SPERRY RAND

SPACE SHUTTLE
DATA MANAGEMENT SYSTEM
REQUIREMENTS ANALYSIS

FACILITY FORM 602

N71-38687
(ACCESSION NUMBER)

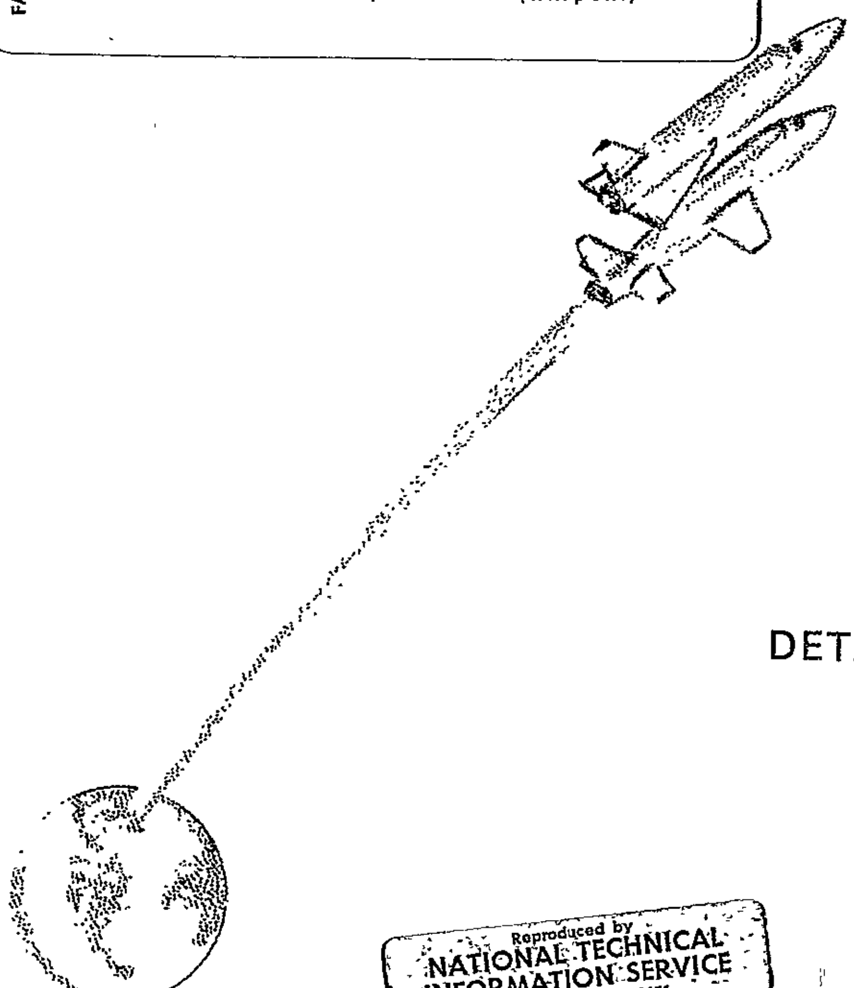
581
(PAGES)

CR-120093
(NASA CR OR TMX OR AD NUMBER)

(THRU)

63
(CODE)

31
(CATEGORY)



FINAL REPORT

CR-120093

VOLUME II
DETAIL REQUIREMENTS

Reproduced by
NATIONAL TECHNICAL
INFORMATION SERVICE
Springfield, Va. 22151

UNIVAC
DEFENSE SYSTEMS DIVISION
St. Paul, Minnesota

REPORT NO
PX 6551-2

UNIVAC

SPACE SHUTTLE BOOSTER DATA MANAGEMENT SYSTEM (DMS) REQUIREMENTS ANALYSIS

Final Report

Volume II

C. L. Lutes, Principal Investigator
K. M. Hoglund, Program Manager

Advanced Systems Requirements
Defense Systems Division
St. Paul, Minnesota

PREPARED FOR

George C. Marshall Space Flight Center
Final Report on
NAS8-30186, Mod 1

24 September 1971

SPERRY RAND

UDI-2536-1 (REV 5/69)

Acknowledgements

Univac performed the Space Shuttle Booster Data Management System Study (NAS8-30186) under the technical direction of Dr. H. Kerner and Mr. Bobby Hodges of NASA's Computational Laboratory, George C. Marshall Space Flight Center, Huntsville, Alabama. The study was performed by Univac's Defense System Division, St. Paul, Minnesota under the Program Management of Mr. K. M. Hoglund. The major technical contributions at Univac were made by Messrs. C. L. Lutes, L. A. Snider and R. E. Tyler. Contributions to the flight control and propulsion system requirements were made by Mr. Steve Osder of Sperry Flight System, Phoenix, Arizona.

Distribution

NASA - MSFC

Code	Copies
A&TS - PR - M	1
S&TS - MS - IL	1
A&TS - T	1
A&TS - MS - I	2
S&E - COMP - RD	6
S&E - COMP - M	1

Overview of Final Report

The final report on the Space Shuttle Booster Data Management System (DMS) for NASA contract NAS8-30186 is in two volumes. The major items covered in Volume I are a discussion of the study results, a description of the booster mission, and a description of the functional requirements of the assumed avionics system. Volume II is devoted to subsystem interface description, subsystem computational requirements and a description of an analysis program generated and used during the study.

The reader should first read section 1 for a summary of the study, the major study conclusions and booster mission description. Section 2 provides a detailed discussion of the DMS configuration and sizing analysis. The reader interested in the detailed description of a particular subsystem is directed to the functional requirements in section 3, the interface requirements in section 4 and the computational requirements in section 5. Section 6 presents evaluation program, reliability analysis and configuration mechanization details. The table of contents will direct the reader to the particular subsystem of interest.

Volume II

TABLE OF CONTENTS

Table on Contents (Cont.)

Volume II

LIST OF FIGURES

List of Figures (Cont.)

List of Figures (Cont.)

List of Figures (Cont.)

4.0 BOOSTER SYSTEMS INTERFACE DESCRIPTIONS

The development of the DMS requires detailed definitions of all subsystems it interfaces with. This section is concerned with the data flow between the DMS and other subsystems while section 5 is concerned with the action of the DMS upon this data. The method of transmission of data between the DMS and other subsystems is highly dependent upon the DMS configuration. There are certain interface parameters which can be defined independent of the DMS configuration. Data transferred between the DMS and other subsystem equipment will be data representing the numerical magnitude of a parameter or coded data indicating the state or an action, or the command of a state or action within subsystem equipment of the DMS. Discrete inputs and outputs are a special form of coded data. Numerical values may originate in or be required by subsystem equipment as analog voltages, modulated ac voltages, or special encoded parallel or serial digital data. Much of this data will require conversion before it can be used internally in any DMS computer. The electronics required for conversion will be assumed part of the DMS and not part of the subsystem equipment. If the DMS is a centralized system then all interface data will be through the data bus system and conversions will occur at the subsystem end of the data bus. If the DMS is decentralized with local computers at the major subsystem elements then special input/output conversion equipment could be implemented as part of the local computer. The requirements to interface directly with a digital computer will be different from the requirements to interface with the data bus system.

In this section those parameters delineated for each interface data element transmitting numerical values are:

range- This is the difference between the minimum and maximum values that the transmitted parameter may portray, e.g., whole value gyro outputs will have minimum and maximum values of -180 and +180 degrees respectively producing a range of 360 degrees.

resolution - This is the size of the smallest change in the parameter which must be transmitted. This is different than accuracy which is a measurement of how closely the transmitted data represents the parameter being measured. For example a rate gyro could have a scale factor error causing measurement inaccuracies of $1^{\circ}/s$ at large rates but have a resolution requirement of $.1^{\circ}/s$. The $1^{\circ}/s$ scale factor inaccuracy would cause a minor variation of the attitude control loop short-period time constant while if the resolution were increased to $1^{\circ}/s$ the attitude control loop would develop a limit cycle oscillation of possibly intolerable magnitude. Resolution is a measurement of the least significant bit value which must be transmitted.

sampling rate - This is a measurement of the rate at which the data must be updated either by the DMS or for use by the DMS. This measurement is dependent upon the intended use of the data. Data items may be used by more than one system and their rate requirements may vary with mission phase. For example the attitude gyros are used by both the strapdown navigation computations and the attitude control system. The sampling rate requirements will be determined from that system having the highest rate requirements. An example of data rate requirements restricted to particular mission phases occurs with the valve commands to the reaction jet system which is used only during coast and the early portion of reentry.

Data Subsystem source or destination - The orginating source or final destination of all data indicates the type of conversion electronics required.

The parameters delineated for each interface data element transmitting coded states is:

Number of possible subsystem states

This is the information required to determine the number of binary bits needed to transmit the status of a subsystem. In any subsystem there is the possibility for some modes to operate only exclusively of other modes, e.g., the TACAN Distance Measuring Equipment (DME) cannot be in a track mode simultaneously with a search mode. There are other modes which may operate independent of one another, e.g., the TACAN may or may not be delivering good bearing data independent of the DMS operation.

Required response time

A change in subsystem status requests an action by the DMS. The action required of the DMS must occur within a set time after the status change. This set time results in specifying either the rate at which status data must be sampled or in some instances specifies that the status must be transmitted on a special priority interrupt line.

This section delineates all of the input/output requirements of the

Structures
Propulsion
Electrical power generation and distribution
Navigation and guidance
Flight control
Operations management

systems. The parameters listed for each input/output data item are those listed above.

4.1 STRUCTURES

Structures include performance monitoring, landing gear deployment, and separation control and monitoring.

4.1.1 PERFORMANCE MONITORING

Structural performance is monitored by vibration, stress, and temperature sensors. This data will be recorded during the mission flight phases for post flight analysis and data reduction. The DMS will be responsible for:

1. Sensor checkout during prelaunch
2. Control of data recording during flight
3. Post flight data reduction

During flight the data generated by the majority of these sensors will not interface with the DMS but only with the recorder through the data bus system. Recording of the sensor outputs are of primary interest during periods of large sensor output values. The data rate requirements for the transmission of vibration data to the recorder will be very high if the number of sensors is large. It is possible for some DMS configurations to have large data rate requirements on the main data bus system which will make the addition of vibration data to the main data bus system prohibitive. Since the recording of vibration data is not critical to the mission (except initial flight test missions where mission objectives are the gathering of vehicle data) redundancy requirements do not apply. For these reasons it is probable that a separate data bus system will be constructed for the purposes of collecting and recording special vehicle data. Such a configuration has the additional advantage of being a nearly independent subsystem with a minimum interface with other systems so that it can easily be removed if its use is found no longer necessary.

Figure 4-1 shows the configuration adapted for this study for the recording of special vehicle data as applied to the vibration sensors. Digital rather than analog recording is assumed because it allows for the use of identical equipment as that required by the DMS data bus system, it exhibits better noise rejection than analog multiplexing, allows for the insertion of sensor identification codes, and provides compatability with data reduction equipment. Each vibration sensor output is connected directly to an analog-to digital converter. The converter output is connected through switching and control electronics to the recorder data bus. There are several methods of mechanizing the recorder data bus, however these mechanization details will not significantly effect the DMS. Data bus addressing and control electronics will communicate with the individual vibration sensor A/D converter and data bus interface electronics through the recorder data bus to command the time shared multiplexing of the converted sensor outputs onto the recorder data bus. The data bus addressing and control electronics is commanded from the main data bus through data bus interface electronics. The commands which can be issued to the data bus addressing and control electronics are

- Standby
- Record (slow speed
- Record (fast speed)
- Test Record Fast
- Test Record Slow

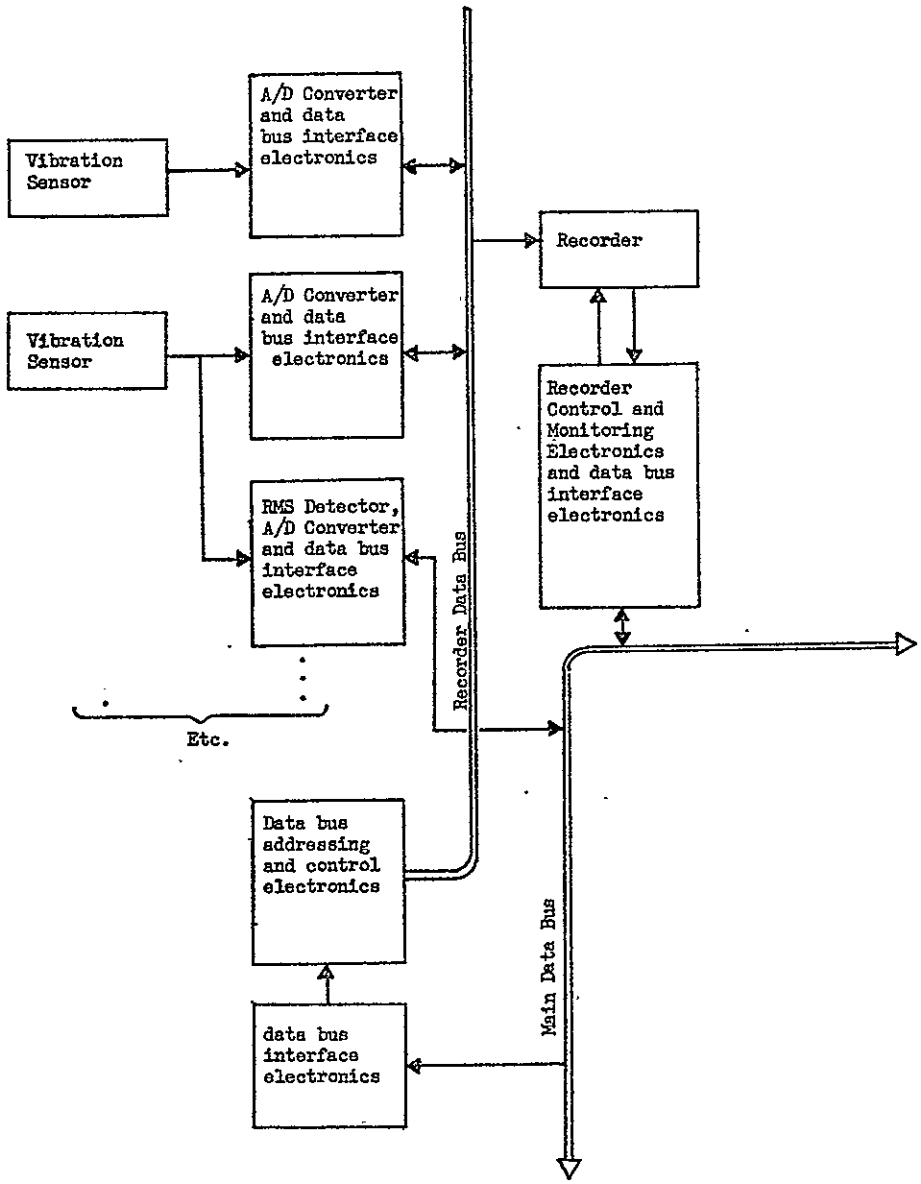


Figure 4-1 Vibration Sensor Recording Block Diagram

These commands will originate in the DMS computers and will be issued as a function of mission phase and required recording rate as determined from sensor measurements. The commands will not be issued at a set iteration rate but only as required.

Figure 4-2 shows typical acceleration power spectral density vibration levels for a liquid fueled booster. The fast recording speed should be capable of accumulating vibration data out to 1300 Hz. which will require at a minimum 2600 samples per second. The range required for each sensor output is ±25g's and a resolution of .05g's. The slow recording speed should be capable of covering the frequency range where the majority of subsystem equipment resonances occur, 30 to 150Hz which will require a minimum sampling rate of 300 samples per second. A convenient ratio between fast and slow recording speeds would be 8:1. For the purposes of this study it will be assumed that there are a total of 15 vibration sensors to be recorded which will be sampled at 3200 samples per second for high speed recording and 400 samples per second for slow speed recording. Each sampling will result in a 10 bit data word. Every 16th sample will be used to accumulate non-vibration data such as temperature and stress sensor outputs which are sampled at a much lower rate. The high speed data rate requirements are then 512,000 bits per second and the low speed rates are 64,000 bits per second.

The determination of recording speed is dependent upon several measurements. One of thse measurements is vibration magnitude. Several of the vibration sensors (for this study three are assumed) will have an interface directly with the main data bus in addition to the recorder data bus. When interfacing with the main data bus the sensor output will be converted to a pseudo RMS value by full wave rectification and filtering. This RMS value will be converted

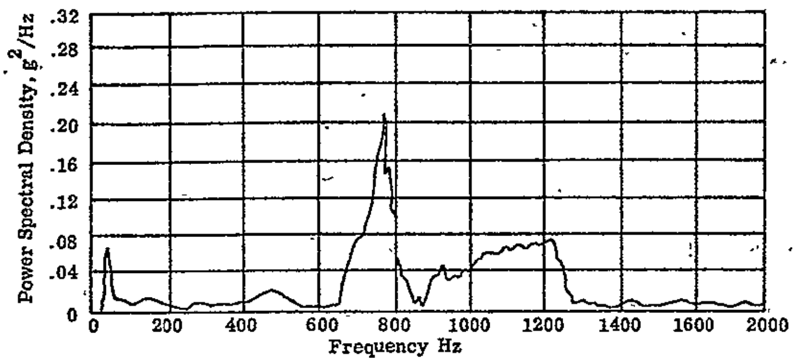


Figure 4-2 Vibration Power Spectral Density[1]

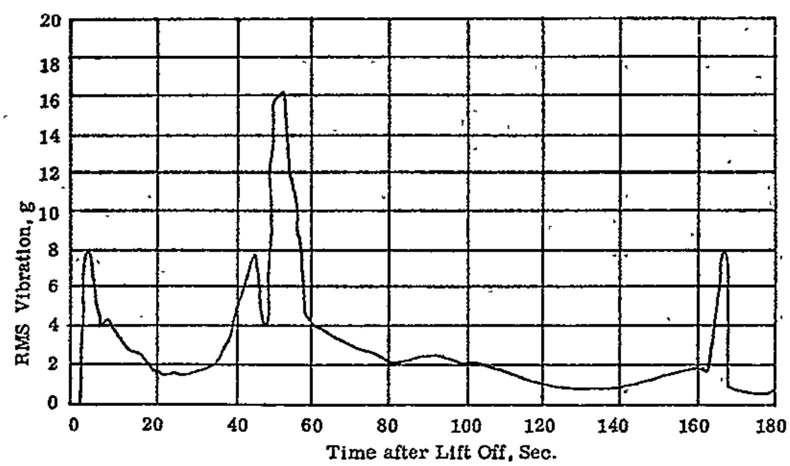


Figure 4-3 Boost Vibration History[1]

to a digital number and transmitted to the DMS computers upon request of the DMS. Figure 4-3 shows a typical RMS vibration history of a liquid fueled booster[1]. Collection of vibration data will be of most interest during periods of large vibration activity. One criteria for determining when high speed recording should occur is when vibration magnitudes exceed a fixed threshold value. If this threshold value could be predetermined early in the space shuttle design phase then the A/D converter in the interface electronics between the vibration sensors and main data bus could consist of a threshold detector producing a single bit conversion, i.e., discrete output. In the interest of flexibility it is assumed that an A/D conversion with a range of 0-16 g's and a resolution of .5 resulting in a 5 bit output word will be used.

When DMS computer issues commands to control the recorder data bus it must also issue commands to control the recorder. The commands which can be issued to the recorder from the DMS for use in this application are:

- Standby
- Record Fast
- Record Slow
- Rewind
- Playback

When ever commands for recording are issued to the data bus addressing and control electonics, commands must also be issued to the recorder so that the data bus and recorder are operating at compatable speeds. These commands are thus also issued as required and not at a fixed iteration rate.

Checkout of this system requires first the recording of data and then the playback and analysis of the recorded data. The major events necessary for checkout are:

on the booster that would cause at least some low magnitude sensor output, such as fueling the booster main propulsion tanks or mating the orbiter to the booster.

<u>Playback</u> Playback compatability with the DMS is primarily contingent upon the ability to read the recorded data at a rate slow enough to allow the computer to process the data. The usual method of providing speed compatability is to record data in records which are short enough to be read into memory in their entirety. After each record is read it is processed with intermediate results stored in a reduced form or outputted. The recorder is stopped after reading each record and waits for the computer to finish its computation and command a new record to be read before the recorder is started again. This requires a gap in the data on the tape to allow room for stopping and starting the tape. To generate a true gap during the recording process requires a loss in data during the gap generation time. A method of writing continous data on the tape and yet allow reading operations to occur as if data was written in records spaced with stop/start gaps is to insert an end of record mark after each record when recording and then immediately start the next record leaving no gaps. Assuming initially the tape is stopped in the middle of a record a read operation is performed after issuance of a read command by ignoring all data from the read electronics until the first end of record mark is found at which time data is read and transmitted to the DMS

until the next end of record mark is encountered at which time the recorder is stopped. Before reading the next record the recorder is commanded to read in reverse until an end of record mark is encountered and the stops. The issuance of the reverse command can be done automatically by the recorder electronics after the recorder stops at the completion of each read command.

As described above data will be recorded at two different speeds. A finite time will be required during the record operation to change the recorder speed during which time no data will be recorded. These gaps will be marked with an extra end of record mark at the beginning of the gap and an end of record mark at the end of the gap. The extra end of record mark at the beginning of the gap will inhibit the reverse read operation associated with the reading of the data record just prior to the gap. Data recorded at high or low speed will appear the same on the tape. When the tape speed is reduced for low speed recording the pulse width of each data bit will be increased so that each data bit occupies the same linear tape length. All data can thus be read at the same speed. The DMS program for reducing the recorded data will required the speed at which the data was recorded. The recording speed can be indicated by special codes in the end of record mark.

The temperature and stress sensors are treated primarily in the same manner as the vibration sensors. The outputs from these sensors is sampled at a slower rate than the vibration sensors and the data recorded interspersed with the vibration data. The actual recording rates for each sensor will be a function of the number of sensors and the recorder capabilities. Select vibration sensor outputs converted to RMS values along with select temperature and stress sensor outputs will be used by the DMS in determining the speed at which data is to be recorded. For the purposes of this study the following assumptions will be made concerning data flow required for the recording of structural sensor data.

<u>Vibration Sensors</u>

Number to be recorded	15
Range for recorder	-25g to +25g
Resolution for recorder	.05g
Word length for recorder	10 bits
Fast recording samples per sensor per second	3200
Slow recording samples per sensor per second	400
RMS outputs to the DMS	3
RMS range	0g to 16g
RMS resolution	.5g
RMS samples per sensor per second	10
RMS word length to DMS	5 bits

<u>Temperature Sensors</u>

Number to be recorded	15
Range for recorder	0^{o}F to 2500^{o}F
Resolution for recorder	20^{o}F
Word length for recorder	7 bits
Fast recording samples per sensor per second	40
Slow recording samples per sensor per second	5
Sensor outputs to DMS	3
Range to DMS	0^{o}F to 2500^{o}F
Resolution to DMS	20^{o}F
Word length to DMS	7 bits
Samples per sensor per second to DMS	10

Stress Sensors

Number to be recorded	5
Range for recorder*	
Resolution for recorder*	
Word length for recorder	7 bits
Fast recording samples per sensor per second	40
Slow recording samples per sensor per second	5
Sensor outputs to DMS	3
Word length to DMS	7 bits
Samples per sensor per second to DMS	10

DMS/Recorder Data Bus Controller

Command word length to Data Bus Controller	3 bits
Command word rate to Data Bus Controller	as required

DMS/Structural Recorder Interface

Command word length to Data Recorder	3 bits
Command word rate to Data Recorder	as required
Record length from recorder	3210 bits

The record assumed here is composed of 321, 10 bit words which are divided into 20, 16 word fields plus 1 end of record word. Each field contains 15 vibration sensor samples and 1 temperature, stress, or other sensor sample. In order to meet the sampling rate requirements of the temperature and stress sensors eacl sensor must be sampled only every fourth record. The end of record word will contain codes indicating whether temperature and stress sensors are recorded in that record.

*Note: Range and resolution of the stress sensors is dependent upon the sensor chosen and the method and location of mounting. It is assumed that measurements should be made to 1% of full scale.

4.1.2 Landing Gear System

The space shuttle booster has three landing gears; a nose gear, a right main gear, and a left main gear. The landing gear are used during the landing phases of either a normal or ferry mission and during the takeoff phase during a ferry mission. Prelaunch ground checkout of the landing gear system must occur after the vehicle is lifted for placement in a vertical position. The signal interface with the DMS for each landing gear is assumed identical. Figure 4-4 shows this signal interface for the nose wheel landing gear system. The primary power source for operating the landing gear system is hydraulic. Modern aircraft sometimes employ a pneumatic supply system for backup, however, with redundant hydraulic systems on the space shuttle it is assumed that a backup pneumatic system will not be required. Each landing gear when stowed is covered by a door to protect the gear from the environment and to give the booster a smooth aerodynamic shape. The doors are automatically locked in position when fully open or fully closed. The DMS can monitor the locked condition and must give an unlock command before the doors can be opened or closed. After the doors are unlocked an open or closed command must be issued by the DMS. The DMS checks door operation by monitoring the door position and the hydraulic actuator hydraulic fluid temperature and pressure. Each landing gear is lowered or raised by the same procedure as the doors are opened or closed, i.e., an unlock command is issued and then a raise or lower command. Monitoring of the gear up and locked or the gear down and locked, the gear position, and the hydraulic actuator fluid temperature and pressure is provided. It is assumed for this study that each landing gear will have four wheels. The four wheels are mounted on a bogie which must be rotated into a stowed position before the landing gear is raised,

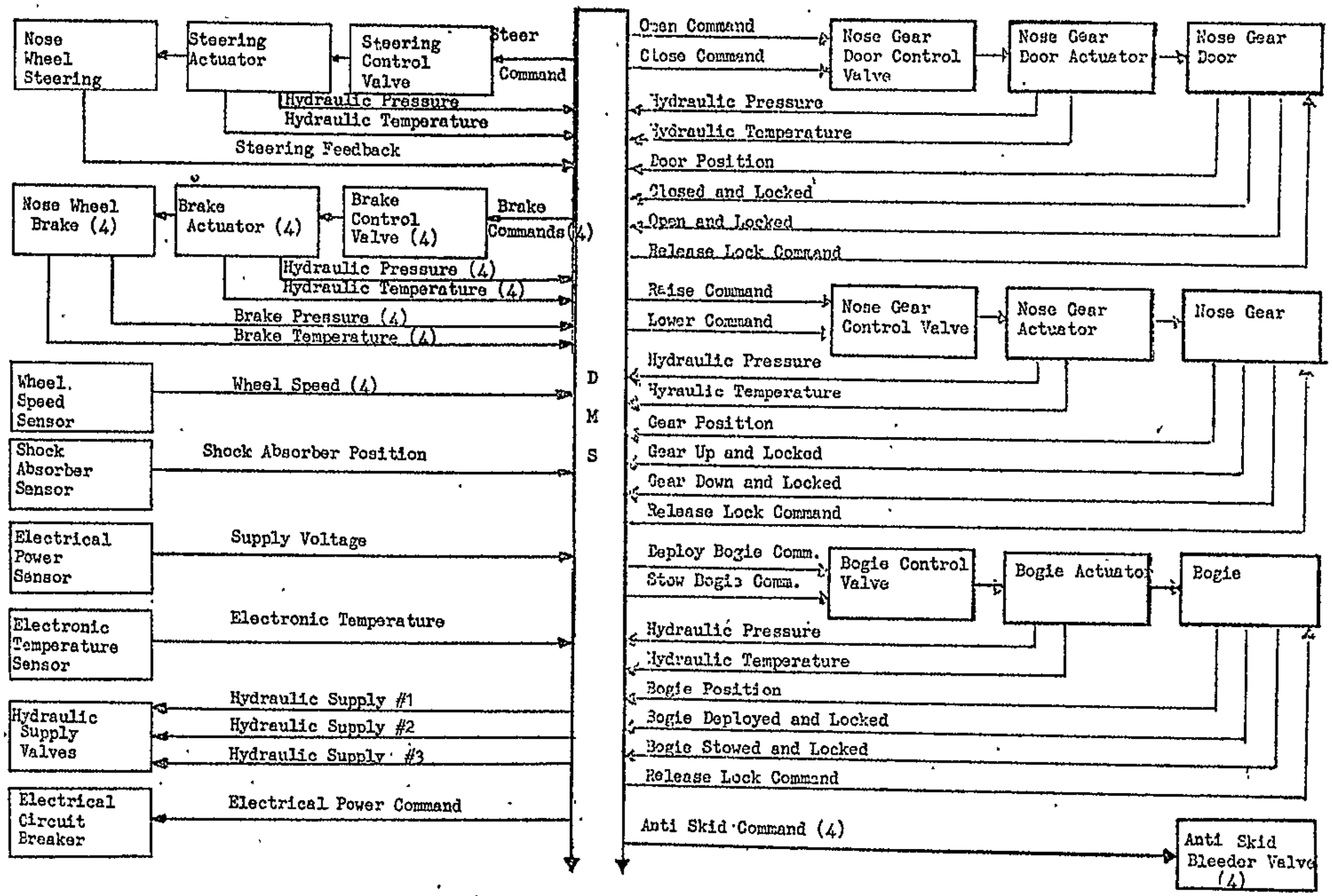


Figure 4-4 Nose Wheel Landing Gear System

and rotated into the operational position after the landing gear is lowered. Before issuing the stow or deploy bogie command a release lock command must be issued. The DMS is provided with signals indicating a locked condition in the stowed or deployed position, the bogie position, and hydraulic actuator fluid temperature and pressure. Each landing gear is capable of being steered. Nose wheel steering is used during the taxi operation. Main gear steering is employed to reduce in magnitude or eliminate the need for the decrab manuever. The DMS is provided with the capability of issuing a steering command and receives signals for monitoring of the steering position and hydraulic actuator fluid temperature and pressure. Each wheel has its own brake which is activated by the DMS. Brake temperature and pressure plus hydrualic actuation fluid temperature and pressure measurements are sent to the DMS. Each wheel has an accurate tachometer attached which measures the rotational speed of each wheel. The speed of each wheel is used to verify touchdown, to activate the anti-skid system, and to determine taxi velocity. Each gear has a shock absorber with its position monitored for the purpose of determining touchdown. If the shock absorber bottoms during landing a warning is displayed indicating a possible overstress of the vehicle during landing. The electrical supply voltage and temperature of the landing gear electronics is monitored. The DMS has control over determining which redundant hydraulic system supplies the landing gear system and can monitor the hydraulic supply pressure to the landing gear system. Each brake has an associate anti-skid bypass valve to remove braking power from any wheel which has stopped while the others are rotating or is slowing more rapidly than the others. Figure 4-5 is a complete list of all interface signals between the DMS and the Landing Gear System.

SIGNAL		SOURCE	TYPE	RANGE	RESOLU-TION	WORD LGTH (BITS)	RATE /SEC
<u>Doors, (Nose, Left and Right Gear)</u>							
Open Command	(3)	DMS	DIS			1	AR
Close Command	(3)	DMS	DIS			1	AR
Hydraulic Actuator Pressure	(3)	LGS	AN	0-3000psi	25psi	1	4
Hydraulic Actuator Temp	(3)	LGS	AN	0-1000°F	20°F	6	2
Position	(3)	LGS	AN	0-90°	3°	5	2
Closed and Locked	(3)	LGS	DIS			1	AR
Open and Locked	(3)	LGS	DIS			1	AR
Release Lock Command	(3)	DMS	DIS			1	AR
<u>Gear, (Nose, Left and Right)</u>							
Raise Command	(3)	DMS	DIS			1	AR
Lower Command	(3)	DMS	DIS			1	AR
Hydraulic Actuator Pressure	(3)	LGS	AN	0-3000psi	25psi	7	4
Hydraulic Actuator Temp	(3)	LGS	AN	0-1000°F	20°F	6	2
Lowering Position	(3)	LGS	AN	0-90°	3°	5	2
Up and Locked	(3)	LGS	DIS			1	AR
Down and Locked	(3)	LGS	DIS			1	AR
Release Lock Command	(3)	DMS	DIS			1	AR
<u>Bogie (Nose, Left and Right)</u>							
Deploy Command	(3)	DMS	DIS			1	AR
Stow Command	(3)	DMS	DIS			1	AR
Hydraulic Actuator Pressure	(3)	LGS	AN	0-3000psi	25psi	7	4
Hydraulic Actuator Temp	(3)	LGS	AN	0-1000°F	20°F	6	2
Position	(3)	LGS	AN	0-90°	3°	5	2
Deployed and Locked	(3)	LGS	DIS			1	AR
Stored and Locked	(3)	LGS	DIS			1	AR
Release Bogie Lock	(3)	DMS	DIS			1	AR
<u>Steering (Nose, Left and Right)</u>							
Nose Wheel Steering	(1)	DMS	AN	-90° to +90°	.25°	10	8
Main Gear Steering	(2)	DMS	AN	-20° to +20°	.25°	8	4
Hydraulic Actuator Pressure	(3)	LGS	AN	0-3000psi	25psi	7	4
Hydraulic Actuator Temp.	(3)	LGS	AN	0-1000°F	20°F	6	2
Nose Wheel Feedback	(1)	LGS	AN	-90° to +98°	.25°	10	8
Main Gear Feedback	(2)	LGS	AN	-20° to 20°	.25°	8	4
<u>Brakes (4 each on Nose, Left and Right Gear)</u>							
Command	(12)	DMS	AN	0-100%	1%	7	4
Hydraulic Actuator Pressure	(12)	LGS	AN	0-3000psi	25psi	7	4
Hydraulic Actuator Temp	(12)	LGS	AN	0-1000°F	20°F	5	2
Pressure	(12)	LGS	AN	0-1000%	1%	7	4
Temperature	(12)	LGS	AN	0-1500°F	20°F	7	2

Figure 4-5 Landing Gear System (LGS)/DMS Interface

4.1.3 SEPARATION CONTROL

The orbiter has primary responsibility for orbiter/booster separation. The booster activity with this function under nominal operating conditions is entirely one of monitoring. The booster and orbiter are attached during the boost phase by 3 explosive bolts. The squibs for these bolts are tested and ignited by the orbiter. The booster monitors the separation procedure by communications from the orbiter and by measurements of the resistance of a wire which becomes severed at separation and the temperature of the squibs. In an emergency abort situation the booster will also have the capability of igniting the explosive bolts. The abort conditions under which the booster has the authority to initiate separation are of the following types. If during powered boost a potentially catastrophic failure in the booster occurs such as a ruptured oxidizer or fuel tank or line where it becomes critical that separation occurs immediately to protect the orbiter from a booster explosion, the booster will initiate separation. This bypasses the time delay required to communicate a separation command to the orbiter to have the orbiter execute the command. If during powered boost a failure occurs calling for separation which is less time critical and booster/orbiter communications have failed the booster will initiate the separation. A failure which shuts down the booster main engines would be of this type. If, at the end of powered boost with the orbiter/booster communications failed, and the orbiter fails to initiate the separation or if the orbiter initiates separation either under abort or nominal conditions and separation does not occur the booster will initiate separation. The signal interface between the DMS and separation system in the booster is shown in figure 4-6. The DMS can issue an arming and separation command to the squib system of each bolt. From each bolt the DMS receives the results of the separation command by discretes indicating the occurence of a temperature pulse caused by the explosion of the squib and by the severance of wire when physical separation of the orbiter and booster occurs. The booster also monitors the temperature of each squib during prelaunch, launch and boost to determine if some heat source has caused the squib temperature to rise to a critical value. It is assumed that all three explosive bolts are serviced by the same electronic package; the DMS has control of electrical power on/off and can measure the supply voltage and electronic temperature. For test purposes the DMS can assume a test configuration in which the electronics simulate all separation responses upon receiving an arming and separation command if the built in test equipment determines that the separation system is operable.

SIGNAL	SOURCE	TYPE	RANGE	RESOLUTION	WORD LGTH (BITS)	RATE /SEC
Bolt #1, #2, #3						
Arming Command (3)	DMS	DIS				AR
Separation Command (3)	DMS	DIS				AR
Temperature Pulse (3)	SS	DIS				16
Separation Resistance (3)	SS	DIS				16
Temperature (3)	SS	AN	0-500°F	10°F	6	1
Electrical Power on Command	DMS	DIS				AR
Electrical Supply Voltage	SS	AN	0-30vdc	.3vdc	7	1
Electronic Temperature	SS	AN	0-500°F	20°F	5	1
Test Configuration Command	DMS	DIS				AR
Test Configuration Accomplished	SS	DIS				1

Figure 4-6 Separation System (SS)/DMS Interface

4.2 Propulsion Systems

The booster has 3 separate propulsion systems. These are the main boost rocket engines, the reaction jet system used during coast and the cruise air breathing engines. Each of these propulsion systems interface with the DMS.

4.2.1 Main Rocket Engines

It is assumed that the booster will contain 12 main rocket engines. Various techniques are being suggested for closed-loop control of the staged combustion cycle for these engines. No single approach has been proven at this time, but the following interface description is consistent with preliminary recommendations made by Pratt and Whitney Aircraft. Potential control points are shown in Figure 4-7. The control technique is to establish an open-loop program which sets the preburner oxidizer valve, main chamber oxidizer valve, and preburner fuel valve in accordance with a stored program that relates valve area and hence valve position with required thrust and mixture ratio. The control system interface in terms of sensor and control signal data rates and accuracies is shown in Figure 4-8.

In addition to controlling the engines the DMS will have control of propellant management and utilization. This function includes sensing the level, temperature and pressure in each tank and controlling solenoid valves. Figure 4-9 lists the interface requirements between the fuel management systems and the DMS.

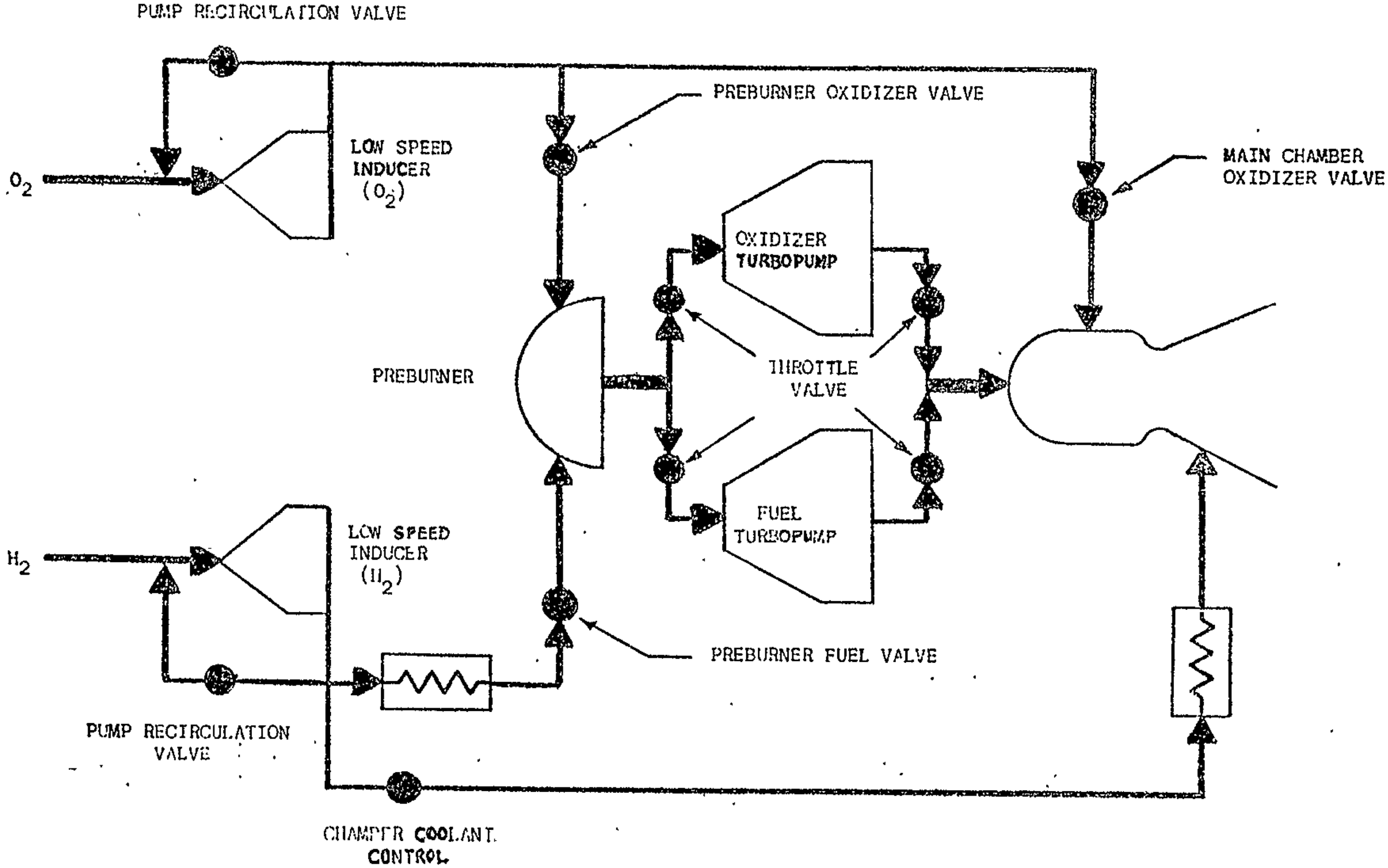


Figure 4-7   Main Engine Control Points

SIGNAL	SOURCE	TYPE	RANGE	RES.	WORD LENGTH	RATE
Main Chamber Oxidizer Valve Command	DMS	AN	0-100%	.1%	10	8
Main Chamber Oxidizer Valve Position	MPS	AN	0-100%	.1%	10	8
Preburner Oxidizer Valve Command	DMS	AN	0-100%	.1%	10	64
Preburner Oxidizer Valve Position	MPS	AN	0-100%	.1%	10	64
Preburner Fuel Valve Command	DMS	AN	0-100%	.1%	10	8
Preburner Fuel Valve Position	MPS	AN	0-100%	.1%	10	8
LOX Speed Inducer Oxidizer Valve Command	DMS	AN	0-100%	.1%	10	64
LOX Speed Inducer Oxidizer Valve Position	MPS	AN	0-100%	.1%	10	64
LOX Turbopump Speed	MPS	AN	0-100%	.1%	10	64
LH_2 Turbopump Speed	MPS	AN	0-100%	.1%	10	64
LOX Flow	MPS	AN	0-100%	.1%	10	8
LH_2 Flow	MPS	AN	0-100%	.1%	10	8
Main Pump Inlet Lox Temperature	MPS	AN	0-100%	.25%	8	4
Main Pump Inlet LH_2 Temperature	MPS	AN	0-100%	.25%	8	4
Main Pump Inlet Fuel Pressure	MPS	AN	0-100%	.1%	10	64
LOX Low Speed Inducer Exit Pressure	MPS	AM	0-100%	.1%	10	64
Heat Exchanger Exit Temperature	MPS	AN	0-100%	.25%	8	8
Eight Preburner Temperatures	MPS	AN	0-100%	.25%	64	2
Main Chamber Skin Temperature	MPS	AN	0-100%	.25%	8	2
Nozzle Coolant Temperature	MPS	AN	0-100%	.25%	8	2
Fifteen Solenoid Valves	DMS	DIS			15	2

Figure 4- 8 Main Propulsion System/DMS Interface

Twelve LH_2 Isolation Disconnect Valves	DMS	DIS			12	AR
Twelve LOX Isolation Disconnect Valves	DMS	DIS			12	AR
Six LH_2 Vent Valves	DMS	DIS			6	AR
Six Lox Vent Valves	DMS	DIS			6	AR
Four LH_2 Fill and Drain Valves	DMS	DIS			4	AR
Four Lox Fill and Drain Valves	DMS	DIS			4	AR
Ten LH_2 Pressurization Valves	DMS	DIS			10	AR
Ten Lox Pressurization Valves	DMS	DIS			10	AR
Two LH_2 Squib Actuated Pre-Valves	DMS	DIS			2	AR
Two Lox Squib Actuated Pre-Valves	DMS	DIS			2	AR
Valve Position Feedback	PMS	DIS			68	1
Ten Lox Level Sensor Otuputs	PMS	AN	0-100%	.25%	80	1
Ten LH_2 Level Sensor Outputs	PMS	AN	0-100%	.25%	80	1
Ten Lox Pressure Sensor Outputs	PMS	AN	0-100%	.25%	80	2
Ten LH_2 Pressure Sensor Outputs	PMS	AN	0-100%	.25%	80	2
Five Lox Pressure Switch Outputs	PMS	DIS			5	2
Five LH_2 Pressure Switch Outputs	PMS	DIS			5	2
Four Lox Temperatures	PMS	AN	0-100%	.25%	32	2
Four LH_2 Temperatures	PMS	AN	0-100%	.25%	32	2

Figure 4- 9 Propellant Management System/DMS Interface

4.2.2 Reaction Jet Propulsion System

Figure 4-10 is a schematic of the reaction jet propulsion system. The figure shows only the hydrogen system; the oxygen system is identical. The reaction jet system uses a gas generator driven turbopump to supply pressurized gaseous hydrogen and oxygen to the reaction jets. There are 16 reaction jets on the booster contained in two rings.

For redundancy three gas generator systems are used to supply the 16 reaction jets. The O_2 and H_2 solenoid valves shown on the rockets are controlled by the flight control system and are not considered part of the reaction jet propulsion system control. Figure 4-11 shows the interface requirements between the reaction jet propulsion system and DMS.

4.2.3 Cruise Engine System

The booster is assumed to have six cruise engines. Thrust level commands to each engine are provided by the autothrottle control law in the flight control system. In addition to throttle commands it will be the responsibility of the DMS to issue start and shutdown discretes, control the fuel air supply to the engines, and to monitor engine performance. Figure 4-12 shows the interface requirements between the cruise engines and DMS.

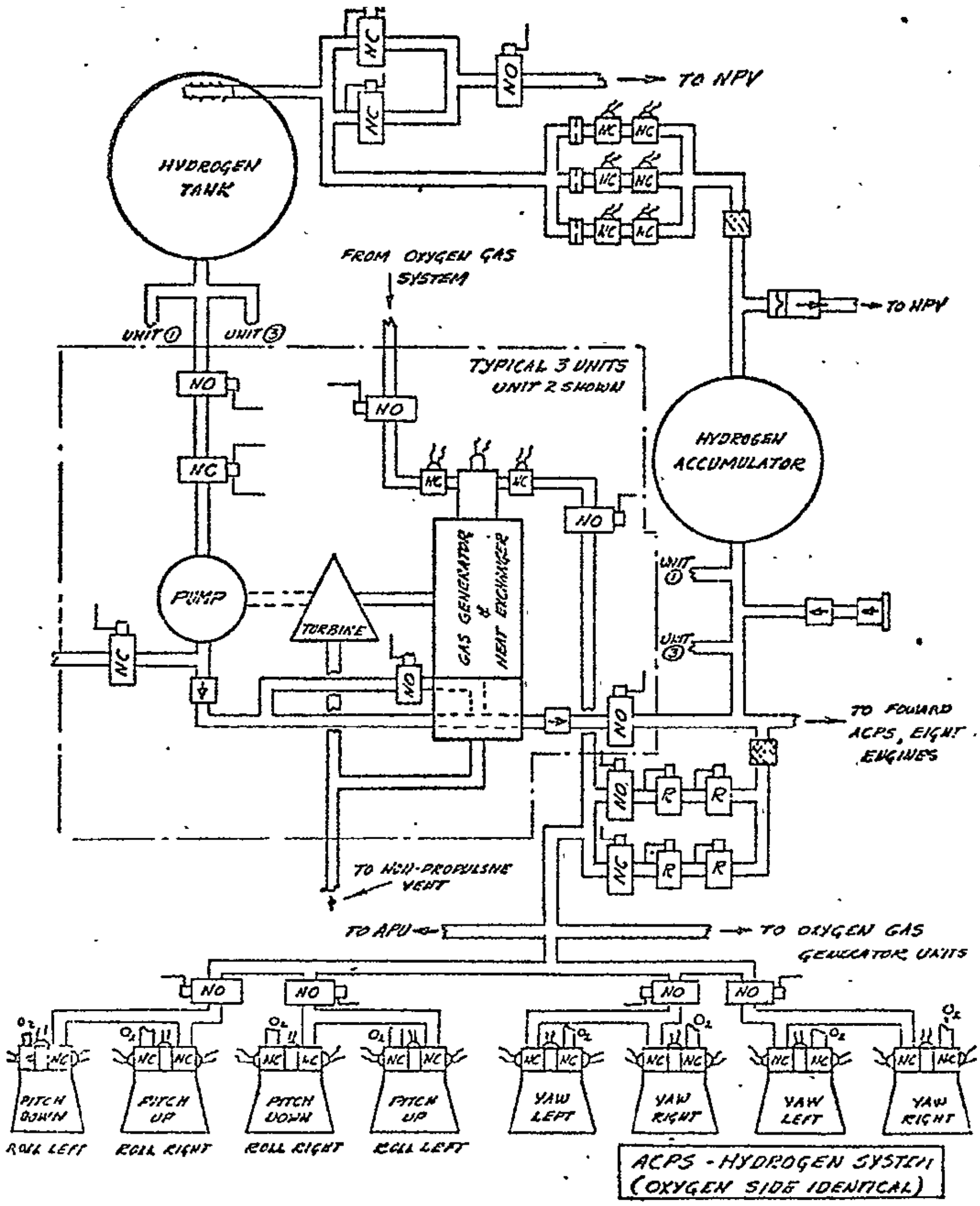


Figure 4-10 Attitude Control Propulsion System (ACPS)

SIGNAL	SOURCE	TYPE	RANGE	RES	WORD LENGTH	RATE
Three GH_2 Gas Generator GH_2 Control Valves	DMS	DIS			3	2
Three GH_2 Gas Generator GO_2 Control Valves	DMS	DIS			3	2
Three GH_2 Gas Generator Ignition Voltages	DMS	DIS			3	2
Three GH_2 Gas Generator Turbine Speed	RJP	AN	0-100%	.25%	24	2
Nine GH_2 Gas Generator Temperatures	RJP	AN	0-100%	.25%	72	2
Three GH_2 Gas Generator Pressures	RJP	AN	0-100%	.25%	24	2
Three GO_2 Gas Generator GH_2 Control Valves	DMS	DIS			3	2
Three GO_2 Gas Generator GO_2 Control Valves	DMS	DIS			3	2
Three GO_2 Gas Generator Ignition Voltages	DMS	DIS			3	2
Three HO_2 Gas Generator Turbine Speeds	RJP	AN	0-100%	.25%	24	2
Nine GO_2 Gas Generator Temperature	RJP	AN	0-100%	.25%	72	2
Three GO_2 Gas Generator Pressures	RJP	AN	0-100%	.25%	24	2
Forty-Eight Propellant Control Valves	DMS	DIS			48	AR
Sixteen Rocket Chamber Temperatures	RJP	AN	0-100%	.25%	128	8
Sixteen Rocket Chamber Pressures	RJP	AN	0-100%	.25%	128	8
GH_2 Tank Pressure	RJP	AN	0-100%	.25%	8	2
GO_2 Tank Pressure	R P	AN	0-100%	.25%	8	2
GH_2 Flow	RJP	AN	0-100%	.25%	8	2
GO_2 Flow	RJP	AN	0-100%	.25%	8	2
Feed Line Temperatures and Pressures (5)	RJP	AN	0-100%	.25%	40	2

Figure 4-11 Reaction Jet Propulsion System/DMS Interface

Engine Start Discretes	DMS	DIS			6	AR
Engine Test Discrete	DMS	DIS			6	AR
Fuel Control Valve Command	DMS	AN	0-100%	.25%	48	4
Air Control Valve Command	DMS	AN	0-100%	.25%	48	4
Ignition Voltage Command	DMS	DIS			6	AR
Ten Flow Control Valves	DMS	DIS			10	4
Four Vent Control Valves	DMS	DIS			4	4
Control Valve Position	CES	AN	0-100%	.25%	48	4
Engine Inlet Pressure	CES	AN	0-100%	.25%	48	4
Engine Inlet Temperature	CES	AN	0-100%	.25%	48	4
Two Turbine Temperatures Per Engine.	CES	AN	0-100%	.25%	96	4
Engine Speed	CES	AN	0-100%	.25%	48	4
Two GH_2 Flow	CES	AN	0-100%	.25%	16	4
Two GH_2 Tank Level	CES	AN	0-100%	.25%	16	4
Two GH_2 Tank Pressure	CES	AN	0-100%	.25%	16	4

Figure 4-12 Cruise Engine System/DMS Interface

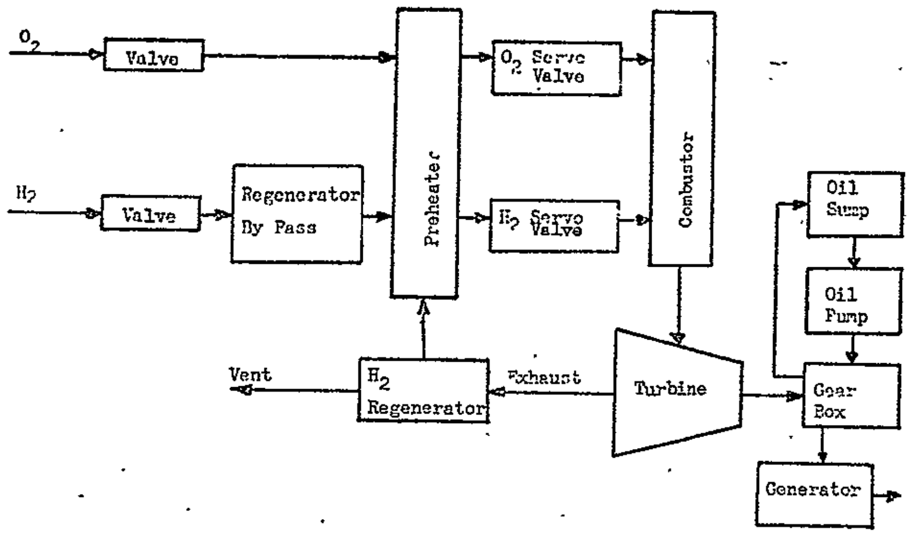
4.3 Electrical Power Generation and Distribution

Electrical power is supplied from two sources, umbilical power during prelaunch and a generator driven from a hot gas generator supplied turbine. The generator produces 115/208V; 400 hertz power which feeds an ac distribution system and converters which generate 28 volts dc which is fed to a dc distribution system.

4.3.1 Electrical Power Generation

Figure 4-13 is a block diagram of the major elements of the electrical generation system. Hydrogen and oxygen are supplied to the system from the main propellent tanks through quad redundant shut off valves and pressure regulators. The hydrogen supply is additionally controlled by a regenerator bypass. Excess hydrogen is collected from the turbine exhaust and mixed with the main supply hydrogen in the preheater. The regenerator bypass controls the total hydrogen supply to the system by limiting the hydrogen supplied from the main supply tanks. The hydrogen and oxygen are preheated before combustion in order to change the hydrogen and oxgyen to the gaseous state and to insure total consumption of the oxygen in the combustion process. The supply's of oxygen and hydrogen from the preheater are controlled by a servo valve which accurately controls the quantity of each to the combustor. The combustor is supplied by an overly rich hydrogen mixture in order to guarantee complete oxygen consumption thus reducing oxidation of the turbine parts during operation. Excess hydrogen is also used to control the combustion temperature. The hot gas is used to drive the turbine. The turbine is connected to the

Figure 4-13 Main Elements of Electrical Generator System



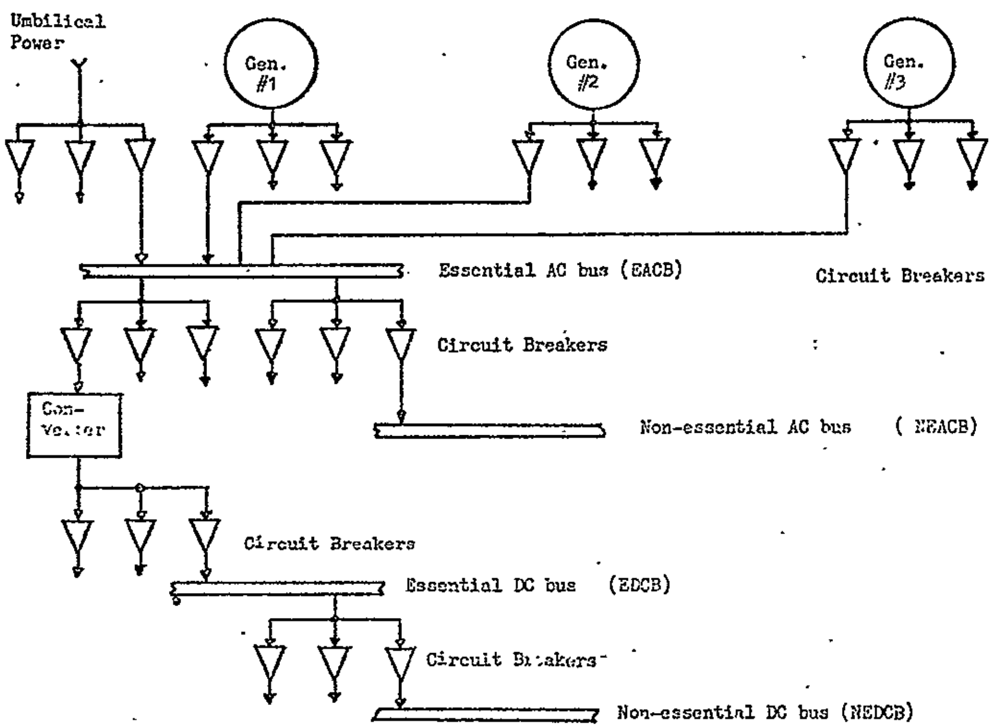


Figure 4-14 Electrical Distribution System

generator through a constant speed drive transmission. The transmission compensates for variable turbine drive speeds and generator load variations. Constant speed is required to provide a fixed frequency output from the generator. Figure 4-15 is a list of the monitoring and control signal interface between the electrical generation system and the DMS. On the space shuttle booster there are three identical electrical generation systems. The signal interface shown in Figure 4-15 is for one system only.

4.3.2 Electrical Distribution System

A diagram of the electrical distribution system including dc converters is shown in Figure 4-14 Each of the three generators and umbilical power is capable of being connected to any one of three essential AC buses through circuit breakers controllable by the DMS. Each of the essential ac buses is capable of being connected to any one of three ac to dc power converters and any one of three non-essential ac busses through DMS controlled circuit breakers. The outputs of each converter are capable of being connected to any one of three essential dc buses through DMS controlled circuit breakers. Each subsystem is connected through DMS ontrolled circuit breakers to the appropriate ac or dc bus. Critical subsystems are capable of being supplied from more than one bus, though not simultaneously. No provision is provided to parallel generators or converter outputs however, generator speed control will attempt to keep all generators synchronized such that an inadvertant paralleling of generators due to circuit failures will not demand excessive currents from the paralleled generators. Protective diodes will be provided to the DMS between umbilical power and each generator in order to parallel each generator with umbilical power for a short period during the switch from external to internal power in order to avoid switching transients. Figure 4-16 is a list of the interface signals between the electrical distribution system and the DMS.

Frequency measurements are synchronized to the computer clock. Two words are read into the DMS for each frequency measurement. These two words are generated from the 400 cycle power line. One input word is the contents of a counter register which continuously counts the positive slope zero crossings of the ac voltage. This counter runs continously, is not reset and overflows are ignored. The second word is a counter counting 40Kc clock signals and reset every time the first counter counts. These two words are sampled four times every second. If the ac supply frequency is exactly 400 cps then the difference in two samples of the first word will be counts and the difference in two samples of the second word will be 0 counts.

The synchronizing voltage is derived by registering the maximum difference between umbilical and generator voltage occuring between any two DMS samplings. If the maximum voltage between the umbilical and generator is small, over $\frac{1}{4}$ second (i.e., the time between two samplings) it is known that the two sources are nearly synchronized in frequency and phase and the switching from one source to the other can occur with little switching transients.

SIGNAL	SOURCE	TYPE	RANGE	RESOLU- TION.	WORD LGTH (BITS)	RATE /SEC
Circuit Breaker Commands						
Umbilical to EACB (3)	DMS	DIS				AR
Generator to EACB (9)	DMS	DIS				AR
EACB to Converter (9)	DMS	DIS				AR
EACB to NEACB (9)	DMS	DIS				AR
Converter to EDCB (9)	DMS	DIS				AR
EDCB to NEDCB (9)	DMS	DIS				AR
Monitoring Signals						
EACB Current (3)	EDS	AN	0-100%	5%	5	4
NEACB Current (3)	EDS	AN	0-100%	1%	5	4
EDCB Current (3)	EDS	AN	0-100%	1%	5	4
NEDCB Current (3)	EDS	AN	0-100%	1%	5	4
Generator Sync Voltage (3)	EDS	AN	-300to+300V	5v	6	4
Umbilical Frequency Coarse	EDS	AN	0-128counts	1count	7	4
Umbilical Frequency Fine	EDS	AN	0-100counts	1count	7	4
Converter Temperature (3)	EDS	AN	-20 to +85°C	5°C	5	4
Converter Voltage (3)	EDS	AN	0 to 30v	.5V	6	4

Figure 4-16 Electrical Distribution System (EDS)/DMS Signal Interface

O_2 and H_2 Shut Off Valve Commands (8)	DMS	DIS				AR
O_2 and H_2 Preheater Pressure (2)	EGS	AN	300-400psi	1psi	7	4
O_2 and H_2 Preheater Temperature(2)	EGS	AN	0-100°F	1°F.	7	4
Regenerator Bypass Valve Command	DMS	AN	Closed-open	1%	7	4
O_2 and H_2 Servo Valve Command (2)	DMS	AN	Closed-open	1%	7	4
O_2 and H_2 Heater (2)	DMS	DIS				4
O_2 and H_2 Combustor Inlet Pressure (2)	EGS	AN	300-400psi	1psi	7	4
O_2 and H_2 Combustor Inlet Temp.(2)	EGS	AN	1000-1400°F	4°F	7	4
Combustor Output Temperature	EGS	AN	1000-1400°F	4°F	7	4
Turbine Speed	EGS	AN	0-70000 rpm	600rpm.	7	4
Turbine Exhaust Temperature	EGS	AN	200-500°F	4°F	7	4
Oil Temperature	EGS	AN	150-190°F	1°F	6	4
Oil Pump Output Temperature	EGS	AN	0-1000psi	10psi	7	4
Oil Quantity	EGS	AN	0-100%	2%	6	4
Oil Control Valve	DMS	AN	0-100%	1%	7	4
Generator Gross Speed	EGS	AN	0-128counts	1count	7	4
Generator Fine Speed	EGS	AN	0-100counts	1 count	7	4
Generator Voltage, ³Phase A,B,C.(3)	EGS	AN	0-150vac	1vac	8	4
Generator Current Phase A,B,C.(3)	EGS	AN	0-50 amps	1amp	6	4
Generator Field Current Control	DMS	AN	0-100%	1%	7	4
Turbine rms Vibration	EGS	AN	$0\text{-}8.10^{-3}$in	2.10^{-3}in	6	4

Figure 4-15 Electrical Generation System (EGS)/DMS Interface

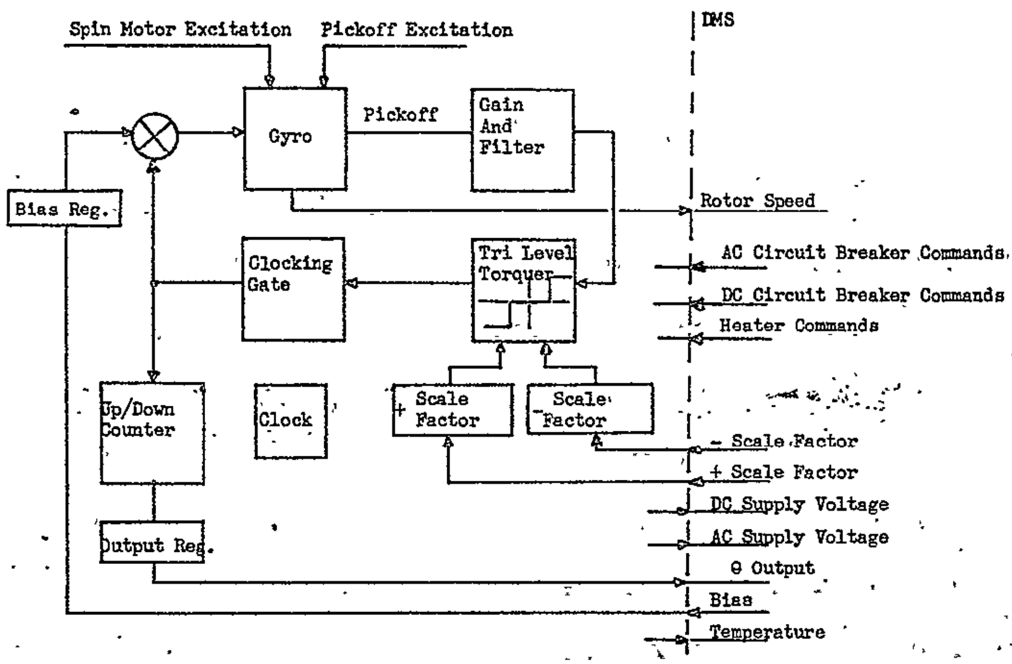
4.4 Navigation and Guidance

Navigation is the process of determining the location and velocity of the vehicle while guidance is the process of determining the desired vehicle position and velocity and the generation of commands to force the desired and actual position and velocity to coincide. Navigation requires sensors which measure either the vehicle's position and/or velocity with respect to a known reference or the vehicle's change in position and/or velocity with respect to its known position and velocity. To perform navigation the DMS computers must interface with navigation instruments. The only interface with external equipment required by the guidance system is an interface with data entry whereby the pilot can enter the values required to specify the desired mission flight path. Data entry is discussed under section 4.7 and thus only the navigation sensor interface requirements will be discussed in this section.

