2-58, 3-52, 3-77, B-3
 OPTMODE, 3-88
 OR condition, 3-32
 order of data cards, 3-50
 ORIGIN, 3-57, A-3, A-7
 output bits assignments in COMMBLK, B-7
 OUTX, 3-25
 overflow condition, 3-33

 parameter files, 2-24, 2-25, 2-72, 3-43, 5-14, 5-15, 5-16, B-1, B-4
 peek function, 2-29
 PERIGEE, C-6
 P frame, 2-81
 PID, FID, 5-12, 5-13, 5-15, 5-16, 5-24
 PIPA errors, 3-75
 PIPTSTOP, A-7
 PITCH, YAW, ROLL, 2-81, 2-82, 2-83, 2-85, 2-86, 2-87, 3-64, 3-67, 3-72, A-11, C-4
 PLOTSW, 5-17
 plotting, B-1, B-2
 plot variables for CSM, 3-63
 plot variables for IMU, 3-75

 plot variables for LEM, 3-67
 position, velocity, A-7
 powered flight, 2-33
 preferred reference frame, 2-81
 prelaunch on earth, 2-35, 2-36
 prelaunch on moon, 2-37
 PRELUNCH program, 2-15, 2-18, 2-23, 2-35 to 2-37, 2-80, 3-52, 3-59, B-3
 print control, 3-55
 procedures, 3-2, 3-36, 3-38
 PRNTCTRL, 3-55
 PROCEED, 3-90
 PROGRAM ALARM, 3-85
 program source card, 3-6, 3-34, 3-35, 3-38, 3-42

 Q, QDOT, A-8
 radar, 2-59
 RCS failures, 2-67
 RCSFUEL, A-11
 reaction control system, 2-41, 2-42
 recipes, 2-13, B-8, B-10
 reentry, 2-33
 reference inertial frame, 2-30, 2-78
 RELEASE DISPLAY, 3-85
 RELEASE MARSROT, 5-5, 5-25
 remarks field, 3-7, 3-32
 rendezvous radar, 2-59
 REPLACE, 3-10, 3-26
 restarts, 4-2
 RESUME, 3-25
 return requests, B-5
 RI, A-7
 RI-TOGO, C-7
 RMODE, 3-103

INDEX

R_M , R_S , A-3
 rollbacks, 3-27, 4-1 to 4-5
 RRADAR program, 2-18, 2-23, 2-59 to 2-61, 3-52, 3-78, B-3
 RT, VT, A-3
 run decks, 1-2, 1-5, 2-2, 2-5, 3-35, 3-38, 3-42, 3-47
 RUPT, 3-25
 R1, R2, R3, 3-85

 SATURN.DATA206, 2-25
 SATURN.FILE 205, 2-25
 SATURN program, 2-15, 2-18, 2-23, 2-38 to 2-40, 3-52, 3-60, B-3, B-4
 SAVE MARSROT, 5-5, 5-25
 SCSMODE, 3-88
 SCS05GSW, 3-88
 service propulsion system, 2-42, 2-43
 SHAFT, A-4
 SIMEPHEM.F6667, 2-25
 SIMFLIGHT, 3-56
 SIMSETUP program, 2-20, 2-21, 2-24, 2-70, 3-52, 3-101
 SIMULATE, 3-4, 3-5, 3-34, 3-35, 3-38 3-42
 simulation set up, 3-101 to 3-104
 SINGLE edit format, 5-10
 slosh, 2-44, 2-53
 SNAP, 3-10, 3-27, 4-1
 SNAPSHOT, 3-4, 4-1, 4-2
 special requests, 2-10, 2-11, 3-7 to 3-30
 special request conditions, 3-31 to 3-33
 SPFILE, 5-13
 SPSFUEL, A-11
 staging forces, 2-53
 STANDBY, 3-10, 3-27
 START, 3-10, 3-28, 4-4
 STARVI, A-2
 STOP, 3-10, 3-28
 subprograms, 3-91 to 3-93
 subroutine function, 2-66
 subset cards for edits, 5-6
 subset edit cards, 5-10
 summary call for edits, 5-16
 summary prints, 3-20
 sway on launch pad, 2-36, 3-59
 synchronization of time, 3-28
 S4BSEP, 3-88

 tape labels for MARSROT, 5-3, 5-4
 target latitude, longitude, 3-55
 TARGET program, 2-18, 2-23, 2-62, 3-52, 3-81, B-3
 TCSGO, 3-10, 3-28
 TELEM, 3-10, 3-29
 TEMPCHK, 3-10, 3-29
 TEXPIRE, B-10
 TFF, C-6
 THRMODEL, 2-40
 throttle, 3-89
 thrust, A-12
 THRUSTSW, 3-90
 T_{IL} , 2-84, A-4
 TIME (astronaut action), 2-64, 3-82
 time in simulation, 2-78, 3-12, 3-28
 time loss, 3-29
 time selection for edits, 5-20, 5-21
 T_{IN} , 2-84, A-10