4.4.1 Strapdown Inertial Navigation

The sensors required for inertial navigation(using a strapdown configuration capable of surviving any three sensor failures) are 6 linear accelerometers and 6 single degree of freedom gyros. Figure 4-17 is a block diagram of a single degree of freedom gyro and its torquing and DMS interface electronics. The gyro contains a balanced rotor driven by a spin motor supplied from the 3 phase 400 cycle essential ac bus. The rotor is mounted in a single gimbal having a pickoff and a torque motor. During operation the pickoff angle is continuously driven to

Figure 4-17 Single Degree of Freedom Gyro Mechanization



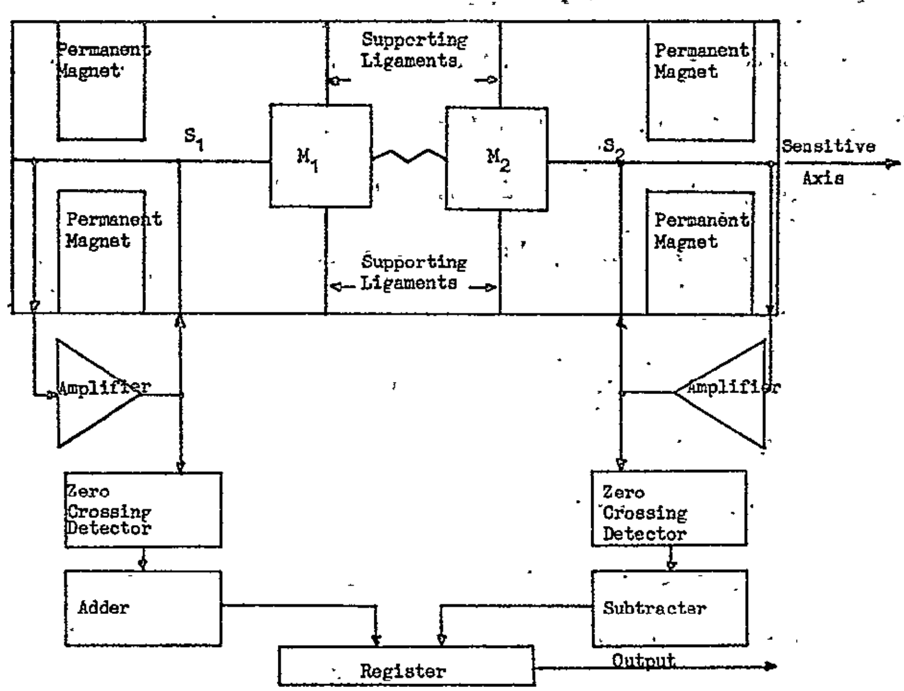


Figure 4-18 Vibrating String Accelerometer

zero by the torquer through an electronic servo loop. If the gyro was perfect the rotational rate about the input axis would be proportional to the torquing current. In an actual instrument there are numerous error sources which are partially corrected in the torquing loop mechanization. In order to provide a digital interface and eliminate effects of torquer nonlinearities the gyro is pulse torqued rather than proportionally torqued. A high precision reference clock is generated. At each clock pulse a decision is made to command either a constant positive, constant negative or zero torque pulse for the next clock period. This decision is based upon the magnitude of the filtered pickoff output. The instrument output is the accumulated algebraic count of the torquing pulses. Special provisions are provided for the elimination of two major error sources, scale factor and bias errors. Scale factor errors originate in torquer nonlinearities and tolerances in the electronic generation of the torquer current. Correction of scale factor errors is accomplished by adjusting the positive and negative torquing pulse current magnitude. Pickoff and torquing current null errors produce a continuous steady state bias while mass unbalance in the gyro causes biases proportional to accelerations experienced by the instrument. These biases are removed by constantly supplying a computed torquing current to the gyro. Scale factor and bias errors are sensitive to temperature which is therefore controlled within narrow margins.

There are several types of accelerometers used in inertial navigation systems. The most commonly used accelerometer in stabilized platform applications in a pendulous integrating gyro accelerometer (PIGA). This instrument has inherent disadvantages when applied to a strapdown navigation system. The instrument output is proportioned to both linear acceleration and angular rates. When mounted on a platform stabilized to inertial space,angular rate inputs to the instrument are zero. In a strapdown application the instrument output would have large angular rate components which would have to be subtracted in order to obtain pure linear accelerations. In designing an accelerometer each design decision is based upon obtaining highly accurate acceleration outputs. If the design is also constrained by a requirement to obtain accurate angular rate outputs a sacrifice in linear acceleration will result. Because of the limitations of the PIGA, a vibrating string accelerometer will be assumed as the strapdown linear acceleration instrument on the space shuttle booster. Figure 4-18 shows the internal mechanization of a vibrating string accelerometer. The heart of the instrument are the two masses, M_1 and M_2 separated by a spring and supported along the instrument sensitive axis by two taut strings, S_1 and S_2. Each mass is also supported by four ligaments, two as shown in the plane of Figure 4-18 and two normal to the plane of figure 4-18. The two masses and strings are made as identical as possible. When the instrument is not being subjected to an acceleration along its sensitive axis the tension on the two strings is the same. An acceleration along the instruments sensitive axis causes an increase in the tension on one string and a decrease in the tension on the other string. The two strings are made to vibrate at their resonant frequency. To do this,permanent magnets are built into the instrument such that each string passes through a strong magnetic field. As the strings vibrate they move in this magnetic field causing a current to be induced in the string. This current is amplified and fed back to the string in such a manner that a sustained vibration at the natural frequency of the string occurs. If T_1 is the tension in string 1, T_2 is the tension in string 2, T_3 the tension in the interconnecting string,and a the acceleration being experienced by both masses along the sensitive axis direction,then the equations relating tension to acceleration for the two masses are:

$$T_3 - T_1 = M_1 a \qquad (1)$$

$$T_2 - T_3 = M_2 a \qquad (2)$$

These two equations assumed that both masses are being simultaneously acted upon by the same accelerations. This is a reasonable approximation in the frequency bandwidth of interest for the instrument being employed as an accelerometer in that the two masses are constrained by the strings to remain with the instrument. Adding equations 1 and 2 yields:

$$T_2 - T_1 = (M_1 + M_2)a \qquad (3)$$

If the tension in the strings under static unaccelerated conditions is T_0 and the change in tension due to an acceleration of magnitude a is ΔT_2

and $-\Delta T_1$ then

$$T_1 = T_0 - \Delta T_1 \qquad (4)$$

$$T_2 = T_0 + \Delta T_2 \qquad (5)$$

and equation 3 becomes

$$\Delta T_1 + \Delta T_2 = (M_1 + M_2)a \qquad (6)$$

In the linear range of the accelerometer

$$\Delta T_1 = K_1 a \qquad (7)$$

$$\Delta T_2 = K_2 a \qquad (8)$$

The resonant frequency of a uniform vibrating string is proportional to the square root of the tension on the string thus

$$f_1 = C_1 \sqrt{T_1} \qquad (9)$$

$$f_2 = C_2 \sqrt{T_2} \qquad (10)$$

substituting equations 4,5,7 and 8 into 9 and 10 yields

$$f_1 = C_1 \sqrt{T_0 - K_1 a}$$

$$f_2 = C_2 \sqrt{T_0 + K_1 a}$$

By Taylor's expansion

$$f_1 = C_1\sqrt{T_0}\left[1 - \frac{1}{2}\left(\frac{K_1 a}{T_0}\right) - \frac{1}{8}\left(\frac{K_1 a}{T_0}\right)^2 - \frac{1}{16}\left(\frac{K_1 a}{T_0}\right)^3 - \frac{5}{128}\left(\frac{K_1 a}{T_0}\right)^4 + \cdots\right] \qquad (11)$$

$$f_2 = C_2\sqrt{T_0}\left[1 + \frac{1}{2}\left(\frac{K_1 a}{T_0}\right) - \frac{1}{8}\left(\frac{K_1 a}{T_0}\right)^2 + \frac{1}{16}\left(\frac{K_1 a}{T_0}\right)^3 - \frac{5}{128}\left(\frac{K_1 a}{T_0}\right)^4 + \cdots\right] \qquad (12)$$

These series are rapidly convergent if $\frac{K_1 a}{T_0}$ and $\frac{K_2 a}{T_0}$ are small. From equations 7 and 8 these terms become $\frac{\Delta T_1}{T_0}$ and $\frac{\Delta T_2}{T_0}$. The accelerometer is manufactured so that the pre-stress on the strings is very high compared to the operating range stress changes making the above series rapidly convergent. Subtracting equations 11 and 12 yields

$$f_2 - f_1 = (C_2 - C_1)\sqrt{T_0} + \frac{\sqrt{T_0}}{2T_0}(C_2K_2 + C_1K_1)a - \frac{\sqrt{T_0}}{8T_0^2}(C_2K_2^2 - C_1K_1^2)a^2$$
$$+ \frac{\sqrt{T_0}}{16T_0^3}(C_2K_2^3 + C_1K_1^3)a^3 - \frac{5\sqrt{T_0}}{128T_0^4}(C_2K_2^4 - C_1K_1^4)a^4 + \cdots \qquad (13)$$

If the instrument is built symmetrical, i.e., the two masses and two strings are identical the

$$M_1 = M_2 = M \qquad (14)$$

$$C_1 = C_2 = C \qquad (15)$$

$$K_1 = K_2 = K \qquad (16)$$

and the equation 13 becomes

$$f_2 - f_1 = \frac{CK}{\sqrt{T_0}}\, a + \frac{CK^3}{8T_0^2\sqrt{T_0}}\, a^3 \qquad (17)$$

With a perfect instrument acceleration would be measured using the equation

$$a = \frac{\sqrt{T_0}}{cK}(f_2 - f_1) - \frac{K^2}{8T_0^2}\, a^3 \qquad (18)$$

The output of the amplifiers supplying excitation current to the vibrating strings is a sine wave. The frequency difference of the two sine waves is generated by taking the difference in the number of zero crossings on the two amplifier outputs over a specified constant time increment. The input to the computer is this accumulated zero crossing difference. With the computer, compensation is made for instrument nonsymmetry are the a^3 term of equation 18 by applying the computation below

$$\Delta V = A_1 I - A_0 - A_2 I^2 - A_3 I^3 \quad (19)$$

The values of A_1, A_2 and A_3 are determined from laboratory calibrations of the instrument and A_0 from laboratory and prelaunch calibrations. I is the instrument output and ΔV the measured change in velocity over the time interval between samplings.

The instrument is highly sensitive to temperature and has its temperature controlled in the same manner as the gyros. Interface signals between the accelerometer and computer include DC power supply monitoring and control signals as are used for the gyro.

The sampling rate and resolution required in interrogating the strapdown accelerometer and gyros and solving the strapdown equations has a major influence on the speed requirements of the DMS. In order to determine the sampling rate and resolution requirements a simplified error analysis on each error source is performed below.

<u>GYRO RESOLUTION</u> In strapdown navigation the gyros are used to determine the direction of all forces acting upon the vehicle.

An estimate of the velocity error generated by gyro resolution is given by the formula

$$E_V/R_\theta = \int_0^{t_f} a(t) \sin(\theta m - \theta)\, dt$$

where the integration limits cover the time from launch to thrust termination and $a(t)$ is total acceleration and $\theta m - \theta$ is the difference between measured and actual orientation. The formula represents a worst case estimate of velocity error in that rather than total acceleration only the component of acceleration normal to the gyros sensitive axis should appear in the formula. Errors in measured orientation, i.e., $\theta m - \theta$ due to gyro resolution will vary throughout flight within the range $-R_\theta$ to R_θ. A worst case analysis assumes that $\theta m - \theta$ is always at an extreme value R_θ. Assuming small values of R_θ such that the sine R_θ is similar to R_θ expressed in radians then performing the integration in the above formula yields

$$E_V/R_0 = V_f R_\theta$$

where V_f is the vehicle final velocity. Expressing R_θ in degrees and assuming a value of V_f = 15000 ft/sec yields

$$E_V/R_0 = 262 R_\theta \text{ ft/sec}$$

<u>ACCELEROMETER RESOLUTION</u> The accelerometer outputs to the computer at each sampling are a measurement of the change in velocity along the instrument sensitive axis direction since the last sampling. An error source in these measurements is the resolution in measuring the velocity change defined as R_V. The navigation solution in the computer resolves the velocity change into inertial coordinates and accumulates the velocities in each inertial direction according to the formula

$$V = \sum \Delta V_m \cos \theta$$

where ΔV_m is the measured velocity change. The accelerometer mechanization which accumulates the difference in the number of zero crossings of two sine waves does not contribute to an accumulating velocity resolution error. The resolution error is generated such that the sum of ΔV_m pulses accumulated in the DMS will never differ more than $2R_v$ from a sum obtained from an instrument having no resolution error. Thus the velocity error due to accelerometer resolution is

$$E_v/R_v = 2R_v \text{ ft/sec}$$

<u>SAMPLING RATE</u> Errors caused by a finite sampling rate are due to the integrations required in the strapdown navigation solution. The basic navigation computation requires the integration of a resolved acceleration. The integration is approximated in the DMS by a summation process. Several integration approximations exist. Rectangular integration will be used to determine the errors caused by sampling because it is the simplest to analyze and will lead to a conservative answer in that other integration schemes have in general increased accuracy. In order to analyze the effects of sampling, a simple boost trajectory must be assumed. The resolved acceleration of the sample trajectory is then integrated over one sampling period and the approximate integral value developed by the DMS subtracted from the true integral result to generate the error accumulated during one sampling period. This error is then summed over the total number of samples in the boost trajectory. The error over a single sampling period is given by

$$E = \int_{t-T}^{t} (a_0 + \dot{a}t)\cos\dot{\theta}t\,dt - T(a_0 + \dot{a}t)\cos\dot{\theta}t$$

where the assumed trajectory is caused by the vehicle experiencing an acceleration composed of a constant term, a_0, plus a constant acceleration rate, $\dot{a}$, and turning at a constant rate, $\dot{\theta}$. The sampling interval is T.

Performing the integration yields

$$E = \frac{a_0}{\dot\theta}\sin\dot\theta t - \frac{a_0}{\dot\theta}\sin\dot\theta(t-T) + \frac{\dot a}{\dot\theta^2}\cos\dot\theta t - \frac{\dot a}{\dot\theta^2}\cos\dot\theta(t-T) + \frac{t\dot a}{\dot\theta}\sin\dot\theta t$$
$$- \frac{(t-T)\dot a}{\dot\theta}\sin\dot\theta(t-T) - T(a_0+\dot a t)\cos\dot\theta t$$

using trig identities and assuming that $\dot\theta$ T is small such that cos $\dot\theta$ T $\doteq$ 1 and sin $\dot\theta$ T = $\dot\theta$ T yields:

$$E = -T^2 \dot a \quad \cos\dot\theta t$$

To accumulate the velocity error due to sampling over the entire boost flight this error must be evaluated for each sampling instant and summed over all sampling periods. A worst case evaluation results if $\dot\theta t$ is assumed to be zero resulting in

$$E_{V/T} = \sum_{i=1}^{T_f/T}(-T^2\dot a) = -T\,T_f\,\dot a$$

where T_f is the total boost time from launch to thrust termination. The change in acceleration, $\dot a$, is caused by the expenditure of vehicle mass as fuel is used while thrust is maintained near constant. A worst case assumption is that the initial acceleration at launch is zero and thus the total achieved final velocity can be attributed to the $\dot a$ term. If this is assumed with a final velocity of 15000 ft/sec and a total flight time, T_f of 200 seconds then

$$E_V/_T = 150T \quad ft/sec$$

In determining the values of R_v, R_0 and T both practical instrumentation constraints and allowable navigation errors must be considered. Any velocity and position errors existing at booster thrust termination can be corrected by the orbiter. The correction must be paid for by a weight penalty in additional fuel expended by the orbiter.

One phase A orbiter design has vehicle properties of

separation weight	W	730,000 lbs
fuel weight	W_f	500,000 lbs
thrust	F	900,000 lbs
specific impulse	I_{sp}	451 seconds

The total velocity increment capabilities of this vehicle is given by

$$V_T = \int_0^{\frac{I_{sp} W_f}{F}} \frac{gF}{W - \frac{F}{I_{sp}} t} \, dt = g I_{sp} \ln \frac{W}{W - W_f}$$

Evaluating this integral using the above numbers gives

$$V_T = 16{,}670 \text{ ft/sec.}$$

Practical numbers for R_v, R_θ and T are .2 ft/sec., .02 degrees and 1/64 sec., respectively. According to the above analysis these numbers will result in a total additive error of 8.0 ft/sec. This results in a loss of .05% of the orbiter velocity capability. This is a very small portion of the total available orbiter velocity capabilities and is assumed acceptable.

Figure 4-19 is a list of all interface signals between a single gyro and accelerometer and the DMS. There are a total of 6 gyros and 6 accelerometers on the booster and thus the total list for the vehicle is 6 times that shown in Figure 4-19

SIGNAL	SOURCE	TYPE	RANGE	RESOLU-TION	WORD LGTH (BITS)	RATE /SEC
Rotor Speed (Gyro)	SSS	AN	$0-400^{r}/s$	$1^{r}/s$	9	4
AC Circuit Breaker Commands (2)	DMS	DIS			2	AR
DC Circuit Breaker Commands (2)	DMS	DIS			2	AR
Heater Command (2)	DMS	DIS			1	4
-Scale Factor (Gyro)	DMS	AN	0-1	.005%	15	AR
+Scale Factor (Gyro)	DMS	AN	0-1	.005%	15	AR
DC Supply Voltage (2)	SSS	AN	0-30vdc	.3vdc	7	4
AC Supply Voltage, Phase A (2)	SSS	AN	0-127v	1v	7	4
AC Supply Voltage, Phase B (2)	SSS	AN	0-127v	1v	7	4
AC Supply Voltage, Phase C (2)	SSS	AN	0-127v	1v	7	4
$\Delta\theta$ Output (Gyro)	SSS	AN	$-1^{\circ}-+1^{\circ}$	.02v	7	64
Bias (Gyro)	DMS	AN	0-1	.005%	15	64
Temperature (2)	SSS	AN	$75-125^{\circ}F$	$.5^{\circ}\%$	7	4
Δ V Output (Accelerometer)	SSS	AN	-3-+3fps	.2fps	5	64

Figure 4-19 Strapdown Sensor System (SSS)/DMS Interface

Pitot Pressure	ADS	AN	0-23psi	.012psi	11	8
Static Pressure	ADS	AN	0-23psi	.012psi	11	8
Angle of Attack Top Pressure	ADS	AN	0-23psi	.012psi	11	8
Angle of Attack Bottom Pressure	ADS	AN	0-23psi	.012psi	11	8
Angle of Attack Right Pressure	ADS	AN	0-23psi	.012psi	11	8
Angle of Attack Left Pressure	ADS	AN	0-23psi	.012psi	11	8
Probe Temperature	ADS	AN	-100 to $50^{\circ}C$	$.25^{\circ}C$	10	8
Supply Voltage	ADS	AN	0 to 30v	.5v	6	AR
Electronic Temperature	ADS	AN	75 to $125^{\circ}F$	$1^{\circ}F$	6	AR
Power on Command	DMS	AN	75 to $125^{\circ}F$	$1^{\circ}F$	6	AR

Figure 4-20 Air Data Sensor (ADS)/DMS Interface

AC Power on Command	DMS	DIS				AR
DC Power on Command	DMS	DIS				AR
AC Supply Voltage, Phase A	MFC	AN	0-127v	1v	7	1
AC Supply Voltage, Phase B	MFC	AN	0-127v	1v	7	1
AC Supply Voltage, Phase C	MFC	AN	0-127v	1v	7	1
DC Supply Voltage	MFC	AN	0-30v	.3v	7	1
Gyro Rotor Speed	MFC	AN	0-127count	1count	7	1
X Axis Gyro Pickoff	MFC	AN	-90 to 90°	1°	8	4
Y Axis Gyro Pickoff	MFC	AN	-95 to 95°	1°	8	4
X Axis Gyro Torquer	DMS	AN	-90 to 90°	1°	8	4
Y Axis Gyro Torquer	DMS	AN	-95 to 95°	1°	8	4
Oscillator Frequency	MFC	AN	0-127 count	1 count	7	1
Bearing Output $K\sin\Psi_{mh}$	MFC	AN	-10 to 10v	.02v	10	4
Bearing Output $K\cos\Psi_{mh}$	MFC	AN	-10 to 10v	.02v	10	4
Electronic Temperature	MFC	AN	$75-125^{\circ}F$	$.5^{\circ}F$	7	1
Heater Command	DMS	DIS				1

Figure 4-21 Magnetic Flux Gate Compass (MFC)/DMS Interface

4.4.2 Air Data Sensors

The air data sensors consist of probes into the atmosphere, measuring static atmospheric conditions and the disturbance of the atmosphere caused by the vehicle motion. All atmospheric data is measured using two probes.combined into a unit called a pitot-static tube which will be placed on the nose of the booster. This probe is used to sense two pressures and a temperature. Pitot pressure is the pressure of the air rammed into a port on the front of the probe. The ports for the static pressure measurement are along the side of the probe and thus provide still air pressure. The temperature sensor measures the heating effect of the compression of the air in front of the probe.

The second probe measures the four pressures used to determine angle of attack and side slip angle. This is a cylindrical probe having four slots, one on each side, one on the top, and one on the bottom. The difference in pressure between the top and bottom slots is used to measure angle of attack, and the difference in pressure between the two side slots is used to measure side slip angle.

Pressure tranducers are used to convert the pressures sensed by the two probes to an analog votlage. An analog to digital converter is then used to convert these voltages and the temperature probe output to a digital format. Figure 4--20is a list of the signals which interface between the air data probes and the DMS.

4.4.3 Magnetic Flux Gate Compass

The magnetic flux gate compass provides data to the DMS regarding the bearing angle with respect to the earth's north magnetic pole. The output of the magnetic flux gate signal is a typical three wire synchro signal. Thus magnetic heading ψ_{mh}, is in the form of two voltages v_1 and v_2 related to magnetic heading by the formulas:

$$v_1 = K \sin \psi_{mh} \tag{1}$$

and

$$v_2 = K \cos \psi_{mh} \tag{2}$$

The magnetic flux gate compass has a sensing element consisting of a triangular soft iron core stabilized in the earth's tangent plane by a gyro. Two magnetic windings, primary and secondary, are wound about the soft iron core. The primary winding is excited by an AC current of 487½ cycles per second and drives the core into magnetic saturation in both directions with desaturation periods between each saturation drive. The secondary winding is connected so that any induction from the primary is suppressed.

The earth's magnetic field is alternately excluded and admitted to the core as the core is alternately saturated and desaturated by the primary. This modulation of the earth's magnetic field is picked up by the secondary winding. The modulation frequency is 975 cycles per second, filtering allows for further rejection of any 487½ cycle per second primary induction into the secondary. The secondary coils are arranged so that a unique output occurs for each angle with which earth's magnetic field intersects the core. Figure 4-21 is a list of the DMS/Magnetic flux gate compass interface signals.

The interface signals consist of commands to switch the ac and dc supply voltages. Analog signals are required to monitor the ac and dc supply voltages. The gyro rotor speed is monitored by a counter which counts pulses which are generated once each rotor revolution. The counter is 7 bits long and read by the DMS once per second. The counter will overflow several times during the second between interrogations by the DMS. The DMS will assume that the rotor speed is within a wide speed band and thus receives a fine measurement within the assumed speed band.

The magnetic flux gate compass gyro is a vertical seeking gyro in a normal aircraft application. Vertical seeking gyros are constructed such that mechanical torques and the gyro automatically keep the gyro's angular momentum vector aligned with the total acceleration vector on the gyro case. The only acceleration sensed by the gyro is the reaction force between the aircraft and gyro opposing gravity in a typical aircraft application during constant velocity cruise. Thus the gyro's angular momentum vector is automatically aligned with the vertical. A vertical seeking gyro is inadequate for the magnetic flux gate compass to be available on the space shuttle during reentry when large aerodynamic accelerations are experienced. Thus for the space shuttle application a standard two axis gyro will be used having pickoffs and torques on the sensitive gyro axis. The DMS will slave the gyro to the outputs of the inertial navigation Euler angle attitudes at a 4 per second rate. The frequency of the $487\frac{1}{2}$ cps oscillator is measured by a counter and read into the DMS at a 1/second rate. The temperature of the electronics and gyro is measured and controlled by a heater command. The two voltages represented by equations 1 and 2 are read by the DMS at four times per second.

4.4.4 TACAN Receiver

The Tactical Air Navigation System (TACAN) is designed to give a continous indication of bearing and distance from an aircraft to a TACAN ground station. There are 126 frequency channels assigned for TACAN usage. Each operational ground TACAN station is assigned a single channel. If more than a single TACAN station is operating on the same channel,the geographic separation will be such as to eliminate all possible ambiquities between the two stations.

The TACAN ground station produces a theoretically infinite number of courses or radials which radiate from the station like spokes from the hub of a wheel. These radials are provided with azimuth intelligence by a comparison of the phase difference between two radiated signals. These signals consist of a pulsed reference phase signal and a variable phase signal which is directional and exhibits a variable phase versus azimuth relationship. The reference and variable phase signal are radiated from an antenna array consisting of a central antenna and two cylinders. The entire antenna rotates at 15 revolutions per second. The inner cylinder has an embedded reflector to distort the radiation pattern of the central antenna into a cardioid pattern. The outer cylinder has nine embedded reflectors resulting in a modulation of the basic sine wave nine times for each 360° phase difference. This modulation technique results in nine electrical degrees being equated to one azimuth degree.

A pulsed reference signal is utilized for comparison with the variable signal (sine wave). This signal is transmitted each time the maximum lobe of the cardioid is at a bearing of magentic east. Within the receiver the phase measurement is made with respect to a variable signal zero crossing point. The output of the receiver phase measurement electronics is the magentic bearing angle from the transmitter to the receiver.

The measurement of distances from the vehicle to the TACAN ground station is accomplished through utilization of an interrogation/response technique. The vehicle transmitts an interrogation pulse to the ground station transponder. The vehicle receiver determines the time required between the original pulse transmission and the receipt of the transponded pulse. Slant range is then computed by knowing the propogation speed of electromagnetic radiation.

Special features are designed into the distance measurement system to enable simultaneous usage by up to 100 vehicles. The vehicle transmitted interrogation pulses occur at a random frequency. The distance measurement receiver operates in a search or track mode. In both modes the receiver looks at a narrow time slot for the transponded return. In the search mode this time slot is adjusted until the ratio of received returns to transmitted pulses is high at which time the track mode is entered. In track the time slot is slowly moved so as to keep the return pulses always centered in the time slot. Because the pulse transmission rate from each vehicle is random the probability of receiving a large number of replies initiated by other aircraft during the time slot period is low.

Figure 4-22 shows the interface requirements between the TACAN and the DMS. The interface includes the normal AC and DC power on commands and primary power monitoring returns. The transmitter required of the distance measuring equipment (DME) will have its frequency crystal controlled. The crystal will be mounted in a temperature controlled oven. The receiver transmitter channel selected by the pilot will be indicated to the DMS. The bearing measuring equipment will return bearing to the DMS and a signal indicating that the system has achieved phase lock on the received bearing signals. The bearing measuring equipment can be tested by commanding a test and supplying a bearing value. The normal bearing output from the TACAN should indicate the test value within some small tolerance band after the bearing valid discrete is set.

Slant range is determined from two TACAN returns: the center of the slot position and the position of the return in the slot. The center of the slot is measured as a time from the transmission of the interrogation pulse. Under normal operation, multiple returns will be received in the slot during the period between two DMS samplings of the position of the return in the slot. The data generated as the return slot position is the algebraic sum of the position of all returns accumulated since the last DMS sampling. The TACAN distance measuring system also delivers the total number of interrogations transmitted and replies received since the last DMS sampling. The DMS commands the center of slot position and a test configuration. In the test configuration a fixed delay of the transmitted interrogation pulse is inserted into the receiver channel creating a known pulse return.

4.4.5 Landing Systems

The space shuttle booster avionics system will include an Advanced Instrument Landing System (AILS) receiver and a standard Instrument Landing System (ILS) receiver. The AILS is included to fulfill the zero-zero landing requirements and the ILS to extend the number of landing fields available for use by the booster.

AILS

The AILS system provides azimuth, elevation and range from the vehicle to the end of the runway. Figure 4-23 gives the AILS/DMS interface requirements. The power to the AILS is controlled by two discretes, one for DC and one for AC power. The supply power voltages and electronics temperature is made available to the DMS for monitoring purposes. Range is obtained by the vehicle's AILS transmitting a pulse and receiving a transponded return from the ground station. The operation of the AILS Range equipment is similar to the operation of the TACAN DME. The ground station returns are tracked by a narrow time slot. The DMS has control of positioning the time slot center. The accumulated sum of the position of the returns in the time slot and the total number of returns received are available to the DMS. The number of pulses transmitted by the AILS is also sent to the DMS.

The AILS operates on an iterative bases. Each 1/5 of a second is divided into 6 equal parts, each 1/30 of a second long. During one of the 1/30 second periods elevation data is received, during another azimuth, and during another range. Elevation data is received by the AILS as a series of pulse pairs. The two pulses constituting a pulse pair are separated by 12 microseconds. The separation between each pulse pair is 40 microseconds when the elevation angle is 0 degrees and the separation increases by 8 microseconds for each degree of elevation angle up to 10°.

Azimuth data is also received as a series of pulse pairs. The separation between the two pulses constituting a pulse pair is 14 microseconds when the vehicle is left of the runway center line, and 10 microseconds when right of the center line. The separation between pulse pairs is 40 microseconds when the vehicle is above the extended runway center line, and increases by 8 microseconds for each degree in azimuth that the vehicle is off the center line up to 5° left or right.

A five per second rate is incompatible with the selected DMS iteration rates. A valid discrete is associated with range, elevation and azimuth returns. These discretes are set when data is being received by the DMS at the end of the $1/30^{th}$ of a second period associated with the particular data type. The discrete is reset when the data is read in to the DMS. The discrete thus informs the DMS if data is being received by the AILS and if the data is fresh.

Two discretes are available to command the AILS into one of two different test configurations. AILS operation is tested by comparing range, elevation and azimuth values received with the test configuration commanded against expected results.

ILS

The Instrument Landing System (ILS) is much less sophisticated then the AILS system. The precision of the system disallows a zero-zero landing capability. All major airports have ILS ground installations. The space shuttle booster is provided with an ILS to assist landing operations for those airports where AILS is not available. The ILS provides azimuth and elevation data plus outer, middle and inner marker beacon range data. Continuous range data is obtained if the airport also has TACAN DME. Figure 4-24 shows the ILS/DMS interface requirements. The interface includes AC and DC power on discretes, monitoring of the supply voltages and the temperature of the electronics. The elevation angle is measured with respect to a line inclined at $2\frac{1}{2}^{\circ}$ with the horizontal. The aximuth angle is measured with respect to the extended runway center line. The marker beacons are three transmitters located along the extended runway centerline which radiate a fan shaped pattern. The vehicle ILS delivers a discrete to the DMS as it passes over the marker beacon. If the signal strength being received by the ILS is sufficient for good reception, a discrete is issued to the DMS. The DMS has the capability of commanding the ILS to assume one of two different test configurations. Each test configuration exercises the ILS equipment producing a fixed elevation azimuth and marker beacon output to the DMS.

SIGNAL	SOURCE	TYPE	RANGE	RESOLU-TION	WORD LGTH (BITS)	RATE /SEC.
DC Power on Discrete	DMS	DIS			1	AR
AC Power on Discrete	DMS	DIS			1	AR
DC Supply Voltage	ILS	AN	0-30v	.3v	7	1
AC Supply Voltage, Phase A	ILS	AN	0-127v	1v	7	1
AC Supply Voltage, Phase B	ILS	AN	0-127v	1v	7	1
AC Supply Voltage, Phase C	ILS	AN	0-127v	1v	7	1
Electronics Temperature	ILS	AN	75-125°F	1°F	6	1
Elevation Angle	ILS	AN	±10	.25°	7	1
Azimuth Angle	ILS	DIS			1	1
Outer Marker Beacon	ILS	DIS			1	1
Middle Marker Beacon	ILS	DIS			1	1
Inner Marker Beacon	ILS	DIS			1	1
Test Configuration Discretes	DMS	DIS			1	AR

Figure 4-24 ILS/DMS Interface

AC Power on Command	DMS	DIS			1	AR
DC Power on Command	DMS	DIS			1	AR
AC Supply Voltage, Phase A	RAL	AN	0-127v	1v	7	1
AC Supply Voltage, Phase B	RAL	AN	0-127v	1v	7	1
AC Supply Voltage, Phase C	RAL	AN	0-127v	1v	7	1
DC Supply Voltage	RAL	AN	0-30v	.3v	7	1
Electronics Temperature	RAL	AN	75-125°F	.5°F	7	1
Altitude	RAL	AN	0-80,000Ft	10ft	13	1
Signal Strength	RAL	AN	0-100%	1%	7	1
Bite Output	RAL	AN			20	1
Test Configuration Command	DMS	DIS			1	AR
Oven Temperature	RAL	AN	75-125°F	.5°F	7	1
Heater Command	DMS	DIS			1	1

Figure 4-25 Radar Altimeter (RAL)/DMS Interface

AC Power on Command	DMS	DIS			1	AR
DC Power on Command	WRA	DIS			1	AR
AC Supply Voltage, Phase A	WRA	AN	0-127v	1v	7	1
AC Supply Voltage, Phase B	WRA	AN	0-127v	1v	7	1
AC Supply Voltage, Phase C	WRA	AN	0-127v	1v	7	1
DC Supply Voltage	WRA	AN	0-30v	.3v	7	1
Electronics Temperature	WRA	AN	75-125°F	.5°F	7	1
Oven Temperature	WRA	AN	75-125°F	.5°F	7	1
Heater Command	DMS	DIS			1	1
Test Configuration Command	DMS	DIS			1	AR
Bite Output	WRA	AN			30	1

Figure 4-26 Weather Radar (WRA)/DMS Interface Requirements

4.4.6 Radar Sensors

The space shuttle booster will contain two radar sensors. These are a radar altimeter and a weather radar. The radar altimeter will be used during the final reentry phase, cruise and landing. During reentry the radar altimeter is used to determine when to deploy the air breathing engines and to control the flight path. During cruise the radar altimeter will be used for possible inertial navigation updates (the pilot must supply local terrain elevation through keyboard entry), and in establishing a minimum altitude. During landing the radar altimeter will be used to establish a minimum altitude and an altitude display to the pilot. Figure 4-25 is a list of the Radar Altimeter/DMS interface requirements. The signals include the AC and DC power on commands, and monitoring signals providing measurements of the AC and DC supply voltages to the Radar Altimeter. The temperature of the Radar Altimeter electronics is also returned. The measured altitude above the terrain and the signal strength of the radar returns are provided to the DMS. The radar altimeter will include built-in test equipment (BITE), which will generate a 20 bit status word for testing by the DMS. For ground checkout, the DMS can command the radar altimeter into a test configuration and test all of its outputs to the DMS against expected values. The radar altimeter requires an accurate time source in order to measure altitude. This time source is generated by a crystal controlled oscillator which is mounted in a temperature controlled oven. The DMS controls the oven temperature by turning a heater on and off.

The DMS interfaces with the weather radar only for the purpose of checkout, temperature control, and operation monitoring. Figure 4-26 is a list of the weather radar/DMS interface requirements. The interface signals include an AC and DC power command from the DMS. Supply voltage magnitudes and the electronics and oven temperatures are made available to the DMS. A heater command from the DMS is used to control the oven temperature. The DMS has the capability of commanding the weather radar into a test configuration. In both the test and operational configuration, a built-in test equipment (BITE) output is generated by the weather radar.

4.4.7 Miscellaneous Flight Instruments

There may exist several independent flight instruments which interface with the DMS only for the purpose of ground checkout and inflight monitoring. The interface assumed for this function will be 7 discrete inputs and 14 analog inputs. The discretes will indicate if the associated instrument has been turned on. Power switches for these instruments will be located on the instrument. Each of the analog inputs will be 7 bits long. The inputs will be sampled at a 1 per second iteration rate.

4.5 Flight Control

The primary function of the flight control system is to maintain the vehicle at the desired attitude. The desired vehicle attitude is determined either from the guidance system or directly from pilot commands. Attitude is controlled by commanding torques on the vehicle. The manner in which torques are produced is dependent upon the mission flight phase; during boost thrust vector control of the main rocket engines is used, during coast the reaction jet control system is used, and during reentry, cruise, and landing aerodynamic control is used. A secondary system generally considered part of the flight control system is the throttle control of the cruise air breathing engines. Figure 4-27 is a list of the flight control input/output requirements.

The pilot and copilot are each provided with a sidestick controller used to command pitch and roll rate. A detent switch is closed when either a pitch or roll input to the sidestick is greater than a set value. If the sidestick controller is below the detent value the pilot commands are added to the automatic guidance commands allowing minor variations about the guidance flight path. If the pilot command is above the detent value, the guidance commands are zeroed giving the pilot unobstructed control of the vehicle.

The pilot and copilot both have access to throttle controls for the cruise air breathing engines (six engines are assumed for the booster).

SIGNAL	SOURCE	TYPE	RANGE	RES.	WORD LENGTH	RATE
Sidestick Controller Pitch Rate Command (2)	FCS	AN	±100%	.2%	10	16
Sidestick Controller Roll Rate Command (2)	FCS	AN	±100%	.2%	10	16
Sidestick Contoller Detent Discretes (2)	FCS	DIS			4	16
Engine Thrust Commands (6)	DMS	AN	0-100%	.1%	10	8
Engine Response Signals (6)	FCS	AN	0-100%	.4%	8	8
Throttle Detent Discretes	FCS	DIS			12	8
Rudder Pedal Command (4)	FCS	AN	0-100%	.1%	10	16
Rudder Detent Discretes (2)	FCS	DIS			2	16
Pitch, Roll, Yaw (PRY) Beep Trim	FCS	DIS			12	8
PRY Rate (4)	FCS	AN	±20°/s	.01°/s	12	32
PRY Rate Gyro Test Command (4)	DMS	DIS			4	1
PRY Rate Gryo DC Voltage	FCS	AN	0-30v	.3v	7	1
PRY Rate Gyro AC Voltage (Phase A, B, & C)(4)	FCS	AN	0-127v	1v	7	1
PRY Rate Gryo Temperature (4)	FCS	AN	75-125°F	1°F	6	1
PRY Rate Gyro Power on Discretes (4)	DMS	DIS			2	AR
PRY Rate Gyro Rotor Speed Status (4)	FCS	DIS			1	1
Acceleration (2)	FCS	AN	±256F/S²	.125F/S²	12	32
Accelerometer Test Command (2)	DMS	DIS			2	1
Accelerometer Status (2)	FCS	DIS			2	1
Pitch + Roll Thrust Vector Servo Command	DMS	AN	±5°	.01°	10	32
Pitch - Roll Thrust Vector Servo Command	DMS	AN	±5°	.01°	10	32
Yaw + Roll Thrust Vector Servo Command	DMS	AN	±5%	.01°	10	32
Yaw - Roll Thrust Vector Servo Command	DMS	AN	±5%	.01°	10	32
Twelve (12) Thrust Vector Position Feedback	FCS	AN	±5%	.01°	10	32
Thrust Vector Servo Power On Discretes	DMS	DIS			14	AR
Twelve (12) Thrust Vector Servo DC Supply	FCS	AN	0-30v	.3v	7	1
Thrust Vector Servo Status	FCS	DIS			28	1
Reaction Jet Commands	DMS	DIS			16	32
Reaction Jet Return Status	FCM	DIS			16	32
Elevon Commands (2)	DMS	AN	±100%	.2%	10	32
Rudder Command	DMS	AN	±100%	.2%	10	32
Flap Command	DMS	AN	0-100%	.4%	8	8
Elevon Position (2)	FCS	AN	±100%	.8%	8	1
Rudder Position	FCS	AN	±100%	.8%	8	1
Flap Position	FCW	AN	0-100%	.4%	8	1
Aerodynamic Controls Power On Discretes	DMS	DIS			5	AR
Elevon DC Supply Voltage (2)	FCS	AN	0-30v	.3v	7	1
Rudder DC Supply Voltage	FCS	AN	0-30v	.3v	7	1
Flap DC Supply Voltage	FCS	AN	0-30v	.3v	7	1
Trim Motor Commands	DMS	DIS			6	8
Trim Motor Status	FCS	DIS			6	8
Pilot/Copilot Station Select Switch	FCS	DIS			1	16
Throttle Positions (6)	FCS	AN	0-100%	.1%	10	8

Figure 4-27 Flight Control System (FCS)/DMS Interface

Each individual throttle position is input to the DMS computers. Each throttle has two detent positions, idle and maximum power, which are made available to the DMS as discretes.

The position of both of the pilot's and copilot's rudder pedals are input to the computer. The difference in the rudder pedal position of the controlling station acts as a command to the rudder during aerodynamic flight. A rudder detent position determines if the pilot command is added to the guidance command or if the guidance command is zeroed.

Trim discretes from switch positions at both the pilot and copilot stations are inputs to the DMS. There are two bits from each station for each vehicle attitude, i.e., pitch, roll and yaw. The two bits contain coding for +, - and 0 commands. The DMS issues commands to the trim motors and tests the trim motor status.

There are four angular rate instruments on the booster, one for sensing roll rate, one for yaw rate, and two for pitch rate. Two pitch rate gyros are used for better suppression of body bending in the pitch axis. Each rate gyro output is made available to the DMS. Four test command discretes can be issued to each rate gyro which cause calibrated torques to be applied to the gyro. The ac and dc supply voltages and electronics temperature for each gyro are input to the DMS. The DMS issues ac and dc power on discretes. An indication of proper gyro rotor speed is input to the DMS as a discrete status bit.

The booster contains two linear accelerometers as special flight control sensors. These instruments are mounted at body stations having the most desirable bending characteristics. One instrument is mounted with its sensitive axis normal to the body X axis and in the vehicle pitch plane. This instrument is referred to as the normal accelerometer. The other instrument is mounted with the sensitive axis normal to the body X axis and in the vehicle yaw plane. This instrument is referred to as the lateral accelerometer. The outputs of each accelerometer are input to the DMS. The DMS can command the accelerometers into a test configuration and receives a status indication from built-in test equipment.

It is assumed that the booster has 12 main rocket engines and that each rocket engine is capable of being vectored. A hydraulic servo on each rocket engine is used to vector the engine. Each engine when vectored causes torque about two axis, either pitch and roll or pitch and yaw. The commands to each thrust vector servo is one of four types, either pitch plus roll, pitch minus roll, yaw plus roll, or yaw minus roll. The DMS issues these four commands which are routed to the appropriate engine. Position feedback signals from each thrust vector servo are returned to the DMS for comparison with the commanded position for test purposes. The computer issues power on discretes and receives dc supply voltage values. A two bit status indicator is returned to the DMS from each servo.

The reaction jet system consists of 16 reaction jets. Each jet receives an on-off command from the DMS and returns an on-off status to the DMS.

There are four aerodynamic surfaces controlled by the DMS. These are the right and left elevon, the rudder, and the flaps. An individual command is issued to each surface by the DMS and the position of the surface returned for testing purposes. A power on command is issued by the DMS to each surface servo and the dc supply voltage value returned to the DMS for testing.

4.6 COMMUNICATIONS

The communication subsystem equipment, outlined in Section 3.6 , provides for two-way flow of information between the booster and ground facilities, booster and orbiter, and booster flight crew. The equipment used is comparable to that found in current spacecraft and large commercial aircraft. The autonomy requirement and the desire to simplify flight crew functions for the short booster mission tends to minimize the external communications. However, during the development and qualification tests of the booster, voice, data, and command capability to the ground will be required for all mission phases. This will assist in the ground personnel tasks of monitoring and analyzing booster performance to confirm the design or to determine required modifications (7). For the operational missions, the booster, if unmanned, will require data and command capability from the ground at all times.

This section discusses interface requirements between the communication subsystem equipment and the data bus and data management computers. Interface details developed in this section are then used in later sections for determining total system input/output, computer memory, and software requirement.

Frequencies for the communications equipment discussed will either be given as an exact figure, or by an abbreviation or letter designation. Abbreviations for radio frequencies follow established standards (2).

4.6.1 VOICE COMMUNICATIONS

The primary voice communications equipment are the VHF and UHF transceivers necessary to talk to MSFN or FAA stations. Voice reception is also available on frequencies associated with the navigation and landing aids; AILS, ILS, and TACAN. The audio tones and identification signals associated with these aids are considered as belonging to the voice communications subsystem. Voice communications capabilities of the S-band transponder are described in Sections 4.6.2 and 4.6.3.

It is anticipated that there will be no requirement to digitize voice signals and carry them on the data bus. This will, then, limit DMS functions as regards the voice communication subsystems to testing, monitoring, and control operations. As complete automatic checkout of the voice communication subsystem and the associated antennas could involve signal simulators that surpass the complexity of the equipment being tested (7), it is assumed that only a gross onboard checkout of communication units will be performed. A detailed checkout will be conducted on the ground during the maintenance cycle using adequate test equipment. Preflight and onboard checkout will use the minimum equipment necessary to provide reasonable assurance of satisfactory equipment operations. The booster checkout system will have the capability of performing RF parameter checks, ranging calibrations, and other similar type functions through DMS programs and onboard test equipment (11). Switch and connector checks are also performed under DMS control. Monitoring functions consist of verifying, displaying or recording the positions of on/off switches controls, frequency selectors, headset connections, and antenna or voice recorder switches. Control functions consist of the remote operation of these communications as required by operational modes.

A modification of current Apollo communication equipment (8) will satisfy voice requirements for

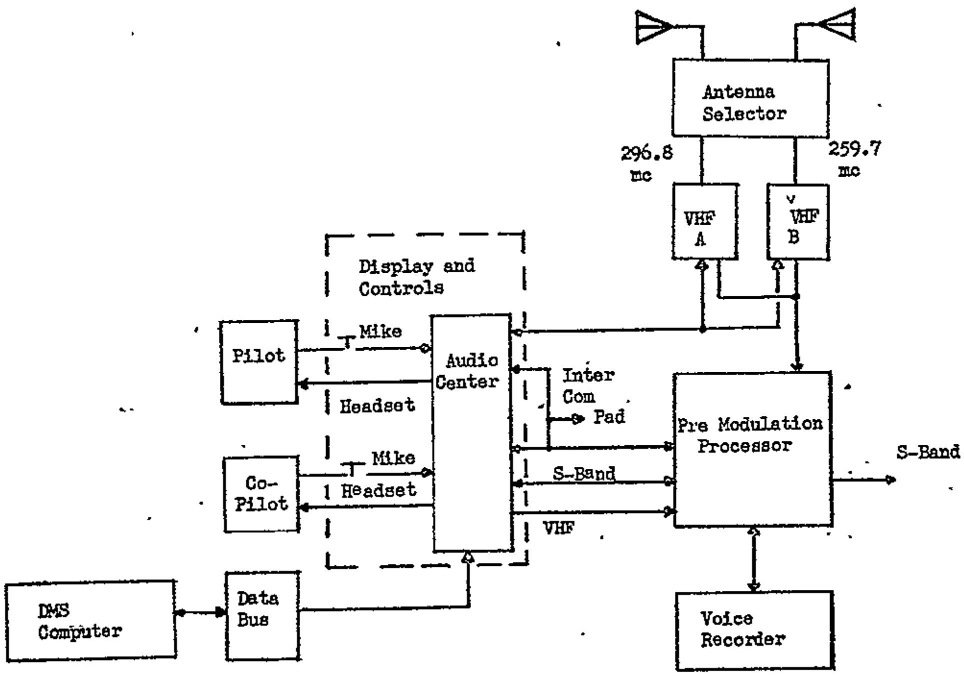
- booster to and from MSFN
- booster to and from orbiter
- intercommunications.

Figure 4-28 shows an arrangement which emphasizes the role of the data bus and the DMS computer in monitoring and controlling communication functions. The following table lists a frequency assignment of VHF A and VHF B transceivers for pre-flight, boost and separation voice:

	From	To	VHF Mode	XMIT	RCV
Primary	Booster	MSFN	Simplex A	296.8	296.8
	MSFN	Booster	Simplex A	296.8	296.8
	Booster	Orbiter	Simplex B	259.7	259.7
	Orbiter	Booster	Simplex B	259.7	259.7
Secondary	Booster	MSFN	Duplex A	296.8	259.7
	MSFN	Booster	Duplex A	296.8	259.7
	Booster	Orbiter	Duplex B	259.7	296.8
	Orbiter	Booster	Duplex B	259.7	296.8

The intercommunication equipment permits in-flight communication between the crew members, and preflight communication between the flight crew and booster servicing personnel. Communication is conducted using hand-held (MIKE 1 and 2) or oxygen mask (MIKE 3 and 4) microphones, and headsets. To use the intercom (I/C), the microphone selector switch is set to I/C and the push-to-talk (PTT) button activated.

Figure 4-28 Voice Communications Block Diagram



SIGNAL		SOURCE	TYPE	RANGE	RESOLU-TION	WORD LGTH (BITS)	RATE /SEC
Voice Communications							
Volume Controls	(12)	CMS	AN	0-28v	.2v	6	1
Mike Audio	(4)	CMS	AN	0 - .5v	.002v	8	1
Radio Master Control	(3)	DMS	DIS			1	1
Headset Jack	(2)	CMS	DIS			1	1
VHF XMIT Cont	(2)	DMS	DIS			1	1
VHF RCVR Cont	(2)	DMS	DIS			1	1
Comm Sel	(5)	DMS	DIS			1	1
Audio Selector	(4)	DMS	DIS			1	1
Norm-Aux.	(7)	DMS	DIS.			1	1
Range Voice Select	(2)	DMS	DIG			2	1
VHF Freq. Select	(3)	DMS	DIG			5	1
UHF Frequency Select	(2)	DMS	DIG			7	1
Comm Test	(5)	DMS	DIS			1	1
MBCN Broad/Sharp	(2)	CMS	DIS			1	1
MBCN Control	(2)	DMS	DIS			1	1
Mike PTT	(4)	CMS	DIS			1	1
Intercom Control	(2)	DMS	DIS			1	1

Figure 4-29 Voice Communications/DMS Interface

An additional VHF communications system will meet voice requirements with MSFN during flyback and is also compatible with FAA enroute and landing facilities. This system provides a capability for 360 channels in the frequency range from 118.0 to 135.95 MHz. It is anticipated that three independent VHF transceivers will be provided: VHF1, VHF2, and VHF3. Control panels for each of these units will become part of the audio center. Current control functions, which will be supervised by the booster DMS, consist of the following:

- COMM SEL switch - Two frequencies can be pre-selected on each VHF unit (making it possible to have a total of six different frequencies selected at one time). On each VHF unit, only one frequency can be used at a time, but the COMM SEL feature permits switching in the other frequency when desired. This eliminates having to dial a new frequency at a critical time.
- Frequency In-use Lights - Indicate the position of the COMM SEL switch (light illuminates on side corresponding to switch position and other light goes out).
- Frequency Selectors - Each of these consist of a window and two knobs. The first digit ("1") of the frequency is permantly marked in the window. The next two digits (18 thru 36) are set by one knob, and the decimal portion (.00 to .95 in increments of .05) is set by the other knob.
- COMM TEST - This pushbutton momentarily disables the squelch control circuits, causing background noise to be heard, thus giving an indication that the receiver section is still functioning.

- NORM - SAT - This switch will have no function for booster operation and should be kept in the NORM position. SAT position pertains to VHF communications via satellite.
- VOL - A knob for volume control of each VHF unit.
- Audio selector - Are On-OFF toggle switches marked to indicate the radio unit to which each pertains.
- NORM - AUX - In case of amplifier failure, moving this switch to the AUX position cuts in a standby amplifier thus restoring interphone communications and radio receiver outputs.
- RANGE-BOTH-VOICE - A three position switch which permits selection of either range or voice signals, or both, from the selected VHF NAV (navigation) receiver.
- Radio Master - Three radio master switches which control the 115V A.C. or 28V D.C. power to (for example):
  - (a) TACAN, Marker Beacon 1, Weather Radar
  - (b) VHF-2, VHF-B, Marker Beacon 2, DME
  - (c) VHF-3, UHF-2, VHF NAV-2, Glide Slope 2

Normally, someunit's power supply is independent of Master Radio switch position (circuit breakers being the only control over power as long as power is supplied to booster systems). These units typically include:

Intercom, VHF A, VHF 1, VHF NAV & GS-1, ATC Transponder

Two UHF units, UHF-1 and UHF-2, are the primary means for voice communications with military agencies. Channels are also available to FAA facilities. This system has a capability of 3500 channels in the frequency range from 225.0 to 399.95 MHz with incremental changes of .05 MHz. Ten of the available channels (those most likely to be used) are preset into each UHF unit.

Controls associated with the UHF communications are:

- CHAN SEL - Selects a preset channel and displays its frequency
- COMM SEL - Same function as VHF comm.
- VOL - Volume control

When either of the VHF NAV sets are in operation and the RANGE-BOTH-VOICE switch is set to BOTH or VOICE, ground to air voice communications or aural identification of VHF navigation facilities being monitored is possible. The characteristics of these received signals are (10):

- The simultaneous ground to air signals are on the same frequency as the ILS localizer(all 50 KHz spaced channels from 108.0 to 119.95).
- The peak modulation depth of the carrier of this communication channel shall not be greater than 40 per cent.
- The audio frequency characteristic of the speech channel shall be flat to within 3 db relative to the level at 1000 hertz over the range 300 to 3000 hertz.
- The localizer provides an identification signal using two or three signals of the Morse code sent at a speed of 7 words per minute.
- The identification signal is repeated six times per minute and its modulating tone is 1020 hertz plus or minus 50 hertz.
- The depth to which the radio frequency is modulated by the identification signal shall be between the limits of 7 and 15 percent.

- The transmission of speech or the identification signal should not interfere with the basic localizer function. When speech is being radiated, the identification signal is suppressed.

- The receivers utilize a two out of five frequency selection system. Frequency selection can be tested and verified by the DMS in comparing A to E pin connections with code requirements for each frequency.

Ground to air communications is also possible from enroute TACAN stations. Each of the 126 TACAN channels is paired with VHF navigational facility in the 108.0 to 117.9 MHz frequency band (10) as in the following example:

TACAN Channel	VHF Channel
97	115.0
98	115.1
99	115.2
100	115.3

DMS tables will provide the complete list of matching frequencies and will set a VHF channel to match the selected TACAN channel.

Vocie signals from AILS ground stations to the booster will be provided on UHF channels associated with each AILS Distanc[e] Measuring Equipment (DME) transponder channel. DMS will provide information on matching frequencies.

As described in Section 3.6, the marker beacon receiver is pretune[d] to receive only a single frequency of 75 MHz, the frequency of airway markers, and ILS outer and middle markers. Both aural and visual indications of aircraft passage over a marker are provided. Current indicators and controls consist of:

- Marker lights - Two sets of indicator lights (one for pilot, one for copilot). Each set consists of an amber, a purple, and a white light. The amber lights respond to ILS middle markers, purple lights to ILS outer markers, and white lights to inner markers or airway markers.
- BROAD/SHARP - A two position switch which governs sensitivity of the marker lights. Normally, a fan marker station is identified when the marker light comes on with the sensitivity control in SHARP position. At the lower altitudes, the light remains on for a relatively short time, but the time it stays on increases with increase in altitude. BROAD position is even more sensitive than SHARP position, receiving signals at a greater distance from the station, and reacting in the same manner with respect to altitude.
- MARKER -ON-OFF control
- VOL - Volume control for beacon audio.

In the ILS, the glide path marker beacons are identified by dots, or dashes repeated at fixed time intervals.

DMS requirements for the voice communications subsystem are summarized in Figure 4-29.

4.6.2 COMMAND SUBSYSTEM

The booster uses, as part of its communication system, a subset of the Apollo S-band communications system. Figure 4-30 lists the typical functional requirements (13) of this Apollo system (used on the Block II Command and Service module), and indicates those capabilities anticipated for inclusion in the booster vehicle. The onboard communications equipment communicates with the Manned Space Flight Network (MSFN) consisting of unified S-band (USB) antenna systems located at remote site data processing (RSDP) units all under the control of Goddard Space Flight Center (GSFC). It is anticipated that sufficient MSFN sites will remain in operation for command and control of shuttle operations (7).

The USB antenna system uses a single carrier frequency in each direction to provide tracking as well as communications with the booster. Communications to both the booster and the orbiter, when mated, can be provided simultaneously within the beamwidth of the single antenna. The system will pack all data acquisition, telemetry, and command control in a single radio frequency band which is 2270 to 2300 MHz for telemetry (down link) and 2090 to 2120 MHz for command (up link). To accomplish this, two sets of frequencies are used, separated by 5 MHZ on each up and down link spectrum.

The remote site data processing organizations have the functions of sampling and handling instrumentation data from the vehicles; accepting, handling, and transmitting digital commands to the booster or orbiter; and processing and displaying mission information. GSFC acts as the primary switching center for the

FUNCTION	PERFORMANCE REQUIRED	BOOSTER REQUIREMENT
Two way Voice	Capability shall exist at all times when in line of sight of station - Normal voice: 90% word intelligibility - Back up voice: 70% word intelligibility	Yes
Telemetry	Continuous capability when in line of sight of station. Uses Pulse Code Modulation (PCM) with design goal of maximum bit error rate of 1 in 1 million.	Yes
Up Data	Continuous capability to transmit discrete commands and complex digital information from earth to spacecraft when in line of sight of station. Probability of rejecting or not properly receiving a correctly transmitted message (decoding, failure) shall be not greater than 1 in a thousand. Probability of accepting an incorrect or false message (decoding error) shall be no greater than 1 in 1 billion.	Yes
Trajectory Measurement	By phase - coherent turn around of pseudo-random noise (RRN) range code. Used when necessary to update trajectory data.	Yes
Television	Justified as a means of giving the public a pictorial real-time account of the progress of Apollo lunar missions.	No
Scientific, Biomedical Telemetry	Three channels. As required by scientific community on a non interference basis to the mission.	No
Data and Voice Dump	Capability to record operational data, voice, and scientific data, and playback stored data when in line of sight of station, simultaneously with real time operational telemetry and voice.	In Flight Play-Back Not Required
Data and Voice Relay	Capability to relay voice or data from other vehicles to earth.	Yes
Emergency Key	Continuous capability when in line of sight of station, in event of equipment failure precluding other communication. At a maximum rate of 25 characters per minute, copy accuracy shall be at least 70%	Yes

Figure 4-30 Apollo S-Band Communications Functional Requirements

flow of data to RSDP's, range instrumentation aircraft, communication satellites and Mission Control Center (MCC), Houston.

The booster S-band equipment, the data flow, and interfaces with the data bus and DMS are shown in Figure 4-31. The functions of the S-band switches are outlined in Figure 4-32. The word lengths (bit requirements) of these switches are also listed. This information is used as a portion of the total S-band and DMS interfaces requirements.

In addition to control and monitoring of switches listed in Figure 4.21, additional up link functions which have an impact on interface requirements include:

- Circuit margin calculations
- Modes of operation control
- Acceptance and management of up-data

The performance of a communication channel depends largely on a judicious choice of modulating parameters (phase deviations, modulation indices, etc.) and system parameters such as transmitter power, receiver sensitivity, and bandwidths. With these parameters, the communication system's performance can be assessed and predicted. The simplest measurement of channel performance

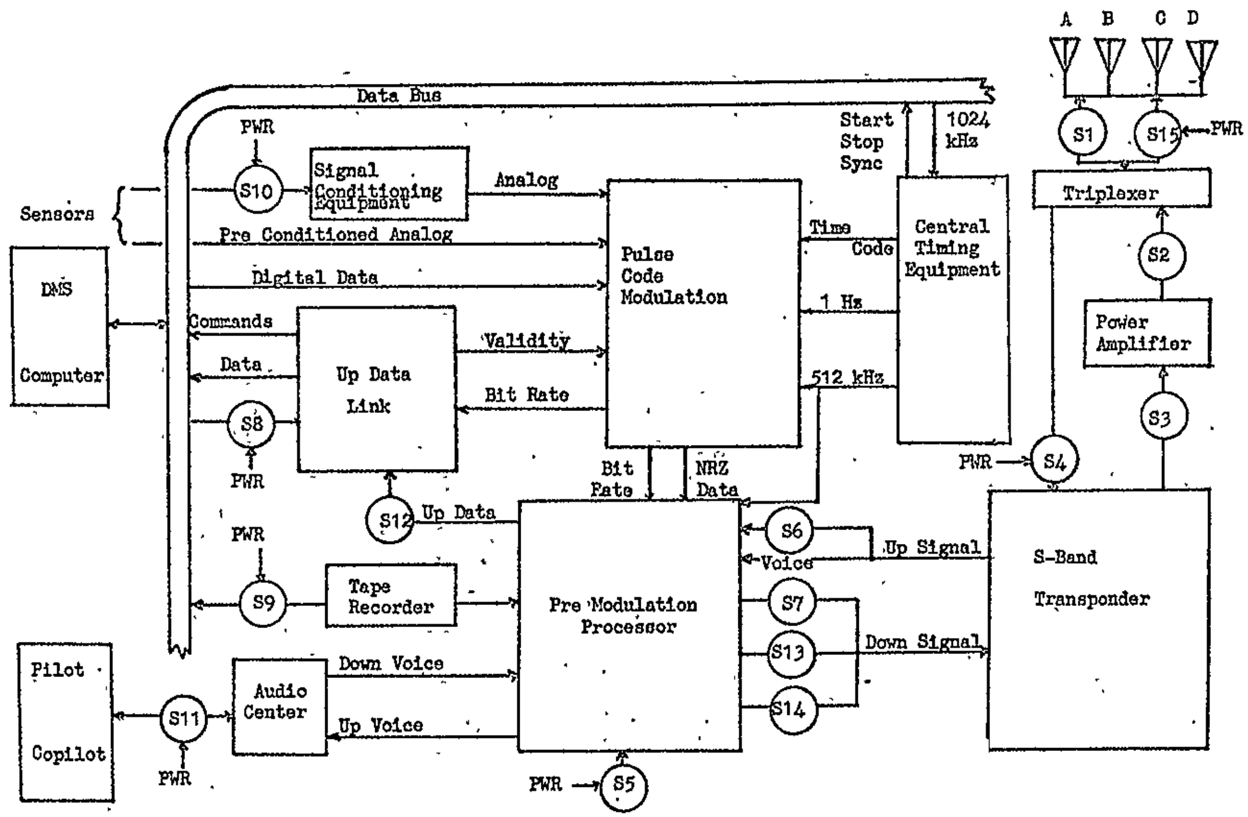


Figure 4-30 S-Band Command and Telemetry Block Diagram

NUMBER	NAME	FUNCTIONAL DESCRIPTION	WORD LENGTH (BITS)
S1	Antenna Selector	Selects one of the four fixed omnidirectional S-band antennas	2
S2	Power Amplifier Control	Controls power - output levels: 5 watts in low power mode or 20 watts in high power mode	4
S3	Power Amplifier select	Select primary or secondary power amplifier unit	1
S4	S-Band Transponder Control	Selects S-band transponder to be used	2
S5	Pre Modulation Processor Power	Controls 28 VDC input to 18VDC power regulator	2
S6	Pre Modulation Processor UP Control	Selects either data or voice back up information from command link. Voice back up is primarily designed for contigency situations	1
S7	Pre Modulation Processor Down Control	Selects down link tape or voice back up information as required	2
S8	Up Data Link Control	Resets the relay assembly associated with the command decoder and controls 28VDC power supply input to up data link unit	2
S9	Telemetry Tape Control	Controls tape motion, circuit enabling, and tape functions	4
S10	Signal Conditioning Equipment Control	Controls power supply to signal conditioning equipment	2
S11	S-Band Operation Control	Selects S-band voice function for pilot or copilot • Controls volume with analog range of 0-28V with resolution of .2V	4 12
S12	Up Data Link Telemetry	Controls up data link decoder outputs to DMS	1
S13	Pulse Code Modulation Telemetry Inputs	Sets telemetry high or low bit rates. Used in conjunction with switches S7 and S14 to select different data modulation levels. Used in conjunction with switch S9 to record telemetry data at high or low speeds	1
S14	Pulse Code Modulation Mode Controls	Used to select primary voice or PCM functions, or secondary relay or key capabilities	4
S15	Ranging	Controls RF track output	1

Figure 4-32 Command and Telemetry Switch Functions

is a comparison of the required signal-to- noise (S/N) ratio in a given bandwidth to yield adequate performance and the actual signal-to-noise ratio in that bandwidth.

The required signal to noise ratio is that S/N necessary to provide the output intelligence of adequate quality. Figure 4-30 defines this quality, for example, in terms of maximum bit error rate for telemetry and miminum word intelligibility for voice. The circuit performance margin (signal margin) is the difference, stated in decibels (db), between the actual signal to noise ratio and the signal to noise ratio required. The equation

$$(S/N)_{\text{actual}} - (S/N)_{\text{required}} \geq 0$$

should be satisfied for sati factory communications performance. The db margin calculations for C-band up signals summarize the system transmitted power, gains, losses, noise spectral densities, predication noise bandwidths, and modulations losses. The equations and the computer program to generate the db margin summaries for each communication mode of operation are described in Section 5.6.2. This program is used during preflight test, and inflight when shifting communication modes or experiencing communications troubles. The parameters required, and their characteristics are listed in Figure 4-33.

Communication modes of operation are designed to aid in conservation of power and optimization of signal margins by transmitting only the intelligence required by the operational situation. Figure 4-34 summarizes the up-link combinations. In mode 6, for example, the PRN code is phase-modulated directly on the carrier and requires approximately 3MHz of bandwidth. The voice is frequency modulated onto a 30kHz subcarrier which in turn is phase modulated onto the carrier.

SIGNAL	SOURCE	TYPE	RANGE	RESOLU-TION	WORD LGTH (BITS)	RATE /SEC
S-Band						
Validity	CMS	DIS			1	AR
Up Data Out of Sync	CMS	DIS			1	AR
Time Word	CMS	DIG			25	AR
GND Word	CMS	DIG			16	AR
RTC Word	CMS	DIG			6	AR
SPC Word	CMS	DIG			27	AR
Test Request Word	DMS	DIG			24	1
CC Assembly ON/OFF	DMS	DIS			1	1
Voltage (6 steps)	DMS	DIG	0-28v		3	1
Calibration Start	DMS	DIS	0-28v		1	1
Enable/Disable	DMS	DIS			1	1
Message Accept	CMS	DIS			1	AR
Frame Sync	CMS	DIG			32	AR
Word Sync	CMS	DIS			1	AR
End of Message	CMS	DIS			1	AR
Serial Start	CMS	DIS			1	AR
Serial Bit Sync	CMS	DIS			1	AR
Serial Stop	CMS	DIS			1	AR
PCM Coder A	CMS	AN	6v	1v	3	1
PCM Coder B	CMS	AN	0	.5v	2	1
PCM Output Reg	CMS	AN	0-6v	1v	3	1
Check Voltage "0"	CMS	AN	0	.5v	2	1
Check Voltage "1"	CMS	AN	6v	1v	3	1
PCM Power Supply	CMS	AN	20.0	2v	5	1
S-Band Switches	DMS	DIG	----		33	1
S-Band Vol Control	CMS	AN	0-28v	0.2v	12	1
DB Margin Summary						
XMIT Power Booster Systems	CMS	AN	0-20w	.1w	9	1
XMIT Power Ground Systems	CMS	AN	0-20kw	.1kw	9	1
Transmitter System Circ. Loss	CMS	AN	-20 to +20db	.1db	9	1
Transmitter Antenna Gain	CMS	AN	-60 to +60db	.5db	8	1
Antenna Pointing Loss	CMS	AN	-60 to _60db	.5db	8	1
Slant Range	DMS	DIG	0-1000n mile	1nm	10	1
Carrier Frequency	CMS	AN	0-2500 MHz	2.5 MHz	10	1
Receiver Antenna Gain	CMS	AN	-40 to +40db	.5	8	1
Receiver Circuit Losses	CMS	AN	-5 to +5 db	.1db	7	1
Antenna Noise Temperature	CMS	AN	200-300K	1k	9	1
Receiver Noise Temperature	CMS	AN	2000-3000K	10k	9	1
Noise Bandwidth	CMS	AN	0-10 MHz	.1MHz	8	1
Modulation Index	CMS	DIG	0-2	.01	9	1

Figure 4-33 Communications System (CMS)/DMS Interface

MODE	INFORMATION	MODULATION TECHNIQUES	SUBCARRIER FREQUENCY
1	Carrier Ranging	PM on carrier	-
2	Carrier voice	FM/PM	30 kHz
3	Carrier up-data	PSK/FM/PM	70 kHz
4	Carrier ranging Voice	PM on carrier FM/PM	- 30 kHz
5	Carrier ranging Up-data	PM on carrier PSK/FM/PM	- 70 kHz
6	Carrier ranging Voice Up data	PM on carrier FM/PM PSK/FM/PM	- 30 kHz 70 kHz
7	Carrier voice Up-data	FM/PM PSK/FM/PM	30 kHz 70 kHz
8	Carrier Voice Back-up	FM/PM	70 kHz

34

Figure 4-34 UpLink Modes

1	Carrier Voice 51.2 KBPS TLM	FM/PM PCM/PM/PM	1.25 MHz 1.024 MHz
2	Carrier PRN 51.2 KBPS Voice	PM on Carrier PCM/PM/PM FM/PM	1.024 MHz 1.25 MHz
3	Carrier PRN Voice 1.6 KBPS	PM on Carrier FM/PM PXM/FM/PM	- 1.25 MHz 1.024 MHz
4	Carrier Voice 1.6 KBPS	FM/PM PCM/PM/PM	1.25 MHz 1.024 MHz
5	Carrier 1.6 KBPS	PCM/PM/PM	1.024 MHz
6	Carrier Key	AM/PM	512 kHz
7	Carrier PRN	PM on Carrier	-
8	Carrier Backup Voice 1.6 KBPS	PM on Carrier PCM/PM/PM	- 1.024 MHz
9	Carrier PRN 1.6 KBPS	PM on Carrier PCM/PM/PM	- 1.024 MHz

Figure 4-35 Down Link Modes

35

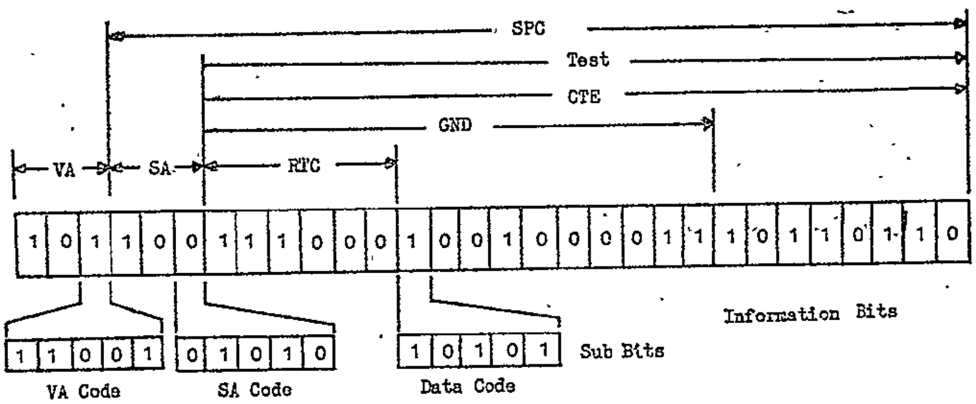
Similarly, the up-data is frequency-modulated onto a 70 kHz subcarrier and then phase-modulated onto the carrier. Mode 8 is an emergency mode in which the backup voice is frequency-modulated onto the up-data channel and then phase-modulated onto the carrier. A computer program correlates communication mode and equipment configuration with the booster operational mission modes.

The acceptance and management of up-data requires a knowledge of command word structure, transmission rates and specific types of commands expected during the booster mission. It is assumed that the command word structure will be similar to the Apollo/Saturn format (8) shown in Figure 4-36 with the information bit interpretation adapted to shuttle mission requirements.

In order to ensure that nonvalid command rejection ratios are met, each information bit is encoded into five sub-bits by the RSDP ground station prior to transmission. The sub-bit scheme shown in the command word structure of Figure 4 -36 is one of the possible arrangements. Here the VA code is different from the SA and data codes. The SA and the data codes are the same with a "1"sub-bit code the complement of the "0" sub-bit code.

Data is transmitted to the vehicle at a rate of 200 information bits per second or 1000 sub bits per second. A phase shift keyed (PSK) frequency modulation (FM) technique is used to transmit the data. The transmitted signal is composed of a reference and an information signal. The information is in phase with the reference for a "1" and 180 degrees out of phase for a "0". Figure 4-37 shows the major equipment of the up-data link unit of the S-band system. The sub bit detector examines five bits and, if coding requirements are met, sends a "1" or "0" to the decoder unit. If the information tone does not match the reference tone, an out of synchronization

Figure 4-36 Command Word Structure



VA;	Vehicle Address	GND;	Guidance and Navigation Data
SA;	System Address	CTE;	Central Timing Equipment
RTC:	Real Time Command.	SPC;	Stored Program Command

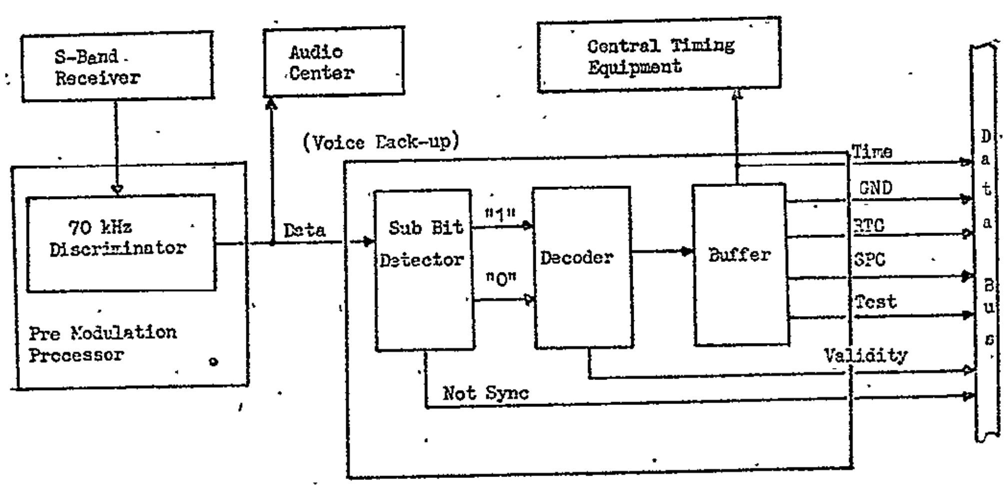


Figure 4-37 Up Data Link Processing

condition is detected and the DMS is notified to initiate recovery procedures. The decoder unit assembles the bits according to codes, and shifts them into the buffer storage until the end of message at which time the information bits are put into the data bus along with a valid message discrete. If coding format requirements are not satisfied, then the DMS initiates a request for retransmission of message. Upon receipt of a valid message, the DMS performs reasonableness tests before telemetering a message acceptance discrete to the ground station.

The command word is configured so that each command will only be accepted if it has the correct vehicle address, systems address, bit structure, and word length. The vehicle address code could be used for example to distinguish between the master, front, middle and rear data processors of the orbiter and booster (7). The system address is used to distinguish between the RTC, GND, TEST, CTE and SPC codes. The additional bits are data bits having a predetermined meaning.

Guidance and navigation (GND) data are commands designed to provide updated information to the selected vehicle. The GND codes can be interpreted as keyboard symbols. The corresponding keys can then be activated to update or vary computer memory contents.

The Test code is a request from ground stations for information such as self-test results, computer memory contents, or status of indicated equipment.

Whenever the ground station receives an indication that the booster's timing system is not accurate (possible in reply to a Test request), the ground station will provide correct timing data to the central timing equipment (CTE) and the DMS. The CTE code has the appropriate number of bits assigned to give seconds, minutes, hours, and days information.

Real time commands (RTC) are commands used to control booster equipment. They are of the ON/OFF variety, such as: Landing Gear UP/DOWN, PCM Mode VOICE/RELAY, Power Amplifier HIGH/LOW, or Ranging ON/OFF. A stored program command (SPC) will incorporate timing and computer instructions. For example: Leave holding pattern at 123452 GMT. (This example illustrates the assumption that MSFN or a FAA system equivalent to USB is available throughout the mission - including control of an unmanned landing).

Figure 4-33 lists interface requirements for both the up and down link data functions. Section 5.6 describes the computer programs required for the validation, processing, and management of uplink information.

4.6.3 TELEMETRY

The telemetry subsystems uses the S-band equipment shown in Figure 4-31 Data flow follows the down signal paths indicated in the diagram. Figure 4-32 shows the combined requirements for command and telemetry switch functions. Analogous to the information provided on up-link functions in section 4.6.2 , this section provides details on the down-link characterstics of

- S-band Down-link spectrum
- Margin Calculations for S-band Down Signals
- Down Link S-band Modes of Operation

Following a discussion of telemetry formatting, synchronization, and data requirements, a list of combined up and down link interface specifications is provided.

The design of spectra to accomodate multiple channels is based on consideration of channel performance, interferences, and equipment limitations. The down link is generally power limited in comparison with the up link. The pseudorandom noise range code (PRN) is transmitted as baseband, directly phase modulating the carrier. It is important to locate the voice and telemetry intelligence in the spectrum for minimum interference to and from the range code. The use of subcarriers permits control of the carrier power reduction and provides the necessary isolation among range code, telemetry, and voice. The optimum location of the subcarriers is in the first null (at 1 MHz) of the range code. The digital PCM, being more susceptible to interference, is placed close to the first null at 1.024 MHz where the total PRN code power is relatively low. The voice subcarrier is located at 1.25 MHz.

The margin calculations for S-band down link signals are shown in Figure 4-38. The discussion in section 4.6.2 of circuit margin calculations and parameters for db margin summaries applies to down link transmissions.

The down link S-band modes of operation are listed in Figure 4-35 . . All these modes are made available to conserve power by using only that spectrum necessary to satisfy the requirements at the time, and to assure at least some limited form of communications in the event of degraded system performance (caused, for example, by failure of a power amplifier or an antenna). Mode 2 is the primary high -data mode designed for critical phases of the mission. In this mode, the PRN code phase modulates the carrier, and the voice frequency modulates a 1.25 MHz subcarrier which then phase modulates the carrier. The telemetry data is a 51.2 kilobit per second (KBPS) PCM wave train which phase modulates a 1.024 MHz subcarrier, which in turn phase modulates the carrier. Mode 3 has essentially the same function with only a 1.6 KBPS telemetry stream on a 1.024 MHz subcarrier. Mode 4 is a reduced activity mode where a minimum amount of data is required by the ground stations for monitoring. The other modes are possible transmission combinations which can be used to optimize circuit margins in contingency situations. The emergency key capability is provided by amplitude modulating a 512 kHz subcarrier which phase modulates the carrier. The backup voice capability is provided by directly phase modulating the carrier with the backup voice signal. (It is anticipated that the orbiter will, in addition to the phase modulated modes of transmission, have a requirement for frequency modulated (FM) modes of S-band transmission. As these modes apply to television, biomedical, and recorder playback transmissions, it is assumed FM communication modes are not required for the booster.)

Booster to MSFN

Transmitter Power (20w)	13 dbm
Transmitter Circuit Loss	6 db
Antenna Gain	-3 db
Polarization Loss	3 db
Path Loss (2500 nm at 2282.5 MHz)	172.8db
Receiver Antenna Gain	44 db
Receiver Circuit Loss	1 db
Noise Spectral Density	204 dbw/Hz
Composite S/N Ratio	75.2 dbw/Hz

Down Voice	
Composite S/N Ratio	75.2
Modulation Loss	9.8
Bandwidth (20kHz)	43.0
Received S/N Ratio	22.4
Desired S/N Ratio	10.0
Signal Margin	14.4

Down Telemetry	
Composite S/N Ratio	75.2
Modulation Loss	9.8
Bandwidth (150kHz)	51.8
Received S/N Ratio	13.6
Desired S/N Ratio	13.0
Signal Margin	.6

Down Range Code	
Composite S/N Ratio	75.2
Modulation Loss	18.7
Bandwidth (1.5 MHz)	17.0
Received S/N Ratio	39.5
Desired S/N Ratio	6.0
Signal Margin	33.5

Down Carrier	
Composite S/N Ratio	75.2
Modulation Loss	12.6
Bandwidth (800 Hz)	29.0
Received S/N Ratio	33.6
Desired S/N Ratio	12.0
	21.6

Figure 4-38 Margin Calculations for S-Band Down Signals

Pulse code modulation (PCM) telemetry is used because of its many advantages: excellent noise rejection properties, very high data accuracy, digital format versatility in handling different data rates, and high bandwidth efficiency. PCM uses time division multiplexing procedures. Following signal conditioning, multiple data channels are sampled sequentially on a repetitive basis for transmission over a single channel. Some definitions applicable to time division multiplexing are: (12)

- Commutation: Sequential sampling, on a repetitive time-sharing basis, of multiple data sources for transmission on a single channel.
- Frame: One complete commutator cycle, including synchronizing signal.
- Frame rate: Number of frames per second.
- Commutation rate: Number of commutator inputs sampled per second.
- Channel sampling rate: Number of times an individual channel (time slot) is sampled per second.
- Sub commutation: Commutation of a channel or time slot to carry a number of parameters that are sampled at a rate that is slower then the frame rate.
- Super commutation: Commutation at a higher rate to carry parameters sampled faster than the frame rate.
- Main frame: Each parameter is transmitted in a distinct time slot referenced to a synchronizing word or channel. A group of these parameters, with its sync words, is called a main frame.
- Master frame: A set of main frames long enough to include one cycle of the slowest sub commutated channel.

Many combinations of the various levels of commutation are possible and have been used in spacecraft applications. Figure 4-39 illustrates the principles involved (16), and is a combination which fits this study's use of sampling rates of powers of 2. Another example is the Gemini format. Apollo formats are designed to output the data in a master frame in one second (51,200 or 1600 bits). The booster telemetry programs will have the capability to shift telemetry formats as required by communication modes and ground station capabilities.

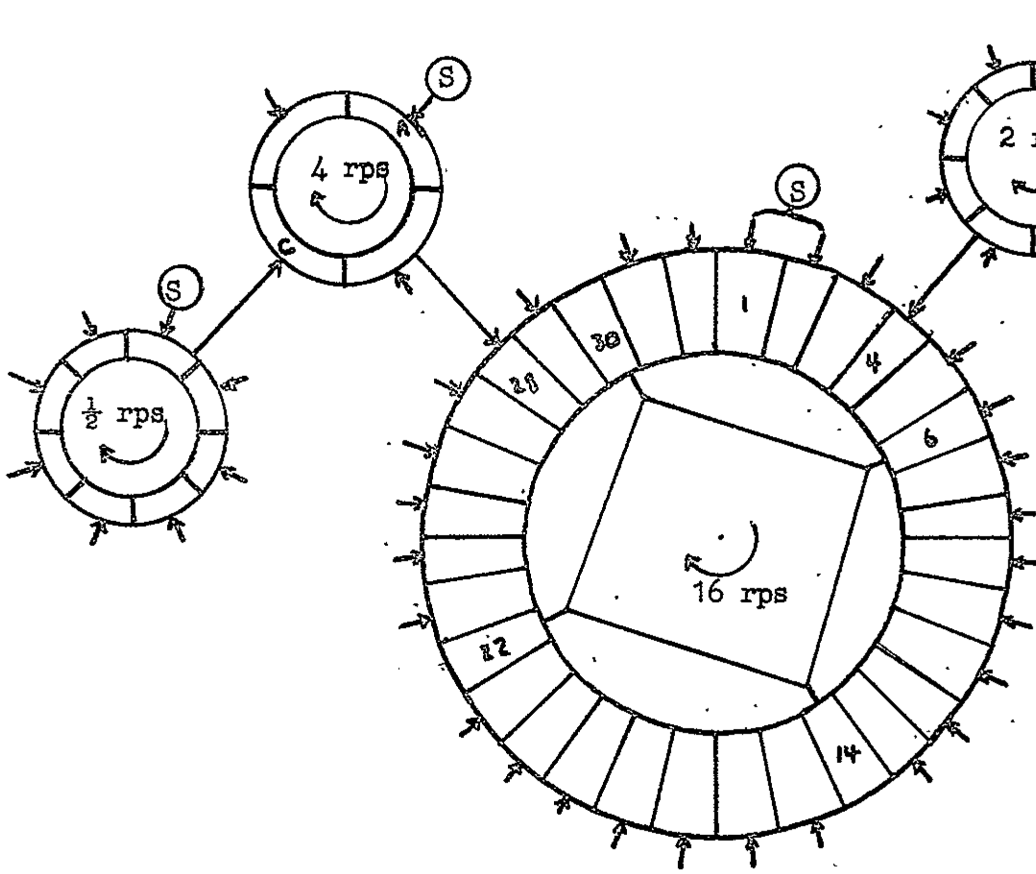
Three elements of synchronization are necessary for PCM systems:

1. Frame synchonization word which marks the beginning of the frame.
2. Word synchronization, provided by a synchronizing bit which marks the beginning of a word.
3. Bit synchronization, required to mark the time for identifying the sense of the digit , and implicit in the pulse train.

Computer programs recognize synchronization discretes or signals and control inputs into the telemetry bit stream as required by the selected format:

- Apollo high speed - 51,200 bits in 1 second
- Apollo low speed - 1600 bits in 1 second
- Gemini - 122,880 bits in 2.4 seconds
- Model - 16,384 bits in 2 seconds

Also associated with the control of down link data is the construction and monitoring of a map for each frame. This map indicates the location of sync words, digital information from the data bus (and its original form of discrete, serial, or analog), and parallel digital or analog words not used by the DMS. A typical example of this map, required for ground decommutation purposes, is shown in Figure 4-40.



S Synchronizing words

Main Frame	32 words (16 frames per second)
Master Frame	64 frames (2 seconds long)
Supercommutation	One 4-word superframe (words6,14,22,30 are tied together to sample sensor 6). 64 words/sec.
Subcommutation	One 8-word subframe (to channel 4) 2 subframes/sec
Sub-subcommutation	One 8-word sub-subframe (to channel 28) 4 subframes/sec.

Figure 4-39 Model for PCM Telemetry Format

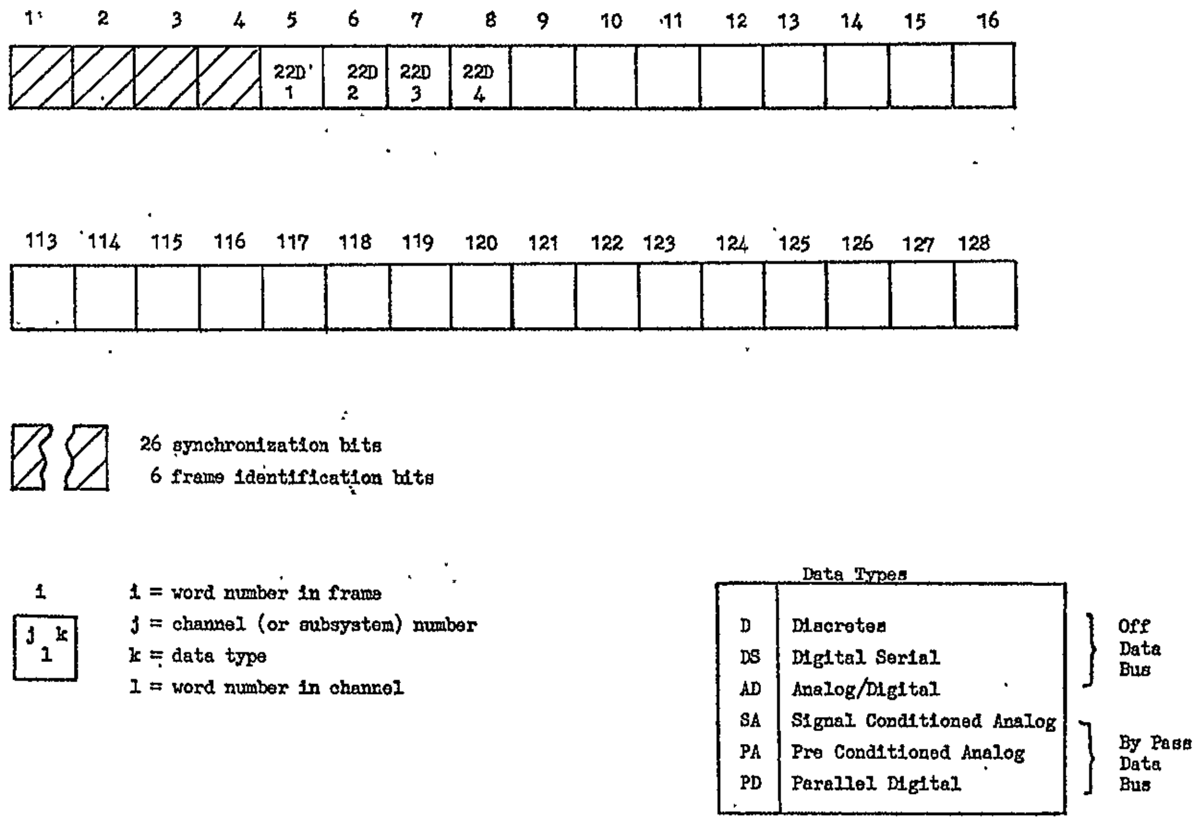


Figure 4-40 Typical PCM Format for a Telemetry Frame

To minimize the total down link requirements, various data compression algorithms are used. Data compression routines may be grouped into three general classes: (1) compression by an encoding or curve fitting method whereby the data may be reconstructed after compression, (2) compression by computing some statistical properties of the data (such as mean or variance) and transmitting this statistical property only, and (3) compression by complete data reduction (performing computations onboard and transmitting only the results of a particular test). The optimal routines are used as dictated by DMS functions.

Telemetry calibration is used during flight and prelaunch checkout to assist in determination of telemetry equipment accuracy. Computer routines can control the Calibrator - Controller assembly which furnishes the calibration inputs to the telemetry components. Control signals under DMS supervision consist of: Calibrator-Controller Assembly ON, Select Calibration voltage (6-steps), Calibration Start, and Calibration gate Enable/Disable.

Figure 4-33 lists the interface requirements for the S-band command and telemetry subsystem. Section 5.6. describes the programs used to:

- Control the formatting and timing of down link data
- Compress data to be telemetered
- Test and calibrate telemetry equipment
- Control communication modes

4.6.4 RECORDING

The recording subsystem uses currently available magnetic tape equipment and techniques to record

- Voice communications from the flight crew to a ground station, to the orbiter, or between the booster pilot and copilot
- Down link data transmitted by the telemetry subsystem for post flight verification of S-band equipment
- Flight data required to satisfy the regulation of the Federal Aviation Administration
- Maintenance data to assist in reconstructing flight history and performing post flight diagnostics
- Flight qualification data to verify structural strain, and vibration parameters for initial flights
- Standard check lists and special symbols for display subsystems

  — in contrast to the above information which is recorded in flight and read back after flight, the display data is assembled prior to flight and is used during preflight and inflight operations. This mass storage function is included in this section in order to combine the read/write and other characteristics of tape system.

For initial booster flights, flight qualification data will require the use of an assigned tape transport while the telemetry, maintenance and flight data records will time share the other tape unit. When the need for flight qualification data becomes minimal, the tape units used for recording digital data can be time shared among the existing requirements.

Recent proposed rules of the Federal Aviation Administration have recommended changes to Federal Aviation Regulations which would: (26)

(1) Increase the recorded flight data required for large jet aircraft certified for flight above 25,000 feet. Figure 4-41 lists these requirements.

(2) Require a device which automatically prevents data erasure after crash impact on flight recorders which erase and re-use tape

(3) Require a device to assist in the location of flight recorders under water

(4) Require a means to correlate the time of recorded data with the time of radio communications between the airplane and air traffic control

Current large commercial aircraft use a cockpit voice recorder system as an aid for incident or accident investigative purposes. It is anticipated that the booster's voice communications will be recorded on a similar system. Inputs to the magnetic tape of the recorder are the pilot's and copilot's microphone inputs plus the general cockpit sound inputs from an area microphone contained in the voice recorder control panel. The control panel is normally located on the overhead panel and includes an area microphone, test meter, test switch and erase switch. The test meter has a pointer and a dial with a green range. When the test switch is pushed, the meter point reads in the green range to indicate that the three input channels

SIGNAL	SOURCE	TYPE	RANGE	RESOLU-TION	WORD LGTH (BITS)	RATE /SEC
Flight Recorder						
Elapsed Time	DMS	DIG		± 4 sec	13	1
Altitude	DMS	DIS	-1000ft - max.	± 100ft	11	1
Airspeed	DMS	AN	60knots-1.2V_D	± 3knots	9	1
Vertical Acceleration	DMS	AN	-3g - 6g	± .1g	7	8
Heading	DMS	AN	360°	± 2°	8	1
Angle of Attack	DMS	AN	-20° - 40°	± .5°	8	2
Pitch Attitude	DMS	AN	± 90°	± 1°	7	1
Pitch Rate	DMS	AN	± 30°/sec	± 3°/sec	5	1
Roll Attitude	DMS	AN	± 180°	± 2°	7	1
Roll Rate	DMS	AN	± 180°/sec	± 3°/sec	6	1
Yaw Attitude	DMS	AN	± 30°	± 2°	5	2
Yaw Rate	DMS	AN	± 190°/sec	± 3°/sec	6	1
Pitch Trim	DMS	AN	Full Range	± .5°	8	1
Pitch Control	DMS	AN	Full Range	± 1°	7	2
Lateral Control	DMS	AN	Full Range	± 2°	6	1
Yaw Control	DMS	AN	Full Range	± 1°	7	2
Engine Thrust	DMS	AN	Full Range	± 2%	6	1
High Lift Devices	DMS	AN	Full Range	± 3°	5.	1
Ambient Air Temp	DMS	AN	-60°c - 55°c	± 2°c	6	1
Voice Recorder						
Parking Brake ON/OFF	DMS	DIS			1	AR
RECORD/PLAY/OFF Switch	DMS	DIG			2	AR
Microphone Closure (2)	RCS	DIS			1	AR
Magnetic Tape Unit						
Function	RCS	DIG			19	
Identifier	RCS	DIG			8	
Record Data	DMS	DIG			16	
Status	RCS	DIG			14	
Busy	RCS	DIS			1	
Read Data	DMS	DIG			16	
Primary Power	RCS	DIS			1	
Tape Handler Power	RCS	DIS			1	
Handler Control	DMS	DIG			2	
Cabinet Temperature	RCS	AN	20-70°C	1°c	6	1
Blower Monitor	RCS	DIS			1	
Alarm Control	RCS	DIS			1	
Model Switch	DMS	DIG			2	
DC Power #1	RCS	AN	-26.5vdc	2.7	4	1
DC Power #2	RCS	AN	26.5	2.7	4	1
DC Power #3	RCS	AN	15.0	0.8	6	1
DC Power #4	RCS	AN	-15.0	0.8	6	1
DC Power #5	RCS	AN	-4.5	0.1	7	1

Figure 4-41 Recorder System (RCS)/DMS Interface

Figure 4-42 shows a block diagram of the magnetic tape subsystem (MTS). The data prepared for magnetic tape storage is recorded using standard recording techniques. Figure 4-43 lists characteristics of the MTS assumed for this study.

In addition, a function and a status word are required to describe the characteristics and capabilities of the tape system. The DMS issues commands to the MTS by means of the function word and control signals. The MTS inspects the function word and performs the specified operations. These operations are of five basic types: read, write, backspace, rewind, and master clear. These basic commands are modified by format, address and other designators. Figure 4-44 describes the characteristics of the function word option. These characteristics are the MTS repertoire of instructions assumed for this study.

At the end of each function performed by the MTS, the MTS control unit will form a status word and put it on the data bus for interpretation by DMS programs. The bit structure of the status word enables the computer program to determine the status of the magnetic tape unit and whether or not the requested operation was completed successfully. If errors occur, then MTS routines will perform the appropriate recovery procedures. Figure 4-45 describes the conditions indicated by the status word.

Figure 4-41 lists the MTS and data bus interface requirements. Computer programs are required to read check list and other information into display memory, and to record telemetry, flight data, maintenance, and flight qualification data. Computer programs are also used for testing the functional capabilities of the MTS.

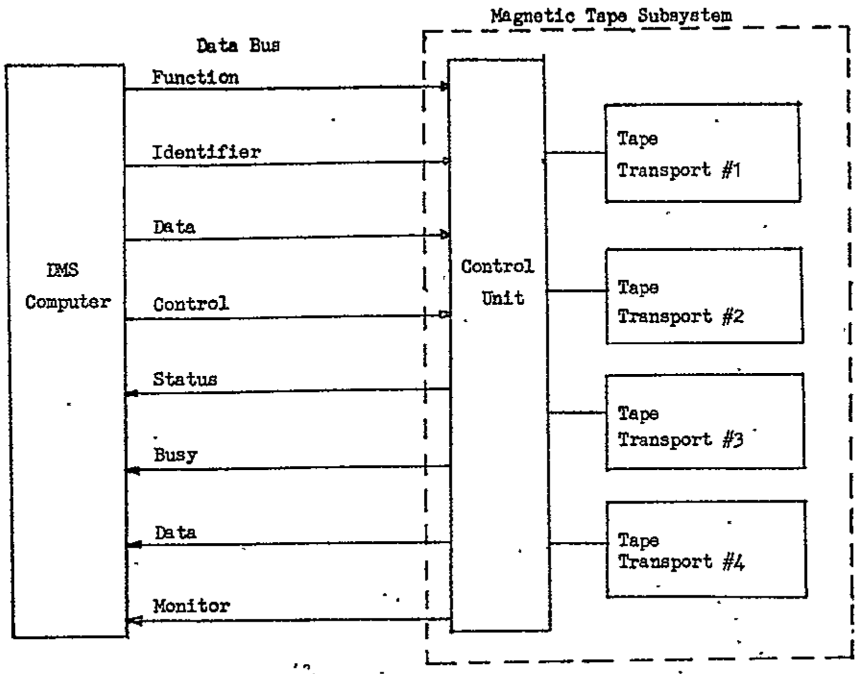


Figure 4-42 Magnetic Tape Subsystem Block Diagram

Modes	Read/Write
Tape Speed	80 inches per second
Density	800 bytes per inch/200 bytes per inch
Interblock Gap	0.6 inch
Format	9-track NRZI
Start-Stop Time	5 milliseconds
Tape Length	2400 feet
Rewind Time	less than 180 seconds
Transfer Rate	64,000 8-bit bytes per second (maximum)
Data Capacity	3×10^8 bits

Figure 4-43 Magnetic Tape Characteristics

FUNCTION	OPERATION
Master Clear	Stops tape motion (except rewind). Sets the tape unit in ready state to accept a new function word. Used when MTS is in illogical state.
Read Forward	Reads one recording according to the format stated in function word. Moves tape in forward direction and transfers 8-bit frames from the tape to MTS control unit. The frames are checked for parity and assembled into proper bit stream for placing computer words on the data bus.
Read Backward	Reads one record backward to the next interrecord gap (back one record) according to format in the function word. Characters are assembled in the same position as in a forward read. Computer words are transmitted in reverse order.
Write Normal Gap	Moves the tape forward and records the normal interrecord gap (IRG). Control unit disassembles the data bus information to be written. If the request is answered, the writing process is continued. Otherwise, an end of write is recognized and recording is terminated. CRC and LRC characaters are added to assist in the reading process.
Write Extended Gap	The selected transport records an extended interrecord gap (XIRG). Other operations are the same as in the normal write.
Backspace	Tape moves in the reverse direction one record. The tape is then properly positioned in the IRG for reading or writing. Parity is checked while backspacing and, if an error occurs, noted in the status word.
Rewind	Rewinds the tape backward to the load point at rewind speed.
Repeat	Indicates to the MTS that tape motion should continue in anticipation of a future command of like function, format and transport selection. Used with read and write, the repeat permits handling more than one record at a time. Used with rewind, the repeat provides automatic recovery as the tape is rewound to the load point then continues to read the first record.
Search Forward	Reads records from the tape in the forward direction and compares the first word of each tape record with an identifier word which is transmitted to the MTS control unit. When a compare is affirmative, the record found is sent to the DMS as in the read forward operation.
Search Backward	Reads records from the tape in the backward direction and compares the first word encountered (the last word of each record) with an identifier word. When a find is made, the read backward operation is performed.
Request Transport Status	No tape operation is performed. MTS sends status word.
Rewind-Clear Write Enable	Performs a normal rewind of the tape to load point and clears the write enable. Selected transport can no longer perform a write function without manual intervention.

Figure 4-44 Magnetic Tape Function Word Instructions

CONDITION	DESCRIPTION
Improper Condition	The improper condition bit is set whenever • Tape transport is not in ready condition • Forward command issued when tape is positioned at end-of-tape • A reverse command is sent when tape is at load point • A write instruction is issued to a tape transport that has no write enable
Transport Ready	The transport ready bit is set if • Power is ON • Magnetic tape reel is mounted and tape is properly loaded • Tape mark detector lamp is operating
No Write Enable	Informs DMS that attempt was made to write on tape with a write enable lockout
XIRG Detected	Indicates that extended interrecord gap has been sensed during the tape read operation
Output Timing Error	Indicates that data, identifier or code words not received during preset time period following receipt of function word by MTS control unit
Input Timing Error	Indicates that DMS has failed to accept a word placed on the data bus by the MTS control before the next word is ready for transmission to the DMS
Incorrect Frame Count	Indicates that the final word of the record was incomplete. Caused by: • One or more characters not properly read or recorded • Bad spots in the tape caused characters to be lost • Reading a record with the wrong format
CRC Error	The Cyclic Redundancy Check (CRC) character is written at the end of each tape block for the possible recovery of single track errors. If a track in error indication is encountered, the CRC error bit is set
LRC Error	The Longitudinal Redundancy Check (LRC) character is written following the CRC as an aid in the detection of read errors. A longitudinal redundancy check bit is written in a track if the longitudinal count is otherwise odd.
Last Tape Motion	At the completion of an operation, the status word indicates the direction of last tape motion (forward or backward).
End of Tape	When the end of tape reflective marker is sensed end of tape bit is set
Low Tape	A pressure-sensitive detector has sensed less than 100 feet of tape remaining on the selected transport reel
Load Point	Indicates that an operation requesting backward motion of the tape is being attempted with the selected tape positioned at load point
Parity Error	A parity bit is formed in each character frame. If this vertical redundance check is in error, the parity error bit is set.

Figure 4-45 Magnetic Tape Status Word Indicators

4.5 BEACONS

The existing air traffic control (ATC) system gathers data on the position of the aircraft by means of an L-band secondary radar system. This system, an outgrowth of the identification, friend or foe (IFF) systems, is known as the air traffic control radar beacon system (ATCRBS). Aircraft transponders are presently required in positive controlled airspace, and the airspace in which transponders are becoming mandatory is being increased. In addition, improvements to the present system are being planned to prevent overload and saturation of current ATCRBS facilities (3).

Characteristics of the present ATCRBS are described in Section 3.5 and in references 2 and 4 . ATCRBS is a radar beacon system in which a ground interrogator transmits a pair of time-coded pulses from a highly directional antenna at 1030 MHZ. The booster transponder in turn replies at 1090 MHZ. The reply consists of up to 14 codes, and two framing, pulses radiated non-directionally. This reply is received by the ground facility and is transmitted to its associated display and digital signal processing equipment. The system is used in conjunction with the primary (skin return) radar. In order that simultaneous presentation of beacon and radar information can be achieved on the same control center display, the radar and beacon antennas are mounted on the same pedestal at the radar site. Range correlation is achieved by transmitting the beacon interrogations in synchronism with the radar at a predetermined time before the radar time, to allow for the required beacon processing time. Booster azimuth information is derived from an antenna mounted azimuth pulse generator.

Interrogation pulse-pair combinations (P_1 and P_3) are shown in Figure 4-46. These are referred to as transponder modes. A pulse (P_2) is transmitted on a omnidirectional antenna in conjunction with the directional pulse pair. P_2 is designed to suppress aircraft responses to interrogator sidelobes, and its effective radiated power is greater than any sidelobe at any azimuth. Transponder modes 3/A and C are used extensively for FAA operated systems. The other modes are used at military or civil sites, or are available for expansion purposes.

Any detected pulse pair which has the correct spacing will cause the booster transponder to reply with a code containing the requested data, identity or altitude. Figure 4-47 defines the pulse nomenclature and spacing, and lists the pulse values for an identification reply code. The delay between the receipt of the interrogation and the transmission of the reply is 3 microseconds. A special position identification pulse is manually initiate by the pilot at the request of the ground controller to resolve ambiguity between identity codes.

Improvements in ATCRBS will evolve from the current beacon system and will have a high degree of compatibility with the present system. Reply requirements from airborne transponders will include azimuth and range information as well as identity and barometric altitude codes.

The identification code functions is independent of the DMS, except for testing. The primary function of the DMS as related to ATCRBS is to encode altitude, range and azimuth information and make it available to the ATC transponder when a request is sent by ground facilities.

Mode	Application
1	Military (IFF)
2	Military (IFF)
3/A	Common (ATC)
B	Civil (ATC)
C	Altitude
D	Expansion (V)

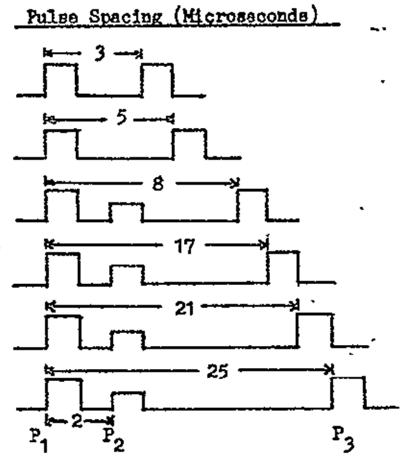


Figure 4-46 Air Traffic Control Radar Beacon System (ATCRBS) Interrogation Modes (UP Link)

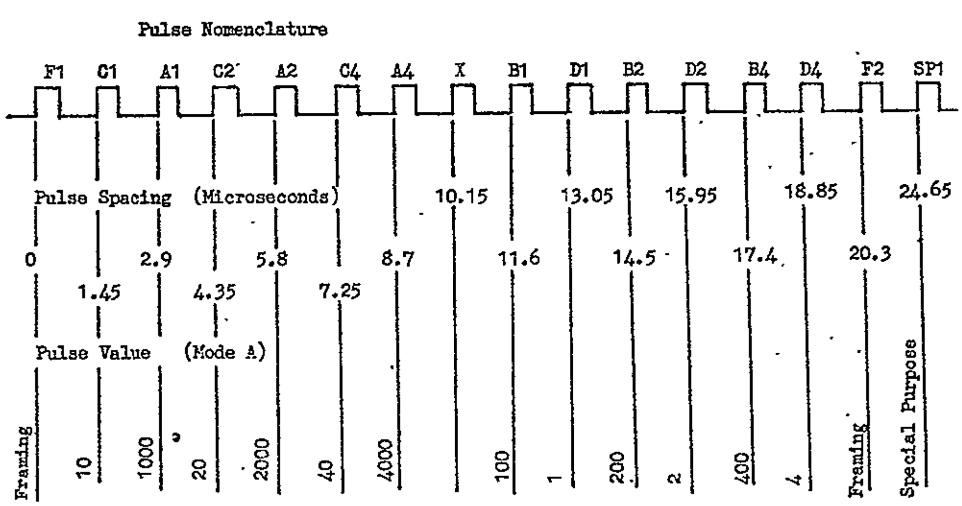


Figure 4-47 Transponder Reply Codes (Down Link)

Altitude accuracy (3) is limited by the quality of the barometric transducer system, its placement on-board, and the maintenance of the system. Altitude reporting accuracy of 250 feet can currently be expected from transport aircraft. Range accuracy is currently ±370 feet, and is due primarily to lack of precise delay control in aircraft transponder systems. For both altitude and range data a resolution of 100 feet can be specified. A height transmission code (5) uses 11 of the transponder reply pulses to encode altitudes from -1000 feet to 127,000 feet. Coarse (500') and Fine (100') increments are used in reporting altitude. A reflected grey code is used to minimize the effects of errors in pulse transmissions. A similar code could be developed for range data to give a maximum value of approximately 20 nautical miles. A resolution of .1758 degrees/bit will provide 360° of azimuth information on the 12 transponder reply codes.

Sampling rates are determined by the sweep rate of the air surveillance radar to which the beacon interrogator is tied. Current systems use a 4-second rate. This 4-second time period could also be used as a response time in which the air data or navigation functions gather, extrapolate and encode requested data to be transmitted at the next radar sweep. An update of requested data each second should provide the accuracy required for ATC control functions.

Figure 4-48 lists flight and maintenance switch functions.

Figure 4-49 shows a block diagram of the ATC transponder system.

Figure 4-50 lists interface requirements.

CONTROL	FUNCTION
HEATER-POWER	Supplies 115 volts ac to the receiver-transmitter when set at the POWER position. In the HEATER position, power is applied to the internal heaters.
REMOTE CONTROL POWER	OFF position: placed receiver-transmitter in standby condition.
RF PROBE	Permits a small amount of outgoing r-f energy to be tapped from the antenna transmission line.
TUNING INDICATOR	Provides a convenient test point for checking receiver frequency
MONITOR	Reads voltage and current of circuits selected by VOLTAGE-CURRENT SELECTOR switch
VOLTAGE-CURRENT SELECTOR	OFF - disconnect MONITOR meter from all circuits CRYSTAL - measures the current output of the crystal mixer REPLY IND RELAY - measures the plate and screen current drawn by the i-f amplifiers REC OSC - measures plate current drawn by local oscillator XMTR - measures average current drawn by transmitter oscillator DETECTOR - measures a portion of the detector dc output voltage IF BIAS - measures positive voltages delivered by B+ rectifier MOD BIAS - measures positive voltages delivered to modulator by bias rectifier MOD HV - measures high voltage applied to modulator MOD SCREEN - measures screen voltage of modulator
ATC TR	Selects primary or secondary transponder
FUNCTION	Four position switch: OFF-STBY-ON-LO SENS. In Off, the transponder is inoperative. In STBY, the transponder is warmed up. LO SENS is used when requested by ground controller.
CODE SELECTION	Sets identification code number.
IDENT	Operates the special position identification pulse. When pushed, it causes the booster's display on the ground facility radar display to brighten sharply.
MODE SELECT	Selects one of four different identification pulse time intervals.
TEST/MONITOR	TEST position causes the transponder to be interrogated by a simulated ground signal. Monitor position shows transponder operation enroute.

Figure 4-48 ATC Transponder Controls

SIGNAL	SOURCE	TYPE	RANGE	RESOLU-TION	WORD LGTH (BITS)	RATE /SEC
ATC Transponder						
Altitude	DMS	DIG	0-128000'	100'	11	1
Range	DMS	DIG	0-128000'	100'	11	1
Azimuth	DMS	DIG	0-360°	.2°	11	1
Altitude Request	BCS	DIS			1	1
Range Request	BCS	DIS			1	1
Azimuth Request	BCS	DIS			1	1
Identification Code	BCS	DIG			12	1
Control	DMS	DIG			3	1
Voltage/Current Monitor	BCS	DIG			5	1
C-Band Transponder						
Control	DMS	DIG			2	1
Incident Power	BCS	AN	1300 watts	10 watts	8	1
Input Signal Level	BCS	AN	50 MHz	1.0MHz	4	1
Input PRF	BCS	AN	10-2000Hz	10%	5	1
Reply PRF	BCS	AN	10-2000Mz	10%	5	1
Reflected Power	BCS	AN	700 watts	10 watts	7	1
Inside Case Temperature	BCS	AN	-20 to 100°c	2°c	7	1
S-Band Transponder						
VCO Monitor		AN	19.6MHz	.5MHz	7	1
IF1 Monitor		AN	9.53MHz	.25MHz	7	1
IF2 Monitor		AN	47.65MHz	1.0MHz	7	1
AGC Monitor		AN	76.25MHz	1.5MHz	7	1
Recovery Beacon						
Control	DMS	DIS			1	1
Data Record Initiate	DMS	DIS			1	1
Housing Ejection	DMS	DIS			1	1

Figure 4-50 Beacon System (BSC)/DMS Interface

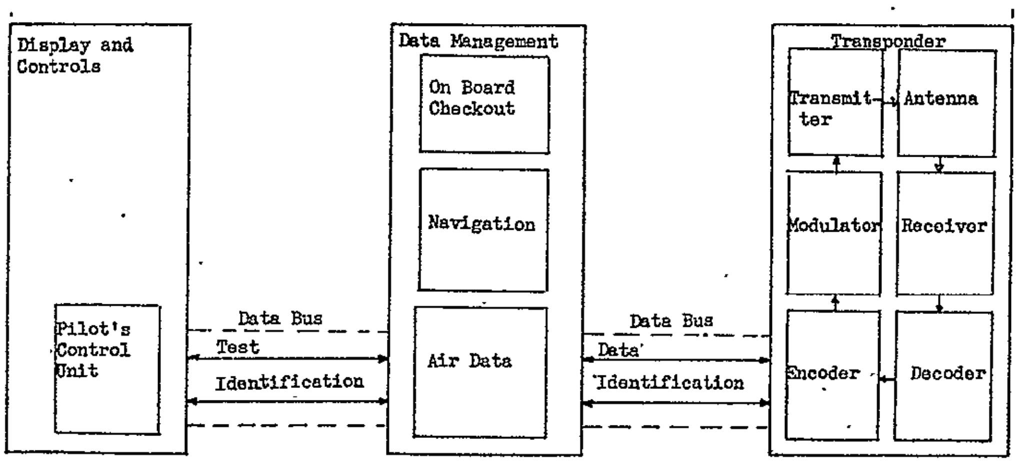


Figure 4-49 ATC Transponder System Block Diagram

Tracking Beacons

Two subsystems, currently in use in the Apollo program, provide tracking information to ground stations. These consist of the PRN (pseudo random noise) ranging subsystem and the C-band radar subsystem. The C-band beacon is intended for use during the launch phase, and the PRN following separation and flyback. These systems, in addition to the ATC transponder, surface radars, and telemetered navigation data from the DMS, provide sufficient equipment to meet redundancy requirements for pinpointing the booster's position during all mission phases.

Pseudo Random Noise Ranging

Ranging consists of filling the up-link and down-link paths between the booster and tracking station with uniformly transmitted cycles of known period, determining the number of cycles in space at the start of ranging acquisition, and subsequently adding or subtracting cycles in accordance with the motion of the booster. The ground based ranging system measures the round-trip propagation time of a signal from the ground transmitter to the booster S-band transponder and back to a ground receiver. The S-band carrier transmission phase modulated by the PRN range code. This code modulation is detected by the transponder and is used to remodulate a down-link S-band carrier (shifted in frequency), which is then received by the ground receiver using the same antenna as used for transmitting.

For the purpose of precisely determining the number of clock cycles, n, a modulation pattern is desired having the following characteristics (14):

(a) A detectable overall periodicity greater than the maximum anticipated round-trip to prevent ambiguous results.

(b) A detectable, fixed, high-frequency periodicity within the overall modulation pattern. This requirement gives high measurement precision.

(c) A two level autocorrelation scheme in which the overall pattern is required to be such that if the pattern is compared with the same pattern displaced by an integral number of bits, the two patterns will match exactly in one relative position, and they will fail to match in all other relative positions.

(d) An essentially balanced transmission having as many 1's as 0's in it (this is not an absolute requirement but balanced use of power in the carrier sidebands gives higher efficiency and better system design).

Requirement (b) above is met by the use of 500 kHz square wave clock transmitting a continuous clock code (CL = 101010...) at a period of 2 microseconds.

Requirement (a) is met by generating the long code required by combining the clock code with other subcodes of relatively prime length. The subcodes (x,a,b,c) have lengths of x = 11, a = 31, b = 7, c = 15. The composite code has length of

$$(2)(11)(31)(7)(15) = 71,610 \text{ bits}$$

which for a bit time of 1 microsecond gives a period of .0716 seconds. This gives a maximum one-way distance between the booster and tracking station of approximately 6700 miles (which is more than sufficient to handle any anticipated booster mission). A transmitter code generator forms a pseudo random code by combining the component codes using a boolean equation.

Requirement (c) is met by matching the transmitted PRN code with the received code. This matching is accomplished by digitally shifting the code components and measuring the correlation indication at each relative shift position until a maximum is obtained. This reading is a measure of the initial range at the start of acquisition. Following acquisition, the tracking station may disable the full code modulation and modulate the S-band carrier with the 2 bit clock code only. This permits a reduction in the required sideband power while continuing to update the range equation:

$$R_t = R_o + \int_o^t \dot{R} \; dt$$

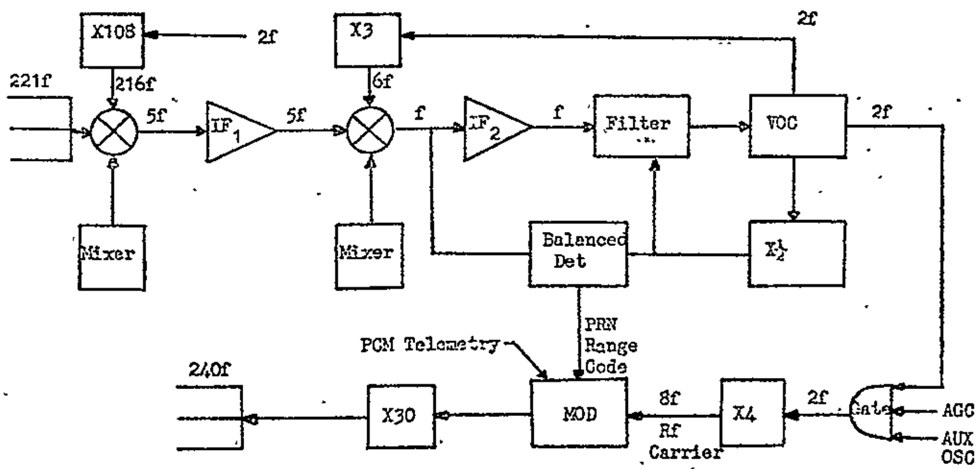
Where R_o is the initial range found at acquisition using the PRN code.

Figure 4-51 shows the portion of the S-band transponder involved in the turnaround of the PRN code or the clock code transmitted by the tracking station. The received frequency is designated 221 f and is nominally 2106.4 MHz so that f equals 9.53 MHz. From the diagram it is seen that the received and transmitted frequencies are internally related by a ratio of 240 to 221. The operation of the S-band transponder can be monitored by test points at the inputs to the first intermediate frequency (IF1), the second intermediate frequency (IF2), and the voltage controlled oscillator (VCO) and verifying the ratio

$$\frac{120VCO}{108VCO+IF1} = \frac{120VCO}{110VCO + IF2} = \frac{2287.5}{2106.4} = \frac{240}{221}$$

The phase modulator (MOD) of the transmitter has two modulation inputs, the PRN range code and the down-link telemtery sub-carrier. The transmitter section of the transponder also receives 2f from the VCO, uses a auxiliary oscillator (AUX OSC) to provide a noise-free carrier, and has an automatic gain control (AGC) to maintain signal input.

Figure 4-51 S-Band Transponder Ranging Diagram



Receiver Characteristics	
Frequency (tunable externally)	5400 to 5900 MHz (set to 5690 ±2 MHz)
Frequency stability	±2.0 MHz
Bandwidth (3 db)	10 MHz
Off-frequency rejection	50 db image; 80 db minimum, 0.15 to 10,000 MHz
Sensitivity (99% reply)	-65 dbm over entire frequency range and all environments
Maximum input signal	+20 dbm
Interrogation code	Single or double pulse
Pulse width	0.2 to 5.0 us (single pulse), 0.2 to 1.0 us (double pulse)
Pulse spacing	Continuously settable between 5 and 12 us (set to 8 ±0.05 us)
Decoder limits	±0.25 us accept, ±0.85 us reject (5 to 12 us)
Transmitter Characteristics	
Frequency (tunable externally)	5400 to 5900 MHz (set to 5765 ± 2 MHz)
Peak power output	400 watts minimum, 700 watts nominal
Pulse width	1.0 ± 0.1 us
Pulse jitter	0.020 us maximum for signals above -55 dbm
Pulse rise time (10% to 90%)	0.1 us maximum
Duty cycle	0.002 maximum
VSWR of load	1.5:1 maximum
Pulse repetition rate	10 to 2000 pps, overinterrogation protection allows interrogation at much higher rates with count-down; replies during overinterrogation meet all requirements
Transponder Characteristics	
Recovery time	50 us single pulse, 62 us double pulse maximum for input signal levels differing by up to 65 db (recovers to full sensitivity with no change in transmitter reply power or frequency with multiple radars interrogating simultaneously)
Fixed delay	Settable 2 ± 0.1 and 3.0 to 0.01 us (set to 3.0 ± 0.01 us)
Delay variation with signal level	50 nanoseconds maximum from -65 dbm to 0 dbm
Power requirements	24 to 30 volts
Primary current drain	0.7 ampere standby; 0.9 ampere at 1000 pps
Weight	2.5 kg (5.5 lbs)

Figure 4-52 C-Band Radar Transponder Characteristics

C-Band Radar

During launch, the booster C-band radar subsystem is interrogated by ground based FPS-16 and similar precision instrumentation radars. These radars have resolutions of 5 meters in range and 0.1 mrad in azimuth and elevation (13). The booster has four C-band flush mounted antennas. A single comparison technique uses portions of four receivers, each connected to a single antenna. The circuitry on the on-board C-band transponder equipment is such that the single transmitter is switched to the antenna receiving the highest signal. The up signal consists of 5690 MHz interrogation pulses received by the four C-band receivers. The signal is combined, compared and decoded (interrogation pulses are coded for a particular vehicle's acceptance) by C-band electronic circuitry equipment. After a 3 microsecond delay, 5765 MHz reply pulses are transmitted. Figure 4-50 lists C-band interface requirements. Figure 4-52 lists characters of the C-Band radar beacon (17).

Recovery Beacon

The recovery beacon is primarily intended as a crash locator. It is anticipated that the equipment used in the booster will be similar to the AN/URT-26 installed in current aircraft such as the C-141 and C-5A (15). The beacon can be ejected from the aircraft manually by the pilot or is ejected automatically through activation of frangible crash detector switches, water activated switches or other devices, mounted in strategic locations in the aircraft. Associated with the recovery beacon are voice and crash data recorders designed to meet FAA recommendations. Figure 4-50 lists beacon subsystem and DMS interfaces. A data request discrete will result in the DMS supplying the information required for crash analysis as described in Section 4.6.4 The beacon ejection discrete will separate the beacon housing panel in the airframe and will also turn on the beacon transmitter if it has not previously been activated.

4.7 DISPLAYS AND CONTROLS

Section 3.7 presented the general functional requirements and a preliminary list of parameters for a display and control subsystem for the booster vehicle. Figure 4-53 shows a functional display and control subsystem. It is anticipated that the displays used on the booster will consist of (7);

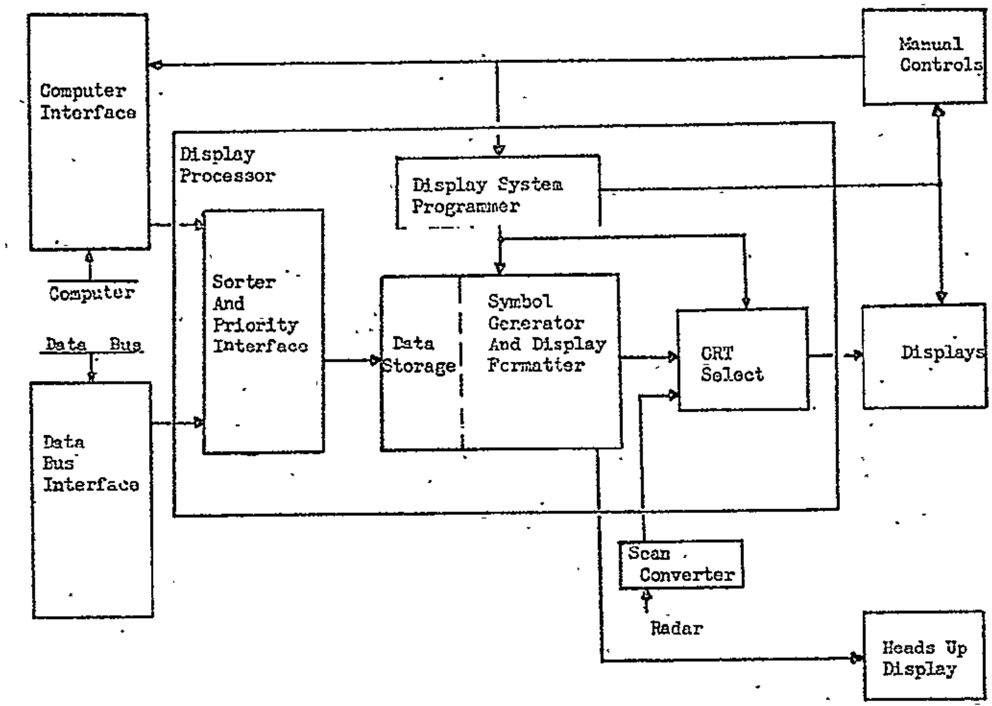
- Electronic Attitude Director Indicator (EADI)
- Heads Up Display (HUD)
- Horizontal Situation Display (HSD)
- Multifunction Display (MFD)
- Electronic-moving Bargraphs
- Dedicated Instruments

and that controls will include

- Flight controllers
- Reprogrammable Switches
- Alphanumeric display keyboard
- Data Entry Keyboard
- Secondary controls

The actual location of these units is a function of human factor engineering and is considered here only to the degree of depicting a possible general arrangement as shown in Figure 4-54. This symmetrical arrangement meets the requirement for operation by one man, and makes it possible to display to the copilot the same information as the pilot sees during periods of critical activity. During other mission phases, the copilot displays can be used to perform subsystem checkout, malfunction isolation, configuration

Figure 4-53 Display and Control Functional Diagram



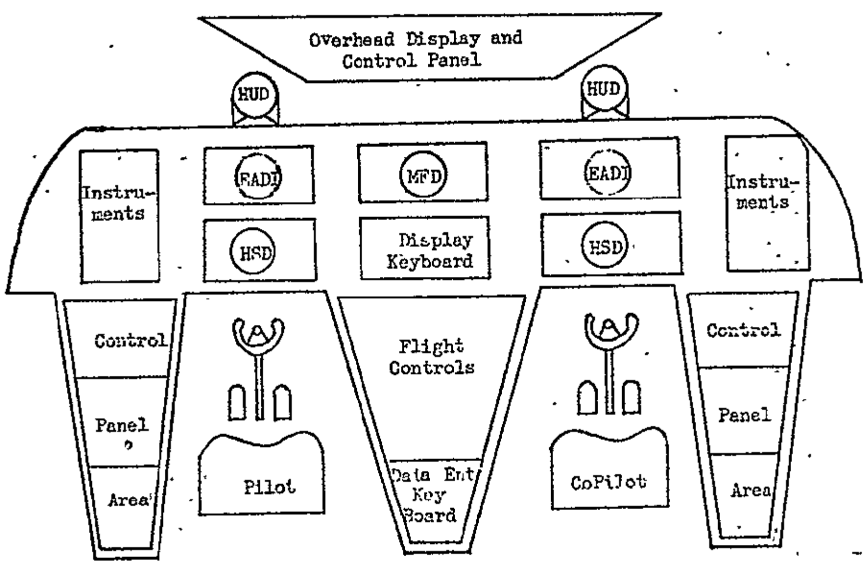


Figure 4-54 Display and Controls Arrangement

monitoring, and other housekeeping activities. The displays can be programmed to present different information during various modes of shuttle flight. Critical events or events can be programmed to take precedence over a non-critical display. Other modes may be controlled manually by the crew, or the formats can be changed automatically by the computer. Programmable redundancy can be incorporated to improve reliability and conform to the redundancy requirements of booster operations. A display usage plan gives the primary dedication of each display in each mission phase, and lists a backup display to take the place of the primary one in case of a display failure. The specific parameters used or functions performed are outlined in sections on each display. The differences in the application of the displays imply different mode and feature selections for the displays. Sections on each display describe these differences. However, the CRT displays have many properties in common. Also existing displays have optional hardware devices. Figure 4-55 lists the characteristics assumed for this study in order to pinpoint hardware and software tasks. The following paragraphs discuss the selected parameters and controls applicable to all displays.

A typical display area for a CRT is 12 inches square. A 10-bit X and Y magnetic deflection system locates the beam at the intersection of a grid of 1024 by 1024 raster points.

Capacity is defines as the amount of information that can be shown on a CRT during a refresh period. For example, the basic character size is 0.16" in height and 0.12" in width with a spacing of 0.04" between alphanumerics and a spacing of 0.07" between lines. This gives 74 positions on each line and 52 lines as the capacity required for a completely alphanumeric display. This can be expressed in terms of the number of characters, bits, or storage words. The CRT beam positioning times and the individual generation times for each type of character are the most significant parameters which affect display capacity. Display capacity is normally less than the storage capacity of the display refresh buffer unit.

Display commands are instructions to generate a presentation on a CRT viewing surface. They specify such things as coordinate end points, modifiers (size, brightness, color of symbols), memory limits, control and sync bits, and conic parameters.

The CRT's are the refresh type which must be regenerated periodically to sustain a flicker free image for the viewer. It is an advantage to use the lowest refresh rate which does not produce any adverse observable effects. The current display is stored in the display refresh buffer unit. No functions, other than read or write, are required of the buffer unit.

The CRT display unit has various monitor status and control functions which will be indicated to the DMS. These include:

- Overflow indicator - which is activated when the display refresh buffer unit does not encounter an end-of-message word during a given display frame period.
- Video gain controls
- Focus controls
- Character space controls

- Master clear controls
- Power supply controls
- Protection circuit status
- Compensation circuit controls

The rate of updating the parameters being displayed on the CRT will be a submultiple of the refresh rate of 32 hertz. Each basic parameter (such as rate of climb, airspeed, angle of attack, etc.) is sampled at a rate necessary to provide a smooth transistion of the display parameter (runway centerline, horizon, airspeed or altitude index).