INDEX

TLIFTOFF, 3-59, B-8
 TLOSS, 3-10, 3-29
 TRACE, 3-10, 3-29, 3-30
 transformation matrices, 2-84, 2-85,
 2-86, 2-87, A-4, A-5, A-10
 TRANSCTL, 3-89
 translation control, 3-89
 TRCON, A-3
 TRIMALPH, A-5
 TRUNNION, A-4
 TRSTOP, 3-10, 3-30
 T_{SI} , 2-84, 2-86, A-5
 T_{SN} , 2-84, A-5
 TZERO, 3-28, 3-57, B-8

 UNEQ conditional, 3-32, 5-7
 UNEQC conditional, 3-33, 5-8
 UNIV.EPHM6768, 2-25
 UNIV.EPHM6869, 2-25
 universe inertial frame, 2-78
 UNIVERSE program, 2-15, 2-16, 2-22,
 2-30 to 2-34, 2-78, 3-52, 3-57,
 B-3, B-4
 unmanned missions, 2-14 to 2-17
 UNUSED, 3-10, 3-30
 UPTTEL FOLLOWS, 3-10, 3-30, 3-87
 UPTTELSW, 3-88

 VAC area, 5-14
 VACFILE, 5-14
 VATM, A-8
 vehicle flag, B-8
 vehicle frame, 2-82
 vehicle orientation, A-11
 vehicle plot flag, B-2
 VERB, 3-85, 3-87
 VERIFY, 2-64, 3-83, 3-86
 V_I , A-8
 VIRGIN, A-11
 VMODE, 3-103

 wait, 2-64, 3-83
 w_n , A-10
 xrate, yrate, zrate, C-4
 1GOSEQ, 3-10, 3-21
 2BCADR, 3-10, 3-15
 2CADR, 3-10, 3-15
 2DEC, 2DEC *, 3-10, 3-18
 2FCADR, 3-10, 3-21
 2SCFILE, 5-14
 1800 input deck, 3-5
 206COEAR.LYBY100, 2-25

DISTRIBUTION LIST

Internal

M. Adams	T. M. Lawton (MIT/MSC)
J. Alekshun	D. Lickly
R. Battin	F. Martin
P. Bowditch/F. Siraco	R. McKern
G. Cherry	James Miller (50)
E. Copps (100)	John Miller
S. Copps	P. Mimno
R. Crisp	J. Nevins
J. Dahlen	J. Nugent
J. Dunbar	R. Ragan
J. B. Feldman	G. Schmidt
P. Felleman	W. Schmidt (MIT/FOD)
J. Flanders	R. Scholten
J. Gilmore	N. Sears
F. Grant	J. Shillingford
Eldon Hall	G. Silver (MIT/KSC)
M. Hamilton	W. Stameris
T. Hemker (MIT/NAR)	J. Suomala
D. Hoag	J. Sutherland
F. Houston	W. Tanner
L. B. Johnson	M. Trageser
M. Johnston	R. Weatherbee
J. Kernan	R. White
A. Kosmala	R. Woodbury
A. Laats	W. Wrigley
L. Larson	Apollo Library (2)
R. Larson	MIT/IL Library (6)

EXTERNAL DISTRIBUTION

NASA/RASPO (1)
AC Electronics (3)
Kollsman (2)
Raytheon (2)
Capt. M. Jensen (AFSC/MIT) (1)

MSC: (21 & 1R)

National Aeronautics and Space Administration
Manned Spacecraft Center
Houston, Texas 77058
Attn: Apollo Document Distribution Office (PA2) (18 & 1R)
C. Frasier (2)
T. Gibson (1)

KSC: (1R)

National Aeronautics and Space Administration
J. F. Kennedy Space Center
J. F. Kennedy Space Center, Florida 32899
Attn: Technical Document Control Office

LRC: (2)

National Aeronautics and Space Administration
Langley Research Center
Hampton, Virginia
Attn: Mr. A. T. Mattson

GAEC: (3 & 1R)

Grumman Aircraft Engineering Corporation
Data Operations and Services, Plant 25
Bethpage, Long Island, New York
Attn: Mr. E. Stern

NAA: (18 & 1R)

North American Aviation, Inc.
Space and Information Systems Division
12214 Lakewood Boulevard
Downey, California
Attn: Apollo Data Requirements
Dept. 096-340, Bldg. 3, CA 99

NAA RASPO: (1)

NASA Resident Apollo Spacecraft Program Office
North American Aviation, Inc.
Space and Information System Division
Downey, California 90241

AC RASPO: (1)

National Aeronautics and Space Administration
Resident Apollo Spacecraft Program Officer
Dept. 32-31
AC Electronics Division of General Motors
Milwaukee 1, Wisconsin
Attn: Mr. W. Swingle