As the Displays and Controls subsystem is indispensable for mission success, system status indication, and flight maneuver efficiency, the equipment of the subsystem will be tested and monitored to ensure proper operation and fault isolation (7). The test points listed in Figure 4-56 are monitored for confidence checks and preflight checkouts. For fault isolation and system performance, program generated test patterns will detect malfunctions or suboptimal performance by display engineering parameters.

Manufactures of CRT's list specifications for testing cathode ray tubes. Tests include visual checks, pattern distortion tests, brightness and resistence measurements, alignment tolerances and line width tests. Pattern distortion measurements are made to check the effects of non orthogonality, keystoning, and pincushion and barrel distortion. This test draws a rectangle on the screen, and measures the trace variation from a perfect rectangle.

Brightness requirements are given in individual test specifications for each tube type. The measurements are made using a standard scanning pattern. Light output is measured by a foot-lambert light meter. Persistence test are also made on a scanned raster. In this test the light intensity is observed at a given number of seconds after applying the raster. Another brightness related test is that of determining buildup factor, e.g., the ratio of the light intensity one second after the fifth raster to the intensity one second after the first raster.

Two line width measurements are made; one in the center of the screen and one near the edge. These measurements are compared to check line width and accuracy and distortion effects.

Tests are made to verify both the absolute and relative accuracy of CRT beam positioning. The absolute accuracy of location of the CRT beam in response to known inputs is specified as a percentage of full scale deflection. If the full scale deflection covers 12 inches and the absolute accuracy required is .5% then the beam must be located within .06 inches of expected value for any input to meet this specification. Similarly, if two beam positions are expected to differ by 2 inches with a relative accuracy of .1% then the difference must be less than .2 (these accuracy tests apply to the position of visual results - other tests will verify that when a specific input is called for, it is displayed).

Repeatibility tests measure the extent to which an element of the visual presentation appears at the same location each time the same positioning information is introduced into the system. Repeatibility errors contribute to both jitter and legibility of the display. Repeatibility is specified as a percentage of full scale deflection (normally in the order of half a line width). By repeating a test pattern, statistics are compiled to determine the repeatibility of each display.

Capacity is tested using other patterns. These patterns verify that a given amount of information can or cannot be generated in the refresh period. These tests also verify the generation time for characters and vectors. Figure 4-56 lists the interface requirements for displays and controls. This list includes the parameters common to all displays, and those for the individual units described in the following sections.

Viewing Surface	12" by 12"
Number of Raster Points	1024 both X,Y
Displayable Elements Capacity	2480
Number of Display Commands	16
Command Types: Single Operand	Yes
Multiple Operand	No
Minimum Interface Word Size	16 bits
Character Generator	Yes
Repertoire (Max.)	48
Character Sizes	2
Character Orientations	2
Character Spacing	2
Character Fonts	1
Vector Generator	Yes
Generator Technique	End point
Brightness Levels	4
Line Widths	2
Colors	3
Rotation Hardware	No
Perspective Hardware	No
Conic Generator Hardware	No
Symbol Blink	Yes
Input Device: Light Pen	No
Table and Styles	No
Alphanumeric Keyboard	Yes
Cursor Control	Yes
Function Keys	Yes
Overlay Plates	No
Refresh Memory	Yes
Refresh Rate	32 hertz
Memory Module	4096-32 bit words
Memory Functions	Read, Write
Hard Copy	No

Figure 4-55 Graphic Display Characteristics

SIGNAL	SOURCE	TYPE	RANGE	RESOLU-TION	WORD LGTH (BITS)	RATE /SEC
CRT Power Supply	DCS	AN	5vdc	.05vdc	8	1
CRT Power Supply	DCS	AN	28vdc	.28vdc	8	1
CRT Power Supply	DCS	AN	14vdc	.15vdc	8	1
CRT Power Supply	DCS	AN	-15vdc	.15vdc	8	1
CRT Power Supply	DCS	AN	10kvdc	200bdc	7	1
Focus Control (7)	DMS	AN				1
A/N Video Gain Control (7)	DMS	AN	0-1000fl	10fl	8	1
Vector Video Gain Control (7)	DMS	AN	0-1000fl	10fl	8	1
CRT Power Control (7)	DCS	AN			1	AR
CRT Master Clear (7)	DCS	DIS			1	AR
Blank/Unblank (7)	DCS	DIS			1	AR
Overflow (7)	DCS	DIS			1	AR
Frame Sync (7)	DMS	DIG			2	AR
CRT Protect (7)	DCS	DIS			1	AR
Data Interrupt Enable	DMS	DIS			1	AR
Data Interrupt	DCS	DIS			1	AR
Input Data Request	DCS	DIS			1	AR
Input Acknowledge	DMS	DIS			1	AR
External Functions Request	DCS	DIS			1	AR
External Function	DMS	DIS			1	AR
Output Data Request	DCS	DIS			1	AR
Output Acknowledge	DMS	DIS			1	AR
Parity Fault	DMS	DIS			1	AR
Vertical Tape Control	DMS	AN	360°	1°	10	1
Scan Converter Control	DMS	DIS			1	1
Scan Converter Radar Select	DMS	DIG			2	1
Scan Converter Power	DCS	AN				
Mode Control	DMS	DIG			4	1
Airspeed Reference	DMS	DIG			11	1
Path Selector	DMS	DIG			2	1
Film Transport Control	DMS	DIS			1	1
Filmslide Control	DMS	DIS			1	1
Filmslide Orient Select	DMS	DIS			1	1
Filmslide Course Control	DMS	AN	360°	1°	10	1
El Lamp Power	DCS	AN	22vdc	1vdc	6	1
EL Circuit Power	DCS	AN	4.5vdc	.25vdc	6	1
El Control	DMS	DIS			1	AR
Function Key (16)	DCS	DIG			4	1
Enter	DCS	DIS			4	1
Clear Enter	DCS	DIS			1	1
Keyset Select	DCS	DIS			1	1
Inhibit RTI	DMS	DIS			1	AR
Computer Power Monitor	DMS	AN	28vdc	$\pm$ 2	5	1
HSD Select	DMS	DIG			2	AR

Figure 4-56 Displays and Controls System (DCS)/DMS Interface

4.7.1 ELECTRONIC ATTITUDE DIRECTOR INDICATOR (EADI)

The EADI integrates into one instrument the electromechanical 8-ball, altitude, vertical speed, airspeed, and Mach number instruments. Through the use of electronically generated symbology, it provides a simultaneous display of critical flight information; heading, airspeed, attitude, altitude, and command and rate information in clear, immediately recognizable form. In addition, radar inputs can be displayed on the EADI.

Figure 4-57 is a functional block diagram of the EADI showing its inputs when the display is being used in a flight director mode during approach to a landing. Information from air data, attitude, and navigation sensors is combined and the appropriate symbology is generated on the CRT. Pitch, roll, and yaw information combines to display an artifical horizon. Altitude and airspeed data update the fixed symbology used to portray runways and traffic patterns. Inputs from the instrument landing system equipment provide flight path data. All this information is validated and, if valid, is used to generate flight director commands to the crew. Failure warning may also be displayed using inputs from the Built in Test Equipment (BITE), or data from software analysis.

Air speed and altitude are the type of data that can be displayed using vertical tape displays. In this display, a fixed pointer is set next to a movable tape which is of sufficient length to contain all likely variations in the parameter being measured. In addition to controlling the movement of the tape as required to display the actual value of the parameter, recommended, caution, and warning indications are also displayed.

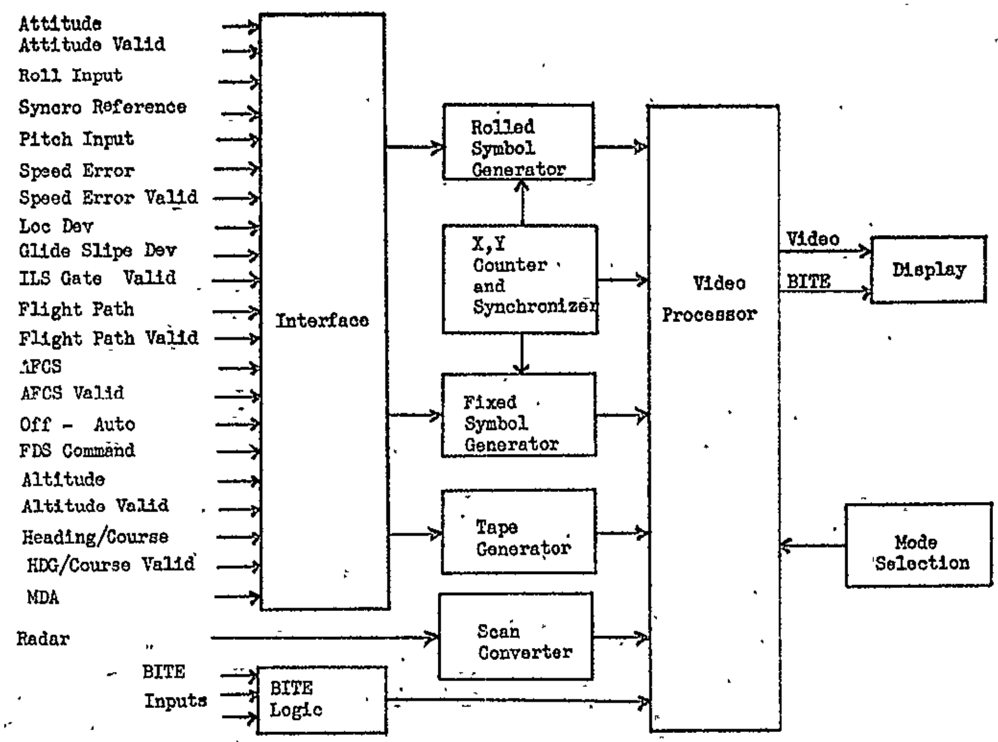


Figure 4.-57 EADI Functional Block Diagram

Weather or doppler radar pictures can be displayed on the EADI. This is made possible by the use of scan converter tubes. The scan converter storage tube is an electrical input-electrical output storage device which enables conversion of slow data rate signals, such as radar scan patterns, into signals read out at high data rates. The high data rate signal (CRT refresh rate) is then used to modulate the electron beam of the EADI CRT.

Special modes and feature selection associated with the EADI are:

- EADI mode control
- Air speed and altitude markers
- Flight command markers
- Vertical tape control
- Scan converter ON/OFF
- Scan converter radar select
- Scan converter voltage monitor

Computer programs associated with the EADI consist primarily of application routines which organize the inputs for a particular mission phase and display the appropriate symbols. Other routines provide control and monitor functions. Interface requirements are listed in Figure 4-56.

4.7.2 HEADS UP DISPLAYS (HUD)

HUD is a windshield projection display which provides collimated virtual images at optical infinity projected within the pilot's field of view as he looks through the windshield. This display permits the superposition of information on the pilot's external vision field in a form that is compatible with his view of the real world through the windshield.

Figure 4-58 shows a possible format for HUD display during an approach. Symbology may vary from that shown. In addition, alpha numerics may be added for critical parameters such as altitude or air speed. In this figure, the symbols are defined as follows:

- Horizon line - is a horizontal reference line that represents a trace of a plane normal to the vertical at the current aircraft altitude. It is space stabilized in pitch and bank to maintain its horizontal orientation at an elevation angle of zero.
- Heading index -represents the runway or reference heading in the proper visual relationship with the actual heading of the aircraft. When the nose of the aircraft moves one degree to the left, the heading index will move one degree to the right. The horizon line and heading index provides an external visual reference to the pilot from which he can obtain pitch, roll, and heading information as in visual flight. These images continuously overlay their counterparts in the real world.

- Runway image - represents the real runway, and is consistent with the runway in position, perspective shape, and size. The runway image overlays the real runway when visibility conditions allow it to be seen.

- Aim point - is a point on the runway where the ILS on-course (or AILS touch down point) intersects the ground plane.

- Deviation image - represents the deviation of the aircraft from the on-course in azimuth and elevation, which is measured by the displacement of the deviation dot from the aim point. In part (a) of Figure 4-58, the aircraft is on-course but is above the glide path by an angle equal to the visual angle between the deviation dot and the aim point.

- Altitude scale - indicates the actual altitude above the runway. The horizon line is the reference against which the altitude is read on the scale. In part (a) of Figure 4-58, the aircraft is about 150 feet above the runway. Alphanumerics are used to display the precise radar altitude. (In other modes, such as IFR enroute, the barometric altitude is displayed).

- Flight path marker - uses a miniature airplane to indicate the direction of the velocity vector of the aircraft. If the direction of flight were to remain constant, the aircraft would strike the ground at the point indicated by the center of the path marker image. In part (a) of Figure 4-58, the aircraft is undershooting the runway.

- Airspeed index - presents the departure of the actual airpseed from the reference value. The top and bottom indices are slaved to the path marker, and represent plus and minus 10 knot limits. In part (a) of Figure 4-58, the aircraft is 5 knots fast.

- Director image - presents lateral and vertical flight control commands. The commands are satisfied when the pilot (or auto-pilot system) flies the circle so that it overlays the aim point on the runway image.

Figure 4-58 shows additional applications of a HUD display. In part (d), the aircraft is maneuvering toward the runway in a left bank. The altitude is between 400 and 450 feet. The aircraft is high and to the right of the on-course and is undershooting the runway. It is below the reference airspeed by about 5 knots. At the flare altitude (b), the runway image configuration changes to eliminate the aim point and the horizontal glide slope intersection line through this point. The centerline of the runway, positioned by the localizer signal, and the runway edges remain. Elevation guidance is provided by the altitude scale and the perspective of the runway edges. For go-around (c), a command circle is shown at the reference climb-out angle above the horizon line. The pilot raises his flight path marker to this circle to increase his flight path angle, while increasing his airspeed to the climb-out reference speed commanded by the airspeed index. The horizontal line above the command circle represents the altitude at which to level out (20).

Software programs are required to activate and control the HUD. Processing and application routines position, rotate, translate, or emphasize symbols for the display. For example, vector end points can be specified by the

X and Y coordinate chart. Controls associated with the HUD consist of:

- Mode selector - which provides a means of selecting the parameters to be displayed in a particular flight mode. Modes used in a current application consist of: (1) OFF, (2) localizer flying outbound, (3) heading, (4) VOR, (5) localizer, (6) approach, and (7) go around. Additional modes will be added to satisfy other booster modes.
- Airspeed reference selector - sets the recommended airspeed for all flight modes.
- Path selector - for selecting one of three ILS glide paths.

Figure 4-56 shows these interface requirements together with those in common with the other display units.

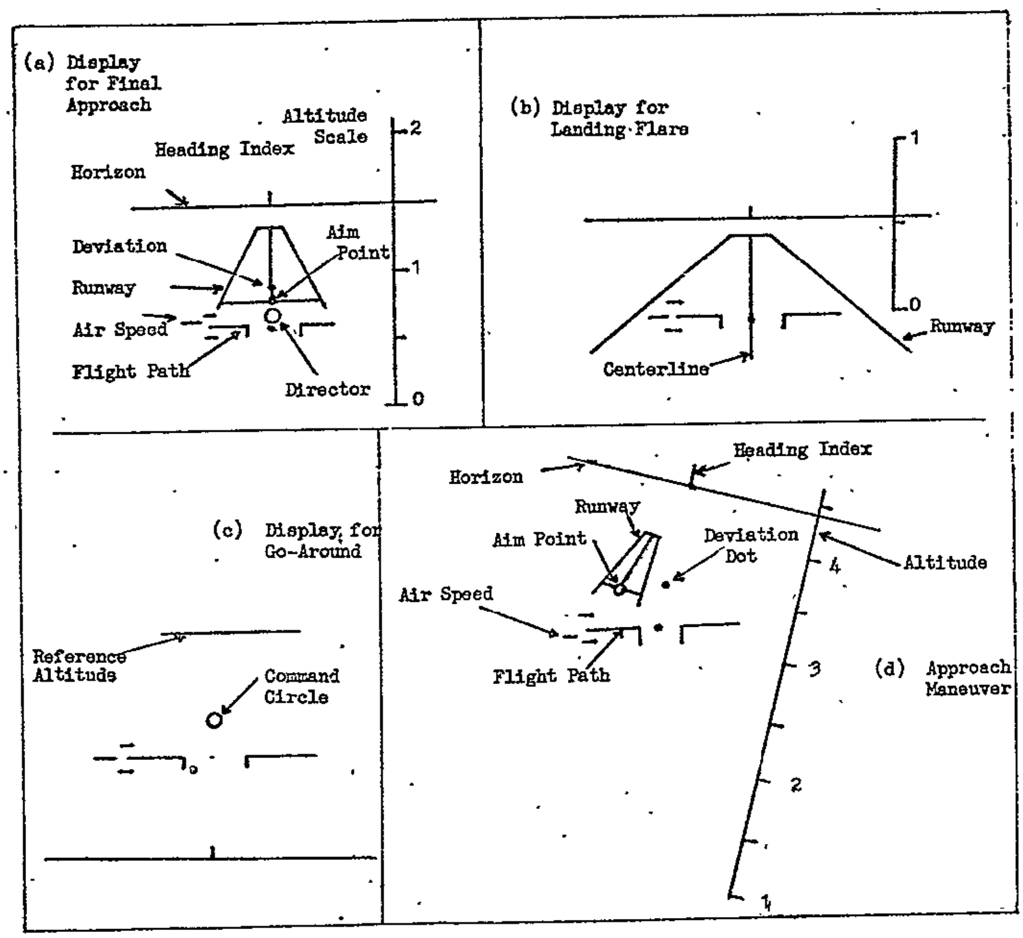


Figure 4-58 HUD Display

4.7.3 HORIZONTAL SITUATION DISPLAY (HSD)

The horizontal situation display offers the capability to display useful information concerning the booster situation in all phases of flight. The HSD also offers the capability to display check lists, procedural instructions , and maintenance information during other periods of shuttle activity. During periods of minimal utilization or during emergencies, the HSD can be used either as a backup for other CRT displays or to display information supplementary to that on other displays, such as during check-out or fault isolation activities (7).

The HSD is primarily a moving map display which can be implemented by various means, such as optronic, remote TV, electronic, or optical. The optronic has the most desirable features in respect to flexibilty and the generation of map, symbol and data information. The optronic display is a rear projection CRT with the ability to present filmslides placed in the transport unit.

Possible display applications for the HSD are maps outlining:

- Launch ground track
- Reentry footprint
- Approach and holding patterns
- Landing and taxi patterns
- Go around patterns
- Takeoff and departure patterns

Film slides are included for primary and alternate fields and can be selected by the crew or by computer signals.

Computer programs will add character, vector and symbolic information to the maps to increase the effective navigational performance and provide positive orientation. This information continously updates the position and path of the booster on the selected map, and provides navigational information such as present position, time to primary and alternate fields, and fuel reserves. Weather radar information can also be displayed on the HSD using the scan converter technique described for the EADI.

Special modes and features applicable to the HSD consist of film transport and filmslide controls:

- Film transport ON/OFF
- Filmslide control: FORWARD/REVERSE
- Filmslide orient - NORTH/COURSE. This control gives the flight crew the option of orienting the selected map with magnetic north at the top, or aligning the direction of the map with the magnetic heading of the booster.

Computer programs are required to display the symbology associated with each chart and to answer requests by the flight crew for navigational information. Interface requirements are shown in Figure 4.-56.

4.7.4 Multifunction Display (MFD)

The size and complexity of the booster vehicle, and the rapid changes in mission phases and vehicle configuration place many demands on the flight crew in anticipating events and making appropriate decisions. The MFD is planned to aid in the task of processing and displaying information so that the crews decision-making capabilities are increased and their workload is decreased. The MFD is a general purpose, time shared, optronic (rear projection CRT) placed conveniently to both the pilot and copilot.

The major applications planned for the MFD are (7);

- Presentation of optimum flight profile information and energy management parameters
- Presentation of vertical profiles for ascent and descent corridors
- Presentation of situation data from onboard systems for crew monitoring of operational performance
- Aid in crew/computer interface in determining alternate courses of action

Suitable graphics, alphanumerics and special symbols are used to present operational data such as the following:

- Vertical navigation aids
  - Ascent profiles
  - Emergency descent profiles
  - Range and cross range landing capabilities
  - Descent profiles

- Fuel management
  - Center of gravity
  - Propellant utilization
- General information
  - Time references
  - Flight data
  - Check lists
  - Radar data
- Safety aids
  - Engine restart envelope
  - Emergency procedures
  - Aerodynamic and heating loads
  - Subsystem failure display
  - Circuit breaker monitoring

Visual and aural alert devices in the form of flashing lights and buzzers are provided to direct the attention of the flight crew to status changes. For example, monitoring and control of system circuit breakers is performed by the displays and controls subsystem. If a circuit breaker is tripped, a buzzer and flashing light near the MFD are activated. The MFD then displays the information on which circuit breaker is involved (as determined by BITE or DMS diagnostics), its location, and the address code for resetting.

Weather or doppler radar pictures can be displayed on the MFD if desired. The display can show the radar video by itself or in connection with flight profiles. Filmslides used with the MFD are primarily the framework (abscissa, ordinate, fixed legends) for graphs and charts used during the mission, but the capability for using the map displays of the HSD exists. Interface requirements are similar to those required for general CRT operations plus the scan converter and film transport controls described in the EADI and HSD sections. Figure 4-56 lists these requirements.

Computer program requirements consist of the application routines associated with check list management, presentation of alternate, emergency, or abort procedures, and updating of flight profile and status charts and graphs.

4.7.5 Electronic-Moving Bargraphs

Some measurements may required continous monitoring and display on dedicated instruments. A computer driven electroluminescent vertical scale indicator (VSI) is a bargraph with scale and parameter indicators which can be controlled and updated by DMS signals. These bargraphs are used for displaying temperatures, pressures, flow rates, and propellant and fuel quantities.

The VSI receive signal data inputs from the DMS. These signals consist of (23):

- A number (7 bits) representing the value of the variable used for the bargraph function
- A scale indication(2 bits)
- A parameter indication (2 bits)
- A data strobe signal

The input signals are gated by a computer command strobe signal into a buffer memory which retains the data until the next updating strobe signal is received. Current circuitry used in VSI's permit the updating at a maximum rate of 40 times per second.

The electronic logic design of the VSI consists of the three functions: (1) bargraph, (2) scale, and (3) parameter. The bargraph is generated from the variable value by decoding it into signals which activate a segmented electroluminescent lamp (EL). The height of the lighted segments (top segment and all below it) is proportional to the value of the variable indicated by the input signal. A single segment pointer capability is also available to light only the top segment if desired.

The scale or multiplier function illuminates various EL areas to show the scale factor selected for the parameter being measured. A 2-bit code permits the selection of a low/medium/high (or X1,X10,X100) scale factor. In addition one of the scale codes can be used to activate a warning lamp if the measurement becomes critical

The parameter indicator code is used to illuminate one of four lamps, each of which is labeled with the name of a parameter which can be measured using the VSI bargraph. This function permits reprogramming of the VSI depending on mission phase. For example, VSI's dedicated to boost engine parameters during launch can be used for jet engine performance parameters during cruise and landing approaches.

For use in program sizing estimates, a total of 8 VSI's are assumed. As each VSI can be reprogrammed to measure 4 different parameters, a total of 32 measurements can be displayed by these bargraphs. In addition, multiple bargraphs can be placed next to VSI scales to indicate repeated parameters. For example, the exhaust temperatures for jet engines 1,2 and 3 could be indicated in adjacent columns using the same scale.

Computer programs are required to activate and control the VSI's in accordance with a mission phase and bargraph usage schedule. Formatting and scaling of DMS input data, together with verification and control of parameter, is performed. Testing of VSI's is conducted using a signal simulator which permits exercising the VSI's statically in all modes with all possible combinations of input siggals. A test routine steps through each VSI checking each instrument for proper operation.

4.7.6 Dedicated Instruments

The computer driven paragraphs of the previous section are dedicated to a particular parameter during a specified mission phase. I is anticipated that a limited number of meters and indicators will be dedicated to one particular measurement throughout the booster mission. These will cintinuously monitor the status of essential combustibles, power supplies, and pressures. It is assumed that 8 meters will be used for this purpose in the booster.

Computer program requirements are limited to test, monitor, and alarm indicating functions.

4.7.7 Flight Controllers

Flight controller devices consist of the various pedals, sticks, yokes, levers or knobs used to control the attitude, direction and speed of flight. Current tests being conducted will resolve the question of whether current aircraft yoke and rudder pedal systems can be replaced by a hand controller similar to that used for space flight. To eliminate cockpit clutter, the control design replaces conventional levers (such as speed brake, wing flap, and landing gear controls) with pushbutton switches which become part of the reprogrammable switch control unit.

Regardless of the physical design of the flight control device, the function of the displays and control subsystem is to indicate the current position of the unit being controlled, provide a means to change its position, monitor the change and provide status and warning information as required.

4.7.8 Reprogrammable Switches

Dedicating a separate switch to every function overcrowds panels, causes operator difficulities, and is, in general, wasteful and inefficient. It is more effective to perform the multitude of switch functions with a small number of switches. To make this possible, each switch must be reprogrammable to perform several functions in such a way that all the choices that the pilot might wish to make at any one time are available to him (22).

Implicit in this concept is that, at any one moment, the pilot's actions are by nature limited to a relatively small number of options, even when he performs complicated tasks. Complicated tasks usually are separated into a sequence of simpler steps. Several solutions to reducing the number of switches are available. In each, the switch function is changed by having the switch interact with the computer. Switch closures are coded inputs to the computer. When an operator depresses one of the switches, a coded pulse is transmitted to the computer. The computer identifies the switch position, interprets its intended function, and performs the indicated operation. Reprogramming of switch functions may be performed manually by flight crew request, or automatically under computer control. Computer control is particularly advantageous for long operational or test sequences requiring the use of many switches.

There are three general methods for organizing switch functions and displaying their title (22):

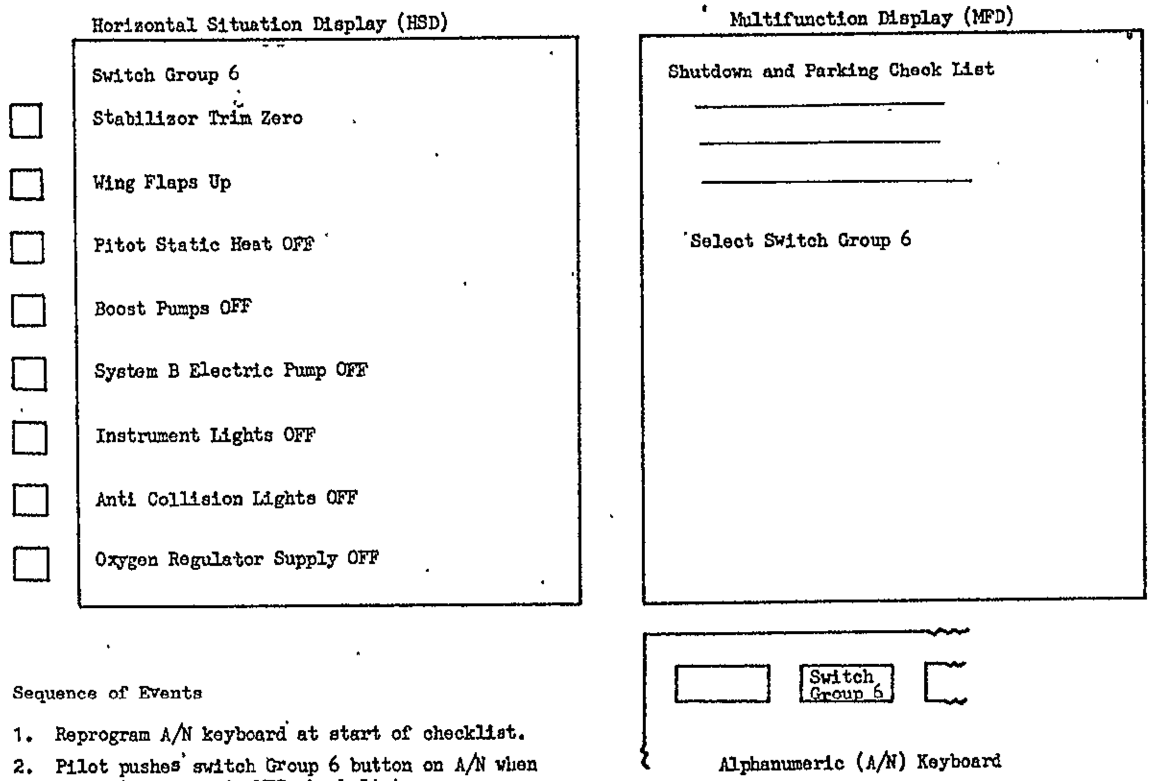
(1) Display on Face of Pushbutton - which displays the programmable functions directly on the pushbutton faces. This method uses a

filmslide that contains the titles for the switch functions. The switch legend is changed by activating appropriate optics and controlling filmslide motion.

(2) Display Next to Pushbutton - which displays the switch function on a CRT adjacent to a vertical column of pushbuttons. Each message is close enough to its corresponding pushbutton so that the face of the button can be left blank. The quantity of messages is not limited as in the first method.

(3) Keyboard Concept - which uses a keyboard similar to the computer or function word display keyboards which identifies the changed switch function on a display near the keyboard. The pushbutton has a fixed legend on its face, usually an alphanumeric or symbolic code. The CRT display shows the switch legend with the switch function spelled out next to it.

Method 2 is assumed to be the primary method to be used in the booster for system switch control. The HSD's of the pilot and the copilot are used as the switch function CRT display. Figure 4-59 shows the use of this technique on assisting check list functions. The techniques of method 1 are used to label the function keys of alphanumeric keyboard. These function keys, used to call up switch function groups or display application programs, vary in their meaning depending on the mission phase. The keyboard concept, method 3, is not used except as dedicated to the data entry keyboard and the alphanumeric and editing keys of the display keyboard.. Computer routines are required to activate and control filmslide usage on the function keys of the display keyboard, and to correlate and display the various switch groupings.



Sequence of Events

1. Reprogram A/N keyboard at start of checklist.
2. Pilot pushes switch Group 6 button on A/N when request appears in MFD check list.
3. Pushbuttons next to HSD are reprogrammed in response to step 2.
4. Pushbutton on HSD are depressed from top to bottom giving the indicated results.
5. When last HSD pushbutton is activated, next step in checklist appears on MFD

Figure 4-59 Display Next to Pushbutton Usage

4.7.9 Alphanumeric Display Keyboard

The alphanumeric display keyboard consists of three major divisions:

- Data keyboard - which generates new data using 43 standard typewriter characters.
- Control keyboard - 12 keys which provide the operator with local control of the console, including editing keys and cursor control keys.
- Function keyboard - 16 function keys are provided which generate computer interrupts. By using upper and lower case codes a total of 32 function codes at a time can be made available. By using a technique similar to that used for reprogrammable switches, these keys can specify different functions depending upon mission mode.

A typical coding scheme uses a modified 7-bit ASCII code for data, control, and function inputs from the keyboard and for external functions from the computer. Figure 4-60 shows a basic block diagram of the keyboard interfacing with character generator, storage, CRT deflection circuits, and control units. Data transfer, timing, and control signal lines are also indicated.

The cursor (⌝) is a displayable character. It is used (instead of light pens or other devices) to indicate a particular location on the display. The operator can enable, clear, or position the cursor by means of the cursor control keys. These functions can also be performed by means of external function code words from the computer. The cursor coordinates may be transferred from the display to the computer or from the computer to the display.

The cursor indicates where the next entry from the keyboard is to be displayed. As characters are selected at the keyboard (or controlled by computer input of keyboard codes), the cursor is advanced one position to the right. At the end of each line, the cursor automatically advances to the beginning of the next line. At the end of the last line, the cursor returns to the home position (line 2 and column 1). Old data appearing at the cursor location is deleted as new data is entered from the keyboard or the computer.

Software programs are required to answer keyboard interrupts, to accept and process keyboard inputs, and to route alphanumeric data from the computer to the appropriate display. For example, when the copilot calls for the takeoff checklist by depressing the proper function code button, the title is displayed on line 1 and the first step in the checkout procedure on line 2 as follows:

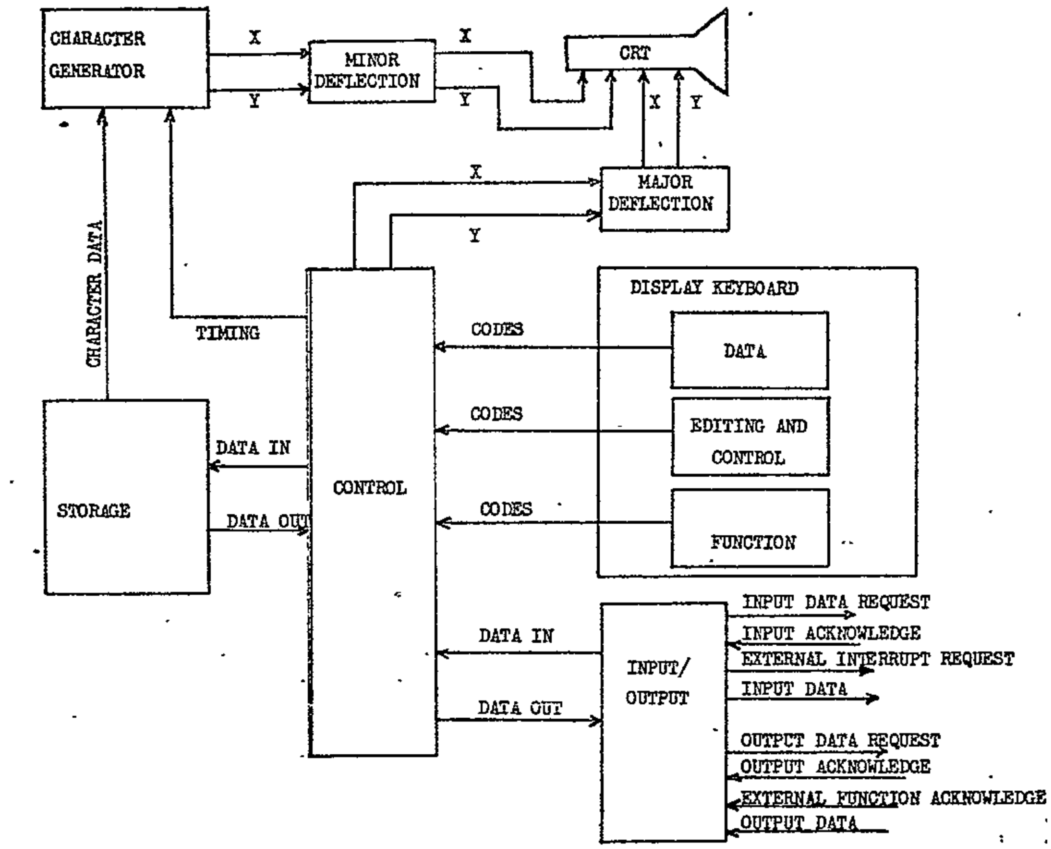
1 TAKEOFF CHECKLIST

2 SET FLAPS TO 15 DEGREES

3 ⌜

After the first step is performed, the operation is verified and the next step is displayed for flight crew action. If a device, or instrument does not pass verification tests then the display indicates this. In addition, the options available are also displayed. By positioning the cursor to the line having the desired option, the pilot can select his course of action.

Figure 4-60 Alphanumeric Display Functional Diagram



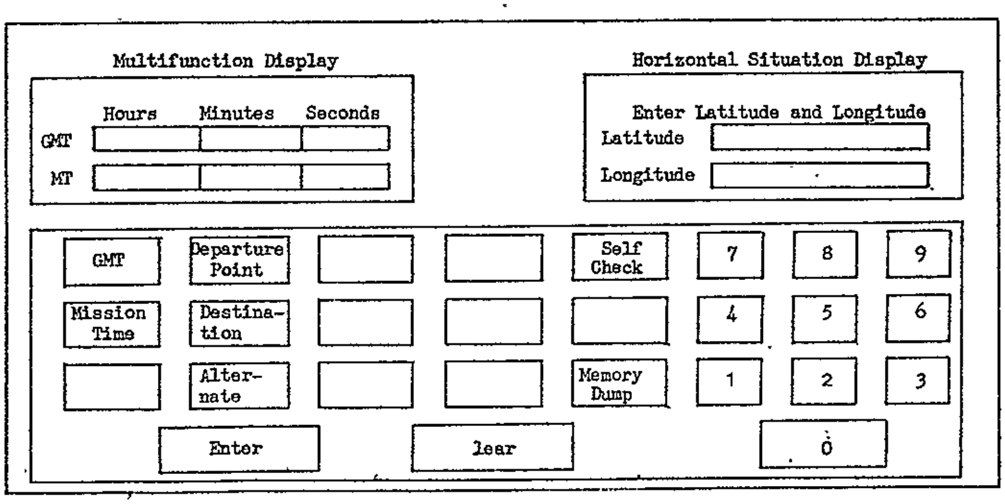


Figure 4-61 Data Entry Keyboard

4.7.10 Data Entry Keyboard

The data entry keyboard provides means for the flight crew to interface with the computer. The keyboard contains function keys which activitate computer programs applicable to mission and inflight testing requirements. Some of these programs need parametric values to complete the function. For this purpose, decimal keys are used to enter numerical data into the computer.

The available CRT's in the display and control subsystem are used as an aid in monitoring flight crew activation of the data entry keyboard, and as a display for the output of various function keys. For example, a portion of the MFD is reserved for Greenwich Mean Time (GMT) and Mission Time (MT) displays. The HSD is used to inform the pilot of action necessary to complete the requested function. The HSD also displays the data which the crew has selected so that it may be verified prior to being entered into the computer. Figure 4.-61 shows an arrangement of the data entry keyboard with its function and data keys, and its interface with the MFD and HSD.

The MFD shows the area reserved for time displays. The HSD shows the display output in response to the flight crew depressing the Destination button. The latitude and longitude is then shown on the HSD as the desired decimal buttons are pushed. If the displayed values are incorrect, depressing the Clear button will remove the numerical parts of the display, and the operator can repeat the selection of the parameter values. When the data is correct, the Enter button is pushed and the numerical data is entered into the computer.

Computer software programs perform the tasks of responding to interrupts caused by data entry keyboard usage, displaying flight crew cues and requested information, controlling peripheral equipment, and formatting and storing data at the proper address. Interface requirements are listed in Figure 4.

4.7.11 Secondary Controls

Various other infrequently used levers, dials, buttons, or knobs are dedicated to miscellaneous subsystem control functions. In general, these devices will be located on the side, overhead or rear panel areas of the cockpit. They include:

- Circuit breakers
- Emergency levers
- Maintenance panel controls
- Communication dials
- Cabin environment controls

The circuit breakers are solid state devices with automatic as well as remote reset and trip capability (7). For each load, the breakers will provide an on-off status which can be programmed through the computer system. Each load has a separate breaker providing overload protection for equipment and wire runs from the power bus to the load. It is estimated that approximately 150 circuit breakers (programmable discretes) will be required for the booster system.

Emergency controls, such as fire extinguisher levers and emergency exit handles, are installed as backup means to ensure crew safety or survival. No interface with the DMS or data bus in anticipated.

Maintenance panel controls may be provided for some onboard maintenance functions which are not integrated with the preflight and operational avionics system. These controls are associated with special test equipment such as oscilloscopes. Interface requirements are limited to controlling and verifying proper positioning of operational or maintenance switching (discretes). It is anticipated that each of the 8 major subsystems will have a small number of controls associated with test points for special checkout procedures.

Twist knobs and thumb wheels are currently the preferred method for communication equipment operation and cabin environment control. The sections on communication and environment describe the interface requirements for these devices.

5.0 DATA MANAGEMENT SYSTEM DESCRIPTION

This section describes in detail the requirements of the DMS in performing the mission functions. The computational requirements of the DMS to satisfy the subsystem functional requirements are developed, and software modules are described. As each software module is developed it is assigned a nemonic name for use as future reference. The DMS is required to solve all of the equations and perform all logical decisions needed for the completion of a successful mission. In accomplishing this a computer program is executed within the DMS.

In order to meet the avionic system functional requirements the DMS must perform computational tasks for each major booster subsystem. The software requirements of the DMS pertaining to the booster systems are described below by flow diagrams, logic diagrams, mathematical equations and verbal description. The requirements of the DMS vary with mission phase. Each of the subprograms described below are required only during certain mission phases as specified in the discussion. It is the task of an executive program to schedule these subprograms during the proper mission phases and at the proper iteration rates. Major mission commitments can only be initiated through the executive by manual command. Certain properties of the overall programming task are dependent upon the DMS configuration, e.g., self test, failure monitoring and reconfiguration, and data bus control. The major booster systems described below are:

Structures

Propulsion

Electrical power generation and distribution

Navigation and Guidance

Flight Control

Communications

Operations Management

5.1 STRUCTURES

Structures include performance monitoring, landing gear deployment, and separation control and monitoring.

5.1.1 PERFORMANCE MONITORING

The performance monitoring task computational requirements include checkout of the sensors, recorder data bus, and interface electronics during the prelaunch mission phase, control of the recording speed during the flight phases, and reduction of the collected data during the post flight phase. The data reduction program required during the post flight phase will be stored in mass storage and will not effect the DMS design. It is assumed that prelaunch checkout programs and in-flight programs will remain resident in the DMS main memory throughout the operational mission. Diagnostic programs and programs to reduce data collected during the mission will reside in mass storage until they are specifically called for by an operator at which time they will overwrite the operational programs. The performance monitoring function requires two software modules which are described below.

A. <u>Performance Monitoring Checkout Program (PMCP)</u>

This program will be scheduled through the executive by the master checkout scheduling program. It will be executed at an iteration rate of 8 times per second. The time at which the program is scheduled during prelaunch is not critical in that loss of the total performance monitoring function is not critical to mission success. However, a better test environment would be present if the performance monitoring testing would occur simultaneously with the occurrence of some prelaunch activity which would stimulate the vibration sensors such as fueling the booster, attaching the orbiter, erecting the booster, testing the landing gear system, etc. There are four types of recordings which must be tested by this program:

Test Record Fast

Test Record Slow

Record Fast

Record Slow

In the test record fast and slow tests a known test signal is recorded on the tape. In the standard record mode actual sensor outputs are recorded. Figure 5-1 is a flow diagram of the total requirements of this program. Program initiation includes the setting of flags and index registers to control the flow through the program. At the beginning of the program electrical power is turned on to the recorder and recorder data bus electronics and a test made to determine if the electrical voltage to the units is within limits. At the end of the program the reverse function is provided of turning the power off and testing for power off. In the rewind operation a timer is set up and the recorder continuously tested for rewind until the timer runs out. If the timer runs out before rewind indication is received an error message is displayed indicating the inability to rewind and the program descheduled. A timer is set for the record operation of sufficient length to allow for the recording of the number of records desired for testing. In the test recorded data functions a timer is set and the replay command issued. If replay is not finished before the timer has run out an error message indicating the inability to replay is issued and the program is descheduled. In testing the recorded data

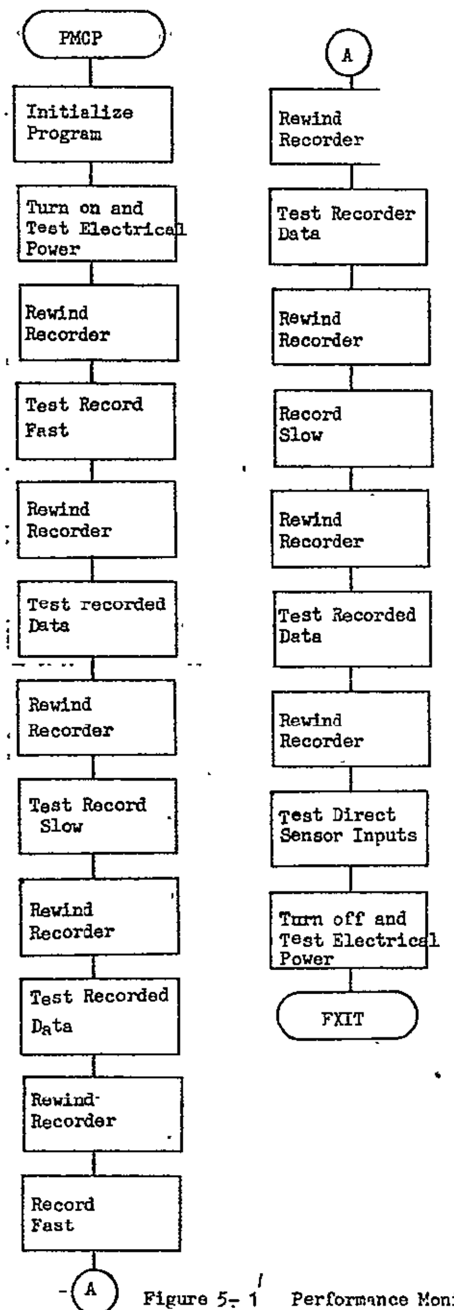


Figure 5-1 Performance Monitoring Checkout Program (PMCP)

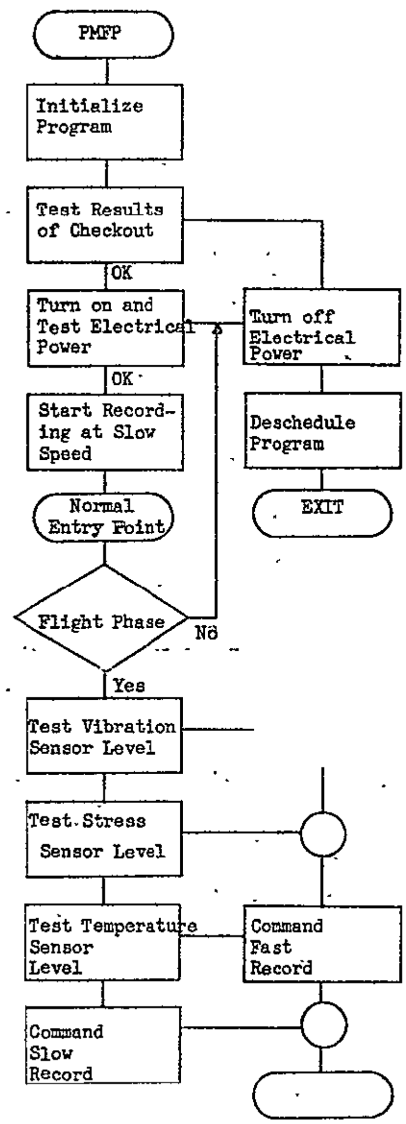


Figure 5-2 Performance Monitoring Flight Program (PMFP)

the vibration data is tested, the end of record word tested for stress and temperature sensor data present which, in turn is tested if indicated present, and the end of record word tested for indication of the proper recorder speed. A counter is set to control the number of records tested. A test is made of the direct input vibration, stress, and temperature sensors. A error indication is issued whenever any test fails which is displayed on one of the multipurpose CRT displays and/or printed. If any test results indicate the inability to control the performance monitoring equipment sufficiently to proceed with the test the program is descheduled. If all tests show proper operation a test completed with proper operation message is issued. A flag is issued to the performance monitoring flight program indicating the ability or inability to record during flight.

B. Performance Monitoring Flight Program (PMFP)

The performance monitoring flight program is scheduled just prior to rocket engine ignition. Its function is to initially turn on the recorders and then to control recorder speed during the flight mission phases, increasing recorder speed during periods of high sensor activity and decreasing recorder speed during periods of low sensor activity. Figure5-2 is a flow diagram of the Performance Monitoring Flight Program. The program contains an initial and normal entry point. In the initial entry the program is initialized by descheduling the initial entry point and scheduling the normal entry point, the test results of the checkout program are tested, the power turned on and tested, and the recorder and recorder data bus started at a slow speed. In the normal entry program the mission phase is tested to determine if the flight phase is over and the vibration, stress and temperature sensors tested. There are three vibration sensors, three stress sensors, and three temperature sensors. In testing for recorder speed the median sensor output value of each sensor type is selected and compared against limit values. If any one of the sensor types is above the limit value the fast recording speed is commanded. If all of them are below the limit values the slow record speed is commanded. If the initial entry tests show that the prelaunch checkout indicated inoperative equipment or that the power cannot be turned on or if the normal entry program indicates that the mission flight phases are finished the power to the recorder and recorder data bus is turned off and the performance monitoring flight program descheduled. This program is run at an 8 per second rate.

5.1.2 LANDING GEAR SYSTEM

The landing gear system computational task includes a full prelaunch checkout of the landing gear system, a partial checkout of the system during flight, the sequence required to lower the gear for landing, and the sequence required to raise the gear after takeoff on a ferry mission. The computational tasks also include nose gear and main gear steering. There are three landing gear, the nose gear, the right main gear, and the left main gear on the space shuttle booster. The operation of all three landing gear is the same except for the steering function. This identity of operation of the three landing gear allows for the sharing of some program functions. A description of each program module is given below.

A. <u>Landing Gear System Full Checkout Program (LGFC)</u>

At some point prior to launch the booster must be lifted from its normal horizontal landing position and errected in a vertical position on the launch pad. During this period the landing gear are not required to support the booster, and can be tested while being commanded through a full raise and lower cycle. Figure 5- 3 is a flow diagram of the landing gear system full checkout program. It is assumed that when the program is entered the landing gear will be in a fully lowered condition and that when testing is over the landing gear will be fully raised. The first step upon entry to the program is to initialize the program. Initialization includes setting a counter and a program control flag. The counter is used to determine the number of cycles through which the gear is to be tested. If the counter is initialized to 1 the gear will be raised only, if to 2 it will be raised, lowered and raised, etc. The flow diagram shows a straight line flow through the program, however, when actually programmed for the computer many exits to the executive will be inter-

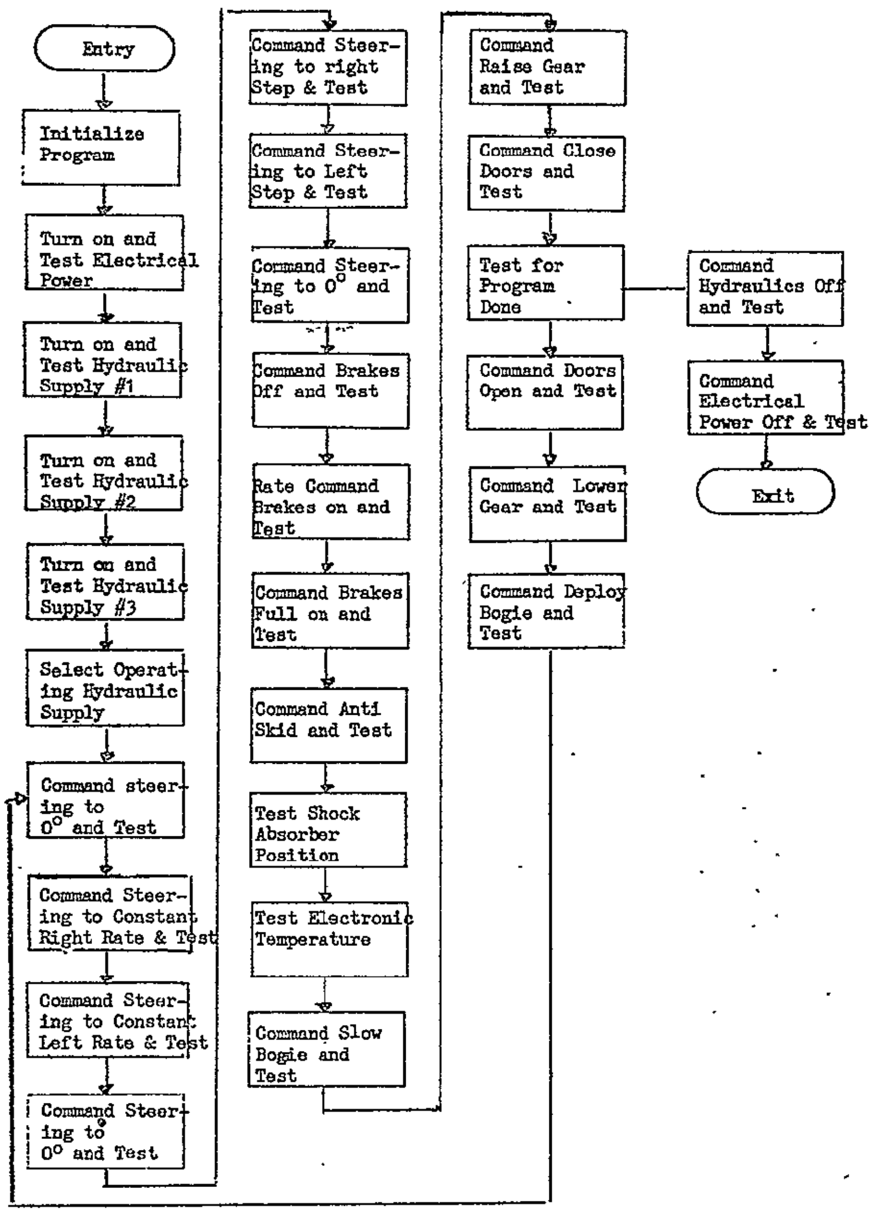


Figure 5-3 Landing Gear System Full Checkout Program

leaved with the program. The program control flag will be advanced as portions of the program are completed and used to determine where to continue computing with each return from the executive. After program initialization electrical power is supplied to all three landing gear and tested. It is assumed that during this portion of the prelaunch testing electrical power will be provided from umbilical ground power and a failure indicates an inability to switch power to the system. If the test fails an error message is issued and testing halted on the failed landing gear. A test philosophy is assumed that testing will continue to what ever degree possible in the event of a failure for the purpose of possibly uncovering additional faults. Next hydraulic supply number 1 is commanded to supply each landing gear system. A delay is then programmed to allow for hydraulic supply transients to decay after which time the supply pressure to each landing gear is tested. Hydraulic supply #1 is then commanded to disconnect and a time delay programmed to allow for the disconnect. The same test sequence is then repeated for hydraulic supply #2 and #3. An operating hydraulic supply is then selected for each landing gear. The steering mechanism for each landing gear is then commanded to 0° and a time delay coded to allow sufficient time for the steering to achieve 0°. After the time delay has elapsed the steering feedback is tested for 0°. If the steering for any gear cannot be commanded to 0° the testing of that gear is halted and an error message issued. Each landing gear steering mechanism is then commanded to turn at a constant right turning rate by continuously incrementing a positioning command. The rate error of each steering mechanism is tested by comparing the commanded position to the feedback position. At the same time the steering mechanism hydraulic pressure and temperature is tested. This test is conducted for a predetermined period of time as controlled by a programmed timer. This test is then repeated for a constant left turning rate. The steering mechanism is then commanded to return to 0° in preparation for a step command test. In the step command test a step steering command to a full right position os applied. Upper and lower bounds as a function of time are developed in the DMS and compared with the actual feedback position. This test thus measures the time response and final positioning error of the steering servo system. The steering system for each landing gear is then tested for its step response to a full left position command in the same manner that the right step command test was performed. The steering mechanism for each gear is then commanded to zero degrees and tested after a time delay. The brakes on each wheel of each landing gear are then commanded off and after a time delay to allow for brake pressure to be removed the hydraulic pressure and temperature and brake pressure and temperature for each brake is tested within limits. A brake rate command of constantly increasing pressure is then commanded with the hydraulic and brake pressures tested to computer generated boundaries over a period of time. The brakes are then commanded to full on and after a delay sufficient to allow for transients to decay the brake and hydraulic pressure and temperature are tested against limit values. The anti-skid valves for each brake of each landing gear are then commanded open and after a delay of sufficient time for transients to decay the brake and hydraulic temperature and pressure is tested. The anti-skid valves are then commanded to close and a delay sufficient to allow for closure awaited before the hydraulic and brake pressure is again tested. The shock absorber position and the temperature of the electronics for each landing gear is then tested. A test is then made of the ability to stow the bogie. The procedure for this test is to set a timer and then issue an unlock command to the bogies of each landing gear. The bogie deployed and locked signal is then interrogated until an unlocked condition occurs or until the timer runs out. If the timer runs out first the test fails and further testing of the failed landing gear is discontinued. After receiving an unlocked indication the stow bogie command is issued and another programmed timer initiated. The bogie position is then continuously compared against computer generated upper and lower limits. At the same time hydraulic pressure and temperature are measured. The timer and bogie stowed and locked signal are continuously monitored. If the timer runs out before the locked signal occurs the test has failed and further testing of the failed landing gear is discontinued. If the locked signal occurs first a check is made to determine if the lock signal occurred between the proper time limits. Tests are then made of the ability to raise each landing gear and close each door. These tests are conducted by the same procedure as the stow bogie test. The counter set up in the program initialization is then decremented and tested to determine if a further cycling of landing gear testing is desired. If further testing is desired a test of opening the doors, lowering the gear and deploying the bogie for each landing gear is conducted in the same manner as the stow bogie test. The program is then returned to the point where the steering system was first commanded to 0°. If no further testing is desired a command is issued to remove the hydraulic supply from each landing gear and after a time delay of sufficient length to allow for the hydraulic supply to be removed a test is conducted to insure its removal. The electric power to each landing gear system is then removed and tested for removal. If all landing gear tests passed a message is issued indicating total test success and the landing gear system full checkout program descheduled.

This tests the total landing gear system except for the wheel speed sensors. To test the wheel speed sensors at this time would require special equipment to rotate the wheels at a fixed rate. This test will be incorporated in the steering programs and thus, automatically tested whenever the vehicle is taxied.

This program is executed at a 4 per second rate.

B. Landing Gear System Partial Checkout Program (LGPC)

This program tests all landing gear functions which can be tested while the landing gear remains in a stowed position. Its primary function is to provide increased confidence in the braking system prior to landing. Figure 5-4 is a flow diagram of the landing gear system partial checkout program. The block diagram depicts the program as a straight line flow with a single entry and exit. The actual program will have multiple entries from the executive program. A control flag will be used to route the flow through the program at each entry from the executive. The initialization program initializes this flag. This test program can be selected by the pilot during any mission phase where the landing gear are raised. The program checks the brakes by applying brake command signals and could result in a

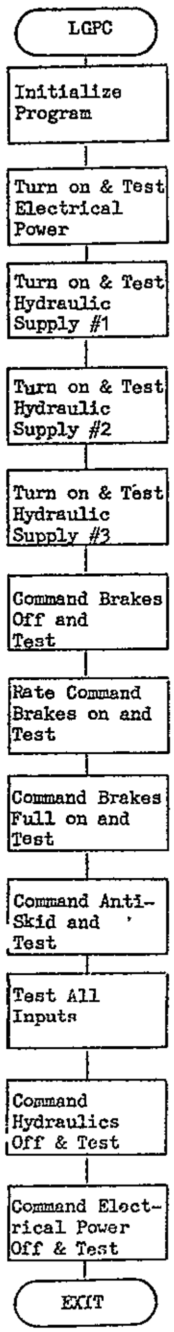


Figure 5-4 Landing Gear Program Partial Checkout Program (LGPC)

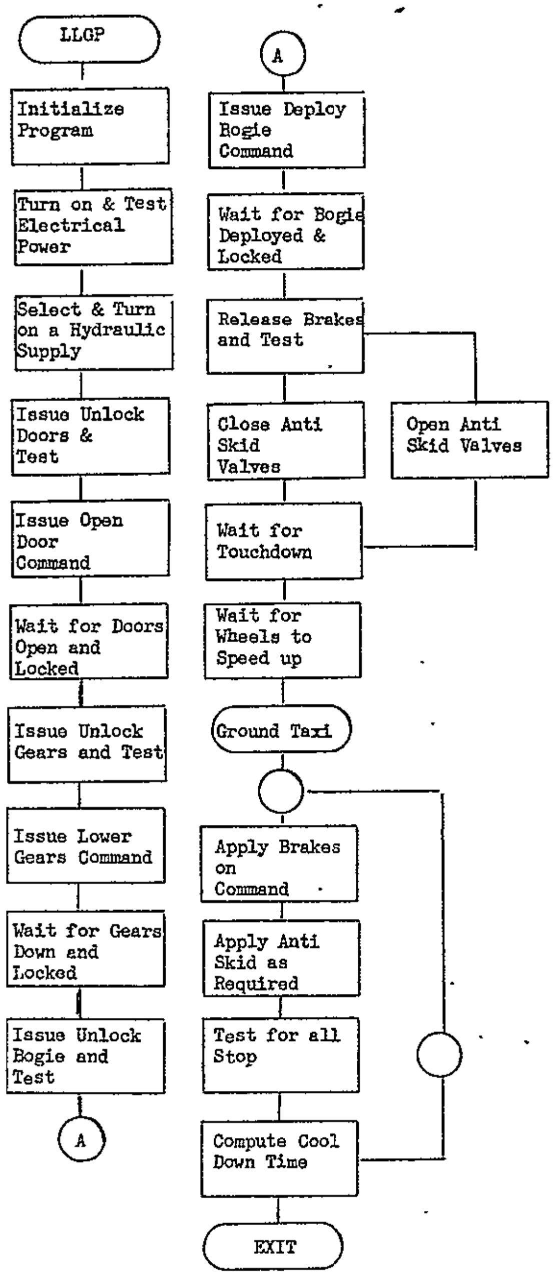


Figure 5-5 Lower Landing Gear Program (LLGP)

accident if run during landing or horizontal takeoff. For this reason at every entry from the executive to this program, a test will be made that the doors are closed and locked; if they are not the program will be automatically deschedule. After initialization the electrical power is turned on and tested in the same manner as done in the Landing Gear System Full Checkout Program (LGFC). The availability of each hydraulic supply and the selection of one supply is tested in the same manner as was done in the LGFC program with the exception that the results of the hydraulic supply system checkout program are tested first to determine if any hydraulic system failure has occured before a hydraulic valve is opened. A hydraulic supply system failure could be caused by a loss of hydraulic fluid or contamination of the hydraulic fluid. By not opening a supply valve to a failed hydraulic system will avoid bleeding the hydraulic fluid from the landing gear system or contaminating the landing gear system hydraulic fluid. After a hydraulic supply has been chosen and connected to the brake system, the brakes and anti-skid valves are tested in the same manner as was done in the LGFC program. After the brakes have been tested, all outputs from the landing gear system to the DMS are tested. The hydraulic supply and electrical power is then turned off and tested for off as was done in the LGFC program. If all tests passed, a message indicating successful results is issued and the program descheduled. Messages for failures are issued as failures are encountered. This program is run at a repitition rate of 4 times per second.

C. Lower Landing Gear Program (LLGP)

This program is initiated by the pilot during approach and remains scheduled through the landing and taxi phases. This program controls the lowering of the landing gear and braking the vehicle once on the ground, including the anti-skid operation. As soon as the landing gear is lowered, this program schedules the landing gear steering program. Figure5-5 is a flow diagram of the Lower Landing Gear Program. The program as shown in the flow diagram has a continuous straight line flow; when programmed returns to the executive must be inserted wherever a wait is indicated, and after each pass through the brake/anti-skid control program loop. Points of return to the program from the executive are controlled by a flag which is initially set in the initialization program. After initialization, an electrical power on command is issued, and a test made for the power on. A hydraulic supply is selected and turned on. The hydraulic supply selected is that operating supply having the least load. After waiting for transients caused by connecting the supply to decay the supply pressure is tested. If the test indicates a failure, a different supply is selected. An unlock door command is then issued and the doors closed and locked discrete interrogated. A timer is set at the time of the issuance of each command in the landing gear lowering sequence. If the timer runs down before the commanded event is completed, an alarm is issued to the pilot and the command continuously reissued. If the commanded event then occurs, the alarm is removed. After the doors are unlocked, an open door command is issued. The door position feedback is continuously monitored and if the opening rate drops below a specified value, the command is reissued. As soon as the door is locked, the commands unlock gears, lower gears, unlock bogie and deploy bogie are issued in sequence, as each issued command is indicated completed. Tests are made and alarms are issued for the lowering of the gears and bogie deployment in the same manner as done for the door opening function. As soon as the doors, gears, and bogies are fully extended and locked, a command is issued to release the brakes and close the anti-skid valves. If the brakes will not release as measured by brake pressure a warning is issued and the anti-skid valves opened. Opening the anti-skid valves causes a hydraulic bypass which should release the brakes. If they still do not release, an alarm is issued. If the brakes do not release except by opening the anti-skid valves, braking during landing will be provided by pulse width modulation of the anti-skid valves. After releasing the brakes, the program waits for a touchdown indication. Touchdown is determined by shock absorber position. After touchdown wheel speed is observed until the rotational speed of the wheels reaches a minimum value at which time the program enters a brake control loop. The normal mode in this loop is to interrogate the pilot's braking command from the rudder pedals and apply a proportional command to the brakes. The speed of each wheel is then interrogated and if any wheel is slowing much more rapidly than the average, or is rotating much more slowly than the average, its anti-skid valve is opened until its rotational speed again assumes a value closer to the average. During the braking operation, the integral of wheel speed times brake pressure is accumulated. After the vehicle is stopped, this accumulated integral for each wheel is used in computing the time required for the brakes to cool before further braking should be applied. Differential braking will be applied to the main landing gear wheels if uneven rudder pedal pressure is applied.

If the brakes for any wheel could not be released except through use of the anti-skid valve, when the wheels were first lowered, the anti-skid valve for that wheel will be pulse width modulated in order to obtain proportional braking pressure. This program will be run at 4 times per second. A special entry point to the braking program is provided for use in ground taxi operations.

D. Landing Gear Up Warning Program

This is a short monitoring program automatically scheduled at the start of the powered cruise mission phase after the airbreathing engines have been deployed and started. It is used to sound an audio alarm to the pilot in the event a landing configuration is commanded before the landing gear are fully lowered. A flow diagram of this program is shown in Figure 5-6 . If either the engines are commanded to idle speed or the flaps are lowered, a test is made to determine if all landing gear doors are open and locked, the gears down and locked, and the bogie deployed and locked, and if not, a command is issued to sound the audio alarm. This program is scheduled at a once per second iteration rate.

E. Raise Landing Gear Program (RLGP)

This program is scheduled by the pilot prior to horizontal takeoff for a ferry mission. The function of the program is to provide braking if needed during takeoff, and to raise the landing gear after takeoff. A flow diagram of the program is shown in Figure 5-7 . Upon initial entry to the program, a flag is set to control flow through the program. A loop similar to the braking loop in the lower landing gear program is then entered in which brake commands are applied proportional to pilot rudder pedal deflection, anti-skid valves are opened based upon wheel speed and rate of change of wheel speed, and

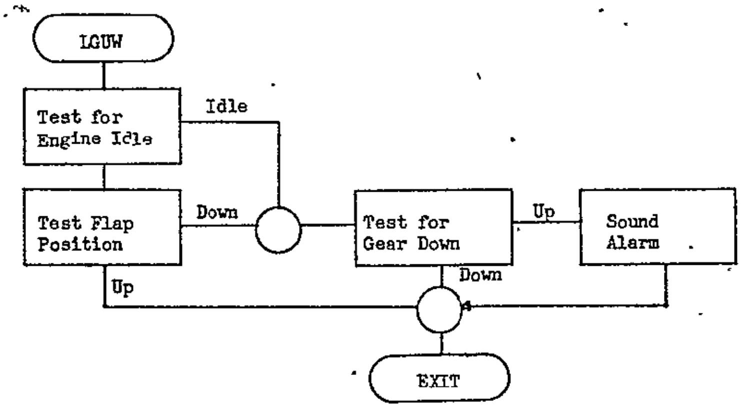


Figure 5-6 Landing Gear Up Warning Program (LGUW)

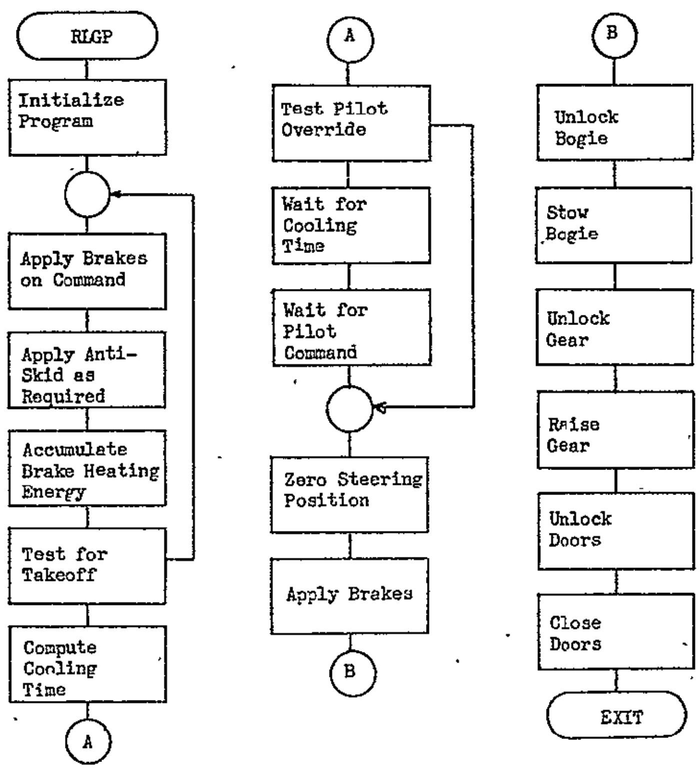


Figure 5-7 Raise Landing Gear Program (RLGP)

the integral of wheel speed times brake pressure for each wheel is accumulated. A test for vehicle takeoff is used to exit the loop. It is assumed that takeoff has occurred when the shock absorber position shows a fully extended shock absorber and a sufficient delay is awaited to allow for any vehicle bounce at takeoff. After takeoff, the time for the braks to cool is computed and unless a pilot override command is issued, this time is allowed to elapse before the normal raise gear command from the pilot is recognized. Upon recognizing a raise gear command, the nose and main gear steering is commanded to zero and the brakes on all wheels are applied to full pressure to stop wheel rotation before raising the gear. After the wheel rotation has stopped, the sequence of unlock bogie, stow bogie, unlock gear, raise gear, unlock doors, and close doors is commanded with each command issued only after the completion of the previously commanded event. A warning is issued to the pilot if any commanded function fails to perform. This program is run at a 4 per second iteration rate.

F. <u>Landing Gear Steering Program (NWSP & MGSP)</u>

This program is scheduled at the same time the lower landing gear program is scheduled, and deschedules itself when takeoff occurs. The steering system for each landing gear is controlled by a closed loop servo system as shown in Figure 5-8. The feedback signal is subtracted from the steering command to form a steering error signal. The error signal is digitally filtered and multiplied by a gain. The assumed filter has a first order numerator and denominator. The filter output is placed on the data bus system and forms the hydraulic value command to the landing gear hydraulic steering servo. Before touchdown the steering command to each landing gear is equal to the

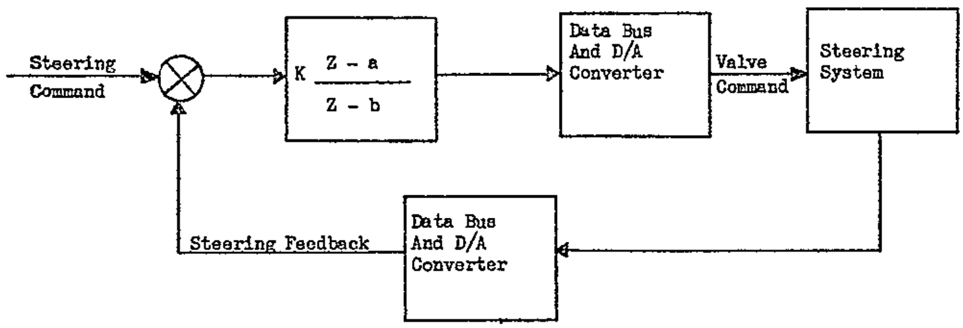


Figure 5-8 Landing Gear Steering Servo

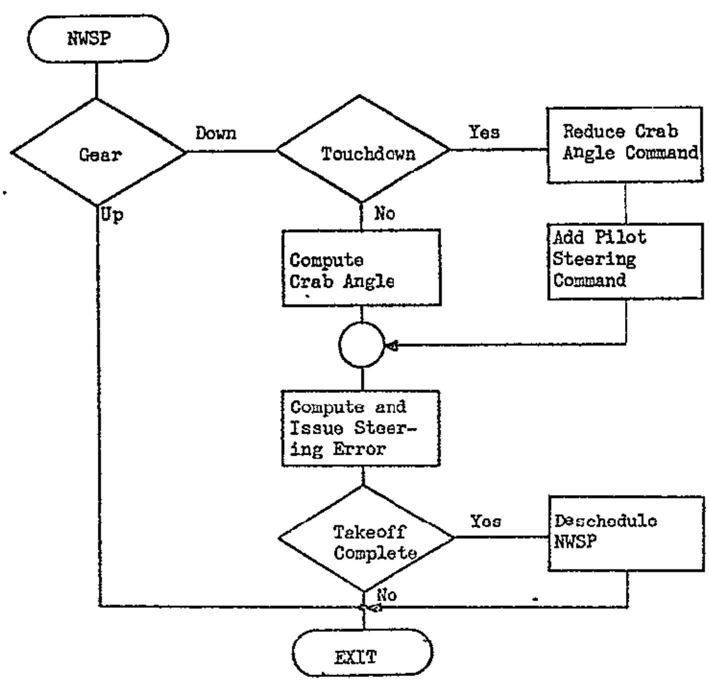


Figure 5-9 Nose Wheel Steering Program (NWSP)

crab angle of the vehicle. The crab angle is the angle between the vehicle longitudinal body axis and the vehicle velocity with respect to the ground. This aligns the wheels of all three landing gear with the vehicle ground velocity eliminating any side or turning forces at touchdown. After touchdown, this steering command is slowly driven to zero as a function of time. For the nose wheel the pilot's steering command from the yoke is added to the decaying pre-touchdown command. Figure 5-9 is a flow diagram of the nose wheel steering program. Upon entry to the program a test is made of the status of the lower landing gear program to determine if the landing gear have been lowered. If not, the program exits to the executive. If the gear has been lowered, a test of the status of the lower landing gear program is made to determine if touchdown has occurred. If touchdown has not occurred, the crab angle steering command is computed. If touchdown has occurrred the crab angle steering command is reduced toward zero and the pilot's steering command added to it. After generating a steering command the steering error is computed by subtracting the steering feedback signal. The steering error is then filtered, a gain applied and output to the landing gear system. The status of the raise landing gear program is then tested to determine if takeoff has occurred. If takeoff has occurred, the landing gear steering program is descheduled. The main wheel steering program is identical to the nose wheel steering program except that the pilot steering command is not added to the steering command after touchdown and two steering errors and filters must be computed, one for each main gear. The nose wheel steering program is run at an 8 per second iteration rate and the main gear steering at a 4 per second iteration rate.

5.1.3 SEPARATION CONTROL

The separation control computational requirements include checkout, monitoring and abort functions. A description of the programs required for separation control is given below.

Prelaunch Checkout (SSPC)

Figure 5-10 is a flow diagram of the prelaunch checkout program for the separation control system. This program is scheduled by the master prelaunch checkout scheduling routine at an 8 per second rate. The flow diagram indicates a straight flow through the program. In the DMS the program will be executed with intermediate returns to the executive program. After the issue of a command at least one return to the executive will occur before the program is reentered. Control of the flow through the program is accomplished with a flag initialized with the first entry to the program. The testing sequence is turned on to test the electrical supply voltage to the electronics. Test for a separation indication on all three bolts by testing the temperature pulse and resistance indicators. Command and test for the test configuration. Issue arming and separation commands and test for test configuration indications of temperature pulse and resistance separation signals. Test squib and electronic temperatures. Turn off separation command, arming command and electrical power. All three bolts are tested simultaneously. If any test fails a message indicating the type of failure is displayed and the turn off procedure of turning off the separation command turning off the arming command, and turning off the power immediately performed. Any indication of a fault in the system will cause an immediate shutdown of the system in an attempt to avoid a possible squib ignition during prelaunch.

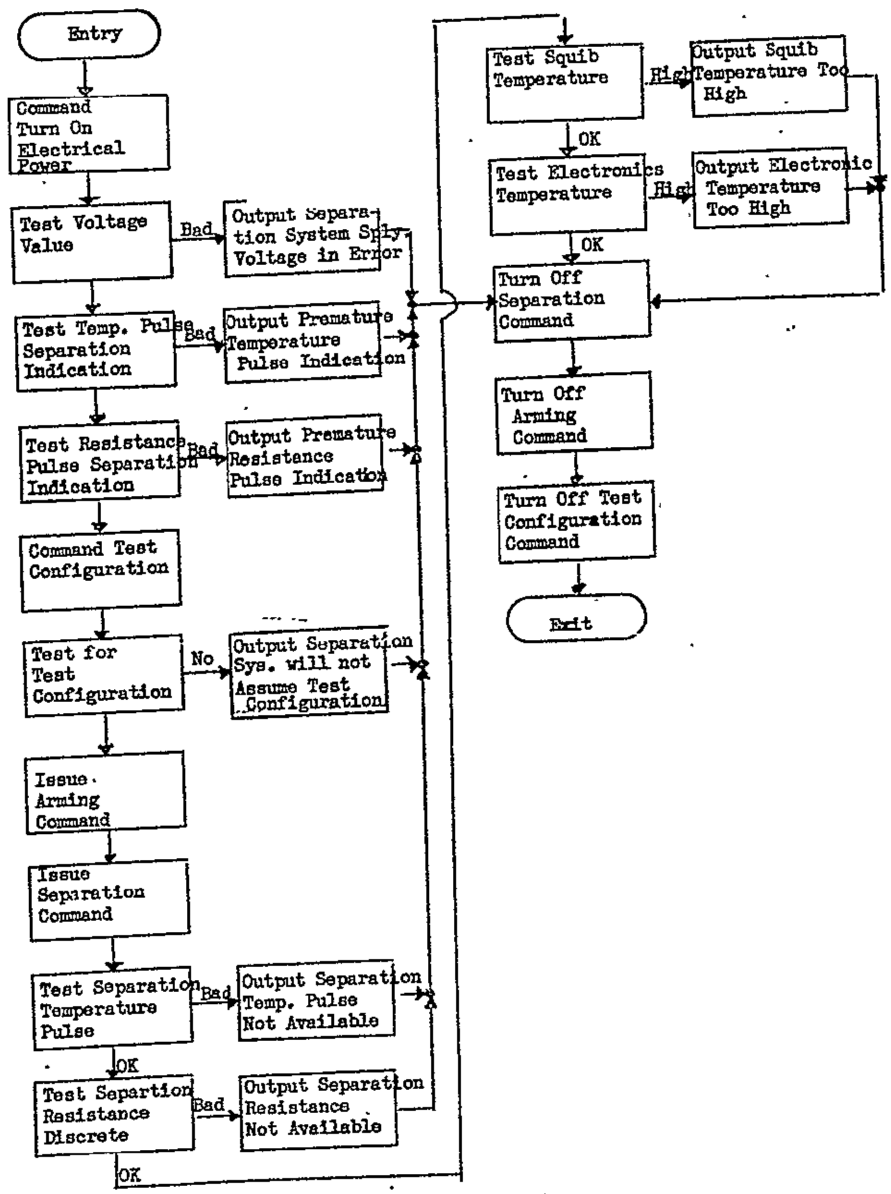


Figure 5-10 Separation System Prelaunch Checkout

Boost Monitoring (SSBM)

The separation system boost monitoring program is a portion of an overall boost system monitoring program initially scheduled just prior to main engine ignition and descheduled at main engine thrust termination time. The program is run at an iteration rate of once per second. A flow diagram of the program is shown in Figure 5-11. On the initial pass through the program the electrical power is turned on. On subsequent passes squib temperature, electronics temperature, and supply voltage is tested. A failure in any of these tests causes the issuance of a warning message and the electrical power to be turned off. Electrical power is turned off in the event of excessive electronics temperature or an out of tolerance supply voltage in order to halt a possible faulty issuance of a separation signal. The temperature pulse and resistance separation indicators are tested with warning messages issued if any separation indications exist. In this program the squib temperatures and separation indicators on all three explosive bolts are checked.

Boost Separation Abort (SSBA)

If any of the monitoring programs show cause for an immediate separation of booster and orbiter the monitoring programs will issue an arm and separation command and schedule the boost separation abort program at a 16/sec iteration rate. The boost separation abort program is shown in Figure 5-12. Upon initial entry to the program a flag is initialized to control the flow of the program with repeated returns from the executive and a timer is set. The supply voltage is then tested to assure that voltage has been applied

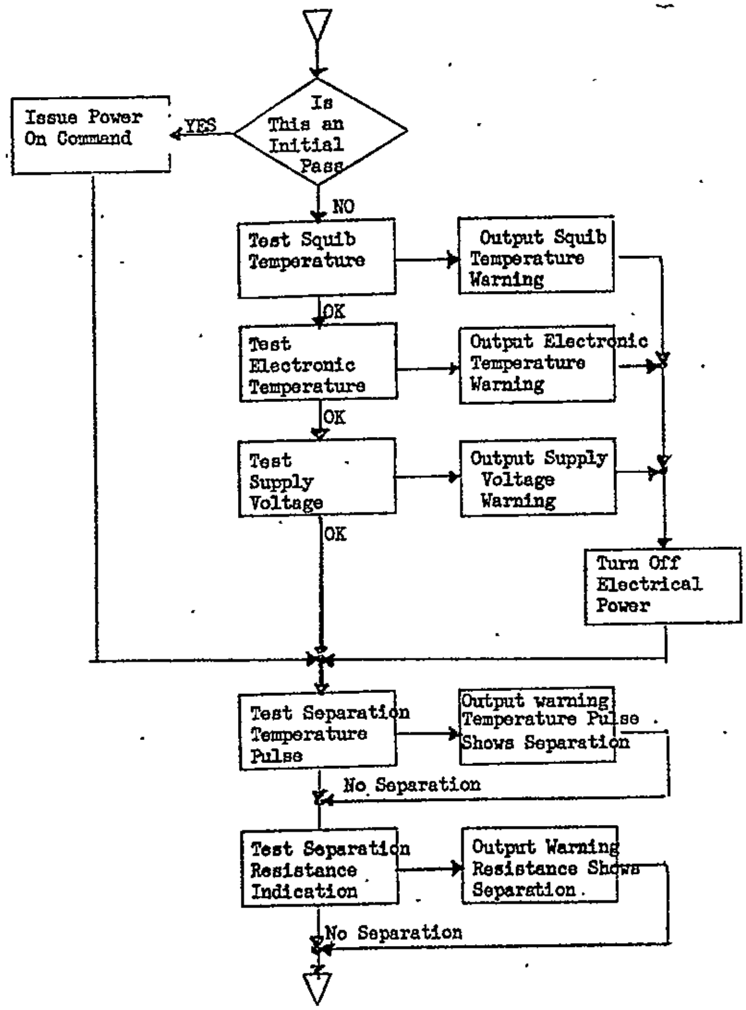


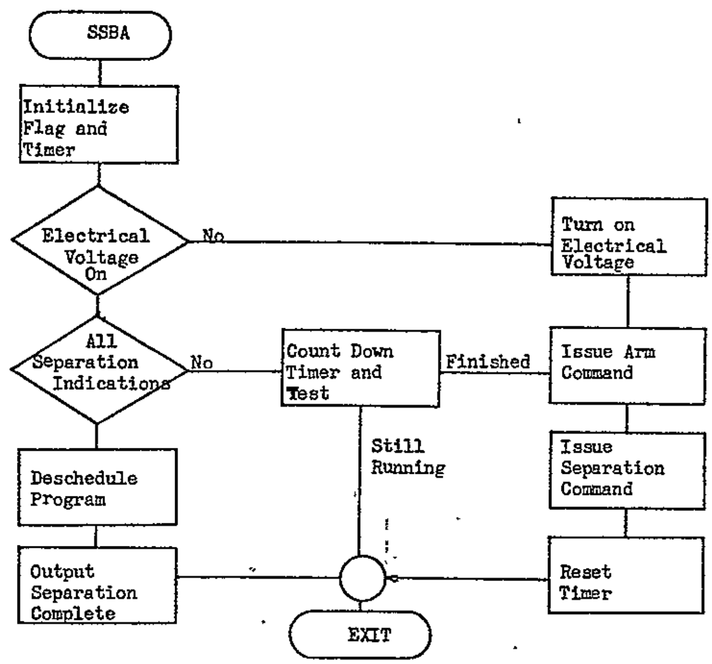
Figure 5-11 Separation System Boost Monitoring

to the separation system electronics. If the voltage is absent it is turned on and the arm and separation commands reissued. The normal path through the program tests the separation temperature pulse and resistance indicators for all three explosive bolts descheduling the program if all separation indications are present. If separation has not occurred the timer is decremented and tested. If the time has run out the arm and separation commands are reissued and the timer reset.

Thrust Termination Separation (SSTS)

At the time of thrust termination when normal separation should occur the orbiter will have primary control of the separation task. If the orbiter equipment fails to accomplish a complete separation the booster will issue separation commands. Figure 5-13 is a flow diagram of the booster program used to monitor the separation sequence and issue separation commands upon failure of the orbiters system. Upon initial entry to the program a flag is set to control flow through the program with multiple returns from the executive and a timer is set. The electrical supply voltage to the separation system electronics is then tested and turned on if absent. The existance of temperature pulse and resistance separation indicators of all three explosive bolts are then tested. If separation has occurred a message is issued and this program descheduled. If separation has not occurred the timer is counted down. When the timer is counted to zero . arm and separation command are then issued and the Boost Separation Abort program scheduled. This program is then descheduled letting the abort program complete the separation testing. This program is scheduled at a 4/sec rate.

Figure 5-12 Boost Separation Abort (SSBA)



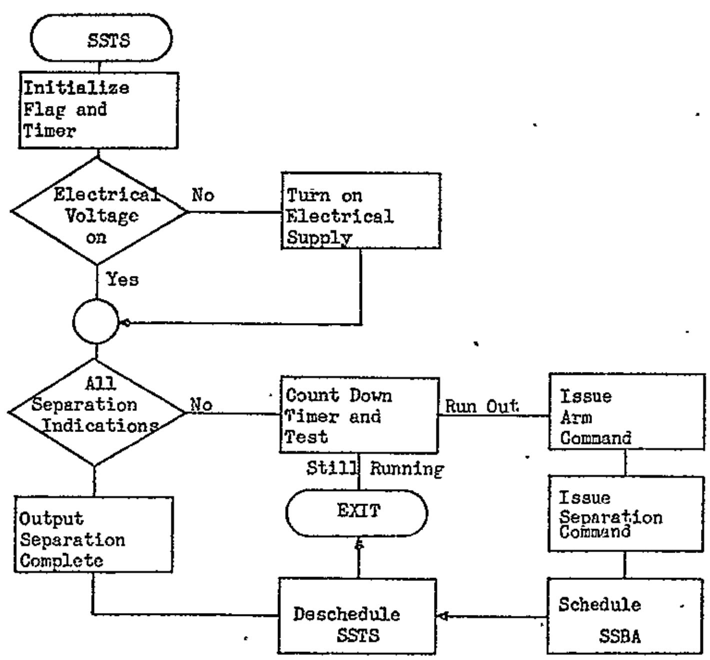


Figure 5-13 Thrust Termination Separation (SSTS)

5.2 Propulsion System

The booster has 3 propulsion systems, the main boost rocket engines, the reaction jet system and the cruise engine system.

5.2.1 Main Rocket Engines

Each rocket engine has four primary control points which are the low speed inducer LOX valve, the main burner LOX valve, the preburner fuel valve, and the preburner LOX valve. Each of these valve positions are controlled by proportional controlled actuators. The commands to each primary control point are designated X_1, X_2, X_3 and X_4 respectively. The engine control equations are non linear functions of the thrust and mixture ratio commands and sensed engine parameters. The non linear functions will be obtained through a table look-up process. Each equation as presented will include a reference to the interation rate required for the equation solution and if a table look-up is required, the number of points which must be stored in the table.

Thrust Command and Mixture Ratio Development

The thrust command and mixture ratio are computed as a non linear function of vehicle velocity and altitude from the formulas

$$V = \sqrt{V_x^2 + V_y^2 + V_z^2} \qquad \text{(2/sec)} \qquad (1)$$

$$T_R = f_1(V) \qquad \text{(10 points) (2/sec)} \qquad (2)$$

$$M_R = f_2(h) \qquad \text{(8 points) (2/sec)} \qquad (3)$$

where V_x, V_y and V_z are inertial velocity components and h inertial altitude from the navigation system.

Main Pump Overspeed Protection

A parameter α_{13} which contributes to the X_1 command is computed from the process

$$Y_1 = N_{PL} - N_{PL\,MAX} \qquad (64/\text{sec}) \qquad (4)$$

$$Y_2 = N_P - N_{P\,MAX} \qquad (64/\text{sec}) \qquad (5)$$

$$Y_{3_n} = K_1 Y_{1_n} + K_2 Y_{1_{n-1}} + K_3 Y_{3_{n-1}} \qquad (64/\text{sec}) \qquad (6)$$

$$Y_{4_n} = K_4 Y_{2_n} + K_5 Y_{2_{n-1}} + K_6 Y_{4_{n-1}} \qquad (64/\text{sec}) \qquad (7)$$

$$Y_5 = \begin{cases} Y_3 & \text{if } Y_3 \geq Y_4 \\ Y_4 & \text{if } Y_4 > Y_3 \end{cases} \qquad (64/\text{sec}) \qquad (8)$$

$$\alpha_{13_n} = K_7 Y_{5n} + K_8 Y_{5_{n-1}} + K_9 \alpha_{13_{n-1}} \qquad (64/\text{sec}) \qquad (9)$$

where N_{pl} is the LOX turbopump speed and N_p the LH_2 turbopump speed. $N_{pl_{max}}$, $N_{p_{max}}$, K_1, K_2 ... and K_9 are constants.

Maximum Preburner Temperature Limiting

A parameter X_{43} is computed which limits the preburner temperature through its contribution to X_4. X_{43} is determined from the process

$$T_{PB_m} = MAX(T_{PB_1}, T_{PB_2}, \cdots T_{PB_8}) \qquad (2/\text{sec}) \qquad (10)$$

i.e., T_{pB_m}, is the maximum value of T_{pB_1} thru T_{pB8} where T_{pB_1} thru T_{pB8} are the preburner temperature sensor outputs.

$Y_6 = f_3(N_P)$ (4 points) (64/sec) (11)

$Y_7 = T_{PB_M} - Y_6$ (64/sec) (12)

$Y_{8_n} = K_1 Y_{7_n} + K_2 Y_{7_{n-1}} + K_3 Y_{8_{n-1}}$ (64/sec) (13)

$X'_{43_n} = K_4 Y_{8_n} + K_5 Y_{8_{n-1}} + K_6 X'_{43_{n-1}}$ (64/sec) (14)

Open Loop Command

The basic open loop command in determined from T_R and M_R by the process

$\dot{\alpha}_{NOM_n} = K_1 (\alpha_{NOM_n} - \alpha_{NOM_{n-1}})$ (2/sec) (15)

$\alpha_{NOM_{n+1}} = \left[K_2 \left[\dot{\alpha}_{NOM_n}\right]_{\pm L_1} + K_3 T_R\right]_{\pm L_2}$ (2/sec) (16)

where L_1 and L_2 are limit valves on the bracketed quantities. α_{NOM_n} and $\alpha_{NOM_{n-1}}$ are initially set to a constant valve.

$B = \begin{cases} f_4(\dot{\alpha}_{NOM}) & \text{for } \dot{\alpha}_{NOM} \geq K \\ 0 & \text{for } \dot{\alpha}_{NOM} < K \end{cases}$ (6 points)(2/sec) (17)

$T_Z = K/(1-B)$ (2/sec) (18)

$Y_{9_n} = K_1 (1-B)\left[(T_Z+1) M_{R_n} - T_Z M_{R_{n-1}}\right] + K_2 Y_{9_{n-1}}$ (2/sec) (19)

$\beta_{NOM_n} = Y_{9_n} + K_3 B - K_4 \beta_{NOM_{n-1}}$ (2/sec) (20)

$Y_{10_n} = (1-B) M_R + K Y_{10_{n-1}}$ (2/sec) (21)

$\beta'_{NOM} = K_1 Y_{10} + K_2 B$ (2/sec) (22)

Main Pump Speed Control

The main pump speed control contribution is computed from

$N_{PREF} = f_5(\alpha_{NOM}, \beta_{NOM})$ (24 points)(2/sec) (23)

$$Y_{11} = N_{PREF} - N_P \qquad (64/\text{sec}) \qquad (24)$$

$$Y'_{42_n} = K_1 Y_{11_n} + K_2 Y_{11_{n-1}} + K_3 X'_{42_{n-1}} \qquad (64/\text{sec}) \qquad (25)$$

$$X_{43} = \begin{cases} X'_{43} & \text{if } X'_{43} < X_{43_{MAX}} \\ X_{43_{MAX}} & \text{if } X'_{43} > X_{43_{MAX}} \end{cases} \qquad (64/\text{sec}) \qquad (26)$$

$$X_{42} = \begin{cases} X'_{42} & \text{if } X_{43} + X'_{42} \leqslant X_{43_{MAX}} \\ X_{43_{MAX}} - X_{43} & \text{if } X_{43} + X'_{42} > X_{43_{MAX}} \end{cases} \qquad (64/\text{sec}) \qquad (27)$$

LOX Flow Trim

Trim compensation for LOX trim is computed from the procedure

$$W'_{L_n} = K_1 W_{L_n} + K_2 W_{L_{n-1}} + K_3 W_{L_{n-2}} \qquad (8/\text{sec}) \qquad (28)$$

where W_L is the sensed LOX flow

$$W_{L_{REF}} = f_6(\alpha_{NOM}, \beta_{NOM}) \qquad (24 \text{ points})(2/\text{sec}) \qquad (29)$$

$$Y_{12} = W_{L_{REF}} - W'_L \qquad (8/\text{sec}) \qquad (30)$$

$$A_{12_n} = K_1 Y_{12_n} + K_2 Y_{12_{n-1}} \qquad (8/\text{sec}) \qquad (31)$$

$$A_{22_n} = A_{12_n}\left(\frac{A_{11_{n-1}}}{A_{21_{n-1}}}\right) \qquad (8/\text{sec}) \qquad (32)$$

LOX to GH_2 Ratio Trim

Trim compensation for oxidizer-fuel ratio trim is computed from the procedure

$$W'_{F_n} = K_1 W_{F_n} + K_2 W_{F_{n-1}} + K_3 W_{F_{n-2}} \qquad (8/\text{sec}) \qquad (33)$$

$$Y_{13_n} = K_4 W'_{F_n} + K_5 W'_{F_{n-1}} + K_6 Y_{13_{n-1}} \qquad (8/\text{sec}) \qquad (34)$$

$$X_{32} = \left[Y_{13_n} + \frac{W'_L}{\beta'_{NOM}} \right]_{\pm L_3} \qquad (8/\text{sec}) \qquad (35)$$

X_{32} is limited for both positive and negative values.

Main LOX Pump Protection

A control term to protect the main LOX pump is computed from

$$Y_{14} = f_7(N_{PL}, W_L') \quad \text{(24 points) (64/sec)} \tag{36}$$

$$Y_{15} = f_8(T_L) \quad \text{(6 points) (4/sec)} \tag{37}$$

where T_L is the main LOX pump inlet temperature

$$Y_{16} = Y_{14} + Y_{15} - P_{LL} \quad \text{(64/sec)} \tag{38}$$

where P_{LL} is the low speed inducer pressure

$$Y_{17_n} = K_1 Y_{16_n} + K_2 Y_{16_{n-1}} + K_3 Y_{17_{n-1}} \quad \text{(64/sec)} \tag{39}$$

$$A_{13} = \begin{cases} Y_{17} & \text{if } Y_{17} \geqslant 0 \\ 0 & \text{if } Y_{17} < 0 \end{cases} \quad \text{(64/sec)} \tag{40}$$

Main Fuel Pump Protection

A control term to protect the main fuel pump is computed from

$$Y_{18} = f_9(W_F, N_P) \quad \text{(32 points) (64/sec)} \tag{41}$$

$$Y_{19} = f_{10}(T_F) \quad \text{(6 points) (4/sec)} \tag{42}$$

where T_F is the main pump inlet fuel temperature.

$$Y_{20} = Y_{18} + Y_{19} - P_F \quad \text{(64/sec)} \tag{43}$$

where P_F is the main fuel inlet pressure

$$Y_{21} = f_{11}(\alpha_{1_{n-1}}, \beta_{NOM}) \quad \text{(24 points) (64/sec)} \tag{44}$$

$$Y_{22} = T_X - Y_{21} \quad \text{(64/sec)} \tag{45}$$

where T_x is the heat exchanger exit temperature

$$Y_{23} = \begin{cases} Y_{MAX} & \text{if } Y_{22} > Y_{MAX} \\ Y_{22} & \text{if } 0 \leq Y_{22} \leq Y_{MAX} \\ 0 & \text{if } Y_{22} < 0 \end{cases} \qquad (64/\text{sec}) \qquad (46)$$

$$Y_{24} = \begin{cases} Y_{23} & \text{if } Y_{23} \geqslant Y_{20} \\ Y_{20} & \text{if } Y_{20} > Y_{23} \end{cases} \qquad (64/\text{sec}) \qquad (47)$$

$$\alpha_{12_n} = K_1 Y_{24_n} + K_2 Y_{24_{n-1}} + K_3 \alpha_{12_{n-1}} + K_4 \alpha_{12_{n-2}} \qquad (64/\text{sec}) \qquad (48)$$

Control Output Computations

The above control computations are summed through a non linear process to construct the primary control outputs to the engines. This process is achieved by computing

$\alpha_1 = \alpha_{NOM} + \alpha_{12} + \alpha_{13}$		(64/sec)	(49)
$A_{11} = f_{12}(\alpha_1, \beta_{NOM})$	(24 points)	(64/sec)	(50)
$A_{21} = f_{13}(\alpha_{NOM}, \beta_{NOM})$	(24 points)	(2/sec)	(51)
$A_3 = f_{14}(\alpha_{NOM}, \beta'_{NOM})$	(24 points)	(2/sec)	(52)
$A_4 = f_{15}(\alpha_{NOM}, \beta_{NOM})$	(24 points)	(2/sec)	(53)
$A_1 = A_{11} + A_{12} + A_{13}$		(64/sec)	(54)
$A_2 = A_{21} + A_{22}$		(8/sec)	(55)
$X_1 = f_{16}(A_1)$	(6 points)	(64/sec)	(56)
$X_2 = f_{17}(A_2)$	(8 points)	(8/sec)	(57)
$X_{31} = f_{18}(A_3)$	(8 points)	(2/sec)	(58)
$X_{41} = f_{19}(A_4)$	(8 points)	(2/sec)	(59)
$X_3 = X_{31} + X_{32}$		(8/sec)	(60)
$X_4 = X_{41} + X_{42} + X_{43}$		(64/sec)	(61)

The above control equations are solved by four computer programs with 2, 4, 8 and 64 per second iteration rates. In addition these four programs must perform monitoring and program initilization functions. Figure 5 - is a flow diagram of the Main Propulsion System Two per second iteration rate program (MPST). On the initial entry to the program the variables for all the main propulsion system programs are initialized and the other main propulsion system programs scheduled. On a normal entry the main chamber skin temperature and nozzle coolant temperature is tested with error messages and error procedures performed if limits are exceeded. All control equations requiring a 2 per second iteration rate are then solved.

Figure 5-15 is a flow diagram of the Main Propulsion System Four per second iteration rate program (MPSF). This program solves those equations which are indicated as requiring a 4 per second iteration rate.

Figure 5-16 is a flow diagram of the Main Propulsion System Eight per second iteration rate program. This program solves those equations which are indicated as requiring an 8 per second iteration rate.

Figure 5-17 is a flow diagram of the Main Propulsion System Sixty-four per second iteration rate program. This program solves those equations which are indicated as requiring a 64 per second iteration rate.

During boost the DMS must control the propellant management system. The performance of this task consists of the continuous monitoring of propellant system valve positions, pressures, temperatures and tank levels and of controlling various time sequenced propellant system configuration changes.

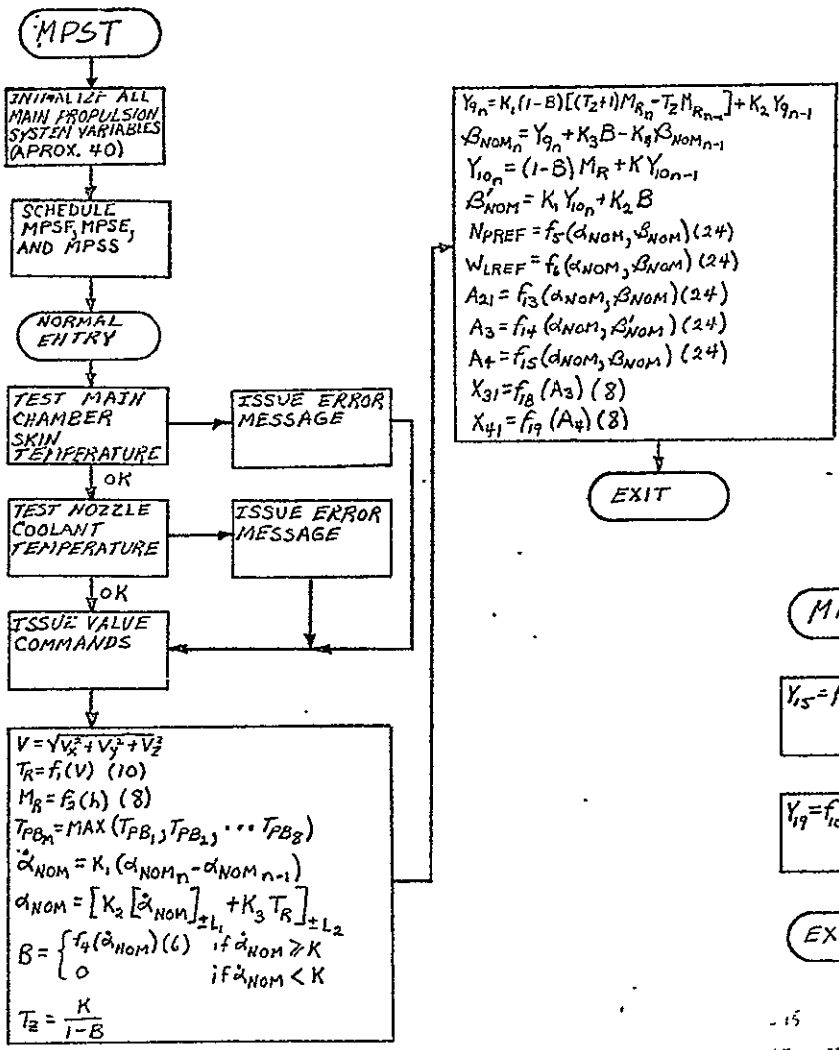


Figure 5-14 Main Propulsion System (2/sec)(MPST)

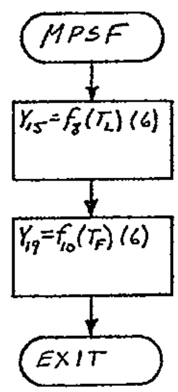


Figure 5-15 Main Propulsion System (4/sec) (MPSF)

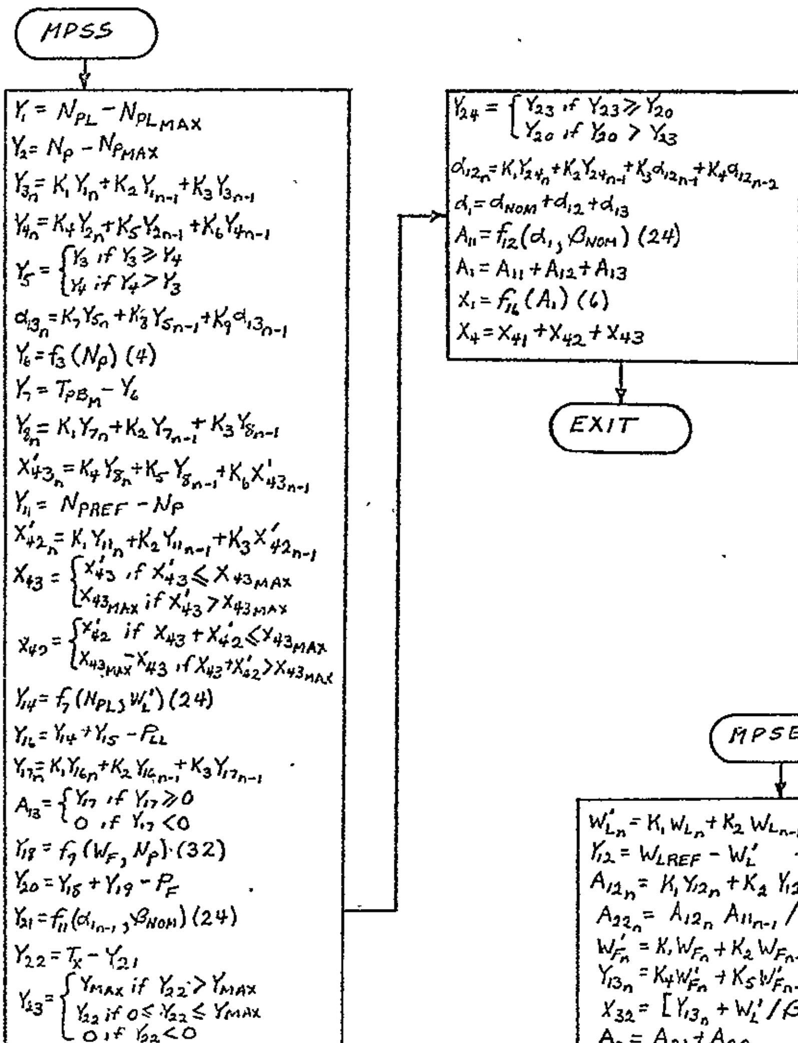


Figure 5-17 Main Propulsion System 64/sec Program (MPSS)

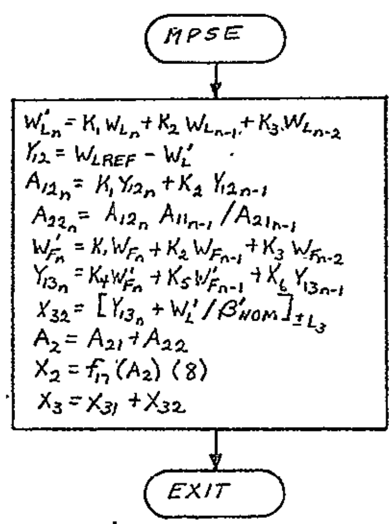


Figure 5-16 Main Propulsion System (8/sec) Program MPSE

The major propellant system configuration sequences which must be performed during the mission are:

prepare for filling

prepare for prelaunch standby

ignition sequence

boost

thrust termination

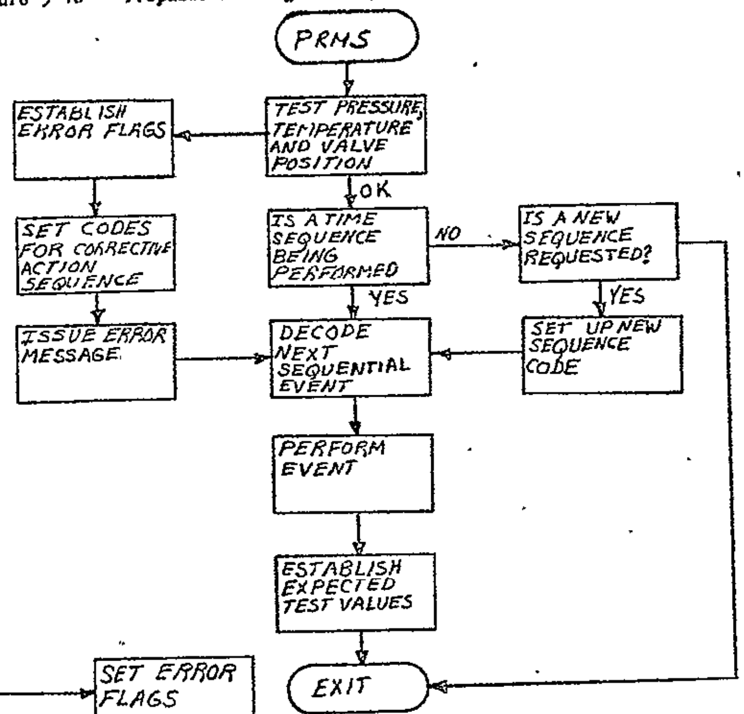
Time sequenced changes are also required in the event of system failures and for draining the tanks after landing or for inflight fuel dump. Figure 5-18 is a flow diagram of the Propulsion Management System (PRMS). Upon entry to the program all propulsion system inputs are tested which includes 68 valve positions, 20 level sensor outputs, 20 pressure valves, 10 pressure switch positions and 64 temperatures. These are tested against an expected value matrix. If an error is detected error flags are established which are used to determine the desired corrective action sequence and error messages to alert the crew are issued. If no errors are encountered a test is performed to determine if any time sequence operation is being performed. Time sequence operations are stored in the memory in a compacted and encoded form. These are decoded and the next required event performed if requested. The test matrix is modified as sequence events are performed. If a sequence is not being performed a test to determine if any new sequences are requested and the new sequence codes set up if a request exists.

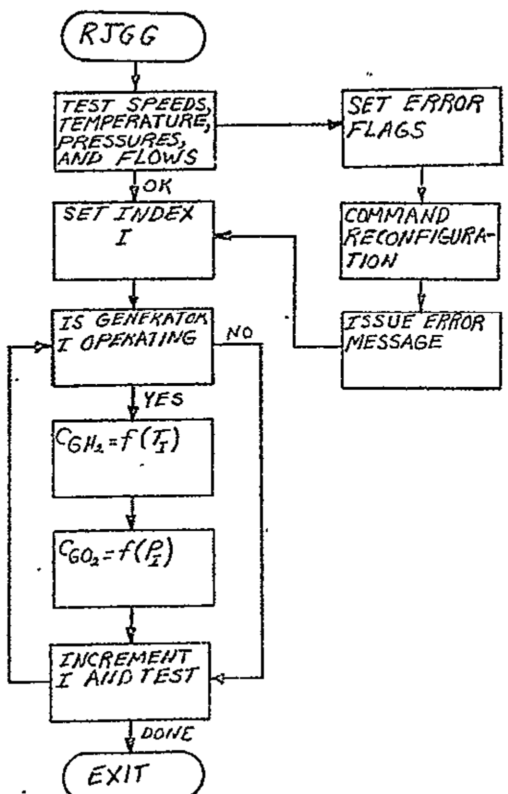
5.2.2 Reaction Jet Propulsion System

The reaction jet propulsion systems consists of three redundant gas generator systems which drive turbines supplying fuel and oxidizer to 16 thrust chambers. The DMS controls the operation of each gas generator system by controlling 20 valves and two ignitor voltages and monitoring 10 speeds, temperatures and pressures. Also included in the reaction jet propulsion system is the control and monitoring of fuel and oxidizer to the gas generator and monitoring of the thrust chamber temperatures and pressures. Control of fuel and oxidizer to the thrust chambers is performed by the coast flight control program. The GH_2 and GO_2 gas generator control valves are modulated controlling the pressure by the flow of oxidizer and the temperature by the flow of GH_2. A fuel rich mixture is maintained to prevent oxidation of generator equipment surfaces. Figure 5-19 is a flow diagram of the Reaction Jet Propulsion System Gas Generator Control Program (RJGG). The program first tests the speeds, temperatures, pressures and flows of the system against a matrix of desired values. If an error exists error flags are set and the system reconfigured if required. The desired value matrix is changed with any reconfiguration and error messages are issued. The fuel and oxidizer commands are then computed for each operating generator.

A thrust chamber monitoring program determines if each thrust chamber is capable of proper operation and sets control flags for the coast flight control program. Figure 5-20 is a flow diagram of the Thrust Chamber Monitoring Program (TCMO).

Figure 5-18 Propulsion Management System (PRMS)





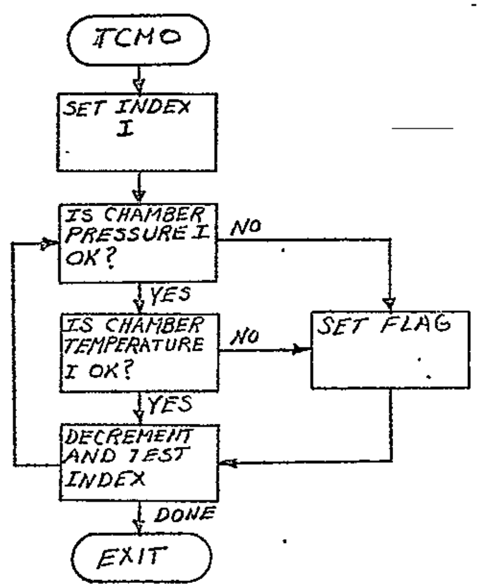


Figure 5-20 Thrust Chamber Monitor Program (TCMO)

Figure 5-19 Reaction Jet Propulsion System Gas Generator Control (RJGG)

5.2.3 Cruise Engine System

The flight control autothrottle equation requests a thrust command from the engines. This thrust command in combination with engine speed and temperature and inlet pressure and temperature is used through a non-linear table look-up process to determine the commands to the fuel and air control valves. It is assumed that six 2 dimensional table look-up functions are sufficient to perform this function. Monitoring functions must be performed in addition to the control function. A total of 42 analog inputs must be monitored for the six engines (i.e., 6 per engine plus 6 fuel flow measurements). The name of the program to perform this function is the Cruise Engine Control and Monitoring Program (CECM).

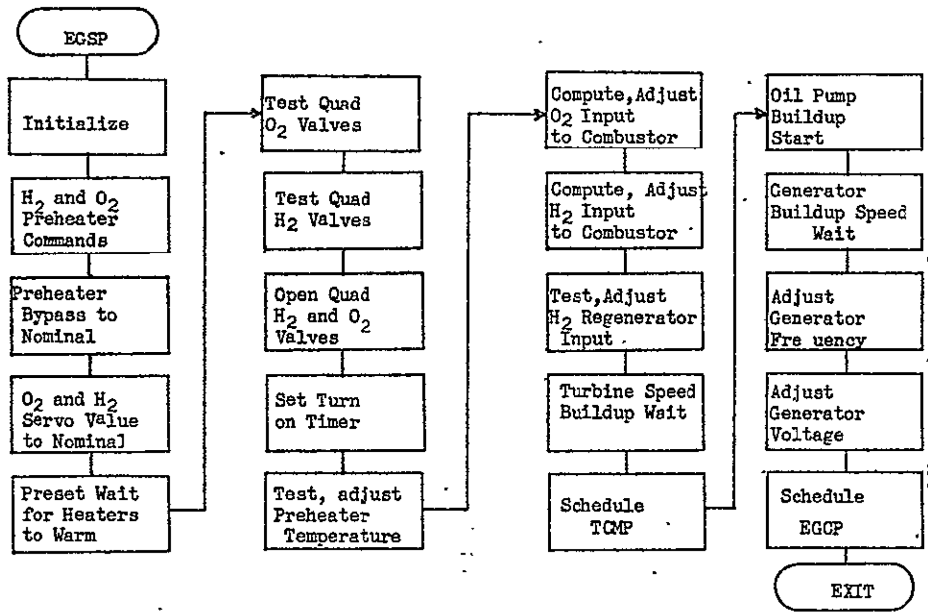
5.3 Electrical Power Generation and Distribution

The DMS must be capable of starting, controlling, testing and monitoring the electrical power generation. In the distribution system the DMS must control the supply route from the generators to the essential ac buses, the non essential ac buses, the converters, the essential dc buses and the non essential dc buses. The following programs are those required for this task.

5.3.1 Electrical Generator Start Program (EGSP) (Figure 5-21)

Upon initial entry to this program a flag is set to control the computational path through the program with multiple exists to and return from the executive. This program is processed at a 4 per second iteration rate. At entry to the program all valves will be closed and heaters turned off. Initially the preheaters will be commanded on and starting value commands issued to the regeneration bypass valve, the O_2 servo valve and the H_2 servo valve. A timer is then initialized and the preheater temperature monitored. If the preheater temperature does not arrive at a predetermined value before the timer runs out the generator is shut down and a heater failure message issued. After achieving the desired temperature the quad H_2 and O_2 valves are tested. Figure 5-22 shows the arrangement of a quad valve set. In order to test a quad valve each valve must be tested to determine that it can be both opened and closed. This is accomplished by commanding a sequence of opening and closing valves and testing for fluid flow. Figure 5-23 shows the command sequence and the desired flow. Flow is tested by measuring the preheater pressure. If the testing detects a faulty valve an error message will be issued, all valves closed and the generator shut down. If both H_2 and O_2 valve quads pass the test all valves in both quads will be commanded open and a timer started which controls the turn on sequence of the turbine and the generator. The H_2 and O_2 preheater temperature is then controlled by

Figure 5-21 Electrical Generation Start Program (EGSP)



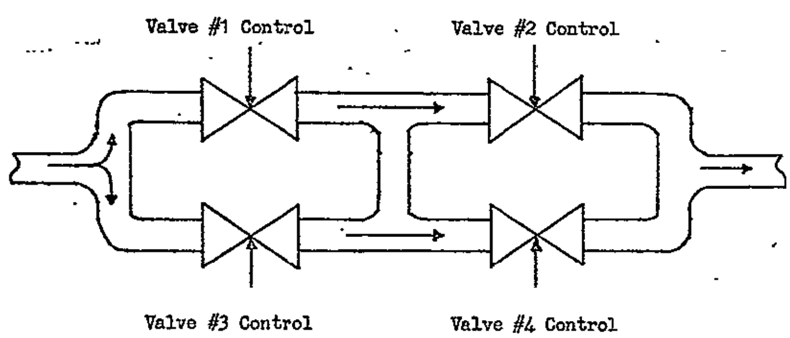


Figure 5-22 Quad Valve Arrangement

Valve	1	Open	Close	Close	Close	Open	Open	Close	Close
	2	Close	Open	Close	Close	Open	Close	Open	Close
Number	3	Close	Close	Open	Close	Close	Close	Open	Open
	4	Close	Close	Close	Open	Close	Open	Close	Open
Flow		No	No	No	No	Yes	Yes	Yes	Yes

Figure 5-23 Quad Valve Test Sequence

measuring the temperature of each heater and turning off the heater if the temperature is above a narrow desired range, and turning the heater on if it is below the desired range. Heater control is adjusted at each return from the executive. In addition to applying heater control the perheater O_2 and H_2 temperature is tested for being outside of a wider range bounding the narrow control ranges with an error message issued and the turbine shut down if the preheater temperature gets out of the wider monitoring range. The total hydrogen supply to the preheater is then controlled by adjusting the Regenerator Bypass valve. The valve position is computed from the value of preheater H_2 pressure. The amount of O_2 to the combustor is then adjusted by commanding the O_2 servo valve. The oxygen supplied to the combustor is based upon turbine speed. Figure 5-24 is a block diagram of the computations required to control the O_2 servo valve. A desired speed (Sd) is generated as a function of time. Speed error between desired and actual speed is computed. The rate of change of actual speed is computer and a valve position error generated as a proportional sum of speed error and actual speed rate. A gain (g) is computed as a polynomial function of actual speed and applied to the valve position error which is then digitaly filtered to form the actual valve command.

The command for the H_2 servo valve must then be computed. The combustor must be supplied with sufficient hydrogen to guarantee complete use of the oxygen in the combustion process. First the quantity of hydrogen required to meet this criteria is computed as a function of the oxygen servo valve command and the combustor oxygen input pressure and temperature. It is assumed that this is done as a two dimensional polynomial curve fit which can mathematicallly be expressed in the form:

$$B_i = A_{4i}(P/T)^4 + A_{3i}(P/T)^3 + A_{2i}(P/T)^2 + A_{1i}(P/T) + A_{0i} \; ; \; i = 0,1,2,3,4$$

$$H_R = B_4 V^4 + B_3 V^3 + B_2 V^2 + B_1 V + B_0$$

where P is the combustor oxygen input pressure, T the combustor oxygen input temperature and V the oxygen servo valve position. The A's are stored constants. H_R is the computed required hydrogen. This value of H_R represents the minimum hydrogen that can be supplied in units of servo valve position. To this is added enough additional hydrogen to achieve a desired combustor output temperature. The computation of the H_2 servo valve position is based upon the solution of the following difference equation:

$$V_n = V_{n-1} + H_{R_n} - H_{R_{n-1}} + K_1(T_d - T_n) - K_2(T_n - T_{n-1})$$

where T_d is the desired temperature, T the measured temperature and the subscripts n and n-1 are standard difference equation notation for present and past parameter values. The H_2 servo valve command is then set equal to V unless V is smaller than H_R in which case it is set equal to H_R.

The process of adjusting preheater temperature, and the O_2 and H_2 servo valve positions is continued until the turbine speed is built up to the desired level at which time the turbine control and monitoring program is scheduled. If the turbine does not reach the desired speed within a preset time or if combustor pressure and temperature profiles exceed limit values an error message will be generated and the generation system shut down.

After arriving at the desired turbine speed the generator will slowly be added as a load to the turbine. This is done by making the generator speed follow a desired speed function computed as a function of time using a polynomial curve fit of the form

$$S_d = A_4 t^4 + A_3 t^3 \pm A_2 t^2 + A_1 t + A_0$$

Oil pump pressure is used to control the generator speed. A command magnitude to the oil pump valve is computed from the difference equation

$$V_n = V_{n-1} + K_1(S_d - S_n) + K_2 (S_n - S_{n-1})$$

where V is the valve command and S the measured generator speed. During the speed buildup the oil temperature and speed profile are tested against limits and the generation system shut down if a malfunction is indicated with appropriate error messages generated. After achieving the desired generator speed as measured by the generator tachometer more accurate speed control is initiated by using the generator frequency output as a speed measurement. The generator output voltage is then adjusted by commanding the generator field current to a value determined from the difference equation

$$I_n = I_{n-1} + K_1(V_d - V_n) + K_2 (V_n - V_{n-1})$$

where I is the field current, V_d the desired generator output voltage and V the measured generator voltage. If voltage or frequency control within prescribed limits can not be reached within a prescribed time the generation system is shut down and an error message issued. Upon achieving voltage and frequency control the generator control program is scheduled and this program descheduled.

5.3.2 Turbine Control and Monitoring Program (TCMP)

Figure 5-25 is a block diagram of the turbine control and monitoring program. This program is scheduled at a 4/sec iteration rate. All program functions are performed with each entry from the executive. First the preheater O_2 and H_2 temperature and pressure are tested against extreme maximum and minimum bounds.

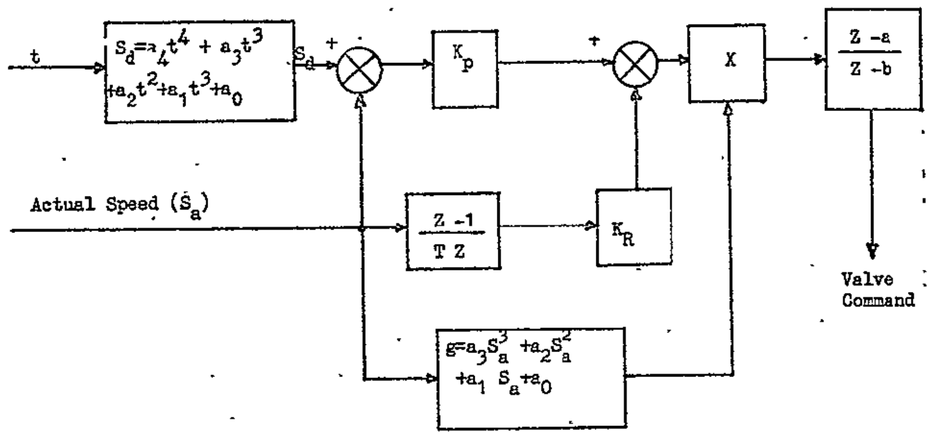


Figure 5-24 Generation of O_2 Servo Valve Command

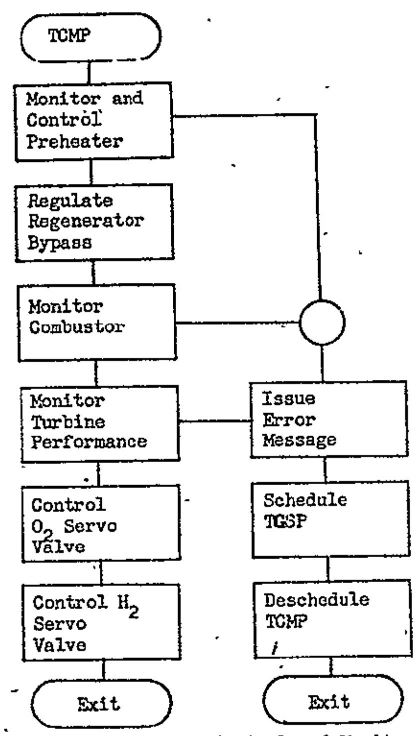


Figure 5-25 Turbine Control and Monitoring Program (TCMP)

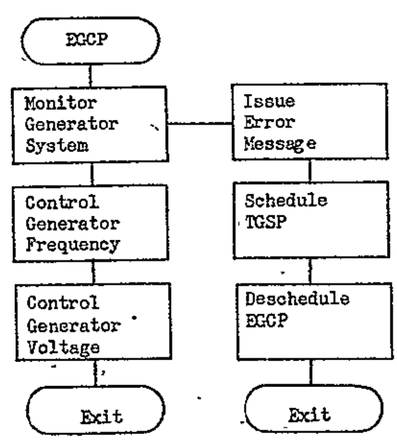


Figure 5-26 Generator Control Program (EGCP)

If any bounds are exceeded the turbine/generator shutdown program is scheduled, an error message issued, and this program descheduled. The O_2 and H_2 preheater temperatures are tested against a narrow temperature range. If either temperature is above its range the associate O_2 or H_2 preheater pressure is then used to determine the regenerator bypass value position according to the formula:

$$V_n = V_{n-1} + X_1 (P_d - P_n) + X_2 (P_n - P_{n-1})$$

where P is the measured preheater H_2 pressure, P_d the desired pressure, V the value command and n the standard differenc equation subscript.

The combustor inlet and outlet temperatures and inlet pressures are then tested against extreme limits with an error message issued, the turbine/generator shutdownprogram scheduled and this program descheduled if any limit is violated. Turbine speed and exhaust temperature and turbine vibration is then monitored against extreme limits with the error message, shutdown scheduling and this program descheduling sequence followed if any limit is violated.

The O_2 servo valve position is computed as a function of turbine speed and the load on the turbine using the equations:

$$D = (E_1 + E_2 + E_3)(I_1 + I_2 + I_3)/9$$

$$V_n = V_{n-1} + K_1 (S_d - S_n) + K_2 (S_n - S_{n-1}) + K_3 (P_d - P_n) + K_4 (D_n - D_{n-1})$$

D is a measure of the electrical load on the generator computed as the average voltage on each phase output times the average current on each phase output.

It is assumed that the power factor for each generator will lie within narrow bounds (most of the electrical power will be supplied from the DC buses) and can be assumed a constant for all loads. The O_2 servo valve position V_n is determined from the sum of its old position V_{n-1} plus the desired position change. The position change is computed from the sum of 4 terms which are the error between the desired and actual turbine speed $(S_d - S_n)$, the time rate of change of turbine speed $(S_n - S_{n-1})$, the error between the desired and actual generator oil pumps control valve position $(P_d - P_n)$ and the rate of change in electrical load on the generator $(D_n - D_{n-1})$.

The H_2 servo valve position is computed as a function of the O_2 servo valve position and the combustor temperature by the equations

$$H_R = B_2 V^2 - B_1 V + B_0$$

$$H_n = H_{n-1} + H_R - H_{R_{n-1}} + K_1 (T_d - T_n) + K_2 (T_n - T_{n-1})$$

where H_R is the H_2 servo valve position required to assure complete oxygen consumption computed from the O_2 servo valve position V. T is the combustor outlet temperature and T_d the desired combustor outlet temperature. The H_2 servo valve will be commanded to a position H unless H is less than H_R in which case the valve will be commanded to a position H_R.

5.3.3 Generator Control Program (EGCP)

A flow diagram of the Generator Control program is shown in figure 5-26. This program is run at an iteration rate of 4 per second. Monitoring, frequency control, and voltage control are performed at each call from the executive.

Monitoring is performed by comparing oil temperature, oil pressure, oil quantity, generator speed, generator voltage, and generator current against extreme limits. An error message is issued, the turbine/generator shutdown program scheduled and this program descheduled if any of the limit values are exceeded.

Generation speed is controlled by controlling the oil pump pressure. Generator speed is measured accurately be measuring the frequency of the generated voltage. The equation used to determine the oil pressure control valve position is

$$P_n = P_{n-1} + K_1(S_d - S_n) + K_2(S_n - S_{n-1}) + K_3(V_n - V_{n-1}) + K_4(D_n - D_{n-1}) + Q$$

where $S_d - S_n$ is the error between desired and measured generator speed, $S_n - S_{n-1}$ is the rate of change of generator speed, $V_n - V_{n-1}$ is the change in turbine speed since the last iterative pass, $D_n - D_{n-1}$ is the change in the electrical load on the generator, and Q is a synchronizing command generated by the power distribution program. D is obtained from the turbine control and monitoring program.

The generator output voltage is controlled by adjusting the generator field current using the equation

$$I_n = I_{n-1} + K_1 (V_d - V_n) + K_2 (V_n - V_{n-1})$$

where V_d is the desired generator voltage, V the average measured generator voltage, and I the computed field current.

5.3.4 Turbine/Generator Shutdown Program (TGSP)

Figure 5-27 is a flow diagram of the turbine/generator shutdown program. Upon entry to the program the generator field current is set to zero, the oil pump is shut down, the hydrogen supply quad valves commanded close, and the O_2 servo valve reduced to a minimum setting which will still support combustion. This is done to burn off as much as possible the excess hydrogen in the system. As soon as the H_2 preheater pressure drops below a preset value the O_2 supply valve is closed. As soon as the H_2 and O_2 preheater pressures drop below a lower preset value the H_2 and O_2 servo valves are closed and the preheater heaters turned off. All turbine and generator functions are monitored and shutdown commands reissued if monitoring indicates a failure to shutdown. Monitoring continues until the turbine speed falls below a preset value at which time a message is issued indicating successful shutdown and this program descheduled. This program is run at an iteration rate of 4 per second.

5.3.5 Electrical Distribution Monitoring and Control (EDMC)

Figure 5-28 is a flow diagram of the electrical distribution monitoring and control program. This program is run at a 4 per second iteration rate. This program includes data tables indicating nominal expected loads for each subsystem, which subsystems are being supplied with power, a priority list indicating the order in which systems can be dropped in case of a shortage in electrical power, and the present status of the distribution system. The program first tests the loads on each supply bus by the procedure shown in Figure 5-29. This testing is conducted on each non essential dc bus, essential dc bus, non essential ac bus and essential ac bus. The converters are then tested by comparing their output voltage and their temperature against limit

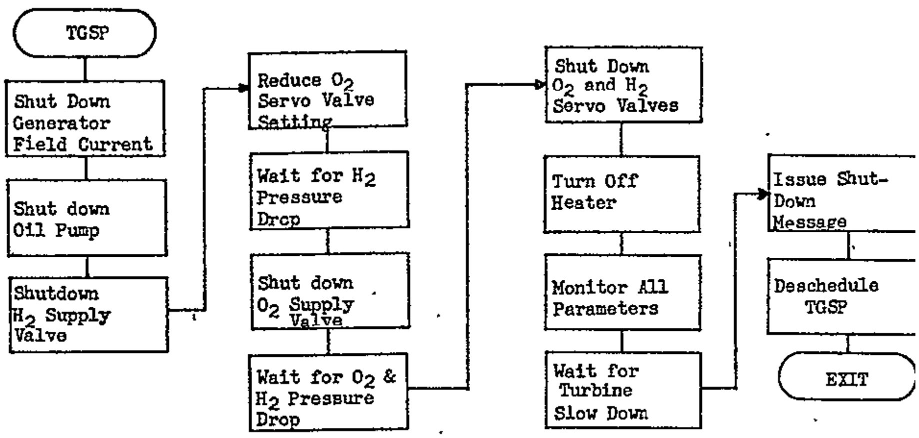


Figure 5-27 Turbine/Generator Shutdown Brogram

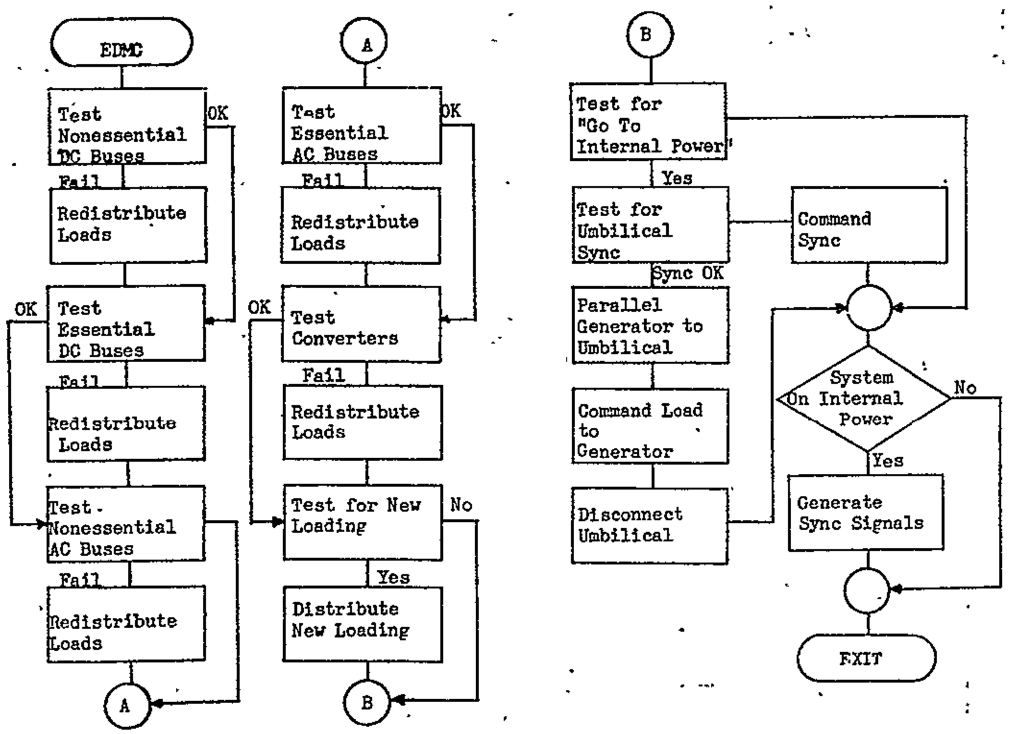


Figure 5-28 Electrical Distribution Monitoring and Control (EDMC)

values. If the limits are exceeded the loads in the converter are redistributed by transferring the total bus load to a different converter if any converter is capable of handling the increased load, otherwise the total dc load is redistributed.

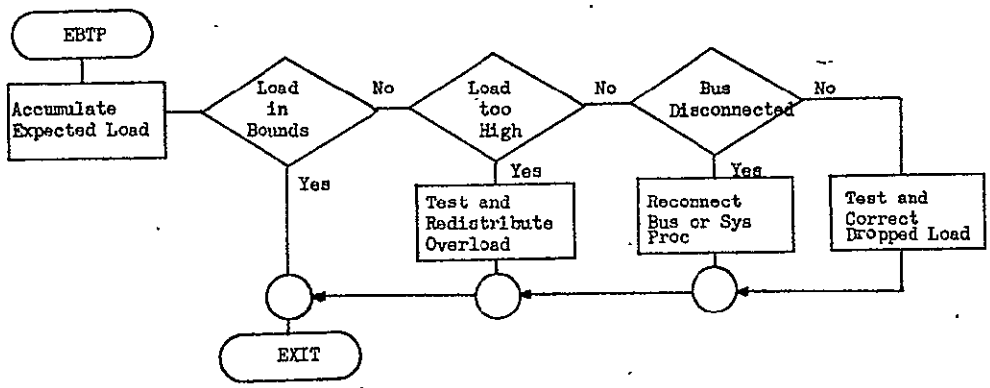
Distribution of the load to the various power busses is controlled by tables stored in the DMS. There is a set of 4 tables, one for each power bus, for each major flight mission phase, i.e., boost, coast, reentry, cruise, and landing. The tables contain a data field for each subsystem requiring electrical power. The data fields are arranged in the order of their priority in being connected. Each data field contains the following information:

1. The current required by the system.
2. Codes which indicate what discretes must be issued in order to connect and disconnect the system, determine how the voltage and/or current to the system can be monitored if possible, and pointers to error message data to be displayed if the system cannot be supplied with power.
3. Status codes which indicate whether or not the system is connected, whether or not connection is desired, which busses the system can be connected to, and why the system is not connected when connection is desired, e. g., power bus loads are too high to allow for connection, connection has been attempted but cannot be connected, or when connected system appears to be shorted.

Figure 5-30 is a flow diagram of the program required to determine how the systems will be connected to the power busses based upon the table data. The data in the table is listed in the order of system priority. At the start of executing the flow diagram of figure 5-30 a pointer is set to the top of the table associated with the data bus and mission phase under consideration. Each system is tested to determine if it should be connected and if it is connected. If it is connected and should not be the system is disconnected; if it should be connected and is not, its error codes are checked to determine if an attempt to previously connect the system has failed or if the system has previously indicated a short circuit. If the error codes do not show that the system cannot be connected the unused power capabilities of the desired bus are compared with the nominal load of the system to be connected. If capability exists for connecting the system to the power bus the system is connected. If not the code word is interrogated to determine if the system can be connected to another power bus of the same type and if so, whether the other bus has power capabilities of handling the system in which case it is connected to the other bus. If it can not be connected to another bus an investigation is made to determine if enough power capability can be made available by disconnecting lower priority systems. If this can be done the lower priority systems are disconnected and the higher priority system connected. This total test sequence is performed for all systems in each table associated with the mission phase being performed.

Referring to figure 5-28 after distributing the load a test is made to determine if the vehicle is being supplied power from the umbilical system and a command to go to internal power has been issued. If this is the case a test for each generator being in sync with the umbilical power is made.

Figure 5-29 Electrical Bus Testing Program (EBTP)



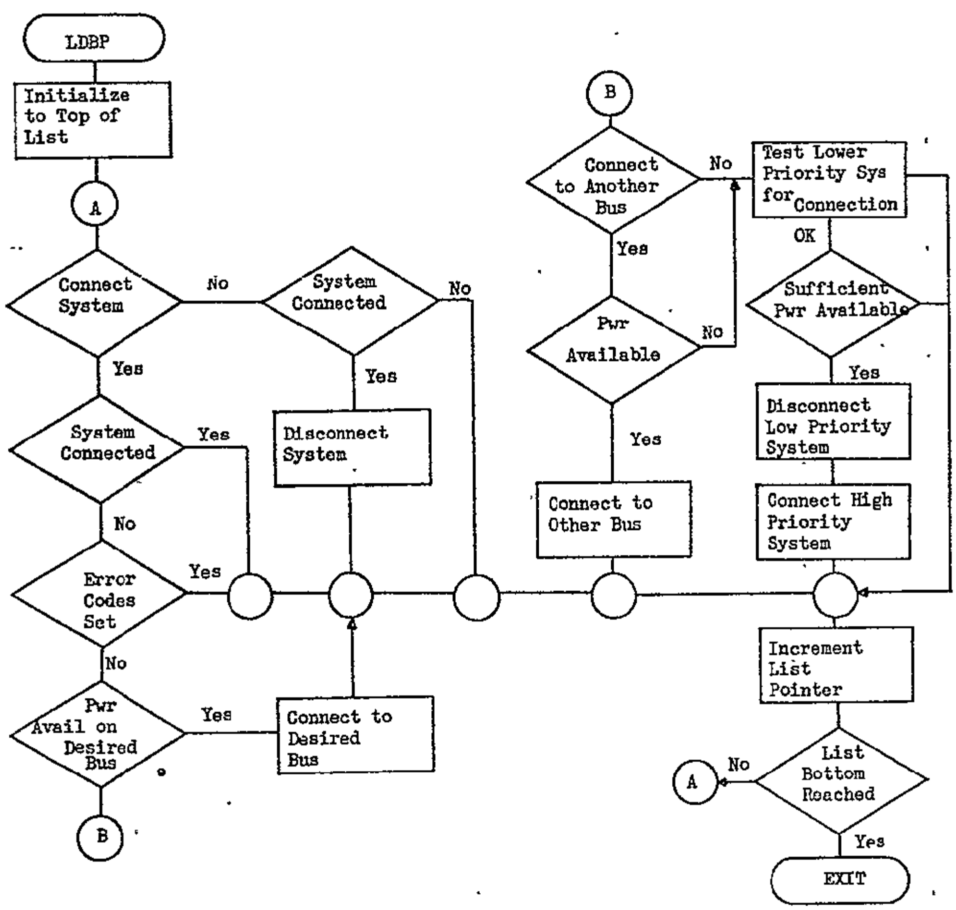


Figure 5-30 Load Distribution Program (LDBP)

If the generator is not in synchronism with the umbilical power a command to either increase or decrease the generator speed is issued. The command is generated from accurate generator and umbilical frequency measurements and constructed so as to command the generator into sync with the umbilical. When sync is achieved the generator and umbilical are paralleled. Commands to increase the generator speed are then issued. As long as the generator and umbilical power are parallel a command to increase the generator speed will not increase the generator speed but will cause the generator to assume a larger portion of the electrical load. As soon as the generator has assumed most of the electrical load the umbilical power is disconnected from the electrical distribution system. This procedure eliminates electrical transients in switching from umbilical to generator power. After switching to internal power on all three generators sync signals are generated to keep all three generator outputs in synchronism.

Referring to figure 5-29, the procedure used in testing each electrical bus system is to first accumulate the expected load on the bus and comparing this value with the actual load. Based upon this testing one of four decision paths are taken; the actual and expected loads are equal within bounds indicating a properly operating system. The actual load is much higher than the expected load indicating an overload condition caused by a malfunctioning system, the actual load is much less than the expected load but not zero indicating one or more subsystems have become disconnected or are malfunctioning, or the actual load is zero and the expected load is non zero indicating that the entire bus has become disconnected.

Figure 5-31 shows the procedure used if testing indicates an overload on an electrical bus. It is possible that the overload is only temporary being caused by a subsystem transient demand, therefore a time delay is incorporated

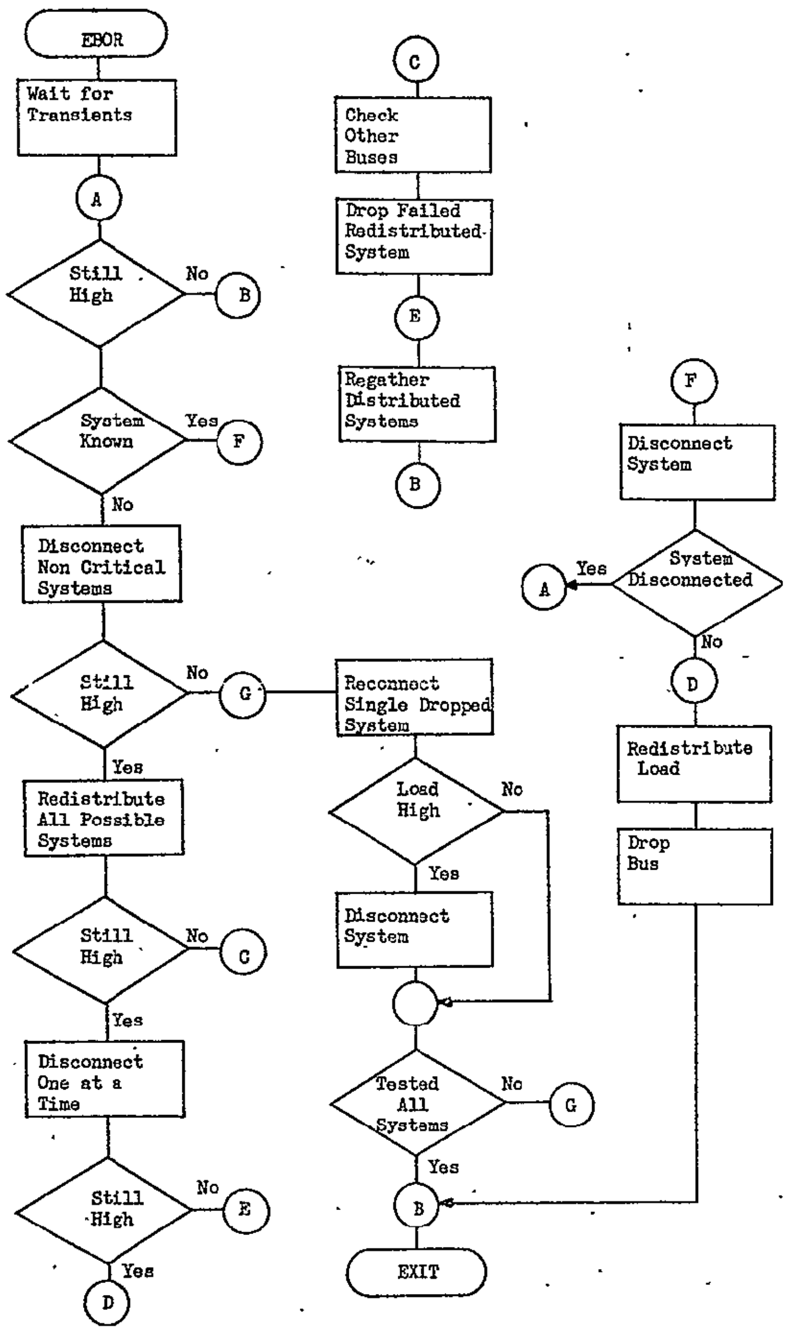


Figure 5-31 Electrical Bus Overload and Redistribution (EBOR)

to allow for any transient demands to disappear. The time delay allowed is dependent upon the overload magnitude. Upon determining that the overload is still present after the time delay has elapsed tests are made to determine if the system causing the overload can be found directly. The capability will exist of directly monitoring the current to some subsystems but not to all subsystems. If an overload is determined to exist in a particular subsystem as determined by measuring the current to that subsystem a command to disconnect the indicated subsystem is issued. A test is then made to insure that the system was disconnected; if the disconnect did not occur all systems wherever possible are redistributed to another bus and the total bus disconntected, and if the disconnect did occur the bus is rechecked to determine if any further overloads exist. If the overlaod exists among those systems that do not have the capability of having their individual currents measured a move complex procedure to isolate the overload is used. The systems are divided into two types, critical and non-critical. The non-critical systems are disconnected and a test for overlaod made. If the overload has vanished the non-critical systems are reconnected one at a time with a check for overload made after each system is reconnected and thos systems dropped which cause the reappearance of an overload when reconnected. If disconnecting the non-critical systems does not eliminate the overload then non-critical systems are reconnected and all critical systems having the capability of being connected to another bus are connected to another bus system. If this transfers the overload to another bus then the systems transferred to that bus are transferred back one at a time until the overloading system or systems are discovered and eliminated at which time all other systems are returned to their original bus. If the overload remained on the original bus after distributing all possible systems the remaining critical systems are disconnected one at a time and a test made to determine if the overload is eliminated. As each system is disconnected it is left disconnected if an overload in the system is indicated, otherwise it is reconnected. If after testing all systems and determining that no individual system is causing an overload yet an overload on the bus still exists,its load is redistributed and the bus disconnected.

Figure 5-32 shows the program flow required if testing indicates that the bus is completely disconnected. There are two possible causes of a disconnected bus which are a failure in the circuit breaker supplying the bus or a major short circuit on the bus or subsystem supplied by the bus which has caused an automatic disconnect of the bus. The procedure used in analyzing the fault is to first disconnect all subsystems from the bus and issue a command to reconnect the bus. If the bus does not reconnect then either the circuit breaker has failed or a short on the bus itself exists. The bus is then commanded to be connected to another available source and tested to determine if the connection was accomplished. If the bus is still disconnected a short to the bus is assumed and its load is redistributed to another bus wherever possible. If the bus can be reconnected either to its original source or a secondary source its loads are then reconnected one at a time. As each load is reconnected a test is made to determine if the bus remains connected. If the bus becomes disconnected a short in the last system to be connected is suspect and that system is dropped from operational status. The bus with its previously tested loads is then reconnected and the procedure of reconnecting a single system at a time continued.

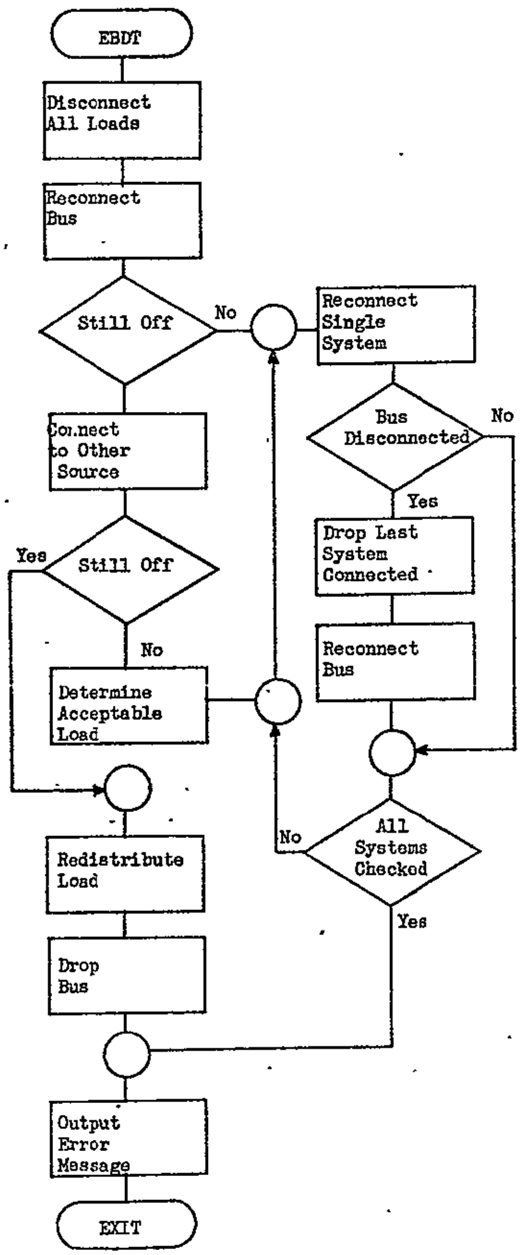


Figure 5-32 Electrical Bus Disconnect Testing and Redistribution (EBDT)

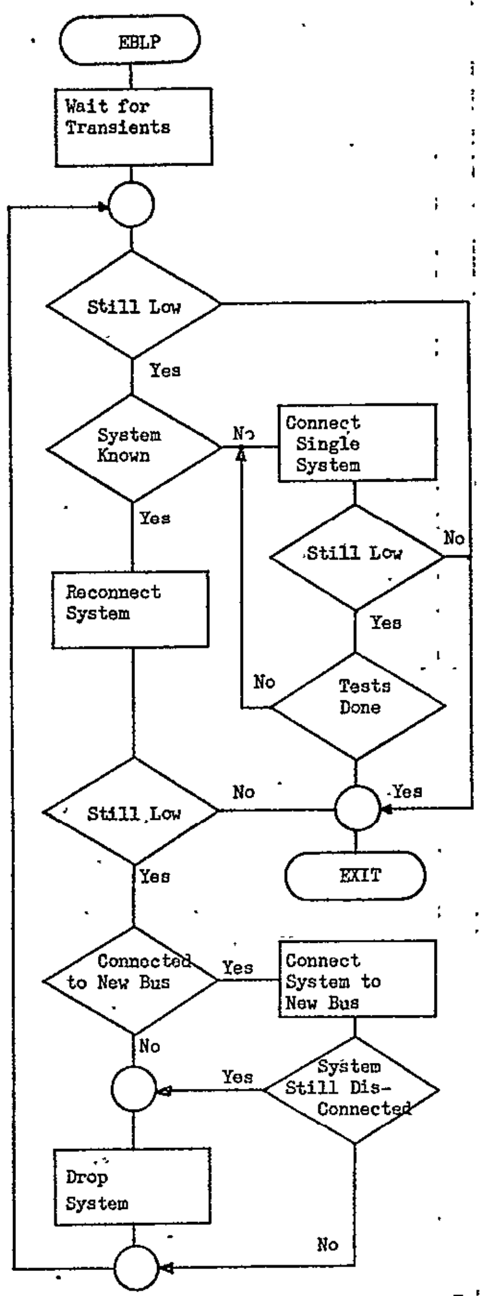


Figure 5-33 Electrical Bus Low Power Drain Test (EBLP)

Error messages are generated as each fault is analyzed. Flags are issued to various subsystem programs to indicate temporary or total disruption of operation.

Figure 5-33 is the program flow executed if a below nominal load is encountered on any power bus. First a time delay is programmed to allow for possible transients indicating low power drain to decay. The assumption made upon finding a permanent below nominal load is that some subsystem has become disconnected from the bus. This disconnect could be caused by a faulty subsystem circuit breaker or a short in a subsystem activating the circuit breaker. Those subsystems having a test point available for monitoring their supply power are first tested. If any are found to be disconnected a command to reconnect them is generated. If they remain disconnected a command to connect them to a different bus is issued for those subsystems having this capability. If the system cannot be reconnected to the original bus or a second bus the system is dropped from operational status. For those systems where primary supply power cannot be monitored a reconnect command is issued. For these systems it is impossible to determine directly if the system is reconnected unless the below nominal load on the bus disappears. If it is impossible to correct the below nominal load it can be expected that some subsystem program will obtain a fault indication and deschedule the faulting system.

5.4 Navigation and Guidance

Navigation and guidance requirements upon the DMS vary greatly with mission flight phase. During boost, navigation is based entirely upon the strapdown inertial navigation system and trajectory guidance is used. During reentry, inertial navigation is updated through use of air data inputs and energy management guidance employed. During cruise and landing, ground based navigation aids are used as a primary navigation source by continously updating the strapdown inertial navigation system with guidance limited to a route point steering system.

5.4.1 Strapdown Inertial Navigation

The sensors for the strapdown inertial navigation system are six acceleo-meters and six single degree of freedom gyros. These are mounted in a configuration such that the instrument sensitive axis are directed to the verticies of a regular dodecahedron. Figure 5-34 shows the orientation of the sensitive axis of each instrument. The angle α shown in the figure is a constant defined by

$$S = \sin\alpha = \sqrt{\frac{5-\sqrt{5}}{10}} \qquad (1)$$

$$C = \cos\alpha = \sqrt{\frac{5+\sqrt{5}}{10}} \qquad (2)$$

An acceleration in the X body axis direction is sensed by accelerometers 1,2,3 and 4 and a rotation about X by gyros 1,2,3 and 4. Before the normal strapdown equations can be solved, the instrument outputs must be trans-formed to the orthogonal body axis coordinates. Because both sets of instruments are mounted identically, the transformations from the non-orthogonal dodecahedron axes to orthogonal body axes is the same for both

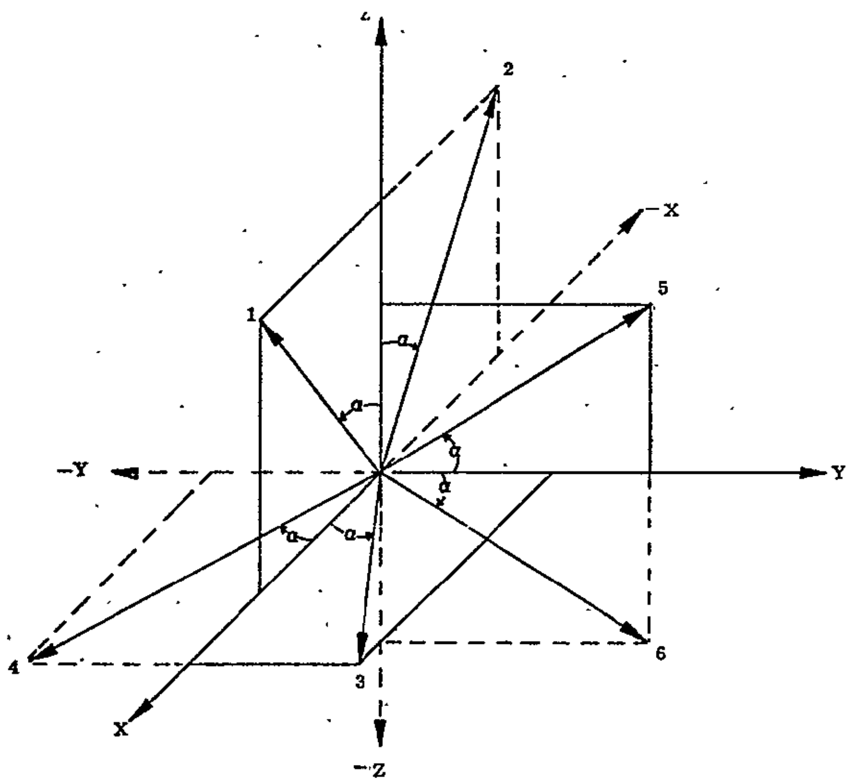


Figure 5-34 Strapdown Sensor Sensitive Axis Orientation

instrument types, i.e., accelerometers and gyros. The accelerometer (or gyro) outputs can be represented as a six vector defined as

$$\vec{I} = \begin{bmatrix} I_1 \\ I_2 \\ I_3 \\ I_4 \\ I_5 \\ I_6 \end{bmatrix} \tag{3}$$

and the transformed body axis output as a three-vector

$$\vec{O} = \begin{bmatrix} O_x \\ O_y \\ O_z \end{bmatrix} \tag{4}$$

Referring to figure 5-34 the geometric relationship between the accelerometer (or gyro) outputs and the body axis outputs is given by the matrix equation

$$\vec{I} = M\vec{O} \tag{5}$$

where

$$M = \begin{bmatrix} S & 0 & C \\ -S & 0 & C \\ C & S & 0 \\ C & -S & 0 \\ 0 & C & S \\ 0 & C & -S \end{bmatrix} \tag{6}$$

A solution for the three components of $\vec{O}$ can be obtained using the outputs of any three instruments. With more than three instruments operating a more accurate solution is achieved by taking a weighted average of the solutions obtained from each set of three instruments. The process of achieving a weighted average can be incorporated into the solution of the output vector, $\vec{O}$, by using the equation

$$\vec{O} = (M^T \delta M)^{-1} M^T \delta \vec{I} \tag{7}$$

where

$$\delta = \begin{bmatrix} \delta_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & \delta_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & \delta_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & \delta_4 & 0 & 0 \\ 0 & 0 & 0 & 0 & \delta_5 & 0 \\ 0 & 0 & 0 & 0 & 0 & \delta_6 \end{bmatrix} \tag{8}$$

and δ_i equals either 1 or 0 dependent upon whether instrument i is functioning or not respectively.

Defining the matrix

$$P = (M^T \delta M)^{-1} M^T \delta \tag{9}$$

reduces equation 7 to

$$\vec{O} = P \vec{I} \tag{10}$$

There are 42 different ways in which 3 or less instruments can fail resulting in the matrix P having 42 possible values. Figure 5-35 gives six possible values of P. In order to determine which transformation should be used at any given time a method must be mechanized which will determine failed instruments. Several methods are available for fault detection. Before a method can be selected the probability of multiple simultaneous failures must be considered, along with the methods capable of handling multiple simultaneous failures. The methods available for fault detection can be classified into two types, either trend analysis or voting. An error analysis will show that trend analysis methods do not have sufficient response or accuracy to determine instrument failures and still achieve mission success. This leaves only voting methods available for the space shuttle booster application in determining the first two failures. If instrument failures are random, multiple simultaneous failures should occur with very low probability. Simultaneous is defined to mean a second (and third) failure occurring before the detection and elimination of the first failure has been accomplished. The probability of two simultaneous random failures occurring is

$$\frac{T_d}{T_m} (P_f)^2$$

and of three simultaneous failures occurring is

$$\left(\frac{T_d}{T_m}\right)^2 \left(P_f\right)^3$$

where T_d is the time required to detect and eliminate a failed instrument, P_f the probability of a single instrument failing, and T_m the total mission time over which P_f is computed. If T_d is one second and T_m is 30 minutes then the probability of two simultaneous random failures is .00056 times the probability of two non-simultaneous failures and the probability of 3 simultaneous random failures is .00000031 times the probability of three non-simultaneous failures. With failure probabilities as small as the above analysis indicates, random simultaneous failures can be ignored. Failures on a boost vehicle are not always random but are caused by some electrical or environmental occurrence. An environmental occurrence, such as a shock force on the vehicle, which would cause failures to occur in the gyros and accelerometers is very likely to affect other major booster systems with catastrophic results. The most common occurrence which will cause simultaneous instrument failures is a transient or failure in an electrical supply source. Since there are three redundant electrical sources, two instruments of each type will be connected to each source. Thus an electrical-caused simultaneous failure will effect only two instruments. For this reason the method used to detect failures should be capable of recognizing two simultaneous failures.

The failure detection mechanization changes as failures occur, i.e., a different mechanization is used with no failures detected, with one failure detected, and with two failures detected. Figure 5-36 is a set of error equations with one error equation for each possible combination of four instruments.

trace δ = 1, 1, 1, 1, 1, 1

$$P = \begin{bmatrix} S/2 & -S/2 & C/2 & C/2 & 0 & 0 \\ 0 & 0 & S/2 & -S/2 & C/2 & C/2 \\ C/2 & C/2 & 0 & 0 & S/2 & -S/2 \end{bmatrix}$$

trace δ = 0, 1, 1, 1, 1, 1

$$P = \begin{bmatrix} 0 & (C-3S)/5 & (7C-S)/10 & (7C-S)/10 & (2C-S)/10 & (S-2C)/10 \\ 0 & 0 & S/2 & -S/2 & C/2 & C/2 \\ 0 & (C+3S)/5 & (2S+C)/10 & (-2S-C)/10 & (7S+C)/10 & (-7S-C)/10 \end{bmatrix}$$

trace δ = 1, 0, 1, 1, 1, 1

$$P = \begin{bmatrix} (3S-C)/5 & 0 & (7C-S)/10 & (7C-S)/10 & (S-2C)/10 & (2C-S)/10 \\ 0 & 0 & S/2 & -S/2 & C/2 & C/2 \\ (3S+C)/5 & 0 & (-2S-C)/10 & (-2S-C)/10 & (7S+C)/10 & (-7S-C)/10 \end{bmatrix}$$

trace δ = 1, 1, 0, 1, 1, 1

$$P = \begin{bmatrix} (7S+C)/10 & (-7S-C)\ 10 & 0 & (3C+S)/5 & (2S-C)/10 & (2S-C)/10 \\ (2C-S)/10 & (S-2C)\ 10 & 0 & (C-3S)/5 & (7C-S)/10 & (7C-S)/10 \\ C/2 & C/2 & 0 & 0 & S/2 & -S/2 \end{bmatrix}$$

trace δ = 1, 1, 1, 0, 1, 1

$$P = \begin{bmatrix} (7S+C)/10 & (-7S-C)/10 & (3C+S)/5 & 0 & (C-2S)/10 & (C-2S)/10 \\ (S-2C)/10 & (2C-S)/10 & (3S-C)/5 & 0 & (7C-S)/10 & (7C-S)/10 \\ C/2 & C/2 & 0 & 0 & S/2 & -S/2 \end{bmatrix}$$

trace δ = 1, 1, 1, 1, 0, 1

$$P = \begin{bmatrix} S/2 & -S/2 & C/2 & C/2 & 0 & -0 \\ (2S+C)/10 & (2S+C)/10 & (7S+C)/10 & (-7S-C)/10 & 0 & (3C+S)/5 \\ (7C-S)/10 & (7C-S)/10 & (2C-S)/10 & (S-2C)/10 & 0 & (C-3S)/5 \end{bmatrix}$$

Figure 5-35 Dodecahedron to Body Axis Transformations

Instruments	Error Equations
1, 2, 3, 4	$E_1 = (I_1 - I_2)C - (I_3 + I_4)S$
1, 2, 3, 5	$E_2 = (I_2 + I_3)C - (I_1 + I_5)S$
1, 2, 3, 6	$E_3 = (I_3 - I_1)C + (I_5 - I_6)S$
1, 2, 4, 5	$E_4 = (I_4 - I_1)C + (I_2 + I_5)S$
1, 2, 4, 6	$E_5 = (I_2 + I_4)C - (I_1 - I_6)S$
1, 2, 5, 6	$E_6 = (I_5 - I_6)C - (I_1 + I_2)S$
1, 3, 4, 5	$E_7 = (I_4 + I_5)C - (I_1 - I_3)S$
1, 3, 4, 6	$E_8 = (I_6 - I_3)C + (I_1 + I_4)S$
1, 3, 5, 6	$E_9 = (I_1 + I_6)C - (I_3 + I_5)S$
1, 4, 5, 6	$E_{10} = (I_5 - I_1)C + (I_4 - I_6)S$
2, 3, 4, 5	$E_{11} = (I_5 - I_3)C + (I_4 - I_2)S$
2, 3, 4, 6	$E_{12} = (I_4 + I_6)C + (I_2 - I_3)S$
2, 3, 5, 6	$E_{13} = (I_2 - I_5)C + (I_3 + I_6)S$
2, 4, 5, 6	$E_{14} = (I_2 + I_6)C + (I_4 - I_5)S$
3, 4, 5, 6	$E_{15} = (I_3 - I_4)C - (I_5 + I_6)S$

Figure 5-36 Error Equations for First Failure Detection

If all instruments are operating perfectly the values of E_1 through E_{15} will be zero. Since the instruments are not perfect but contain small bias, resolution, and scalefactor errors, the values of E_1 through E_{15} will not be zero. Since the instrument inputs are incremental changes in velocity and attitude, very large scalefactor and bias errors must exist before any E_i would have a value larger than produced by an acceptable resolution error. The summation of the resolution errors over several samples will never be larger than twice the instrument resolution while the summation of the errors caused by bias and scalefactor will increase. The summation implemented will be time weighted, increasing the influence of the most recent instrument inputs. The form of time weighted summation which will be used is

$$F_{i_n} = E_{i_n} + a\, F_{i_{n-1}} \qquad (11)$$

where a is the weighting factor, F_i is the weighted summation of E_i and the subscripts n and n-1 the standard difference equation notation. This difference equation has a Z transformation representation of

$$F_i = \frac{Z}{Z-a} E_i \qquad (12)$$

Referring to figure 5-36 each error equation is the form

$$E_i = (I_b \pm I_c)\, C \pm (I_d \pm I_e)\, S \qquad (13)$$

where b,c,d, and e take on various integer values from 1 through 6. Substituting equation 13 into 12 and noting that S and C are constants and that Z transforms are distributive yields:

$$F_i = \left(\frac{Z}{Z-a} I_b \pm \frac{Z}{Z-a} I_c\right) C \pm \left(\frac{Z}{Z-a} I_d \pm \frac{Z}{Z-a} I_c\right) S \qquad (14)$$

Defining J_i such that

$$J_i = \sum_{n} \cdot I_i \tag{15}$$

and substituting into equation 14 yields

$$F_i = (J_b \pm J_c)C \pm (J_d \pm J_e)S \tag{16}$$

By implementing equation 16 rather than equation 12, reduces the number of filters which must be mechanized from 15 (one for each E_i) to 6 (one for each instrument). The equation mechanized are then those of figure 5-36 with each I_i replaced by J_i where J_i is the weighted summation of I_i.

If recognition of two simultaneous failures was not required only 4 of the error equations of figure 5-36 would have to be mechanized. Since two simultaneous failures must be recognized all 15 equations of figure 5-36 must be mechanized. The mechanization must perform two functions which are failure detection and failure isolation. Failure detection can be mechanized using only two of the equations of figure 5-36. The two equations used must contain inputs from all six instruments. An equation pair of this type is the first and sixth equation. These two equations implemented using the weighted summation of instrument outputs are

$$F_1 = (J_1 - J_2)C - (J_3 + J_4)S \tag{17}$$

$$F_6 = (J_5 - J_6)C - (J_1 + J_2)S \tag{18}$$

A failure is detected when F_1 and/or F_2 become significantly different from zero. A parameter G_i is mechanized using the equation

$$G_i = \begin{cases} 1 \text{ if } |F_i| \geq f \\ 0 \text{ if } |F_i| < f \end{cases} \qquad (19)$$

A failure is detected if either G_1 or G_6 is 1. Upon detecting a failure the failure must be isolated to one or two instruments. To isolate the failure all 15 values of G_i are computed. Figure 5-37 shows the expected values of G_i for each single instrument failure and two instrument failure combinations. With very soft failures initially, only a few values of G_i will become a 1. If the value of f in equation 19 is set small enough very soft failures which do not produce one of the G_i patterns of figure 5-37 will not create a navigation error of enough significance to be detrimental to mission success. The mechanization will thus isolate a failure only if one of the G_i patterns of figure 5-37 is generated.

Upon detecting and isolating a failure using the above procedure either one or two instruments will be eliminated as inputs to the navigation equations by selecting a new P matrix. This leaves either five or four instruments on which failure detection and isolation must be performed. A different procedure will be used for the two situations, i.e., four or five instruments remaining.

If five instruments are still operating fault detection is based upon computing two values of G_i. The values of G_i computed are dependent upon which five instruments remain operational. Figure 5-38 gives which values of G_i are computed for each possible combination of five operating instruments. One or more failures in the remaining five instruments will cause either one or both computed values of G_i to become 1. Upon detecting

Failed Instruments	Values of G_i for i =														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0
2	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1
3	1	1	1	0	0	0	1	1	1	0	1	1	1	0	1
4	1	0	0	1	1	0	1	1	0	1	1	1	0	1	1
5	0	1	0	1	0	1	1	0	1	1	1	0	1	1	1
6	0	0	1	0	1	1	0	1	1	1	0	1	1	1	1
1,2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
1,3	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1
1,4	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1
1,5	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1
1.6	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1
2,3	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1
2,4	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1
2,5	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1
2,6	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1
3,4	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1
3,5	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1
3,6	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1
4,5	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1
4,6	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
5,6	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Figure 5-37 Expected Values of G_i for all Single and Dual Instrument Failures

Operating Instruments 1,2,3,4,6

Compute $G_1, G_3, G_5, G_8, G_{12}$

Failed Instruments	Expected Pattern G_1	G_3	G_5	G_8	G_{12}
1	1	1	1	1	0
2	1	1	1	0	1
3	1	1	0	1	1
4	1	0	1	1	1
6	0	1	1	1	1
Multiply	1	1	1	1	1

Figure 5-39 Isolation Patterns for Five Instruments

Operating Instruments	Detection Equations
1,2,3,4,5	G_1, G_2
1,2,3,4,6	G_1, G_3
1,2,3,5,6	G_2, G_3
1,2,4,5,6	G_4, G_5
1,3,4,5,6	G_7, G_8
2,3,4,5,6	G_{11}, G_{12}

Figure 5-38 Error Detection Equations Required for Each Five Operating Instrument Combination

Operating Instruments	G_i required for Fault detection
1,2,3,4	G_1
1,2,3,5	G_2
1,2,3,6	G_3
1,2,4,5	G_4
1,2,4,6	G_5
1,2,5,6	G_6
1,3,4,5	G_7
1,3,4,6	G_8
1,3,5,6	G_9
1,4,5,6	G_{10}
2,3,4,5	G_{11}
2,3,4,6	G_{12}
2,3,5,6	G_{13}
2,4,5,6	G_{14}
3,4,5,6	G_{15}

Figure 5-40 G_i Required for Fault Detection with four Operating Instruments a fault, the faulty instrument or instruments must be isolated. With five instruments it is impossible (without external aids) to isolate two failed instruments. A trend analysis will be used to determine fault isolation if two out of five instruments fail. It is possible by investigating only instrument outputs to isolate a single failure if only one failure exists, and to detect the existance of multiply failures. Figure 5-39 indicates which values of G_i must be computed for a possible combination of five operating instruments in order to perform single failed instrument isolation and multiple failed instrument detection. The figure also shows the values of each G_i required to isolate a single instrument failure or detect a multiple instrument failure. There are five values of G_i which must be computed for each possible combination of five operating instruments. With a single failure four of the five values of G_i will be equal 1 and the other value 0. If testing determines that less than four values of G_i have a value of 1 then the failure is very soft and no action is taken until at least four values of G_i become 1. A multiple failure has occurred if all five computed values of G_i are 1. Upon isolating a single failure the failed instrument is eliminated and the proper P matrix chosen.

If a multiple failure is detected the failed instruments must be isolated by measurements directly on the instrument and trend analysis. This same procedure must be used to isolate a single failed instrument from four remaining instruments. With four remaining instruments, a failure is detected by computing a single value of G_i. The value of G_i used for failure detection for each possible combination of four instruments is shown in figure 5-40. A failure is detected if the single value of G_i computed is 1. If a multiple failure with five operating instruments or a single failure with four operating instruments is detected a third failure has occurred reducing the mission from operational status to that of safely returning the booster and crew. This reduction in mission status reduces the overall accuracy requirements of the navigation system. The action taken in isolating the failed instrument is dependent upon mission phase. The first action taken in any mission phase is to look at the individual instrument outputs and test points. The most common failure mode for any instrument will cause either a zero output or a full scale output. All four or five instruments will be investigated for the occurence of this type failure. If the orbiter is still attached, communication with the orbiter can provide sufficient data to isolate the failure. The data transmitted from the orbiter will be body axis delta velocities or delta attitude changes, i.e., the outputs of equation 10 as mechanized in the orbiter. Body axis for the orbiter and booster are defined independently for each vehicle. When the vehicles are attached to one another during boost these two axis will not coincide. A constant transformation matrix from orbiter body axis to booster body axis will be applied to the delta velocities or delta attitudes received from the orbiter. All possible combinations of three operating instruments are used with their appropriate P matrix transformations to obtain a set of transformed instrument output vectors for comparison with the expected values received from the orbiter. There are 10 possible combinations of three instruments for a multiple failure among five instruments and 4 possible combinations of three instruments for a single failure out of four instruments. If the detected failures exist, only one instrument combination will correlate with the orbiter values. The correlation should be very close if the instrument failures being isolated are gyros. The correlation will not be as close if the failed instruments being investigated are accelerometers. The reduction in correlation for the accelerometers is caused by rotational coupling through the separation distance between the orbiter and booster instrument packages. The major components of rotational coupling can be removed by filtering, since large magnitude rotational rates will be of short duration.

If failure isolation cannot be established by direct measurements upon the instruments and orbiter/booster separation has already occurred, failure isolation will be performed by trend analysis. Trend analysis is achieved by mathematically constructinga model of the vehicle, applying approximations to all forces and torques on the vehicle to the model, computing body axis rates and velocities, and comparing these with the instrument outputs. The general six degree of freedom equations of motion for a rigid body in body axis coordinates are

$$\Sigma \vec{F} = m\dot{\vec{V}} + \vec{\omega} \times \vec{V} \tag{20}$$

$$\Sigma \vec{T} = [I]\dot{\vec{\omega}} + \vec{\omega} \times [I]\vec{\omega} \tag{21}$$

where $\Sigma \vec{F}$ is the total force vector in the vehicle having components (ΣF_x, F_y, ΣF_z), m is the vehicle mass, $\vec{V}$ is the vehicle vector velocity having components (u,v,w), $\vec{\omega}$ is the vehicle angular velocity vector with respect to inertial space having components (p,q,r), $\Sigma \vec{T}$ is the total torques on the vehicle having components ($\Sigma T_x, \Sigma T_y, \Sigma T_z$) and [I]is the vehicle inertia dyadic. All vectors are expressed in body axis components. Due to symmetry of the vehicle inertia dyadic is assumed to have the form

$$I = \begin{bmatrix} I_{xx} & 0 & -I_{xz} \\ 0 & I_{yy} & 0 \\ -I_{xz} & 0 & I_{zz} \end{bmatrix} \tag{22}$$

Expanding equations 20 and 21 yields:

$$\Sigma F_x = m(\dot{u} + qw - rv) \quad (23)$$
$$\Sigma F_y = m(\dot{v} + ru - pw) \quad (24)$$
$$\Sigma F_z = m(\dot{w} + pv - qu) \quad (25)$$
$$\Sigma T_x = I_{xx}\dot{p} - I_{xz}\dot{r} - I_{yy}qr - I_{xz}qp + I_{zz}qr \quad (26)$$
$$\Sigma T_y = I_{yy}\dot{q} + I_{xx}rp - I_{xz}(r^2 - p^2) - I_{zz}rp \quad (27)$$
$$\Sigma T_z = I_{zz}\dot{r} - I_{xz}\dot{p} - I_{xx}qp + I_{xz}qr + I_{yy}qp \quad (28)$$

The vehicle and mission maneuvers are usually designed to minimize the coupling between longitudinal and lateral motions. This allows the above equations to be simplified and separated into two groups, one describing longitudinal motion and the other lateral motion. The longitudinal equations are generated by setting $v = r = p = 0$ in equations 23, 25, and 27. This results in the equations

$$\Sigma F_x = m(\dot{u} + qw) \quad (29)$$
$$\Sigma F_y = m(\dot{w} - qu) \quad (30)$$
$$\Sigma T_y = I_{yy}\dot{q} \quad (31)$$

The lateral equations are generated by setting $u = u_o$, $W = W_o$, and $q = 0$ in equations 24, 26 and 28 yielding:

$$\Sigma F_y = m(\dot{v} + ru_o - pw_o) \quad (32)$$
$$\Sigma T_x = I_{xx}\dot{p} - I_{xz}\dot{r} \quad (33)$$
$$\Sigma T_z = I_{zz}\dot{r} - I_{xz}\dot{p} \quad (34)$$

Equations 29 through 34 must be integrated in order to obtain estimates of u,v,w,p,q and r. In order to integrate the equations the forces and torques must be computed and the equations must be solved for $\dot{u},\dot{v},\dot{w},\dot{p},\dot{q}$, and $\dot{r}$. Solving the equations for the highest order derivatives yields:

$$\dot{u} = \frac{1}{m} \sum F_x - q w_0 \tag{35}$$

$$\dot{v} = \frac{1}{m} \sum F_y - r u_0 + p w_0 \tag{36}$$

$$\dot{w} = \frac{1}{m} \sum F_z + q u_0 \tag{37}$$

$$\dot{p} = \frac{I_{xz} \sum T_z + I_{zz} \sum T_x}{I_{xx} I_{zz} - I_{xz}^2} \tag{38}$$

$$\dot{q} = \frac{\sum T_y}{I_{yy}} \tag{39}$$

$$\dot{r} = \frac{I_{xz} \sum T_x + I_{xx} \sum T_z}{I_{xx} I_{zz} - I_{xz}^2} \tag{40}$$

Forces and torques on the vehicle have their source from gravity, aerodynamics, and engine thrust. Gravity produces no torques on the vehicle and produces the same acceleration forces on the accelerometer test masses as it does on the vehicle. Even through gravity has a major influence upon the vehicle motion it does not contribute to either the acceleromter or gyro outputs. Since the object of solving equations 35 through 40 is to simulate instrument outputs, the gravitational force is not included in the force model. Aerodynamic and engine thrust forces however must be included.

Aerodynamic forces and torques are defined as functions of dynamic pressure, q_i, angle of attack, α, and side slip angle, β. Three velocity components u', v', and w' are used to define these parameters. u', v' and w' are the components of the vehicle velocity with respect to the air mass expressed in body axis components. Then

$$\alpha = \tan^{-1} w'/u' \tag{41}$$

$$\beta = \tan^{-1} v'/u' \tag{42}$$

$$q_i = q_0 (u'^2 + v'^2 + w'^2) \tag{43}$$

The strapdown navigation equations are solved in an earth centered inertial coordinate frame. The navigation solution results in a vecocity vecotr, $\vec{V}_I$, and a position vector, $\vec{r}_I$. Defining $\vec{V}_I$ and $\vec{r}_I$ as having components

$$\vec{V}_I = \begin{bmatrix} u_I \\ v_I \\ w_I \end{bmatrix} \tag{44}$$

$$\vec{r}_I = \begin{bmatrix} X_I \\ Y_I \\ Z_I \end{bmatrix} \tag{45}$$

then

$$\begin{bmatrix} u' \\ v' \\ w' \end{bmatrix} = [C]^T \left\{ \begin{bmatrix} u_I \\ v_I \\ w_I \end{bmatrix} + \Omega_e \begin{bmatrix} -Y_I \\ X_I \\ 0 \end{bmatrix} \right\} \tag{46}$$

where Ω_e is the earth's rotational rate and $[C]$ is the transformation matrix from body to inertial coordinates. Winds are not included in equation 46. It is possible that the final mechanization may include winds in the model with wind data being supplied through pilot inputs, air data computations or filtered steering error data. Angle of attack during reentry will be large, requiring the arctangent computation indicated in equation 41. Side slip angle will always be small allowing for equation 42 to be replaced with:

$$\beta = v'/u' \tag{47}$$

The term q_σ in equation 43 must be computed from the vehicle altitude, h. Altitude is obtained from the equation set.

$$L = \tan^{-1} \frac{Z_I}{\sqrt{X_I^2 + Y_I^2}} \tag{48}$$

$$r_0 = r_e \left[1 - \frac{e}{2}(1 - \cos 2L) + \frac{5}{16} e^2 (1 - \cos 4L)\right] \tag{49}$$

$$h = \sqrt{X_I^2 + Y_I^2 + Z_I^2} - r_o \tag{50}$$

Equation 48 computes the vehicle's geocentric latitude from inertial position. Equation 49 is used to determine the local geoid radius based upon local geocentric latitude, the earth's equational radius, r_e, and the earth's ellipticity, e. Once h is obtained q_o is found from

$$q_o = K_1 e^{-K_2 h} \tag{51}$$

where K_1 and K_2 are constants.

The aerodynamic contributions to body axis forces and torques are described by

$$F_{x_{aero}} = q_i S\,(\alpha C_{x\alpha} + u C_{xu}) \tag{52}$$

$$F_{y_{aero}} = q_i S\,(\beta C_{y\beta} + p C_{yp} + r C_{yr}) \tag{53}$$

$$F_{z_{aero}} = q_i S\,(\alpha C_{z\alpha} + u C_{zu} + C_{zo}) \tag{54}$$

$$T_{x_{aero}} = q_i S\,(\beta C_{\ell\beta} + r C_{\ell r} + p C_{\ell p}) \tag{55}$$

$$T_{y_{aero}} = q_i S\,(\alpha C_{m\alpha} + u C_{mu} + q C_{mq}) \tag{56}$$

$$T_{z_{aero}} = q_i S\,(\beta C_{n\beta} + p C_{np} + r C_{nr}) \tag{57}$$

Equations 52 through 57 include 17 aerodynamic coefficients which are highly dependent upon Mach number. The values of these coefficients will be approximated using curve fit functions of Mach number. Mach number is computed from

$$M = \frac{K}{\sqrt{q_o}} \tag{58}$$

A secondary variable, a, will be formed from

$$a = \frac{1}{\sqrt{|1 - M^2|}} \tag{59}$$

a will be limited to a maximum value of q_{max} in the region of M = 1. For simplicity, only the curve fit formula for $C_{X\alpha}$ is given here. The formulas for the other 16 coefficients are of similar form. For $C_{X\alpha}$ the formula is

$$C_{X\alpha} = C_{X\alpha_0} + q\, C_{X\alpha_1} + q^2\, C_{X\alpha_2} \tag{60}$$

If a third accelerometer or gyro failure is encountered while the main rocket engines are burning, isolation of the failed instrument is achieved by communication with the orbiter as has been previously described. Thus trend analysis for the purpose of failed instrument isolation will not be required during main engine thrusting, eliminating the need for including main engine forces in the vehicle model. The air breathing engines produce forces along the X body axis and torques about the Z body axis. The engine control computers will generate estimates of the thrust of each engine. These estimates are transmitted to the DMS computers where the computations

$$F_{X\,eng} = T_1 + T_2 \tag{61}$$

$$T_{Z\,eng} = \ell_e (T_1 - T_2) \tag{62}$$

are made to generate the F_x and T_z engine contributions. T_1 and T_2 are the received engine thrust estimates and ℓ_e is half the separation distance between the two engines.

The only additional terms which must be added to the vehicle model are the torques generated by the attitude control system. The attitude control system computes pitch, roll and yaw commands (defined as q_c, p_c and r_c respectively), multiplies these commands by a variable gain, and issues the commands to the reaction jets or the areodynamic control surfaces. The variable gain is adjusted to maintain a nearly constant control effectiveness. The torque on the vehicle lags the generation of the torque command. These torques will be added to the vehicle model as a solution to the equations.

$$T_{x_{cont_n}} = K_x T_{x_{cont_{n-1}}} + M_x p_c \tag{63}$$

$$T_{y_{cont_n}} = K_y T_{y_{cont_{n-1}}} + M_y q_c \tag{64}$$

$$T_{z_{cont_n}} = K_z T_{z_{cont_{n-1}}} + M_z r_c \tag{65}$$

The subscripts n and n-1 are standard difference equation notations. The values of K_x, K_y, K_z, M_x, M_y, and M_z are chosen differently if the actual torques are generated using reaction jets or aerodynamic surfaces. To adjust for gain variations in aerodynamic surface control effectiveness the values of M_x,M_y and M_z will each be formed from a formula of the type:

$$M_x = A_{x0} + M A_{x1} + M^2 A_{x2} \tag{66}$$

where A_{xo}, A_{x1}, and A_{x2} are constants and M is Mach number as previously given.

The above equations representing a model of the navigation instrument outputs for use in trend analysis isolation of a failed instrument are only representative of the equations which must be used during flight. They are included for the purpose of sizing the computational requirements of the strapdown navigation task. A lengthy development and simulation study will be required to define the equation set to be mechanized.

A solution to the equations is formed by evaluating the right hand side of equations 35 through 40 and integrating. A solution is developed every T seconds. An estimate of the value of T to be used is 1/8 second. To evaluate the right hand side first, the forces and torques are evaluated using the above described procedure. Values of mass, inertias, u_o and W_0 vary slowly and will be evaluated at a slow rate which will be assumed to be once per second for the purposes of this sizing study. Mass and inertias will be formed as polynomial curve fit functions of fuel remaining or expended as determined by the propulsion programs. Curve fitting for mass is

$$m = m_0 - f_e \tag{67}$$

where m_0 is launch mass and f_e is fuel expended. For the inertias each curve fit formula will be of the form:

$$I_{xx} = I_{xx_0} + f_e I_{xx_1} + f_e^2 I_{xx_2} + f_e^3 I_{xx_3} \tag{68}$$

u_o and W_o are obtained from:

$$\begin{bmatrix} u_o \\ v_o \\ w_o \end{bmatrix} = \begin{bmatrix} C \end{bmatrix}^T \begin{bmatrix} u_I \\ v_I \\ w_I \end{bmatrix} \tag{68}$$

The values of p,q,and r required in the evaluation of the right hand side of equations 35 through 40 are the values resulting from the previous integration of the equations. The equations will be integrated using the digital integration formula:

$$u_n = u_{n-1} + \frac{T}{2}(3\dot{u}_{n-1} - \dot{u}_{n-2}) \tag{69}$$

for the integration of $\dot{u}$ and similar formulas for the integration of $\dot{v}$, $\dot{w}$, $\dot{p}$, $\dot{q}$, and $\dot{r}$. The subscripts n and n-1 are standard difference equation notation and $\dot{U}_{n-1}$ and $\dot{U}_{n-2}$ are the present and previous right hand side evaluation of equation 35.

The outputs of the accelerometers and gyros are transformed to body axis multiplying each output set by their appropriate P matrix. Defining these body asis components as ΔU_b, ΔV_b , ΔW_b, $\Delta \phi_b$, $\Delta \theta_b$ and $\Delta \psi_b$ the trend analysis outputs used for comparison with each body axis instrument output are computed from the formulas

$$\Delta u = (u_n - u_{n-1}) T_i / T \tag{70}$$

$$\Delta v = (v_n - v_{n-1}) T_i / T \tag{71}$$

$$\Delta w = (w_n - w_{n-1}) T_i / T \tag{72}$$

$$\Delta \phi = (3p_n - p_{n-1}) T_i / 2T \tag{73}$$

$$\Delta \theta = (3q_n - q_{n-1}) T_i / 2T \tag{74}$$

$$\Delta \psi = (3r_n - r_{n-1}) T_i / 2T \tag{75}$$

where T is the time interval between trend analysis solutions and T_i is the time interval between instrument processing. The trend analysis program will be executed continously even with all instruments functioning. While the instruments are operating properly, i.e., before a failure detection indicates there are three or less instruments of either type operating properly, an error for each term will be generated by the equations

$$\mathcal{E}_1 = \Delta u_b - \Delta u \tag{76}$$

$$\mathcal{E}_2 = \Delta v_b - \Delta v \tag{77}$$

$$\mathcal{E}_3 = \Delta w_b - \Delta w \tag{78}$$

$$\mathcal{E}_4 = \Delta \phi_b - \Delta \phi \tag{79}$$

$$\mathcal{E}_5 = \Delta \theta_b - \Delta \theta \tag{80}$$

$$\mathcal{E}_6 = \Delta \psi_b - \Delta \psi \tag{81}$$

While the instruments are operating properly, as defined above, force and torque correction terms to the trend analysis model will be generated using the formulas:

$$F_{xcorr_n} = F_{xcorr_{n-1}} + K_a \mathcal{E}_1 \tag{82}$$

$$F_{ycorr_n} = F_{ycorr_{n-1}} + K_a \mathcal{E}_2 \tag{83}$$

$$F_{zcorr_n} = F_{zcorr_{n-1}} + K_a \mathcal{E}_3 \tag{84}$$

$$T_{x\,corr_n} = T_{x\,corr_{n-1}} + K_g\,\mathcal{E}_4 \tag{85}$$

$$T_{y\,corr_n} = T_{y\,corr_{n-1}} + K_g\,\mathcal{E}_5 \tag{86}$$

$$T_{z\,corr_n} = T_{z\,corr_{n-1}} + K_g\,\mathcal{E}_6 \tag{87}$$

These force and torque corrections will be added to the force and torques used in evaluating the right hand side of equations 35 through 40. K_a and K_g are constants. When a failure is detected which indicates only three accelerometers or gyros are operating each possible combination of three instruments are formed for the failed instrument type. The outputs of each combination is transformed independently to body axis. K_g is set equal to zero if the failure is a gyro and K_a if the failure is an accelerometer. For each possible combination of three instruments an error value is computed from either

$$\mathcal{E}_a = \mathcal{E}_1^2 + \mathcal{E}_2^2 + \mathcal{E}_3^2 \tag{88}$$

or

$$\mathcal{E}_g = \mathcal{E}_4^2 + \mathcal{E}_5^2 + \mathcal{E}_6^2 \tag{89}$$

dependent upon an accelerometer or gyro failure repsectively. The combination having the smallest value of E_a or E_g is used in the strapdown computations. This procedure is repeated each sampling period until all but one value of E_a or E_g exceeds a limit value. When this occurs all instruments except those three contributing to the value of E_a or E_g which does not exceed the limit are eliminated and K_a or K_g reset to their proper value.

The output of the dodecahedron to body axis transformations are two vectors having components ($\Delta u_b, \Delta v_b, \Delta w_b$) and ($\Delta \phi_b, \Delta \theta_b, \Delta \psi_b$) where

$$\begin{bmatrix} \Delta u_b \\ \Delta v_b \\ \Delta w_b \end{bmatrix} = \int_t^{t+T} \begin{bmatrix} \dot{u}_b \\ \dot{v}_b \\ \dot{w}_b \end{bmatrix} \qquad (90)$$

and

$$\begin{bmatrix} \Delta \phi_b \\ \Delta \theta_b \\ \Delta \psi_b \end{bmatrix} = \int_t^{t+T} \begin{bmatrix} p_b \\ q_b \\ r_b \end{bmatrix} \qquad (91)$$

T is the time between instrument outputs and the right hand side vector of equations 90 and 91 are the vehicles acceleration and angular rotational rates with respect to inertial space expressed in a body axis coordinate system. Inertial navigation requires the solution of the equation

$$\bar{a} = \ddot{\bar{R}} - \bar{G} \qquad (92)$$

where $\bar{a}$ is the accelerometer's measured acceleration in inertial coordinates, $\bar{R}$ the vehicle's position in inertial coordinates and $\bar{G}$ the gravitational acceleration in inertial coordinates. The solution of equation 92 requires the transformation of the accelerometer outputs from body to inertial coordinates which is given by

$$\bar{a} \triangleq \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} = [C] \begin{bmatrix} \dot{u}_b \\ \dot{v}_b \\ \dot{w}_b \end{bmatrix} \qquad (93)$$

where $[C]$ is the direction cosine transformation matrix. In terms of Euler angles the direction cosine matrix is given by

$$[C] = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \qquad (94)$$

Performing the indicated matrix multiplications yields

$$[C] = \begin{bmatrix} \cos\psi\cos\theta - \sin\theta\sin\psi\sin\phi & \sin\psi\cos\phi & -\sin\psi\sin\theta - \sin\psi\sin\phi\cos\theta \\ -\sin\psi\cos\theta - \sin\theta\cos\psi\sin\phi & \cos\psi\cos\phi & \sin\psi\cos\theta - \cos\psi\sin\phi\cos\theta \\ \cos\phi\sin\theta & \sin\phi & \cos\phi\cos\theta \end{bmatrix} \qquad (95)$$

In order to simplify the notation the elements of $[C]$ are defined as

$$[C] \triangleq \begin{bmatrix} C_{11} & C_{21} & C_{31} \\ C_{12} & C_{22} & C_{32} \\ C_{13} & C_{23} & C_{33} \end{bmatrix} \qquad (96)$$

The matrix $[C]$ represents the transformation between body coordinates and inertial coordinates thus for any vector $\bar{X}_b$

$$\bar{X}_I = [C]\,\bar{X}_b \qquad (97)$$

taking the derivitive of equation 97 yields

$$\dot{\bar{X}}_I = [\dot{C}]\,\bar{X}_b + [C]\,\dot{\bar{X}}_b \qquad (98)$$

It is also true that $\dot{\bar{X}}_I$ expressed in body axis coordinates is given by

$$[C]^T \dot{\bar{X}}_I = (\dot{\bar{X}}_I)_b = \dot{\bar{X}}_b + [\omega]\bar{X}_b \tag{99}$$

where

$$[\omega] = \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix} \tag{100}$$

and $[\omega]\bar{X}_b$ is termed the contribution due to Coriolis. Multiplying equation 99 to $[C]$ and equating the result to equation 98 yields

$$[C]\dot{\bar{X}}_b + [C][\omega]\bar{X}_b = [\dot{C}]\bar{X}_b + [C]\dot{\bar{X}}_b \tag{101}$$

simplifying equation 101 noting that $\{[C][\omega]\}^T = -[\omega][C]^T$ yields

$$[\dot{C}]^T = -[\omega][C]^T \tag{102}$$

Equation 102 is used to propogate the direction cosine matrix $[C]$ during flight. Expanding $[C]^T$ using Taylor's series and difference equation notation yields:

$$[C]^T_{n+1} = [C]^T_n + [\dot{C}]^T_n \Delta T + \frac{1}{2}[\ddot{C}]^T_n \Delta T^2 + \cdots \tag{103}$$

Taking the derivitive of equation (102) yields

$$[\ddot{C}]^T = -[\dot{\omega}][C]^T - [\omega][\dot{C}]^T \tag{104}$$

Substituting equation 104 into equation 103 for $[\ddot{C}]^T$ and equation 102 into the result for $[\dot{C}]^T$ yields:

$$[C]^T_{n+1} = [C]^T_n - [\omega]_n[C]^T_n \Delta T - \frac{1}{2}[\dot{\omega}]_n[C]^T_n \Delta T^2 + \frac{1}{2}[\omega]_n[\omega]_n[C]^T_n \Delta T^2 \tag{105}$$

Approximating $[\dot{\omega}]_n$ by

$$[\dot{\omega}]_n = \{[\omega]_n - [\omega]_{n-1}\}/\Delta T \tag{106}$$

and defining

$$[\Delta\phi] \triangleq \begin{bmatrix} 0 & -\Delta\psi & \Delta\theta \\ \Delta\psi & 0 & -\Delta\phi \\ -\Delta\theta & \Delta\phi & 0 \end{bmatrix} \approx \begin{bmatrix} 0 & -r\Delta T & q\Delta T \\ r\Delta T & 0 & -p\Delta T \\ -q\Delta T & p\Delta T & 0 \end{bmatrix} = [\omega]\Delta T \tag{107}$$

then equation 105 becomes

$$[C]^T_{n+1} = [C]^T_n - \frac{3}{2}[\Delta\phi]_n [C]^T_n + \frac{1}{2}[\Delta\phi]_{n-1}[C]^T_n + \frac{1}{2}[\Delta\phi]_n[\Delta\phi]_n[C]^T_n \tag{108}$$

the matrix $[\Delta\phi]$ is constructed from the incremental instrument outputs after they have been transformed to body axis coordinates. The matrix $[C]^T$ represents a rotational transformation between two orthogonal coordinate systems and should exhibit the property

$$[C][C]^T = [I] \tag{109}$$

where

$$[I] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{110}$$

Because of computational errors caused by a finite computer word length (round off errors) and integration approximations (truncation errors) the matrix $[C]$ will not exhibit the properties of equation 109. Using equation 109 to define an error matrix yields

$$[\mathcal{E}] = [C][C]^T - [I] \tag{111}$$

A better approximation of the true transformation matrix from body to inertial coordinates is formed from

$$[C]^T - \frac{1}{2}[C]^T[\mathcal{E}] \tag{112}$$

The process of propogating the transformation matrix then becomes

$$[C']^T_{n+1} = \left\{[\Delta\phi]_n\left(\frac{1}{2}[\Delta\phi]_n - \frac{3}{2}[I]\right) + \frac{1}{2}[\Delta\phi]_{n-1} + [I]\right\}[C]^T_n \tag{113}$$

$$[C]^T_{n+1} = [C']^T_{n+1}\left\{\frac{3}{2}[I] - \frac{1}{2}[C']_{n+1}[C']^T_{n+1}\right\} \tag{114}$$

Equation 113 is formed by replacing $[C]^{T}_{n+1}$ with $[C']^{T}_{n+1}$ on the left hand side and partitioning the right hand side of equation 108. The partitioning reduces the number of steps required in a general purpose computer solution. Equation 114 is formed by setting expression 112 equal to $[C]^{T}_{n+1}$ and substituting equation 111 into the result.

The transformation matrix $[C]$ is applied to the incremental accelerometer outputs after they have been transformed to body axis by use of the appropriate $[P]$ matrix. This transformation yields:

$$\overline{\Delta U} \triangleq \begin{bmatrix} \Delta U \\ \Delta V \\ \Delta W \end{bmatrix} = [C] \begin{bmatrix} \Delta U_b \\ \Delta V_b \\ \Delta W_b \end{bmatrix} \tag{115}$$

Navigation requires the solution of equation 92 for $\bar{R}$ and $\dot{\bar{R}}$. Solving equation 92 for $\ddot{\bar{R}}$ yields

$$\ddot{\bar{R}} = \bar{a} + \bar{G} \tag{116}$$

The Taylor's series expansion for $\dot{\bar{R}}$ using difference equation notation is:

$$\dot{\bar{R}}_{n+1} = \dot{\bar{R}}_n + \ddot{\bar{R}}_n \Delta T + \frac{1}{2} \dddot{\bar{R}}_n \Delta T^2 + \cdots \tag{117}$$

Approximating $\dddot{\bar{R}}$ in equation 117 by

$$\dddot{\bar{R}}_n = (\ddot{\bar{R}}_n - \ddot{\bar{R}}_{n-1})/\Delta T \tag{118}$$

yields

$$\dot{\bar{R}}_{n+1} = \dot{\bar{R}}_n + \frac{\Delta T}{2}(3\ddot{\bar{R}}_n - \ddot{\bar{R}}_{n-1}) \tag{119}$$

Substituting equation 116 into 119 yields

$$\dot{\bar{R}}_{n+1} = \dot{\bar{R}}_n + (3\bar{a}_n - \bar{a}_{n-1})\frac{\Delta T}{2} + (3\bar{G}_n - \bar{G}_{n-1})\frac{\Delta T}{2} \tag{120}$$

The best available approximation to $\bar{a}$ is $\overline{\Delta U}/\Delta T$.

Thus equation 120 becomes

$$\dot{\bar{R}}_{n+1} = \dot{\bar{R}}_n + \frac{3}{2}\,\overline{\Delta u}_n - \frac{1}{2}\,\overline{\Delta u}_{n-1} + (3\bar{G}_n - \bar{G}_{n-1})\frac{\Delta T}{2} \qquad (121)$$

In order to solve equation 121 gravitational acceleration, G, must be evaluated in inertial coordinates. The solution for $\dot{\bar{R}}$ will be generated at a very fast rate, estimates used for this study are 64 times per second. $\bar{G}$ will change very little between two sequential solutions of $\dot{\bar{R}}$ making

$$\bar{G}_n = \bar{G}_{n-1} \qquad (122)$$

an approximation having little effect upon the solution accuracy of $\dot{\bar{R}}$.
Substituting equation 122 into 121 yields

$$\dot{\bar{R}}_{n+1} = \dot{\bar{R}}_n + \frac{3}{2}\,\overline{\Delta u}_n - \frac{1}{2}\,\overline{\Delta u}_{n-1} + \bar{G}_n\,\Delta T \qquad (123)$$

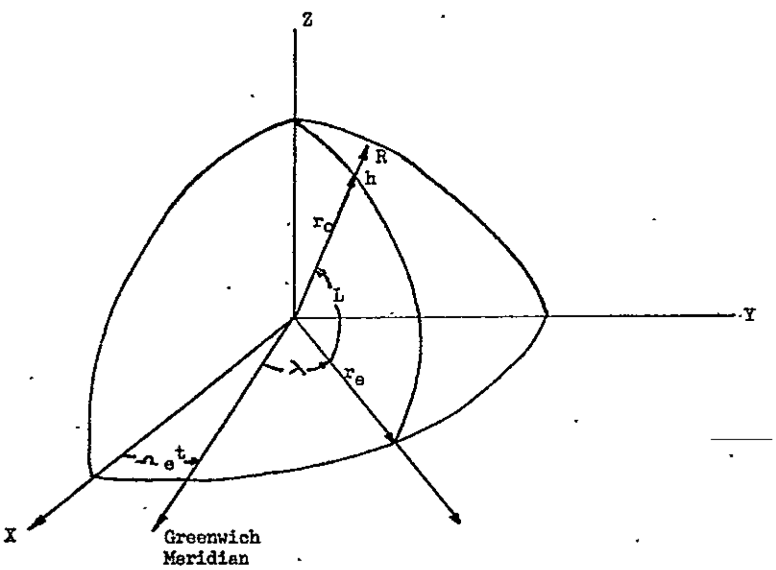


Figure 5-41 Inertial and Navigational Relationships

Figure 5-41 shows the geometric relationship between inertial coordinates and geocentric latitude and longitude. The inertial coordinate frame is represented by the axis system X, Y, Z. This coordinate frame is established at a point in time prior to vehicle launch such that the Greenwich meridian and the inertial X axis coincide. The Greenwich meridian will rotate toward the east as the mission progresses due to earth rate, Ω_e. The gravitational acceleration, $\bar{G}$, is directed toward the center of the earth along the $\bar{R}$ vector. The gravity vector is

$$\bar{G} = \frac{g_e r_e^2}{|\bar{R}|^3}\,\bar{R} \qquad (124)$$

where g_e is the earth's gravitational attraction on the earth's surface at the equator and r_e is the earth's equatorial radius. The generation of the absolute value of $\bar{R}$ requires the square root of the dot produce $\bar{R} \cdot \bar{R}$. Since the magnitude of $\bar{G}$ varys slowly the interation rate used for its solution will be slower than that used for navigation integrations. At a 2 per second rate the value of g_m defined as

$$g_m = \frac{g_e r_e^2}{|\bar{R}|^3} \tag{125}$$

will generated. The integration to propagate $\dot{\bar{R}}$ at a 64 per second rate will be mechanized using the formula:

$$\dot{\bar{R}}_{n+1} = \dot{\bar{R}}_n + \tfrac{3}{2}\overline{\Delta u}_n - \tfrac{1}{2}\overline{\Delta u}_{n-1} + g_m \bar{R}_n \Delta T \tag{126}$$

The value of $\bar{R}$ is obtained by integrating $\dot{\bar{R}}$ using the formula

$$\bar{R}_{n+1} = \bar{R}_n + \tfrac{3}{2}\dot{\bar{R}}_{n+1}\Delta T - \tfrac{1}{2}\dot{\bar{R}}_n \Delta T \tag{127}$$

During boost the primary navigation parameters required by the guidance system are $\bar{R}$ and $\dot{\bar{R}}$. During the reentry, cruise, and landing flight phases the the navigation parameters required are latitude, L, longitude λ , altitude, h, vertical velocity, V_z, horizontal velocity, V_h, and horizontal velocity vector heading angle ψ_V. The components of $\bar{R}$ and $\dot{\bar{R}}$ are defined as

$$\bar{R} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{128}$$

and

$$\dot{\bar{R}} = \begin{bmatrix} u \\ v \\ w \end{bmatrix} \tag{129}$$

An earth fixed coordinate set is defined as a right hand cartesian set having the positive Z_e axis along the earths north polar axis and the

X_e axis in the equatorial plane intersecting the equator at the Greenwich meridian. The space shuttle position and velocity in the earth's coordinate frame are determined from

$$\bar{R}_e = \begin{bmatrix} X_e \\ Y_e \\ Z_e \end{bmatrix} = \begin{bmatrix} X\cos\Omega_e t + Y\sin\Omega_e t \\ -X\sin\Omega_e t + Y\cos\Omega_e t \\ Z \end{bmatrix} \tag{130}$$

and

$$\dot{\bar{R}}_e = \begin{bmatrix} u_e \\ v_e \\ w_e \end{bmatrix} = \begin{bmatrix} u\cos\Omega_e t + v\sin\Omega_e t - \Omega_e Y \\ -u\sin\Omega_e t + v\cos\Omega_e t + \Omega_e X \\ w \end{bmatrix} \tag{131}$$

Longitude is determined from

$$\lambda = \tan^{-1} Y_e/X_e \tag{132}$$

Latitude is determined from

$$L = \tan^{-1} \frac{Z_e}{X_e\cos\lambda + Y_e\sin\lambda} \tag{133}$$

Altitude, h, is determined by subtracting the local earth radius from the magnitude of $\bar{R}e$. The earth's shape is closely approximated as an oblate spheroid having a local geocentric radius of

$$r_o = r_e(1 - e\sin^2 L) \tag{134}$$

where r_e is the equatorial radius and e the earth's ellipticity. The magnitude of $\bar{R}e$ can be determined from

$$|\bar{R}_e| = Z_e \sin L + (X_e\cos\lambda + Y_e\sin\lambda)\cos L \tag{135}$$

Then the altitude, h, is given by:

$$h = |\bar{R}_e| - r_o \tag{136}$$

Defining the velocities V_N , V_E, V_z as the northerly horizontal velocity, easternly horizontal velocity and vertical velocity respectively with the vertical velocity positive when the vehicle is decending then

$$\begin{bmatrix} V_N \\ V_E \\ V_z \end{bmatrix} = \begin{bmatrix} -\sin L \cos\lambda & -\sin L \sin\lambda & \cos L \\ -\sin\lambda & \cos\lambda & 0 \\ -\cos L \cos\lambda & -\cos L \sin\lambda & -\sin L \end{bmatrix} \begin{bmatrix} u_e \\ v_e \\ w_e \end{bmatrix} \tag{137}$$

The horizontal velocity heading angle, ψ_V, is then

$$\psi_V = \tan^{-1} \frac{V_E}{V_N} \tag{138}$$

and the horizontal velocity magnitude is

$$V_h = V_E \sin\psi_V + V_N \cos\psi_V \tag{139}$$

Latitude and longitude coordinates used in maps are geodetic rather than geocentric. Therefore all displays of latitude and longitude to the pilot should be in geodetic coordinates. The transformation between geocentric latitude and longitude (L and λ) and geodetic latitude and longitude (L_g and λ_g) is

$$L_g = \tan^{-1}[(1-e)\tan L] \tag{140}$$

$$\lambda_g = \lambda \tag{141}$$

The strapdown inertial navigation system must also provide various attitude references to the flight control system. These are Euler angle attitudes between inertial coordinates and body axis during boost and between earth fixed locally level axis and body axis during reentry, cruise and landing. Also angle of attack, α , and side slip angle, β, must be computed. During boost Euler angles are computed from the direction cosine matrix. By inspection from equation 95 and 96 the Euler angles are:

$$\psi = \tan^{-1} C_{21}/C_{22} \tag{142}$$

$$\Theta = \tan^{-1} C_{13}/C_{33} \tag{143}$$

$$\phi = \tan^{-1} C_{23}/(C_{13}\sin\Theta + C_{33}\cos\Theta) \tag{144}$$

During reentry, cruise, and landing a direction cosine matrix between body axis and locally level coordinate frame is constructed from:

$$[D] = \begin{bmatrix} -\sin L\cos\lambda & -\sin L\sin\lambda & \cos L \\ -\sin\lambda & \cos\lambda & 0 \\ -\cos L\cos\lambda & -\cos L\sin\lambda & -\sin L \end{bmatrix} \begin{bmatrix} \cos\Omega_e t & \sin\Omega_e t & 0 \\ -\sin\Omega_e t & \cos\Omega_e t & 0 \\ 0 & 0 & 1 \end{bmatrix} [C] \tag{145}$$

The only components of $[D]$ that must be computed are D_{21}, D_{13}, D_{23}, and D_{33}. The values of ψ, Θ and ϕ are then

$$\psi = \tan^{-1} D_{21}/D_{22} \tag{146}$$

$$\Theta = \tan^{-1} D_{13}/D_{33} \tag{147}$$

$$\phi = \tan^{-1} D_{23}/(D_{13}\sin\Theta + D_{33}\cos\Theta) \tag{148}$$

To generate the angle of attack and side slip angle the vehicle velocity with respect to the earth, $\dot{\bar{R}}_e$, must be transformed to body axis coordinates. This is achieved by

$$\begin{bmatrix} u_{eb} \\ v_{eb} \\ w_{eb} \end{bmatrix} = [C]^T \left\{ \begin{bmatrix} u \\ v \\ w \end{bmatrix} + \begin{bmatrix} -\Omega_e Y\cos\Omega_e t - \Omega_e X\sin\Omega_e t \\ -\Omega_e Y\sin\Omega_e t + \Omega_e X\cos\Omega_e t \\ 0 \end{bmatrix} \right\} \tag{149}$$

Then

$$\alpha = \tan^{-1} w_{eb}/u_{eb} \tag{150}$$

and

$$\beta = \tan^{-1} v_{eb}/u_{eb} \tag{151}$$

During reentry, cruise and landing other navigational aids will be used to update the strapdown inertial navigation solution. The measurements made by other navigation systems will provide new values of λ, L, h, Ψ_V, α, β, and ψ. The strapdown inertial navigation system contains 15 integrators. These integrators contain 9 direction cosine values, 3 velocity vector components, and 3 position components. Significant errors in the strapdown inertial navigation system will exist as errors in the integrator outputs. Each parameter which can be determined by some navigational aid other than the strapdown system can also be developed from the strapdown system integrator outputs. Using longitude as an example this can be expressed as

$$\lambda = f_\lambda([C], \dot{\bar{R}}, \bar{R}) \tag{152}$$

A matrix, [Q] can be constructed from partial derivitives of each navigation parameter with respect to each integrator output. The matrix will be of the form

$$[Q] = \begin{bmatrix} \frac{\partial\lambda}{\partial C_{11}} & \frac{\partial\lambda}{\partial C_{12}} & \cdots & \frac{\partial\lambda}{\partial C_{33}} & \frac{\partial\lambda}{\partial u} & \frac{\partial\lambda}{\partial v} & \frac{\partial\lambda}{\partial w} & \frac{\partial\lambda}{\partial X} & \frac{\partial\lambda}{\partial Y} & \frac{\partial\lambda}{\partial Z} \\ \frac{\partial L}{\partial C_{11}} & \ddots & & & & & & & & \vdots \\ \vdots & & & & & & & & & \\ \frac{\partial\psi}{\partial C_{11}} & \cdots & & & & & & & & \frac{\partial\psi}{\partial Z} \end{bmatrix} \tag{153}$$

The equations for each parameter in the matrix are:

$$\frac{\partial\lambda}{\partial C_{ij}} = \frac{\partial\lambda}{\partial u} = \frac{\partial\lambda}{\partial v} = \frac{\partial\lambda}{\partial w} = \frac{\partial\lambda}{\partial Z} = 0 \tag{154}$$

$$\frac{\partial\lambda}{\partial X} = \frac{-X_e}{X_e^2 + Y_e^2}\left[\sin\Omega_e t + \frac{Y_e}{X_e}\cos\Omega_e t\right] \tag{155}$$

$$\frac{\partial\lambda}{\partial Y} = \frac{X_e}{X_e^2 + Y_e^2}\left[\cos\Omega_e t - \frac{Y_e}{X_e}\sin\Omega_e t\right] \tag{156}$$

$$\frac{\partial L}{\partial C_{ij}} = \frac{\partial L}{\partial u} = \frac{\partial L}{\partial v} = \frac{\partial L}{\partial w} = 0 \tag{157}$$

$$\frac{\partial L}{\partial X} = -\frac{X_e\cos\lambda + Y_e\sin\lambda}{X_e^2 + Y_e^2 + Z_e^2} Z_e\left[X_e\cos\Omega_e t - Y_e\sin\Omega_e t\right] \tag{158}$$

$$\frac{\partial L}{\partial Y} = -\frac{X_e \cos\lambda + Y_e \sin\lambda}{X_e^2 + Y_e^2 + Z_e^2} Z_e (X_e \sin\Omega_e t + Y_e \cos\Omega_e t) \tag{159}$$

$$\frac{\partial L}{\partial Z} = \frac{X_e \cos\lambda + Y_e \sin\lambda}{X_e^2 + Y_e^2 + Z_e^2} \tag{160}$$

$$\frac{\partial h}{\partial C_{ij}} = \frac{\partial h}{\partial u} = \frac{\partial h}{\partial v} = \frac{\partial h}{\partial w} = 0 \tag{161}$$

$$\frac{\partial h}{\partial X} = \frac{1}{|R_e|}(X_e \cos\Omega_e t - Y_e \sin\Omega_e t) + 2e \sin L \cos L \frac{\partial L}{\partial X} \tag{162}$$

$$\frac{\partial h}{\partial Y} = \frac{1}{|R_e|}(X_e \sin\Omega_e t + Y_e \cos\Omega_e t) + 2e \sin L \cos L \frac{\partial L}{\partial Y} \tag{163}$$

$$\frac{\partial h}{\partial Z} = \frac{Z_e}{|R_e|} + 2e \sin L \cos L \frac{\partial L}{\partial Z} \tag{164}$$

$$\frac{\partial \psi_V}{\partial C_{ij}} = 0 \tag{165}$$

$$\frac{\partial \psi_V}{\partial u} = \frac{-V_N}{V_E^2 + V_N^2}\left[\sin\lambda \cos\Omega_e t + \cos\lambda \sin\Omega_e t + \frac{V_E}{V_N} \sin L (\sin\lambda \sin\Omega_e t - \cos\lambda \cos\Omega_e t)\right] \tag{166}$$

$$\frac{\partial \psi_V}{\partial V} = \frac{V_N}{V_E^2 + V_N^2}\left[\cos\lambda \cos\Omega_e t - \sin\lambda \sin\Omega_e t + \frac{V_E}{V_N} \sin L(\cos\lambda \sin\Omega_e t + \sin\lambda \cos\Omega_e t)\right] \tag{167}$$

$$\frac{\partial \psi_V}{\partial W} = \frac{-V_E}{V_E^2 + V_N^2} \cos L \tag{168}$$

$$\frac{\partial \psi_V}{\partial X} = \frac{V_N}{V_F^2 + V_N^2}\Big\{-(u_e \cos\lambda + V_e \sin\lambda)\frac{\partial \lambda}{\partial X} + \Omega_e \cos\lambda + \frac{V_E}{V_N}\Big[(u_e \cos L \cos\lambda$$
$$+ V_e \cos L \sin\lambda + W_e \sin L)\frac{\partial L}{\partial X} + \sin L(V_e \cos\lambda - u_e \sin\lambda)\frac{\partial \lambda}{\partial X} + \Omega_e \sin L \sin\lambda\Big]\Big\} \tag{169}$$

$$\frac{\partial \psi_V}{\partial Y} = \frac{V_N}{V_E^2 + V_N^2}\Big\{-(u_e \cos\lambda + V_e \sin\lambda)\frac{\partial \lambda}{\partial Y} + \Omega_e \sin\lambda + \frac{V_E}{V_N}\Big[(u_e \cos L \cos\lambda + V_e \cos L \sin\lambda$$
$$+ W_e \sin L)\frac{\partial L}{\partial Y} + \sin L(V_e \cos\lambda - u_e \sin\lambda)\frac{\partial \lambda}{\partial Y} - \Omega_e \sin L \cos\lambda\Big]\Big\} \tag{170}$$

$$\frac{\partial \psi_V}{\partial Z} = \frac{V_E}{V_E^2 + V_N^2}(u_e \cos L \cos\lambda + V_e \cos L \sin\lambda + W_e \sin L)\frac{\partial L}{\partial Z} \tag{171}$$

$$\frac{\partial \alpha}{\partial C_{21}} = \frac{\partial \alpha}{\partial C_{22}} = \frac{\partial \alpha}{\partial C_{23}} = \frac{\partial \alpha}{\partial Z} = 0 \tag{172}$$

$$\frac{\partial \alpha}{\partial C_{11}} = \frac{-W_{eb}}{W_{eb}^2 + u_{eb}^2}(u - \Omega_e Y \cos\Omega_e t - \Omega_e X \sin\Omega_e t) \tag{173}$$

$$\frac{\partial \alpha}{\partial C_{12}} = \frac{-W_{eb}}{W_{eb}^2 + u_{eb}^2}(V - \Omega_e Y \sin\Omega_e t + \Omega_e X \cos\Omega_e t) \tag{174}$$

$$\frac{\partial \alpha}{\partial C_{13}} = \frac{-W W_{eb}}{W_{eb}^2 + u_{eb}^2} \tag{175}$$

$$\frac{\partial \alpha}{\partial C_{31}} = \frac{u_{eb}}{W_{eb}^2 + u_{eb}^2}(u - \Omega_e Y \cos\Omega_e t - \Omega_e X \sin\Omega_e t) \tag{176}$$

$$\frac{\partial \alpha}{\partial C_{32}} = \frac{u_{eb}}{W_{eb}^2 + u_{eb}^2}(V - \Omega_e Y \sin\Omega_e t + \Omega_e X \cos\Omega_e t) \tag{177}$$

$$\frac{\partial \alpha}{\partial C_{33}} = \frac{W U_{eb}}{W_{eb}^2 + U_{eb}^2} \tag{178}$$

$$\frac{\partial \alpha}{\partial U} = \frac{U_{eb}}{W_{eb}^2 + U_{eb}^2}\left(C_{31} - \frac{W_{eb}}{U_{eb}} C_{11}\right) \tag{179}$$

$$\frac{\partial \alpha}{\partial V} = \frac{U_{eb}}{W_{eb}^2 + U_{eb}^2}\left(C_{32} - \frac{W_{eb}}{U_{eb}} C_{12}\right) \tag{180}$$

$$\frac{\partial \alpha}{\partial W} = \frac{U_{eb}}{W_{eb}^2 + U_{eb}^2}\left(C_{33} - \frac{W_{eb}}{U_{eb}} C_{13}\right) \tag{181}$$

$$\frac{\partial \alpha}{\partial X} = \frac{U_{eb}\Omega_e}{W_{eb}^2 + U_{eb}^2}\left[\left(C_{32}\cos\Omega_e t - C_{31}\sin\Omega_e t\right) + \frac{W_{eb}}{U_{eb}}\left(C_{11}\sin\Omega_e t - C_{12}\cos\Omega_e t\right)\right] \tag{182}$$

$$\frac{\partial \alpha}{\partial Y} = \frac{U_{eb}\Omega_e}{W_{eb}^2 + U_{eb}^2}\left[\left(-C_{31}\cos\Omega_e t - C_{32}\sin\Omega_e t\right) + \frac{W_{eb}}{U_{eb}}\left(C_{11}\cos\Omega_e t + C_{12}\sin\Omega_e t\right)\right] \tag{183}$$

$$\frac{\partial \beta}{\partial C_{31}} = \frac{\partial \beta}{\partial C_{32}} = \frac{\partial \beta}{\partial C_{33}} = \frac{\partial \beta}{\partial Z} = 0 \tag{184}$$

$$\frac{\partial \beta}{\partial C_{11}} = \frac{-V_{eb}}{U_{eb}^2 + V_{eb}^2}\left(U - \Omega_e Y\cos\Omega_e t - \Omega_e X\sin\Omega_e t\right) \tag{185}$$

$$\frac{\partial \beta}{\partial C_{12}} = \frac{-V_{eb}}{U_{eb}^2 + V_{eb}^2}\left(V - \Omega_e Y\sin\Omega_e t + \Omega_e X\cos\Omega_e t\right) \tag{186}$$

$$\frac{\partial \beta}{\partial C_{13}} = \frac{-V_{eb} W}{U_{eb}^2 + V_{eb}^2} \tag{187}$$

$$\frac{\partial \beta}{\partial C_{21}} = \frac{U_{eb}}{U_{eb}^2 + V_{eb}^2}\left(U - \Omega_e Y\cos\Omega_e t - \Omega_e X\sin\Omega_e t\right) \tag{188}$$

$$\frac{\partial \beta}{\partial C_{22}} = \frac{U_{eb}}{U_{eb}^2 + V_{eb}^2}\left(V - \Omega_e Y\sin\Omega_e t + \Omega_e X\cos\Omega_e t\right) \tag{189}$$

$$\frac{\partial \beta}{\partial C_{23}} = \frac{U_{eb} W}{U_{eb}^2 + V_{eb}^2} \tag{190}$$

$$\frac{\partial \beta}{\partial U} = \frac{U_{eb}}{U_{eb}^2 + V_{eb}^2}\left(C_{21} - \frac{V_{eb}}{U_{eb}} C_{11}\right) \tag{191}$$

$$\frac{\partial \beta}{\partial V} = \frac{U_{eb}}{U_{eb}^2 + V_{eb}^2}\left(C_{22} - \frac{V_{eb}}{U_{eb}} C_{12}\right) \tag{192}$$

$$\frac{\partial \beta}{\partial W} = \frac{U_{eb}}{U_{eb}^2 + V_{eb}^2}\left(C_{23} - \frac{V_{eb}}{U_{eb}} C_{13}\right) \tag{193}$$

$$\frac{\partial \beta}{\partial X} = \frac{U_{eb}\Omega_e}{U_{eb}^2 + V_{eb}^2}\left[\left(C_{22}\cos\Omega_e t - C_{21}\sin\Omega_e t\right) - \frac{V_{eb}}{U_{eb}}\left(C_{12}\cos\Omega_e t - C_{11}\sin\Omega_e t\right)\right] \tag{194}$$

$$\frac{\partial \beta}{\partial Y} = \frac{U_{eb}\Omega_e}{U_{eb}^2 + V_{eb}^2}\left[-\left(C_{21}\cos\Omega_e t + C_{22}\sin\Omega_e t\right) + \frac{V_{eb}}{U_{eb}}\left(C_{11}\cos\Omega_e t + C_{12}\sin\Omega_e t\right)\right] \tag{195}$$

$$\frac{\partial \psi}{\partial C_{11}} = \frac{\partial \psi}{\partial C_{12}} = \frac{\partial \psi}{\partial C_{13}} = \frac{\partial \psi}{\partial C_{31}} = \frac{\partial \psi}{\partial C_{32}} = \frac{\partial \psi}{\partial C_{33}} = \frac{\partial \psi}{\partial U} = \frac{\partial \psi}{\partial V} = \frac{\partial \psi}{\partial W} = 0 \tag{196}$$

$$\frac{\partial \psi}{\partial C_{21}} = \frac{D_{22}}{D_{21}^2 + D_{22}^2}\left[\sin L\left(\sin\Omega_e t\sin\lambda - \cos\Omega_e t\cos\lambda\right) + \frac{D_{21}}{D_{22}}\left(\cos\Omega_e t\sin\lambda + \sin\Omega_e t\cos\lambda\right)\right] \tag{197}$$

$$\frac{\partial \psi}{\partial C_{22}} = \frac{D_{22}}{D_{21}^2 + D_{22}^2}\Big[-\sin L\left(\cos\Omega_e t \sin\lambda + \sin\Omega_e t\cos\lambda\right) + \frac{D_{21}}{D_{22}}\left(\sin\Omega_e t\sin\lambda - \cos\Omega_e t\cos\lambda\right)\Big] \tag{198}$$

$$\frac{\partial \psi}{\partial C_{23}} = \frac{D_{22}\cos L}{D_{21}^2 + D_{22}^2} \tag{199}$$

$$\begin{aligned}\frac{\partial \psi}{\partial X} = \frac{D_{22}}{D_{21}^2 + D_{22}^2}\Big[&\left(C_{21}\cos\Omega_e t + C_{22}\sin\Omega_e t\right)\Big(\sin L\sin\lambda\frac{\partial\lambda}{\partial X} - \cos L\cos\lambda\frac{\partial L}{\partial \lambda} \\ &+ \frac{D_{21}}{D_{22}}\cos\lambda\frac{\partial\lambda}{\partial X}\Big) + \left(C_{22}\cos\Omega_e t - C_{21}\sin\Omega_e t\right)\Big(-\cos L\sin\lambda\frac{\partial L}{\partial\lambda} \\ &- \sin L\cos\lambda\frac{\partial\lambda}{\partial X} + \frac{D_{21}}{D_{22}}\sin\lambda\frac{\partial\lambda}{\partial X}\Big) - C_{23}\sin L\frac{\partial L}{\partial X}\Big]\end{aligned} \tag{200}$$

$$\begin{aligned}\frac{\partial \psi}{\partial Y} = \frac{D_{22}}{D_{21}^2 + D_{22}^2}\Big[&\left(C_{21}\cos\Omega_e t + C_{22}\sin\Omega_e t\right)\Big(-\cos L\cos\lambda\frac{\partial L}{\partial Y} + \sin L\sin\lambda\frac{\partial\lambda}{\partial Y} \\ &+ \frac{D_{21}}{D_{22}}\cos\lambda\frac{\partial\lambda}{\partial Y}\Big) + \left(C_{22}\cos\Omega_e t - C_{21}\sin\Omega_e t\right)\Big(-\cos L\sin\lambda\frac{\partial L}{\partial Y} \\ &- \sin L\cos\lambda\frac{\partial\lambda}{\partial Y} + \frac{D_{21}}{D_{22}}\sin\lambda\frac{\partial\lambda}{\partial Y}\Big) - C_{23}\sin L\frac{\partial L}{\partial Y}\Big)\end{aligned} \tag{201}$$

$$\begin{aligned}\frac{\partial \psi}{\partial Z} = \frac{D_{22}}{D_{21}^2 + D_{22}^2}\Big[&-\cos L\cos\lambda\left(C_{21}\cos\Omega_e t + C_{22}\sin\Omega_e t\right) \\ &- \cos L\sin\lambda\left(C_{22}\cos\Omega_e t - C_{21}\sin\Omega_e t\right) - \sin L\Big]\frac{\partial L}{\partial Z}\end{aligned} \tag{202}$$

Using the [Q] matrix as defined by equation 153 a derivitive equation can be written as:

$$\overline{d\lambda} \triangleq \begin{bmatrix} d\lambda \\ dL \\ dh \\ d\psi_r \\ d\alpha \\ d\beta \\ d\psi \end{bmatrix} = [Q]\begin{bmatrix} dC_{11} \\ dC_{12} \\ \vdots \\ dC_{33} \\ du \\ dv \\ \vdots \\ dZ \end{bmatrix} \triangleq [Q]\,\overline{dC} \tag{203}$$

To update the strapdown navigation system the vector $\overline{d\lambda}$ is formed by evaluating the functions

$$\begin{aligned} d\lambda &= f(\lambda, \lambda_m) \\ dL &= f(L, L_m) \\ &\vdots \\ d\psi &= f(\psi, \psi_m) \end{aligned} \tag{204}$$

where λ is longitude developed from the strapdown system and λ_m is longitude determined from some other independent nav-aid source. The errors of the various navigation sources have different characteristics. Strapdown errors have the characteristic of increasing with time while other navigation sources on the space shuttle have limits on their total error magnitude. The function required by equation 204 will be mechanized in the form

$$f(\lambda, \lambda_m) = K_\lambda |(\lambda_m - \lambda)| (\lambda_m - \lambda) \tag{205}$$

where K_λ is a constant selected according to the navigation source of λ_m. This makes the magnitude of the function proportional to the square of the difference between the two navigation system solutions while maintaining the sign of the difference. In order to apply an update to the navigation system equation 203 must be solved for $\overline{dC}$. Since equation 203 represents 7 linear equations in 15 unknowns many solutions to equation 203 exist. The one which will be used for this application is

$$\overline{dC} = \frac{1}{q^2} [Q]^T \overline{d\lambda} \tag{206}$$

Matrix $[Q]^T$ contains many zero terms which will reduce the time required to compute equation 206.

q^2 is evaluated by taking the sum of the squares of all non zero elements of $[Q]$.

The output of a single degree of freedom gyro is

$$\omega_m = (1 + \Delta K_g)\omega + B_g \tag{207}$$

where ω_m is the measured angular rate, ω is the angular rate of the instrument with respect to inertial space about the instruments sensitive axis, K_g represents the instruments scale factor error and B_g the instruments bias error. In a similar manner the accelerometer output is given by:

$$a_m = (1 + \Delta K_a) a + B_a \tag{208}$$

where a_m is measured acceleration, a actual acceleration, K_a scale factor error, and B_a bias error. During prelaunch the angular rate input to each gyro is given by

$$\omega = \Omega_s + \rho_s \tag{209}$$

where Ω_s is the component of earth rate along the gyros sensitive axis and ρ_s is the angular rate component of the vehicle with respect to the earth along the instruments sensitive axis. ρ_s is caused by movement of the vehicle due to launch site winds and loading of fuel, passengers, and stores on the vehicle. The acceleration experienced by the vehicle is given by

$$a = -g_s + a_s \tag{210}$$

where g_s is the acceleration due to gravity and a_s the distrubance accelerations. A negative sign on g_s is used because the accelerometer does not measure the acceleration due to gravity but the opposing force of the earth stopping the vehicle from being accelerated by gravity.

During prelaunch and pre ferry mission phases the strapdown inertial navigation system must be initialized and calibrated. At the launch site ground equipment is used to measure and transmit accurate values of vehicle attitude with respect to a local earth coordinate system. The attitude data transmitted to the vehicle is in the form of three Euler angles ϕ_L, θ_L, and ψ_L. In addition to vehicle attitude with respect to locally level coordinates launch site latitude and longitude and time are available from keyboard entry. Time is accumulated in the computer by the executive program but initial keyboard entires are used to sync the computer time with Greenwich standard time. From these inputs the direction cosine matrix is initialized by using the equation:

$$[C] = [\Omega t][\lambda][L][\psi_L'][\phi_L'][\theta_L'] \tag{211}$$

$$[\Omega t] = \begin{bmatrix} \cos\Omega t & -\sin\Omega t & 0 \\ \sin\Omega t & \cos\Omega t & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{212}$$

$$[\lambda] = \begin{bmatrix} \cos\lambda & -\sin\lambda & 0 \\ \sin\lambda & \cos\lambda & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{213}$$

$$[L] = \begin{bmatrix} \sin L & 0 & -\cos L \\ 0 & 1 & 0 \\ \cos L & 0 & \sin L \end{bmatrix} \tag{214}$$

$$[\psi_L'] = \begin{bmatrix} \cos\psi_L' & \sin\psi_L' & 0 \\ -\sin\psi_L' & \cos\psi_L' & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{215}$$

$$[\phi_L'] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi_L' & -\sin\phi_L' \\ 0 & \sin\phi_L' & \cos\phi_L' \end{bmatrix} \tag{216}$$

$$[\theta_L'] = \begin{bmatrix} \cos\theta_L' & 0 & -\sin\theta_L' \\ 0 & 1 & 0 \\ \sin\theta_L' & 0 & \cos\theta_L' \end{bmatrix} \tag{217}$$

The values of ψ_L', ϕ_L' and θ_L' are derived from ψ_L, ϕ_L, and θ_L respectively, by filtering. The filtering is accomplished by mechanizing the difference equations

$$\psi'_{L_n} = K_1 \psi'_{L_{n-1}} + K_2 \psi_{L_n} \tag{218}$$

$$\phi'_{L_n} = K_1 \phi'_{L_{n-1}} + K_2 \phi_{L_n} \tag{219}$$

and

$$\theta'_{L_n} = K_1 \theta'_{L_{n-1}} + K_2 \theta_{L_n} \tag{220}$$

During the initialization and calibration procedure the direction cosine matrix must be continously updated as ψ_L', ϕ_L', θ_L' and t changes. Earth rate and gravity are transformed to body axis using the formulas

$$\bar{g}_b = [C]^T [\Omega t][\lambda][L] \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \tag{221}$$

and

$$\bar{\Omega}_b = [C]^T \begin{bmatrix} 0 \\ 0 \\ \Omega \end{bmatrix} \tag{222}$$

where g is the local acceleration due to gravity found from

$$g = \frac{g_e r_e^2}{(r_o + h)^2} \tag{223}$$

and Ω is earth rate. r_e and g_e are the same as defined for equation 124, r_o is found from equation 134 and h is the launch site altitude above sea level obtained from data entry through the keyboard. The components of $\bar{g}_b$ and $\bar{\Omega}_b$ along each instruments sensitive axis is then generated by the equations:

$$\bar{g}_s = [M]\,\bar{g}_b \tag{224}$$

and

$$\bar{\Omega}_s = [M]\,\bar{\Omega}_b \tag{225}$$

where the matrix [M] is defined by equation 6. Bias and scale factor errors are removed from the outputs of each instrument during flight. For the gyros this error removal is accomplished by loading a bias, B_g, and a positive and negative scale factor S_{g+}, S_{g-} register in the gyro. The accelerometer errors are removed by applying the formula

$$I = A_1 I_{in} - A_o - A_2 I_{in}^2 - A_3 I_{in}^3 \tag{226}$$

where I_{in} is the raw incremental accelerometer imput to the computer and I is the corrected accelerometer output. For each gyro the bias value is computer from

$$B_g = B_o + B_1 a_n + B_2 a_n^2 \tag{227}$$

where B_o, B_1, and B_2 are constants with different values for each gyro and

a_n is equal to ΔV_b for gyros 1 and 2, ΔW_b for gyros 3 and 4, and Δu for gyros 5 and 6. Initially A_0, A_1, A_2, A_3, B_0, B_1, B_2, S_{g+} and S_{g-} are set to values determined from laboratory calibrations of each instrument. During prelaunch and pre-ferry navigation calibration the values of B_0, S_{g+}, S_{g-}, A_0, and A_1 are adjusted. Equations 224 and 225 represent the expected instrument outputs. For each accelerometer a sum of N errors between actual and expected outputs are accumulated by

$$\mathcal{E}_a = \sum_1^N (g_s \Delta T - I_a) \tag{228}$$

and likewise for the gyros using formulas of the type

$$\mathcal{E}_g = \sum_1^N (\Omega_s \Delta T - I_g)$$

where I_a and I_g are the corrected accelerometer and gyro incremental inputs. From equations 208 and 207 the accumulated errors are

$$\mathcal{E}_a = -(\Delta K_a g_s + B_a) N \Delta T \tag{229}$$

and

$$\mathcal{E}_g = -(\Delta K_g \Omega_s + B_g) N \Delta T \tag{230}$$

Because g_s and Ω_s are constants unless the vehicle is moving with respect to the earth, there are no characteristics to distinguish between error contributions from scale factor and bias errors. In a platform system, the platform can be torqued to a new orientation to change the values of g_s and Ω_s for each system. This cannot be done for a strapdown system. Analysis of expected error sources will be used to determine the proportioning of the accumulated error to bias and scale factor sources. The instruments will be calibrated using the formulas:

$$\Delta K_a = \frac{-K_1 \mathcal{E}_a}{N \Delta T}\, g_s \tag{231}$$

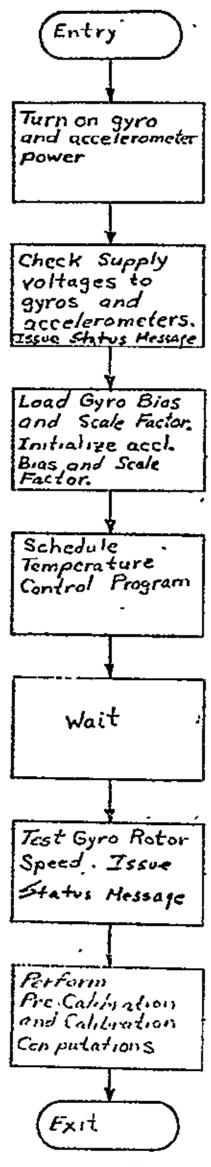
$$B_a = -\frac{\varepsilon_a}{N\Delta T} - \Delta K_a g_s \quad (232)$$

$$\Delta K_g = \frac{-K_2 \varepsilon_g}{N\Delta T} \Omega_s \quad (233)$$

$$B_g = -\frac{\varepsilon_g}{N\Delta T} - \Delta K_g \Omega_s \quad (234)$$

where K_1 and K_2 are constants having a value between 0 and 1.

This completes the description of the computations required by the strapdown inertial navigation systems. These equations are distributed between several programs dependent upon mission phase and solution rate requirements. Figure 5-42 is a flow diagram of the Strap Down inertial navigation start up program which is scheduled during pre launch or pre ferry and run at a 16 times per second rate. Upon entry to the program, power is commanded to the gyros and accelerometers and then tested. The Gyro bias and scale factor registers are then loaded and the temperature control program scheduled. A wait is then programmed to allow the gyro rotor to come up to speed and then the gyro rotor speed tested. If either power supply voltage or rotor speed tests indicate a failed instrument, appropriate error messages are issued and flags set to select the proper P matrix. Pre calibration and calibration computations are then performed. Figure 5-43 is a flow diagram of the pre calibration and calibration procedure. Before calibration is started, the accelerometers and gyros are tested to determine if they have reached their nominal temperature valve. If a nominal temperature for all instruments is not achieved within a preset period, appropriate error messages are generated. Vehicle attitude readings from ground equipment optics (or keyboard entries for a ferry mission) are sampled and used to initialize the ψ'_L, ϕ'_L, and θ'_L filter equations. During prelaunch, a wait is then programmed during which



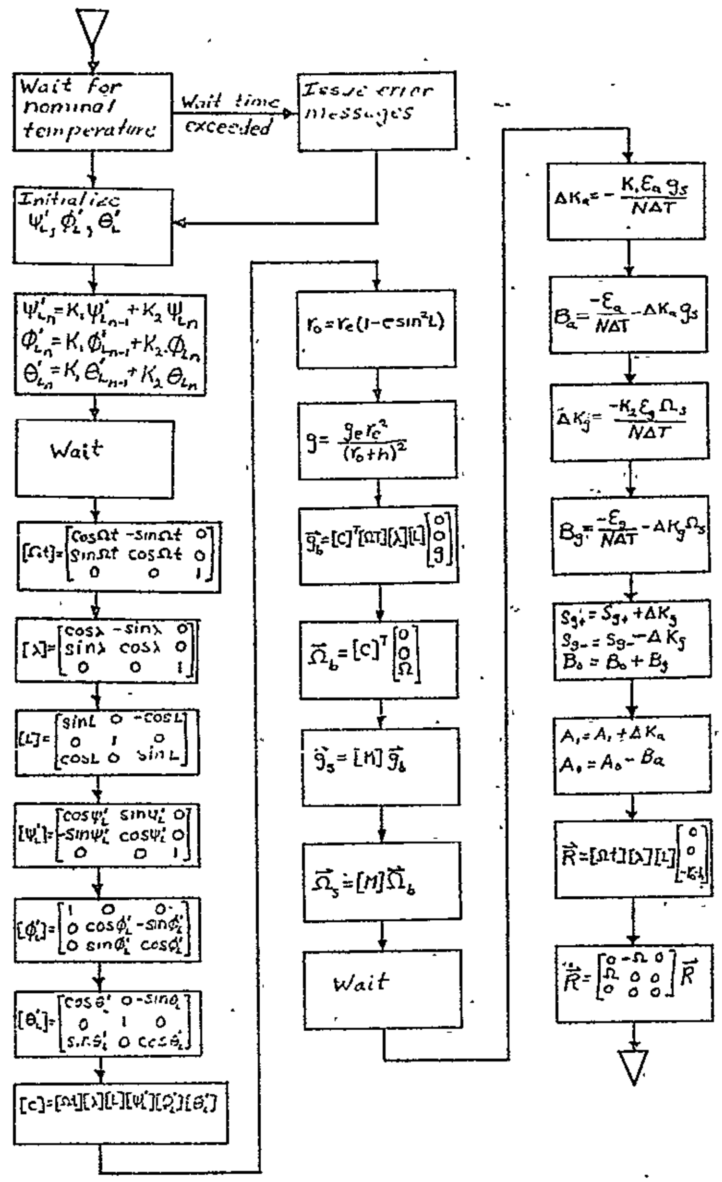


Figure 5-42 Strapdown Start-Up Program (SDSU)

Figure 5-43 Pre-Calibration and Calibratic Computations time the optical attitude inputs are filtered. The direction cosine matrix [C] is then generated by computing the individual rotational transformation matrices from inertial coordinates to body axis coordinates, forming the matrix product. The gravitational and earth rate vectors are then formed and transformed to the body axis system. The gravity and earth rate components along each instruments sensitive axis are then formed. A wait is then programmed during which time the SD64 program accumulates E_a and E_g. During this time, the direction cosine matrix and the gravitational and earth rate vectors are continously updated. At the end of the wait period, the bias and scale factor errors for each instrument are computed and corrections made to the initialized valves. The inertial position and velocity vectors $\bar{R}$ and $\dot{\bar{R}}$ are computed and initialized.

Figure 5-44 is a flow diagram of the main strapdown inertial navigation program. This program is executed at a 64 times per second iteration rate. First bias and scale factor corrections are applied to each accelerometer output. A flag is then tested to determine if the system is in the process of being calibrated or in flight program usage. During calibration, ε_a and ε_g for each instrument is computed and the program exited. If the program is being executed for the flight mission, each instrument output is filtered to provide fault detection inputs. Body axis outputs $\bar{O}_a$ and $\bar{O}_g$ are then generated by multipying each set of

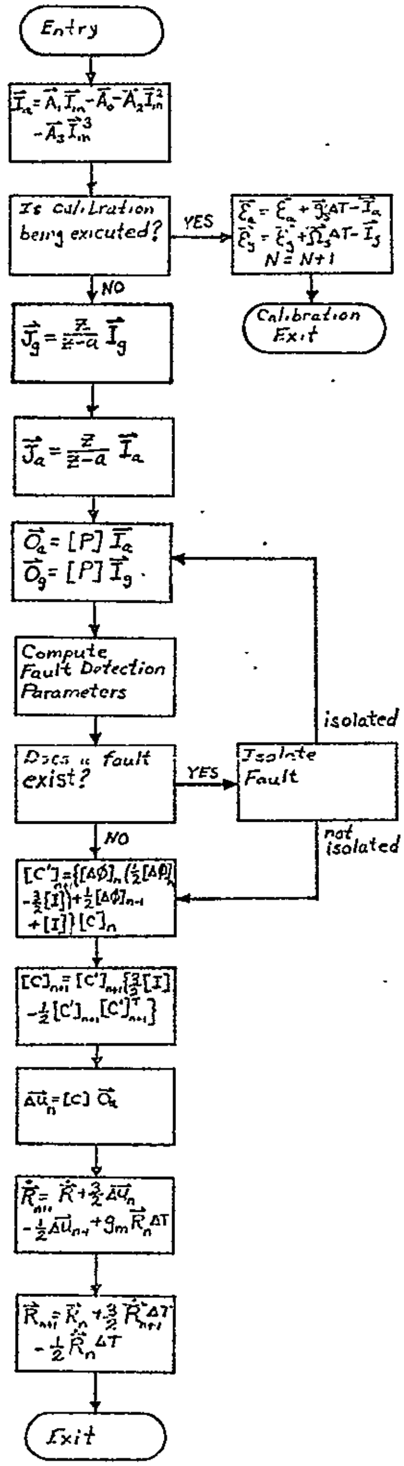


Figure 5-44 Main Strapdown Inertial Navigation Program

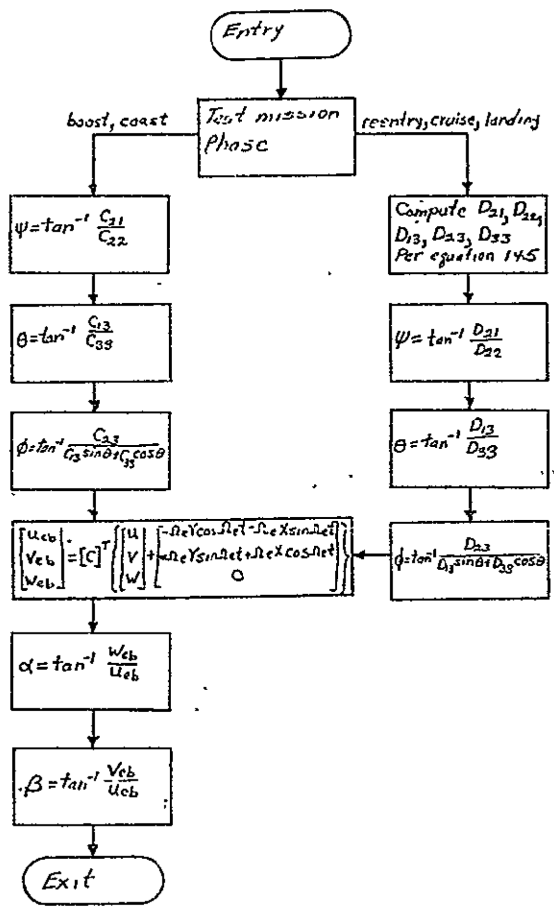


Figure 5-46 Strapdown Attitude Reference Generation

instrument outputs by the appropriate [P] matrix. Fault detection parameters are then computed and tests performed to determine if a fault exists. If a fault does exist, fault isolation computations are performed. If the fault can be isolated, a new [P] matrix is selected and a new set of body axis outputs computed. The body axis gyro outputs are then used to generate the updated direction cosine matrix. The accelerometers body axis outputs are then rotated to inertial space using the direction cosine matrix and integrated twice to form $\dot{\bar{R}}$ and $\bar{R}$.

Figure 5-45 shows a more detailed flow diagram of the failure detection and isolation computations required of the main strapdown inertial navigation program. The program indicated by Figure 5-45 must be executed twice, once for detection and isolation of gyro failures and once for accelerometer failures. First a test is made to determine if all six instruments are operating or if previous tests have reduced the number to 5 or 4. If six instruments are operating F_1, F_6, G_1 and G_6 are computed and G_1 and G_6 tested. A fault is detected if either G_1 or G_6 is a 1. If a fault is detected, direct measurements on each instrument are made. These direct measurements include power supply voltages, output magnitudes to determine if the instrument output is zero or full scale, measurements of rotor speed for the gyros, and comparison with the orbiter outputs if the mission phase is boost. If a fault cannot be isolated by direct measurements, then all 15 values of G_i are computed and tested to determine if their values match one of the patterns of Figure 5-37. If a fault cannot be isolated, an appropriate error message is issued. If a fault is isolated a new [P] martix is selected and an error message issued.

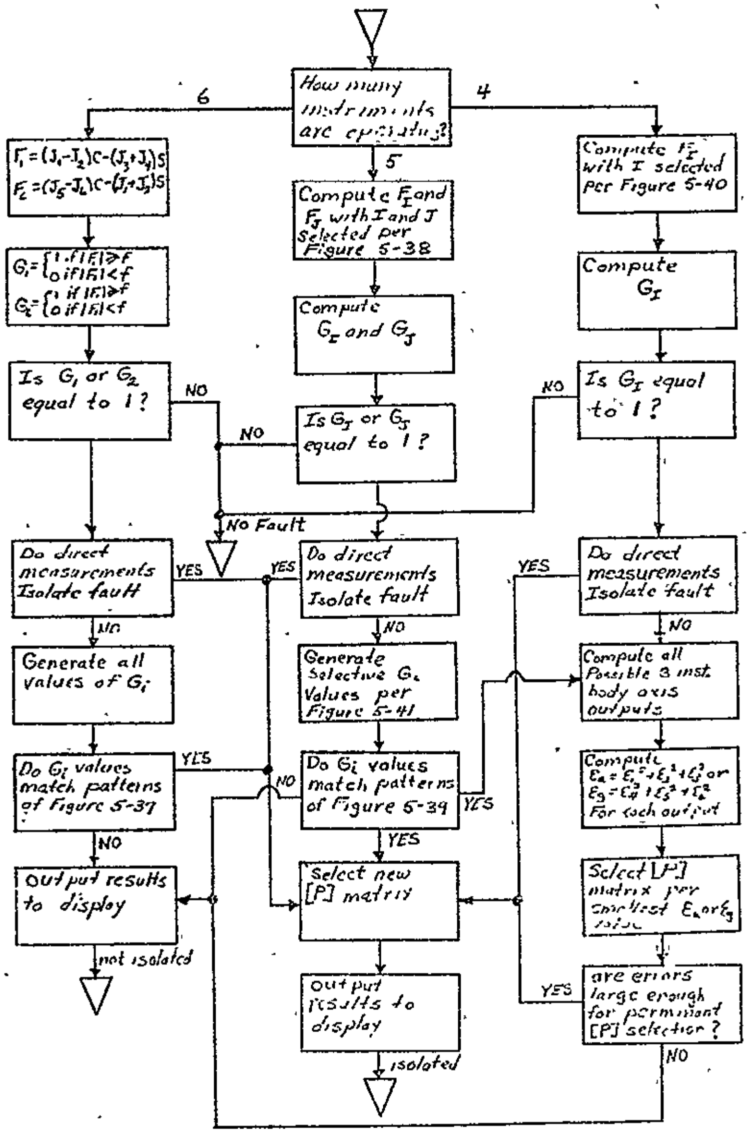


Figure 5-45 Error Detection and Isolation

If the original testing showed only 5 instruments operating, then two values of F are evaluated. The values of F evaluated are chosen according to which instruments are operating as described by Figure 5-38. Two values of G are constructed from the two F values and tested. If either value is 1, a fault has been detected and direct measurements on the instrument are first performed to isolate the fault. If the fault cannot be isolated in this manner, 5 values of G are computed. The 5 values of G computed are dependent upon which instrument has been previously isolated as indicated by Figure 5-39. If all six values of G are one, a multiple failure is indicated and a branch to the four instrument fault isolation procedure is made. If the failed instrument is isolated, a new P martix is selected and an error message generated.

If the initial testing showed only 4 instruments operating, one value of F is generated according to Figure 5-40. A value of G is computed from the value of F and tested. If the value of G is 1 a fault is detected. Fault isolation consists of first performing direct measurements upon the 4 remaining instruments. If the fault cannot be isolated by direct measurements, then all possible three instrument outputs are transformed to body axis and compared with the trend analysis outputs. The errors between trend analysis and instrument outputs are used to select a [P] matrix and isolate the failed instrument.

Figure 5-46 is the flow diagram of the program (SD32) required to generate attitude reference data from the Strapdown system. The program will be run at a 32 per second iteration rate. On entering the program, a test is made to determine which mission phase exists. If the mission phase is boost or coast the Euler angle attitudes θ, ψ, and ϕ with respect to inertial space are generated. If the mission phase is reentry cruise or landing, the direction cosine matrix terms from body to locally level coordinates are computed and the Euler angle attitudes θ, ψ, and ϕ with respect to local level coordinates are evaluated. The velocity of the vehicle with respect to the earth is then computed and angle of attack α and side slip angle β generated.

Figure 5-47 is a flow diagram of the program (SD16) required to generate trend analysis values for fault isolation of a failed instrument with only four instruments operating. The program will be executed at a 16 per second iteration rate. Upon entering the program dynamic pressure Mach number and aerodynamic coefficients are computed. Then the aerodynamic and airbreathing engine forces and torques are computed. The control surface effectiveness is then computed, followed by computations of the aerodynamic control torques. The vehicles mass and inertia and correction forces and torques are computed to make the strapdown system and trend analysis outputs converge. The vehicle forces and torques are then summed and the linear and angular accelerations formed. These are integrated and the angle of attack, side slip angle and altitude computed.

NOT REPRODUCIBLE

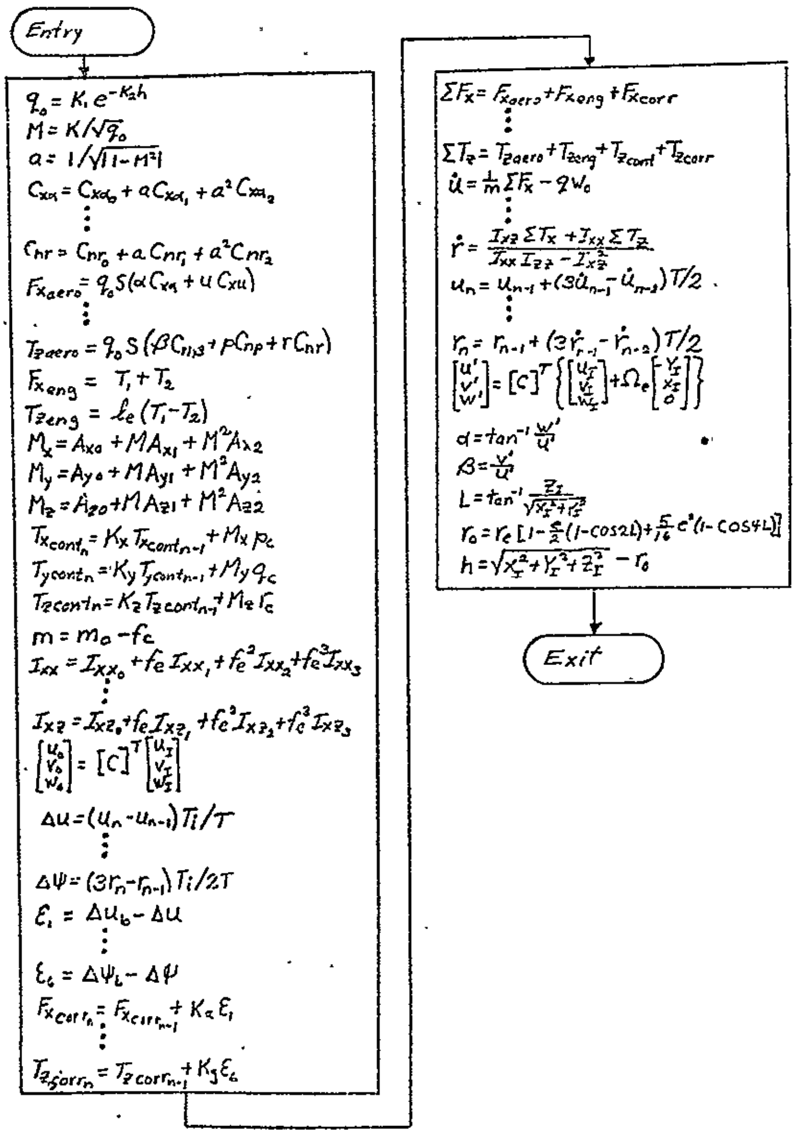


Figure 5-47 Strapdown System Trend Analysis Program

Figure 5-48 is a flow diagram of the program (SD02) required to form navigation parameters and achieve strapdown system updating. The program is run at a 2 per second iteration rate. Upon entry to the program the local gravitational constant g_m is computed. Mission phase is then tested and the program exited if the mission phase is boost. If the mission phase is coast, reentry, cruise, or landing the vehicle position and velocity is transformed to an earth fixed coordinate frame. Geocentric latitude and longitude and the vehicle's altitude is then computed. The vehicle's velocity is then transformed to a locally level coordinate system. Heading and vertical velocity are then generated. Geodetic latitude and longitude is computed. The [Q] matrix is then generated by forming the required partial derivitives. The $\overline{d\lambda}$ vector is evaluted and the strapdown navigation integrators are updated.

Figure 5-49 is a flow diagram of the program (SD01) required to control the temperature of the strapdown gyros and accelerometers. This program is executed at a one per second iteration rate. Upon entering the program,indexes are set up to pick up the temperatures of each instrument one at a time. These temperatures are compared with two limit values. The instruments heater is turned on if the temperature is below a lower limit and turned off if the temperature is above an upper limit. Each instrument is tested in the same manner.

NOT REPRODUCIBLE

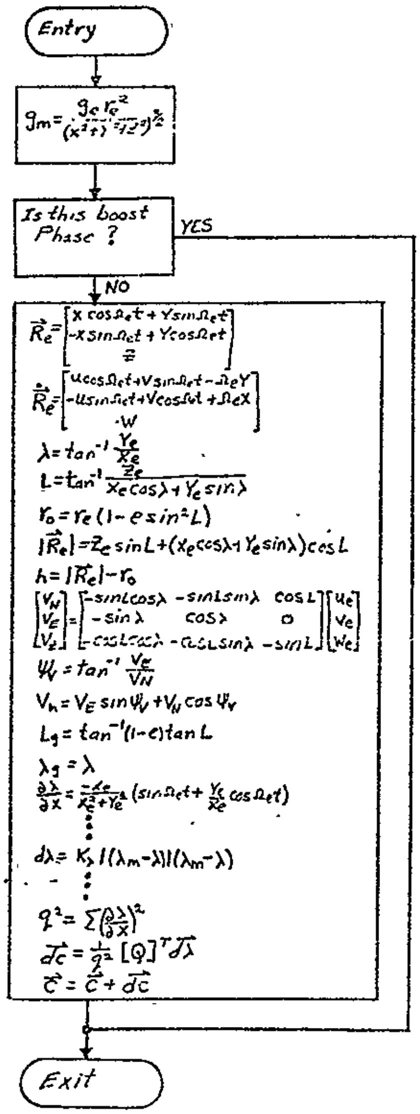


Figure 5-48 Strapdown Navigation Parameters and Updating Program

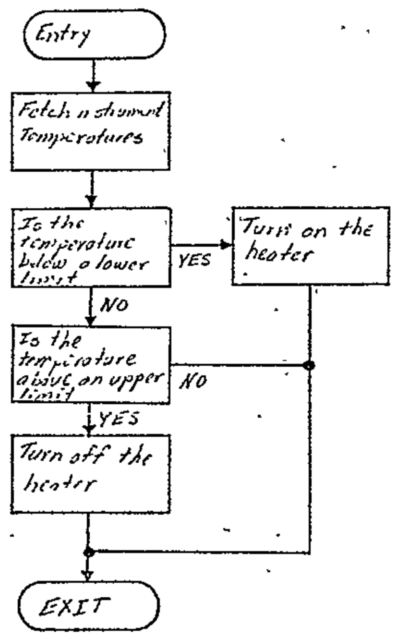


Figure 5-49 Strapdown Sensor Heater Control Program

5.4.2 Air Data System

The air data system converts the air data sensor pressures and temperature to altitude, velocity, angle of attack, and side slip angle data for use in updating the strapdown inertial navigation system and for various pilot displays. The raw data from the pressure sensors must be corrected for installation anomalies. This correction is achieved using the formulas

$$P_S' = a_{S0} + a_{S1}\bar{P}_S + a_{S2}\bar{P}_S^{\,2} + a_{S3}\bar{P}_S^{\,3} \qquad (1)$$

$$P_P = a_{P0} + a_{P1}\bar{P}_P + a_{P2}\bar{P}_P^{\,2} + a_{P3}\bar{P}_P^{\,3} \qquad (2)$$

$$P_{\alpha T} = a_{\alpha T0} + a_{\alpha T1}\bar{P}_{\alpha T} + a_{\alpha T2}\bar{P}_{\alpha T}^{\,2} + a_{\alpha T3}\bar{P}_{\alpha T}^{\,3} \qquad (3)$$

$$P_{\alpha B} = a_{\alpha B0} + a_{\alpha B1}\bar{P}_{\alpha B} + a_{\alpha B2}\bar{P}_{\alpha B}^{\,2} + a_{\alpha B3}\bar{P}_{\alpha B}^{\,3} \qquad (4)$$

$$P_{\beta R} = a_{\beta R0} + a_{\beta R1}\bar{P}_{\beta R} + a_{\beta R2}\bar{P}_{\beta R}^{\,2} + a_{\beta R3}\bar{P}_{\beta R}^{\,3} \qquad (5)$$

$$P_{\beta L} = a_{\beta L0} + a_{\beta L0}\bar{P}_{\beta L} + a_{\beta L2}\bar{P}_{\beta L}^{\,2} + a_{\beta L3}\bar{P}_{\beta L}^{\,3} \qquad (6)$$

Where P'_s is corrected static pressure, P_p corrected pitot pressure, $P_{\alpha T}$ corrected pressure from the top angle of attack probe slot, $P_{\beta R}$ corrected pressure from the right slot, and $P_{\beta L}$ corrected pressure from the left slot. The "barred" pressures are raw sensor data and the a's are calibration constants. Static pressure must be corrected for speed and angle of attack. This correction is achieved using the formula

$$P_S = P_S'(1 + a_{M1}M + a_{M2}M^2 + a_{M3}M^3)(1 + a_{\alpha 1}\alpha + a_{\alpha 2}\alpha^2 + a_{\alpha 3}\alpha^3) \qquad (7)$$

where M is speed expressed in Mach number and α is computed angle of attack.

True dynamic pressure, Q, is obtained from:

$$Q = P_p - P_s \tag{8}$$

True dynamic pressure is a measure of the local change in pressure caused by the vehicle distrubance upon the air mass.

Altitude is computed from one of two formulas dependent upon the vehicle altitude. These two formulas are

$$H = 145447 - 76188.9\, P_s^{.19025} \quad for \ -2000 < H < 36089 \tag{9}$$

and

$$H = 75611.7 - 20805.9 \ln P_s \quad for \ 36089 < H \tag{10}$$

Mach number is computed from one of two formulas dependent upon the value of Mach number. These two formulas are

$$M = \left[5\left(\frac{Q}{P_s}+1\right)^{1/3.5} - 5\right]^{1/2} \quad for \ M \leq 1 \tag{11}$$

and

$$M^{2.8} - f_1 M^2 + f_2 = 0 \quad for \ M > 1 \tag{12}$$

where

$$f_1 = .1342\left(\frac{Q}{P_s}+1\right)^{0.4} \tag{13}$$

and

$$f_2 = .01917\left(\frac{Q}{P_s}+1\right)^{0.4} \tag{14}$$

If M is greater than 1 then equation 12 must be solved for M. A solution for M is achieved by estimating the value of M and improving the estimate using the formula

$$M = M_o - \frac{M_o^{2.8} - f_1 M_o^2 + f_2}{2.8\, M_o^{1.8} - 2 f_1 M_o} \qquad (15)$$

where Mo is the estimated value of M. Normally the estimated value of M will be the previous value of M determined on the last iterative pass through the program. On the first pass through the program during reentry an initial estimate of Mo ≈ 4 will be made. During the initial pass through the program equation 15 will be solved several times using the former solution as the estimated value of Mo for each subsequent solution. Whenever $|M - Mo| > .01$ multiple solutions of equation 15 will be used. The number of multiple solutions used during any one program pass will be limited to five.

An air data computer generally has several air speed outputs. For the space shuttle booster, it is assumed that indicated air speed, equivalent air speed and true air speed will be computed. Each of these speed calculations are employed for different purposes in the mission.

Calibrated air speed, V_c, is computed in a manner very similar to that employed in obtaining Mach number. If V_c is below 661.47 knots it is computed from the formula

$$V_c = 295.808 \left[\left(\frac{Q}{29.921} \right)^{1/3.5} - 1 \right]^{1/2} \qquad (16)$$

and if V_c is greater than 661.47 knots it is obtained by solving the equation

$$V_c^{2.8} - g_1 V_c^2 + g_2 = 0 \qquad (17)$$

where

$$g_1 = 1042\left(\frac{Q}{29.921} + 1\right)^{0.4} \tag{18}$$

and

$$g_2 = 64.6 \times 10^6\left(\frac{Q}{29.921} + 1\right)^{0.4} \tag{19}$$

The calibrated air speed at which the booster will stall is the same at any altitude or temperature. Thus V_c is compared against the stall speed constant, V_{cs}, and a warning issued if the vehicle approaches stall.

Equivalent air speed is computed from the formula

$$V_e = KM\sqrt{P_s} \tag{20}$$

Vehicle safety limits are generally expressed as tabulations of equivalent airspeed versus temperature and Mach number versus temperature. These limits will be computed using the formulas

$$V_{e_{LIM}} = a_{V0} + a_{V1} T_i + a_{V2} T_i^2 + a_{V3} T_i^3 \tag{21}$$

and

$$M_{LIM} = a_{M0} + a_{M1} T_i + a_{M2} T_i^2 + a_{M3} T_i^3 \tag{22}$$

If either V_e or M exceed their computed limit value a warning will be issued. T_i is the temperature sensor input.

True air speed is computed from the formula

$$V_T = KM\sqrt{\frac{T_i}{1 + .2M^2}} \tag{23}$$

True air speed is displayed to the pilot and used for inertial navigation updates.

Indicated angle of attack and side slip angles are computed from the formulas:

$$\alpha_i = K \frac{P_{\alpha T} - P_{\alpha B}}{P_{\alpha T} + P_{\alpha B}} \tag{24}$$

and

$$\beta_i = K \frac{P_{\beta R} - P_{\beta L}}{P_{\beta R} + P_{\beta L}} \tag{25}$$

Indicated angle of attack and side slip angle are then corrected for installation anomalies by the formulas:

$$\alpha = K_{\alpha 0} + K_{\alpha 1}\alpha_i + K_{\alpha 2}\alpha_i^2 \tag{26}$$

and

$$\beta = K_{\beta 0} + K_{\beta 1}\beta_i + K_{\beta 2}\beta_i^2 \tag{27}$$

Figure 5-50 is a flow diagram of the program (ADCP) used to generate air data outputs. This program is run at an 8 per second iteration rate.

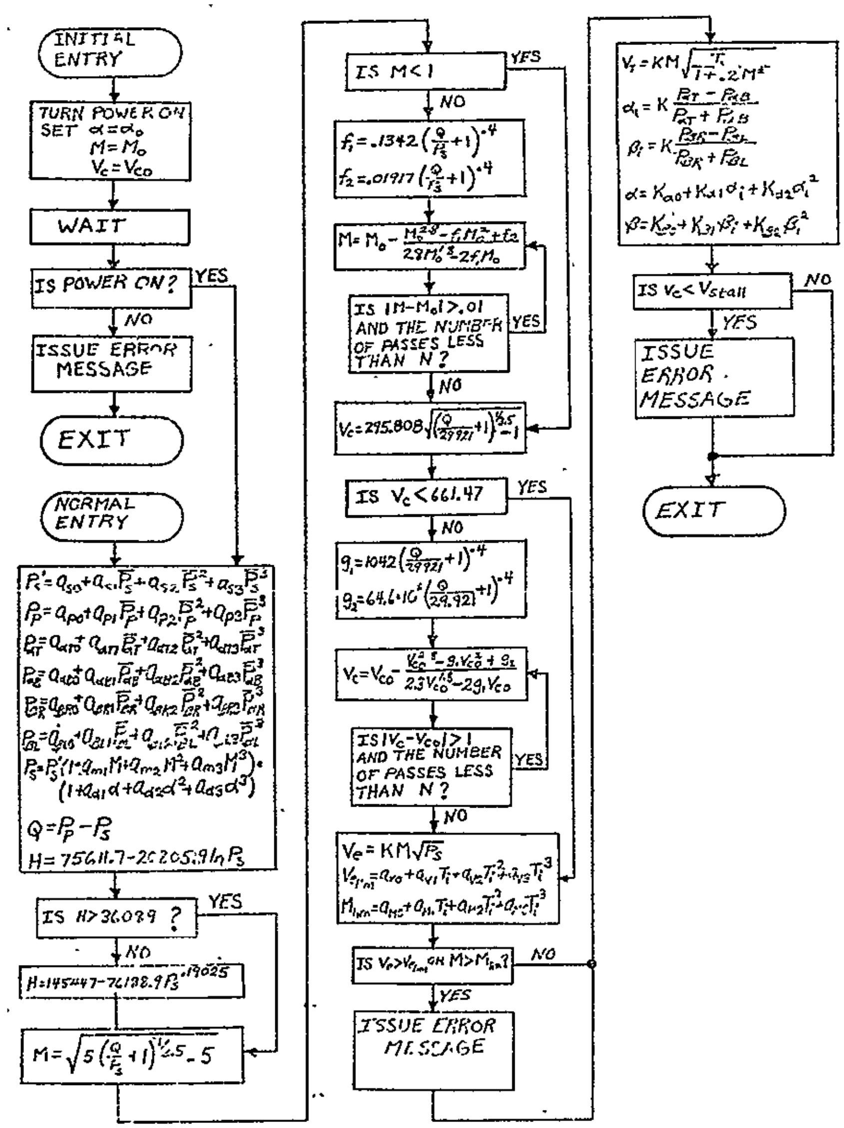


Figure 5-50 Air Data Computer Program

5.4.3 Magnetic Flux Gate Compass

The magnetic flux gate compass provides a measure of the angle between the vehicle body X axis and the earth's north magnetic pole. The following tasks are performed by the DMS system in monitoring and controlling the magnetic flux gate compass

1. Ground checkout
2. In-flight monitoring
3. Temperature control
4. Magnetic heading computations
5. Magnetic declination correction
6. Gyro torquing

Magnetic heading, ψ_{mh}, is delivered to the DMS as two data input items ν_1 and ν_2 which are related to ψ_{mh} by the equations

$$\nu_1 = K \sin \psi_{mh} \tag{1}$$

and

$$\nu_2 = K \cos \psi_{mh} \tag{2}$$

The angle ψ_{mh} is thus determined from

$$\psi_{mh} = \tan^{-1} \frac{\nu_1}{\nu_2} \tag{3}$$

This angle must be corrected for magnetic declination. Magnetic declination is a complex empirical function of latitude and longitude. Near the poles magnetic declination experiences significant daily, seasonal and random changes resulting in unpredictable declination values. Because of the complexity of the magnetic declination function of latitude and longitude correction values must be generated in the DMS by table look up. Because of the unpredictability of magnetic declination in the polar regions not all of the earth's surface can be covered. However, over 85% of the earth's surface can be covered to within $.1^{\circ}$ accuracy using a table having 2500 entries or over 70% of the earth's surface covered to 1° accuracy with a table having 250 entries. It is assumed that two look up tables will be used, one containing 100 points covering the expected mission area to $.1^{\circ}$ accuracy. One hundred points will allow a square coverage area having approximately 2500 mile sides. The second look up table will contain 250 entries generating 1° accuracy over 70% of the earth's surface. The magnetic flux gate compass will not be used in the polar regions due to its inaccuracy.

In order to define the table loop up operation the following parameters must be defined:

λ	longitude of the space shuttle location
L	latitude of the space shuttle location
L_n	North latitude limit of coarse table entries
L_{min}	South latitude limit of five table entries
L_{max}	North latitude limit of five table entries
λ_{min}	West longitude limit of five table entires
λ_{max}	East longitude limit of five table entries
ΔL	latitude difference between table entries
$\Delta \lambda$	longitude difference between table entries
L_o	closest table entry latitude value less than
λ_o	closest table entry longitude value less than

There are different values of ΔL, $\Delta\lambda$, L_0, and λ_0 for each table and the value of L_0 and λ_0 are dependent upon the values of L and λ respectively. If the magnetic declination is a function of λ and L it can be represented by the functiorial notation

$$\psi_d = f(\lambda, L) \qquad (4)$$

The differential of ψ_d is given by:

$$d\psi_d = \frac{\partial f(\lambda, L)}{\partial \lambda} d\lambda + \frac{\partial f(\lambda, L)}{\partial L} dL \qquad (5)$$

The value of ψ_d at the point λ, L is determined by an approximate application of equation 5 which is

$$\psi_d = f(\lambda_0, L_0) + \frac{f(\lambda_0 + \Delta\lambda, L_0) - f(\lambda_0, L_0)}{\Delta\lambda}(\lambda - \lambda_0)$$
$$+ \frac{f(\lambda_0, L_0 + \Delta L) - f(\lambda_0, L_0)}{\Delta L}(L - L_0) \qquad (6)$$

The heading angle ψ_h is then determined from

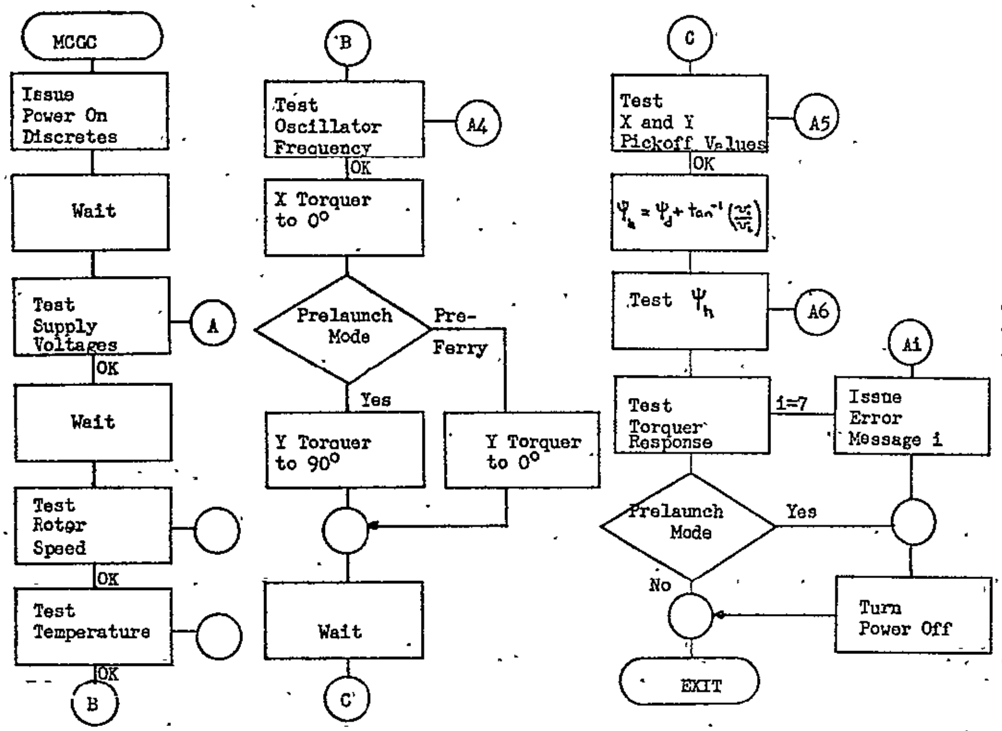
$$\psi_h = \psi_{mh} + \psi_d \qquad (7)$$

Before equation 6 is employed tests must be made to determine which table should be used.

Figure 5-51 is a flow diagram of the magnetic flux gate compass ground checkout program (MCGC). Upon entry to the program the ac and dc power discretes are issued. A wait is then programmed allowing sufficient delay for power turn on transients to decay and the supply voltages to the magnetic flux gate compass system then tested. An additional wait is then programmed to allow for the gyro rotor speed and temperature to reach nominal value. The rotor speed, temperature and oscillator frequency are then tested. The X torquer loop is then commanded to 0°. A test is then made to approximately determine the vehicle attitude by testing mission phase for prelaunch or preferry. If the mission phase is prelaunch, the Y torquer loop is commanded to 90°, if preferry to 0°. A wait is then programmed to allow for torquer loop transient decay. The X and Y torquer loop pickoff values are then read and tested. The compass outputs $\mathcal{V}_1$ and $\mathcal{V}_2$ are read and vehicle heading computed and tested. The local magnetic declination and actual vehicle heading are input through the keyboard for this test. The gyro torquer loop response is then tested by issuing a set of step commands and reading the torquer loop pickoff outputs at selected time points and testing against stored response data. Mission mode is then tested and the power to the magnetic flux gate compass commanded off if the mode is prelaunch. If any of the tests fail, error messages are issued and the power turned off. This program is run at a 1 per second iteration rate.

Figure 5-52 is a flow diagram of the magnetic flux gate compass temperature control and monitoring program (MCTM). This program is scheduled for both prelaunch and preferry checkout and during the flight modes using the magnetic flux gate compass. The program is run at a 1 per second iteration rate. A temperature control cycle is run with every entry to the program. This cycle consists of comparing the measured temperature with an upper and lower temperature value, T_1 and T_0.

Figure 5-51 Magnetic Flux Gate Compass Checkout (MCGC)



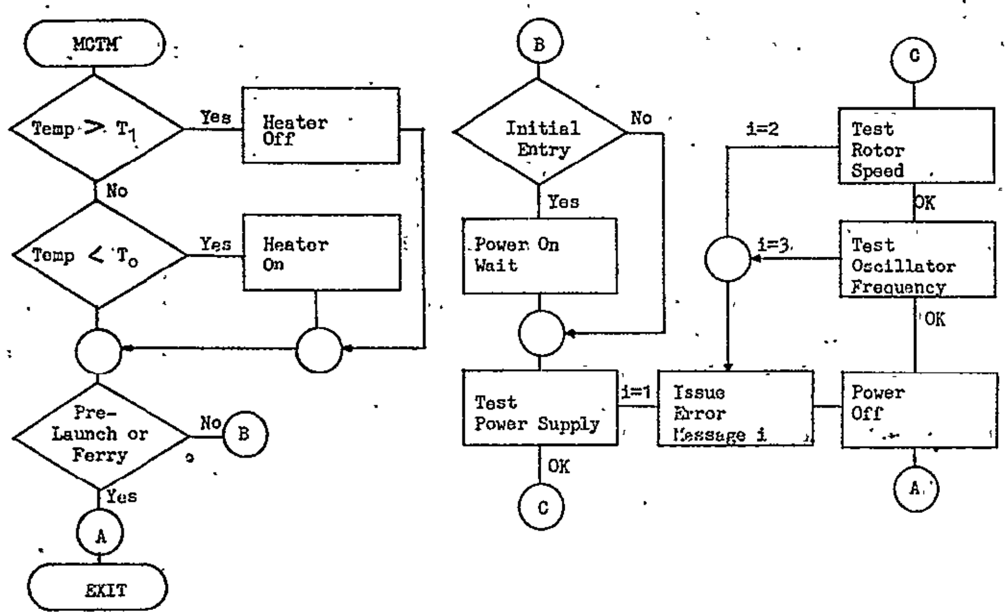


Figure 5-52 Magnetic Flux Gate Compass Temperature Control and Monitoring (MCTM)

If the temperature is above the upper level T_1, the heater is turned off and if below the lower level T_0, the heater is turned on. The mission mode is then tested and if the mode is prelaunch or preferry, the program is exited. If it is a flight mode a test is made to determine if this is an initial entry to the program. If it is an initial entry, the power is turned on and a wait programmed to allow the gyro rotor speed and temperature to achieve nominal values. Otherwise the power, rotor speed and oscillator frequency is tested. If any test fails an error message is issued and the power commanded off.

Figure 5-53 is the flow diagram of the magnetic flux gate compass heading program (MCHP) run at a 4 per second iteration rate. This program is assigned the tasks of slaving the compass gyro to the inertial navigation gyro outputs and computing the vehicle heading. In order to test the gyro torquing loops a model output is computed and compared with the gyro pickoff outputs. If the comparison fails an error message is issued and the power to the magnetic flux gate compass is turned off. With proper operation determined, the values of Θ and $\emptyset$ from the strapdown inertial navigation systems are issued as commands to the two torquing loops. The model outputs for the next pass comparison are then generated using the formulas for the pitch torquing loop of:

$$\mathcal{E}_n = \Theta - \Theta_{PO_{n-1}} \tag{8}$$

$$T_n = K_1 T_{n-1} + K_2 \mathcal{E}_n \tag{9}$$

$$\Theta_{PO_n} = \Theta_{PO_{n-1}} + K_3 T_n \tag{10}$$

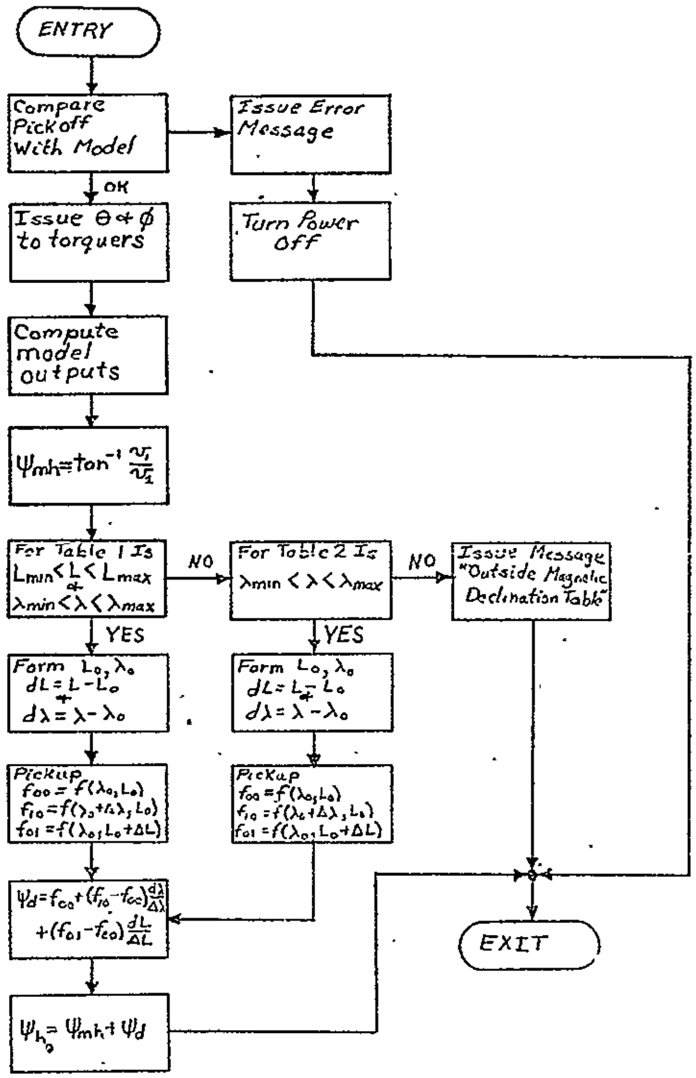


Figure 5-53 Magnetic Flux Gate Compass Heading Program (MCHP)

where θ_{po} is the modeled pickoff output, and K_1, K_2 and K_3 are constants defined by the torquing loop dynamics. Magnetic heading, ψ_{mh}, is then computed according to equation 3. A test is then made to determine if the vehicle latitude and longitude is within the fine magnetic declination table. Actual latitude and longitude, λ and L , are tested against the table limits of λ_{max}, λ_{min}, L_{max}, and L_{min} in making this determination. If the vehicle is determined to be outside the fine table area, a test is made to determine if it is in the coarse table area. It is assumed that the coarse table covers the entire earth except for the two polar regions. Thus only a test of latitude is required. If the vehicle position is not covered by either table a message is issued. The magnetic flux gate compass is maintained operable so that outputs will automatically be generated if the vehicle enters either table area.

Upon determining that the vehicle is within a table area the value of magentic declination for the vehicle position must be computed. The same table look up procedure is used for the fine and coarse tables. Values of $f(\lambda, L)$ are stored in sequential memory locations starting at location A_o with the following order:

$$f(\lambda_{min}, L_{min})$$
$$f(\lambda_{min}+\Delta\lambda, L_{min})$$
$$f(\lambda_{min}+2\Delta\lambda, L_{min})$$
$$\vdots$$
$$f(\lambda_{max}, L_{min})$$
$$f(\lambda_{min}, L_{min}+\Delta L)$$
$$\vdots$$
$$f(\lambda_{max}, L_{max})$$

Using the bracket notation [X] to denote the largest integer less than X then

$$L_0 = [L/\Delta L]\Delta L$$
$$\lambda_0 = [\lambda/\Delta\lambda]\Delta\lambda \qquad (11)$$

and the address of $f(\lambda_0, L_0)$ is

$$A(f_{00}) = A_0 + \left[\frac{\lambda - \lambda_{min}}{\Delta\lambda}\right] + B\left[\frac{L - L_{min}}{\Delta L}\right] \qquad (12)$$

where

$$B = \left[\frac{\lambda_{max} - \lambda_{min}}{\Delta\lambda}\right] + 1 \qquad (13)$$

and the addresses of $f(\lambda_0+\Delta\lambda, L_0)$ and $f(\lambda_0, L_0+\Delta L)$ are:

$$A(f_{10}) = A(f_{00}) + 1 \qquad (14)$$

$$A(f_{01}) = A(f_{00}) + B \qquad (15)$$

The value of ψ_d is then formed according to equation 6 and ψ_h according to equation 7.

5.4.4 TACAN Receiver

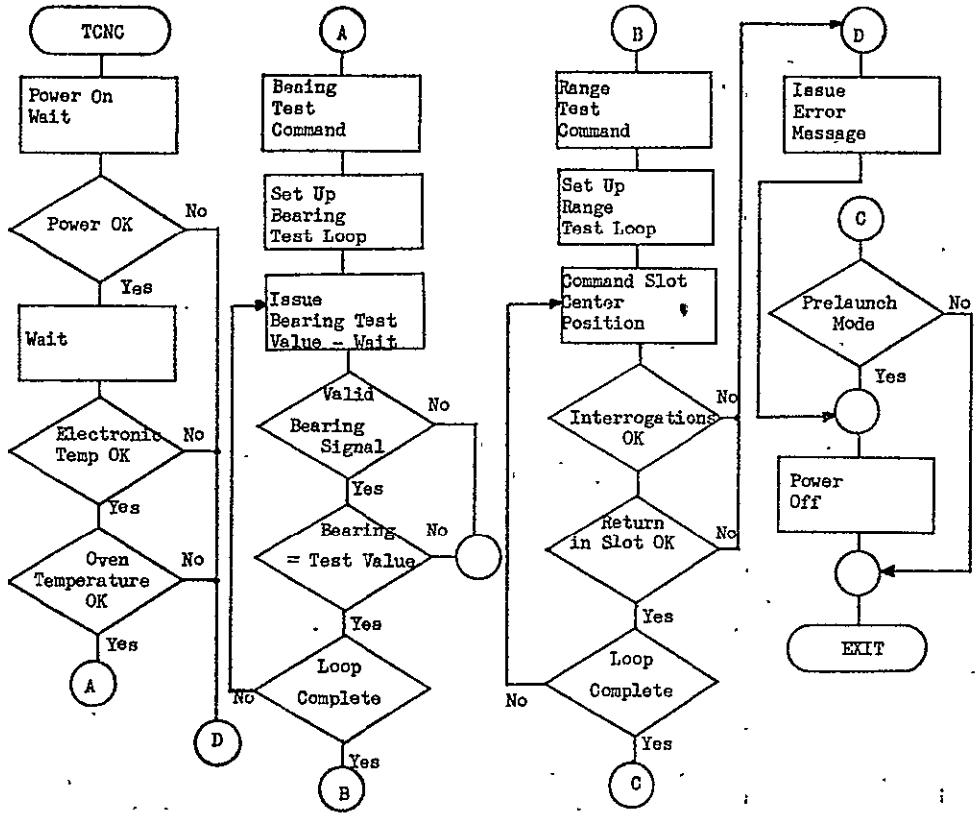
The TACAN receiver is a major navigational source during cruise flight. Its output is used to continously update the inertial navigation system and is a primary navigation source. The navigation data produced by the TACAN receiver is bearing and slant range to a TACAN ground station. This raw navigation data is converted to vehicle latitude, longitude and horizontal velocity. The tasks required of the DMS by the TACAN are:

1. Checkout
2. Station identification
3. Bearing determination
4. Slant range determination
5. Horizontal range determination
6. Return Pulse tracking
7. Navigation Data Generation

Figure 5-54 is a flow diagram of the TACAN ground checkout program (TCNC). Upon entering the program the power is turned on and a wait programmed to allow for power transients to decay before testing the supply power. Another wait is then programmed to allow for the temperature of the crystal oven and electronics to stabilize before testing. The bearing test command is then issued and a programming loop initialized to test the TACAN bearing system at several bearing values. The loop consists of issuing a bearing test value, waiting a length of time for the bearing electronics to phase lock on the test signal, testing for the existance of the bearing valid discrete, comparing the returned bearing value with issued value, and testing for loop completion. The distance measuring equipment is tested in a manner similar to the bearing equipment test. The range test discrete is issued to command the distance measuring equipment into a test configuration and a program loop initialized. Within the loop the slot center position is commanded to a preset value and a wait programmed. The number of interrogations transmitted and the number of replies received are then tested against a prestored value. The position of the slot center and the position of the return within the slot is tested. After completion of the distance measuring equipment testing loop a mission mode test is performed. If the mission mode is prelaunch, power is turned off. The failure of any test causes an error message to be issued and the power to be turned off. The program is run at a 1 per second iteration rate.

Figure 5-55 is the TACAN monitoring and temperature control program (TCNM). This program is scheduled at a 1 per second iteration rate whenever the TACAN checkout program or the TACAN Navigation program is scheduled. The program has two entries, a normal entry and an initial entry point. The initial entry is not used with the TACAN checkout program. Entering the program at the initial entry causes the power on discretes to be issued followed by a programmed wait to allow for power transients to decay. The initial entry program then joins with the normal entry program and the supply voltages and electronics temperature are tested. The crystal oven temperature is then tested against extreme bounds to determine a possible failure in the heater control. If any of these tests fail, an error message is generated and the TACAN power turned off. If the tests indicate

Figure 5-54 TACAN Checkout (TCNC)



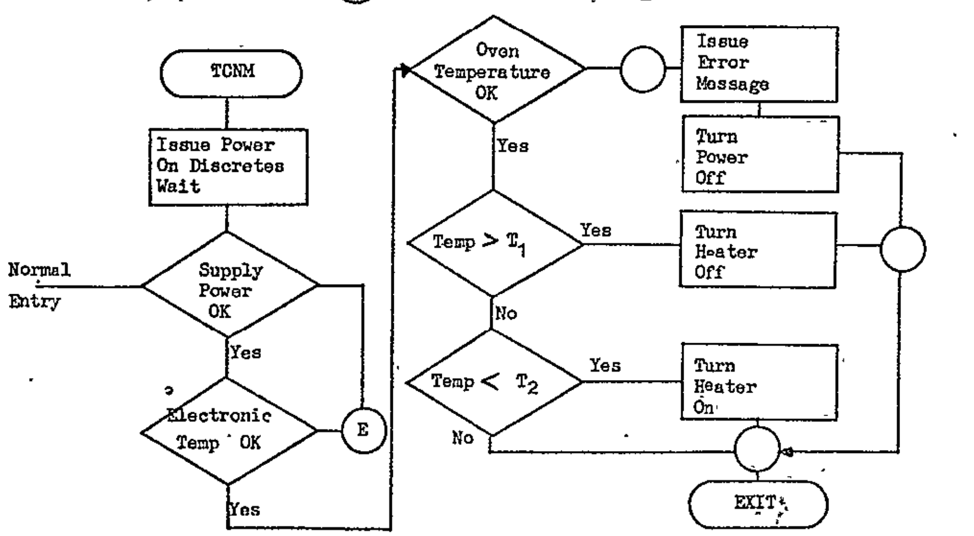


Figure 5-55 TACAN Monitoring and Temperature Control (TCNM)

the TACAN equipment is properly functioning, oven temperature control programming is executed. Temperature is controlled by turning the heaters off if the oven temperature is above a value T_1 and on if below a value T_2.

Figure 5-56 is a flow diagram of the TACAN navigation program (TCNN) which is run at a four per second iteration rate. Upon entering the program, a test is made to determine if the channel select knobs on the receiver have changed since the last pass through the program. If they have changed, the program is forced to the search mode. The channel number is then used to select station parameters from a table. The data stored for each station in the table is:

L_s latitude of the station

λ_s longitude of the station

h_s altitude of the station above sea level

ψ_d magnetic declination at ground station

plus the channel number. It is assumed that a mission plus alternatives for the booster will not require more than 30 stations in the table. There may be more than one station in the table with the same channel number. To locate the desired table entry a search through all table entries is made for those stations having the same channel number as that selected by the receiver. If the search determines that there are no table entries having the receiver channel number it is assumed that the pilot is in the process of selecting a channel with the channel selector setting on some intermediate value when sampled.

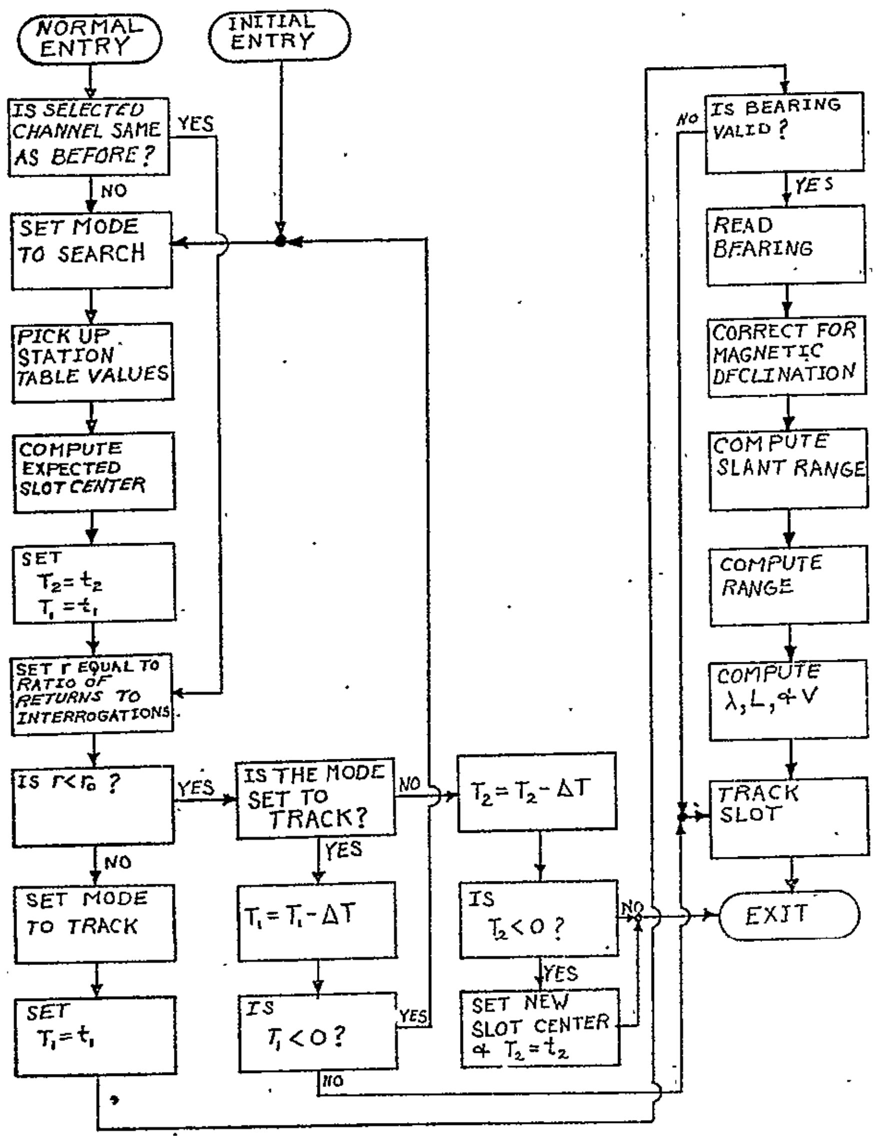


Figure 5-56 TACAN Navigation Program (TCNN)

The TACAN ground station receives a pulse from the aircraft and retransmitts the pulse. The time between the transmission of the original pulse by the TACAN set on the vehicle and the reception of the return slot is given by:

$$T_S = \frac{2d}{C} + T_d \qquad (5)$$

where C is the speed of light and T_d is the time between the ground station reception of the pulse and its retransmission.

Two timers, T_1 and T_2 are also set in the program branch executed after finding a new receiver channel has been selected. Testing is then performed which generates a three way branch. One branch is taken if the operating mode is search and the ratio of return pulses to interrogation pulses is less than a constant value, r_0. If this branch is taken the receiver is in the process of looking for the return time slot. The timer T_2 controls how long the receiver searches each possible slot position before trying a new slot position. If the return to interrogation pulse ratio is not found to be larger than r_0 by the time T_2 seconds have elapsed a new slot position is commanded. If T_s is the original slot position as computed by equation 5 then the sequence of slot positions is T_s, $T_s + \Delta T$, $T_s - \Delta T$, $T_s + 2\Delta T$, $T_s - 2\Delta T$, ... etc. A maximum and minimum slot position limit is tested and only those slot positions between the limits are issued to the TACAN receiver. If the entire range is covered without finding a sufficient return to interrogation pulse ratio an error message is issued and the process is restarted.

If the operation mode is track and an insufficient return to interrogation ratio is found for a period of T_1 seconds the search mode is entered. If the return to interrogation pulse ratio is greater than r_o the mode is set to track and the T_1 counter is reset. A test is performed to determine if bearing is valid. If bearing is valid both range and bearing is available and all TACAN navigation outputs can be computed. The bearing angle is read from the TACAN receiver and corrected for magnetic declination by computing:

$$\psi_h = \psi_m + \psi_d \tag{6}$$

where ψ_m is the bearing angle from the TACAN receiver and ψ_d is the ground station magnetic declination angle.

Slant range is computer from the formula

$$d_s = \frac{C}{2}\left[T_s + \frac{T_c}{N_R} - T_d\right] \tag{7}$$

where T_s is the position of the slot center, T_c is the accumulated return position within the slot, N_R the number of returns received, T_d the delay time between the ground station receiving the pulse and transmitting a return, and C the speed of light. Horizontal range is computed and converted to a central earth angle using the formula:

$$\sigma = \frac{1}{r_L}\sqrt{d_s^2 - (h - h_s)^2} \tag{8}$$

where the local radius r_L is found from

$$r_L = h_s + \frac{r_e}{\sqrt{1 + \varepsilon^2 \, SIN^2 L_s}} \tag{9}$$

The longitude of the vehicle is found from

$$\lambda = \lambda_s + \sin^{-1}\left[\frac{\sin \psi_h \sin \sigma}{\cos L_s}\right] \tag{10}$$

and the latitude of the vehicle from

$$L = \sin^{-1}\left[\frac{\cos \sigma}{\sqrt{1-\sin^2\psi_h \sin^2\sigma}}\right] - \tan^{-1}\left[\frac{\sqrt{\cos^2 L_s - \sin^2\psi_h \sin^2\sigma}}{\sin L_s}\right] \tag{11}$$

The vehicle horizontal velocity is then computed using a flat earth model. The velocity magnitude is found from

$$V = \frac{r_L}{\Delta T}\sqrt{(\lambda_n - \lambda_{n-1})^2 \cos L_n + (L_n - L_{n-1})^2} \tag{12}$$

where the subscripts n and n-1 are standard difference equation notation. Vehicle heading is determined from

$$\psi = \tan^{-1}\left[\frac{(\lambda_n - \lambda_{n-1})\cos L_n}{L_n - L_{n-1}}\right] \tag{13}$$

On each pass through the program the TACAN receiver DME time slot is adjusted so that it will track the returns. The slot center position T_s is computed by solving the two difference equations.

$$T_{\Delta_n} = T_{\Delta_{n-1}} + K_2 \frac{T_c}{N_R} \tag{14}$$

and

$$T_{s_n} = T_{s_{n-1}} + T_{\Delta_n} \tag{15}$$

If the ratio of the number of returns to interrogations is less than r_o then T_c/N_R is assumed zero.

For this case a wait is programmed and if the same channel remains selected after the wait is over, an error message is issued. The closest station is selected if it is determined that more than one table entry has the same channel number as the receiver. The closest station is the one which has the minimum value of:

$$(\lambda - \lambda_S)^2 + (L - L_S)^2 \tag{1}$$

where λ and L is the longitude and latitude of the vehicle as determined from the inertial navigation program.

The expected line of sight distance from the vehicle to the station is then computed for use in estimating the position of the receiver return slot. To accomplish this, first the earth's central angle between the vehicle and ground station is computed from:

$$\sigma = \cos^{-1}[\sin L \sin L_S + \cos L \cos L_S \cos(\lambda - \lambda_S)] \tag{2}$$

A spherical earth model having a radius equal to the local earth radius at the ground station plus the ground station elevation is used to determine the ground distance between the vehicle and ground station. This is determined by computing:

$$d_h = \sigma\left[h_s + \frac{r_e}{\sqrt{1+\varepsilon^2 \sin^2 L_S}}\right] \tag{3}$$

where r_e is the earth's equatorial radius and ε is the ellipticity of the earth. The line of sight distance is then computed from

$$d = \sqrt{d_h^2 + (h - h_S)^2} \tag{4}$$

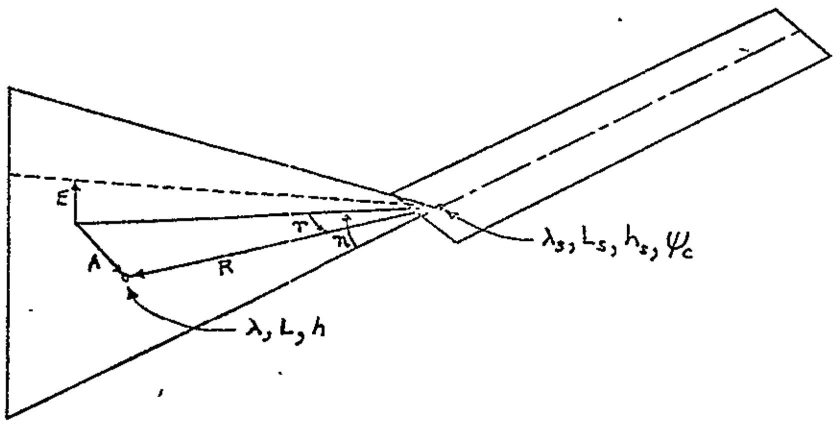
5.4.5 Landing Systems

The space shuttle booster avionics system will include an Advanced Instrument Landing System (AILS) and an Instrument Landing System (ILS). The AILS is the more sophisticated system providing zero-zero landing capabilities. The ILS system is included to assist landing at airports not equipped with AILS ground equipment. Only one of the two systems will be used for any particular landing. The systems are activated during landing approach. Both systems are checked out during pre-launch and pre-ferry. These systems are the primary source of navigation data during final approach and landing.

<u>AILS</u>

Three programs are used in conjunction with the AILS. These are a ground checkout program, an inflight monitoring program, and the landing systems navigation program. Figure 5-57 is a representation of the parameters associated with landing navigation. The output of the AILS is the elevation angle η, the azimuth angle Υ and line of sight range from the vehicle to the end of the runway R. The location of the end of the runway in coordinates of longitude λ_s, latitude L_s, elevation h_s, and the bearing of the runway center line ψ_c will be stored in the computer. It is assumed that 20 values is sufficient for all the airports required of any mission plus mission alternates. It is required that the DMS compute the vehicle position in navigation coordinates λ, L, and h and with respect to the desired landing flight path E and A. The inertial navigation system will be updated so that it can provide a reference during final approach in the event of a sudden AILS failure.

Figure 5-57 Landing Navigation Parameters



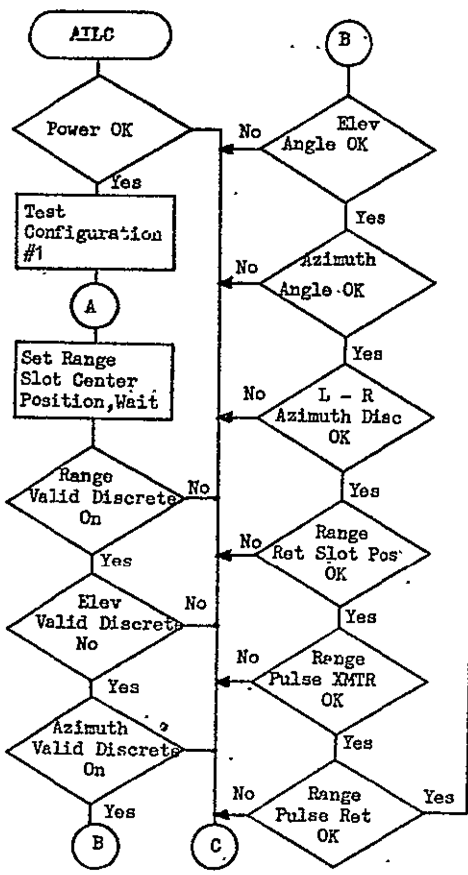


Figure 5-58 AILS Ground Checkout (AILC)

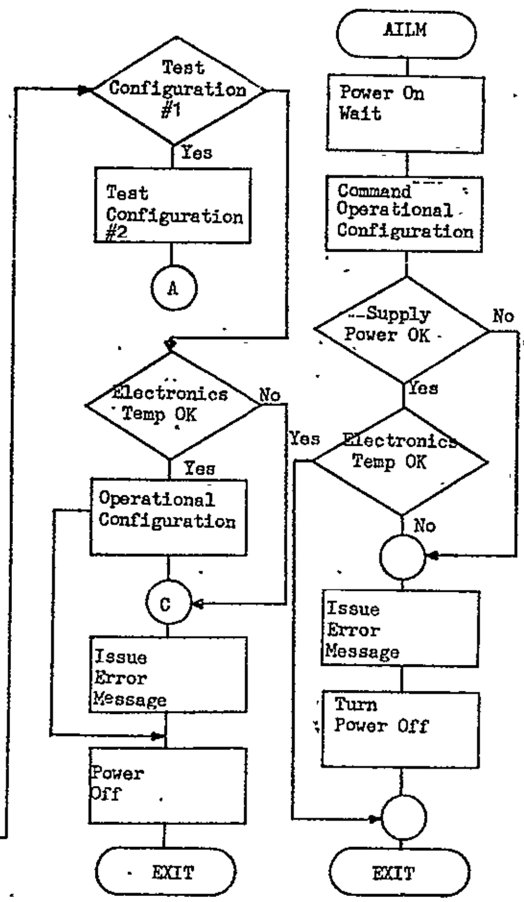


Figure 5-59 AILS Monitoring (AILM)

Because of the short range of operation of the AILS (25 nautical miles) a flat earth model can be used for the required computations. To convert distance values to earth central angles a local radius value is required. This is determined from

$$r_L = h_s + \frac{r_e}{\sqrt{1 + \varepsilon^2 \sin^2 L_s}} \tag{1}$$

The longitude of the vehicle is determined from:

$$\lambda = \lambda_s + \frac{R \cos \eta \cos(\psi_c - \Upsilon)}{r_L \cos L_s} \tag{2}$$

and the latitude from:

$$L = L_s + \frac{R \cos \eta \sin(\psi_c - \Upsilon)}{r_L} \tag{3}$$

The altitude of the vehicle is given by:

$$h = h_s + R \sin \eta \tag{4}$$

The two control parameters are determined from

$$A = R \sin \Upsilon \tag{5}$$

and

$$E = R \cos \Upsilon \sin(\eta_c - \eta) \tag{6}$$

where η_c is the commanded flight path angle. The desired flight path is not a straight line but becomes tangent to the runway at zero range and tangent to a constant descent line at intermediate and large ranges from the runway. A function which exhibits these properties is:

$$\eta_c = \frac{K_1 R^2}{K_2 R^2 + K_3 R + K_4} \qquad (7)$$

In the event of the AILS failing during final approach, the inertial navigation system can be used for landing. The AILS failure must have occurred after an accurate inertial navigation update has been obtained using the AILS. Landing must occur within a short time after this update to insure that inertial navigation system drifts are not significant. To provide this capability, the landing system parameters E, A, and R must be computed from inertial navigation outputs. This is done using the formulas:

$$R = \sqrt{r_L^2 (L - L_s)^2 + r_L^2 (\lambda - \lambda_s)^2 \cos^2 L_s + (h - h_s)^2} \qquad (8)$$

and

$$A = \frac{(\lambda - \lambda_s)\cos L_s \sin \psi_c - (L - L_s)\cos \psi_c}{\sqrt{(L - L_s)^2 + (\lambda - \lambda_s)^2 \cos^2 L_s}} \qquad (9)$$

$$E = \left[(\lambda - \lambda_s)\cos L_s \cos\psi_c + (L - L_s)\sin\psi_c\right]\left[\sin\eta_c - \frac{(h - h_s)\cos\eta_c}{\sqrt{(L - L_s)^2 + (\lambda - \lambda_s)^2 \cos^2 L_s}}\right] \qquad (10)$$

Figure 5-58 is a flow diagram of the AILS ground checkout program (AILS). Upon entry to the program power on discretes are issued. A wait is then programmed to allow for power transients to decay and the supply power tested. Test configuration number 1 is then commanded and a pointer established to reference test configuration number 1 test limits. The range slot center position is commanded to the expected return position and a wait programmed to give sufficient time for the AILS to generate the proper returns. A test is then made of the range valid discrete, the elevation valid discrete, the azimuth angle, the left right azimuth discrete, the position of the range return in the slot, the number of range pulse transmissions, and the number of range pulse returns. A test is then made to determine if the AILS is commanded to test configuration number 1. If it is, test configuration number 2 is commanded, a pointer established to reference test configuration number 2 test limits,and a branch made to the point in the program where the range slot center position is set. If the test shows that the AILS is not commanded to test configuration number 1 the electronics temperature is tested and the AILS is commanded back to the operational configuration. The power is then commanded off and the program exited. If any test in the program indicates a failure, an error message is issued and the power is turned off. This program is run at a 1 per second iteration rate.

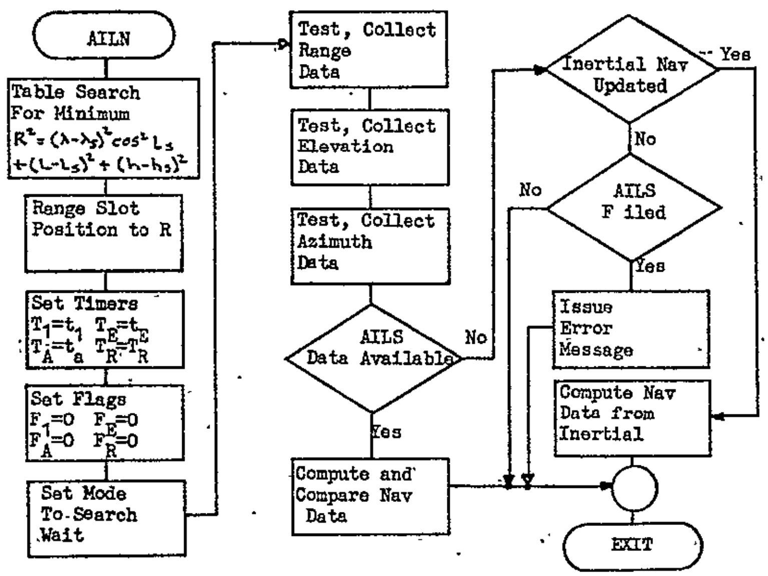
Figure 5-59 is a flow diagram of the AILS inflight monitoring program (AILM). This program is run at a 1 per second iteration rate. Upon entry to the program the power on discretes are issued and a wait programmed to allow for power transients to decay. The AILS is then commanded to an operational configuration and the power and electronics temperature tested. If either test fails an error message is issued and the power turned off.

Figure 5-60 is a flow diagram of the main AILS navigation program. This program is run at a 4 per second iteration rate. The program has an initial and normal entry point. Figure 5-64 includes a flow diagram of the initialization program. The airport parameters of latitude, longitude, elevation, and runway bearing are extracted from the airport data table. The data extracted is for the closest airport to the vehicle when the AILS is energized. Pilot override to select a different airport will be available. The center of the range slot position is then commanded according to the formula:

$$T_S = \frac{2R}{C} + T_d \qquad (11)$$

where R is the range between the vehicle and the airport, C the speed of light and T_d the time delay between the ground station reception of a range pulse and its retransmission. Four timers are then initialized, T_1 which is used to control the time allowed for the testing of each range slot position for an acceptable return before the next slot position is tested, and T_A, T_E, and T_R which are used to time the maximum period allowable between valid azimuth, elevation and range returns respectively. Four flags are then initialized, F_1, F_A, F_E and F_R. F_1 is used to indicate whether or not the inertial navigation system has been sufficiently updated for use in the event of an AILS failure. F_A has three possible states 0, 1, and 2 used to indicate respectively that azimuth data has not been received since entry to the program,

Figure 5-60 AILS Navigation (AILN)



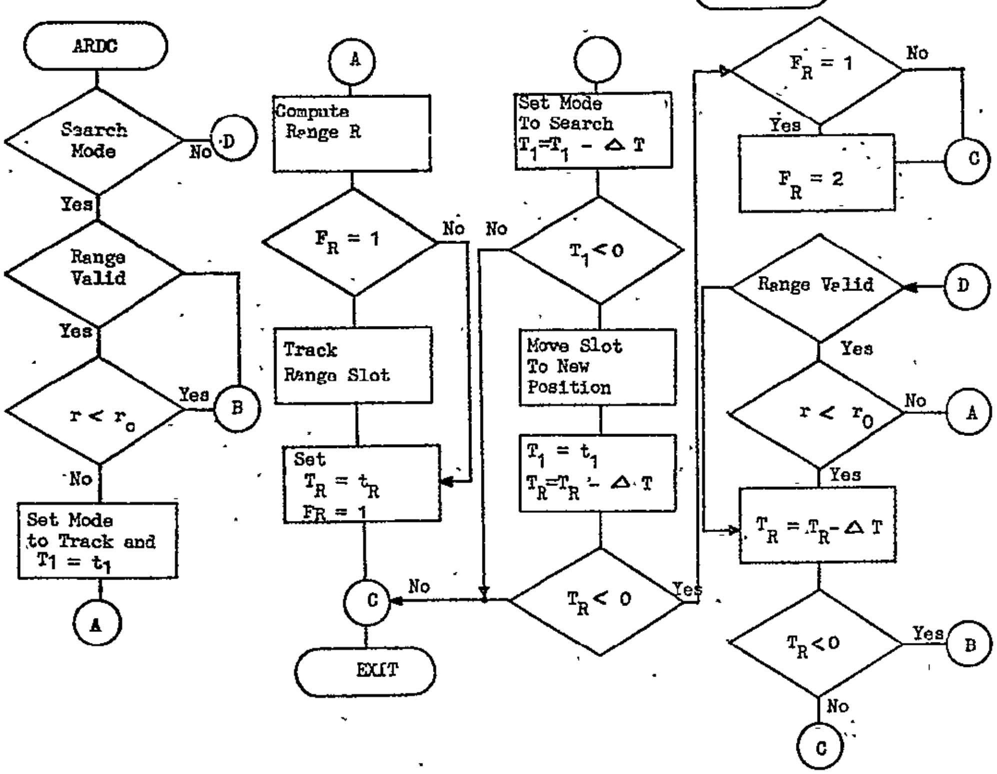


Figure 5-61 AILS Range Data Collection and Testing (ARDC)

azimuth data is OK, and azimuth data has previously been received but not recently enough to be accurate. F_E and F_R are identical to F_A except they are associated with elevation and range respectively, rather than azimuth. A mode flag is set to indicate the range equipment is in a search mode and a wait programmed to allow for the AILS to start generating outputs.

Referring to Figure 5-60 upon completing the initialization programming, the initial entry and normal entry programming coincide. The collection and testing of range data is then performed. A detailed flow diagram of the collection and testing of range data is shown in Figure 5-61. First a test is performed to determine if the operational mode is search or track. If the mode is track a test of the range valid discrete is made. If the range is valid the ratio of return pulses to interrogation pulses is compared against a prestored constant. If the ratio is above the constant value, range is computed from

$$R = \frac{C}{2}\left[T_s + \frac{T_c}{N_R} - T_d\right] \tag{12}$$

where T_s is the commanded range slot center position, T_c is the accumulated position of the returns in the slot, N_R is the number of returns, T_d is the ground station delay between reception and transmission of the pulse, and C is the speed of light. The slot center is then moved to track the return if the range flag (F_R) is 1. The range slot is tracked using the formula

$$T_{s_n} = T_{s_{n-1}} + \frac{2}{C}\left(R_n - R_{n-1}\right) \tag{13}$$

If the range flag is not equal to 1 the value of R_{n-1} is not known, and thus insufficient data is available to evaluate the new range slot position. In either case the timer T_R is reset and the range flag set to 1.

If the initial track mode testing indicated either range invalid or the return to interrogation ratio inadequate, the range timer (T_R) is decremented and tested. If the timer has not run out, no further action is taken until the next pass through the program. If the timer has run out, the mode is changed to search and the normal search path with no range data available is taken.

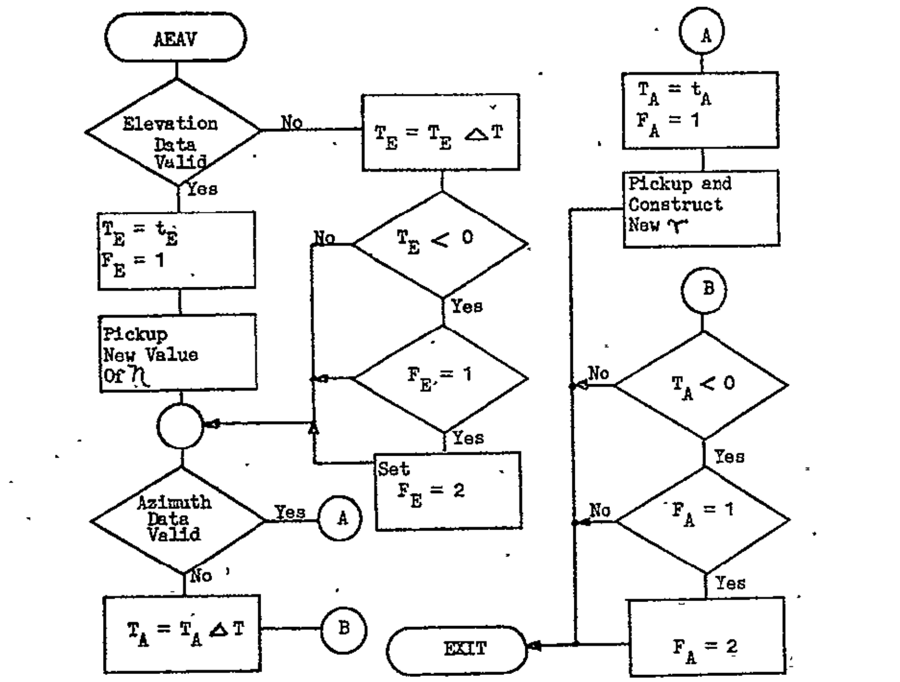
If the initial test showed that the operating mode was track, a test of the range valid discrete and the return to interrogation pulse ratio is performed. If good range data is available the mode is set to track, the T_1 counter is reset, and the previously described computation of range and range slot tracking is performed. If valid range data is not available the timer T_1 is decremented and tested. If the timer has not run out no further action is taken until the next pass through the program. If the T_1 timer has run out the slot is moved to a new position. If the position of the slot determined by the initialization programming or the position of the slot when track was lost is T_{so}, then the slot is repositioned according to the sequence: T_{so}, $T_{so} + \Delta T_s$, $T_{so} - \Delta T_s$, $T_{so} + 2 \Delta T_s$, etc.

The T_1 timer is then reset and the T_R timer decremented and tested. If the timer T_R has run out and the range flag F_R is 1, the range flag is set to 2.

Figure 5-62 is a detailed flow diagram of the testing required to determine if the AILS generated Elevation and Azimuth data is valid. First the elevation valid discrete is tested. If elevation data is valid the timer T_E is reset and the flag F_E set to 1. The elevation angle from the AILS is then picked up and stored for later use in the navigation calculations. If the elevation data is not valid the timer T_E is decremented and tested. If the timer has run out the flag F_E is tested and set equal to 2 if it was previously 1. The azimuth angle is treated the same as the elevation angle with the exception of a signed azimuth angle constructed from the azimuth magnitude input and the left-right discrete input.

Figure 5-63 is a detailed flow diagram of the AILS navigation computations and testing required by the remaining portion of the program of Figure 5-60. First the flags F_R, F_A, and F_E are tested. If any flag is set to zero the AILS is still in an initial operating configuration and no navigation data can be generated. If all three flags are 1, navigation data is available and values of λ, L, h, A, η_c, and E are generated by computing equations 2,3,4,5,7, and 6 respectively. The error between the AILS and inertial navigator data is compared with a small value E_o. The inertial navigation update program will

Figure 5-62 AILS Elevation and Azimuth Valid Testing (AEAV)



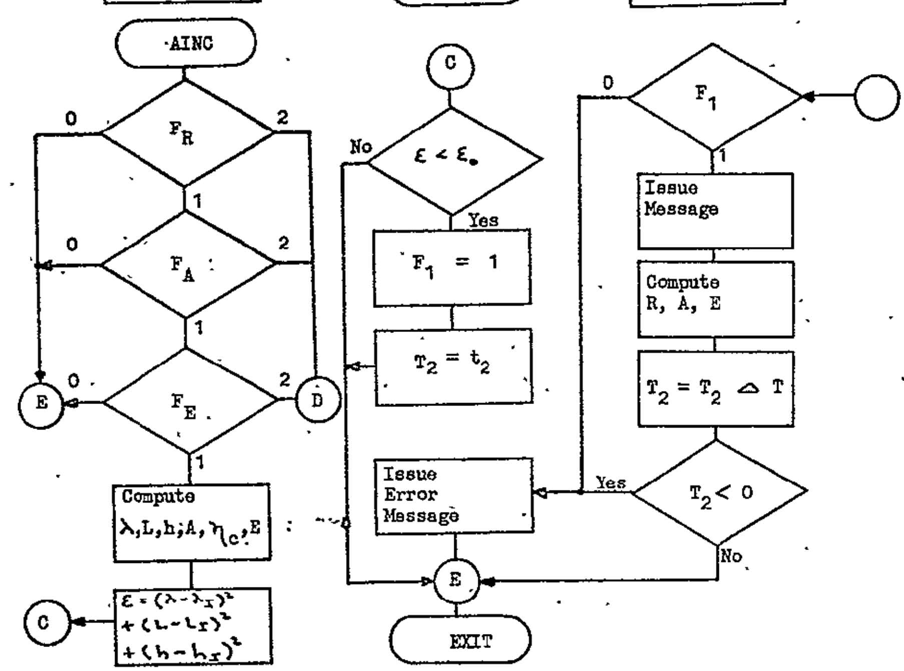


Figure 5-63 AILS Navigation Computations (AINC)

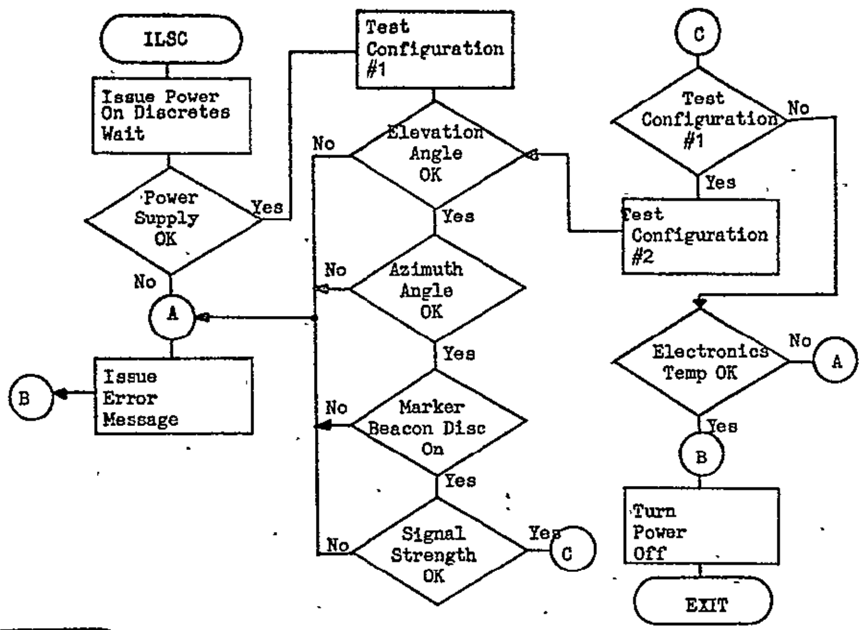
drive the inertial navigation outputs to coincide with the AILS outputs such that after a few AILS inputs the comparison should be less than E_o. When this occurs the flag F_1 is set to 1 and the timer T_2 is set. If any of the flags F_R, F_A or F_E are equal to 2, an indication that the AILS has stopped receiving data, a test is made of flag F_1 to determine if the inertial navigation outputs are sufficiently updated for use as landing data. If F_1 is a 1, a message is issued to inform the pilot that landing navigation data is now being provided from the inertial navigation system. Values of R, A, and E are then computed using equations 8,9 and 10 respectively. The timer T_2 is then decremented and tested. If T_2 is less than zero or if F_1 is zero, an error message is issued indicating landing navigation data is unavailable.

ILS

Since the ILS does not offer enough accuracy to achieve zero-zero landings a completely automatic landing cannot be made using ILS. The ILS will provide data to the pilot and commands to the pitch and yaw control systems. Two programs are required for control and use of the ILS. These are a ground checkout program and an in-flight program. Both programs are run at a 1 per second iteration rate.

Figure 5-64 is a flow diagram of the ILS ground checkout program (ILSC). Upon entry to the program the AC and DC power on discretes are issued and a wait programmed to allow for power transients to decay. The

Figure 5-64 ILS Ground Checkout (ILSC)



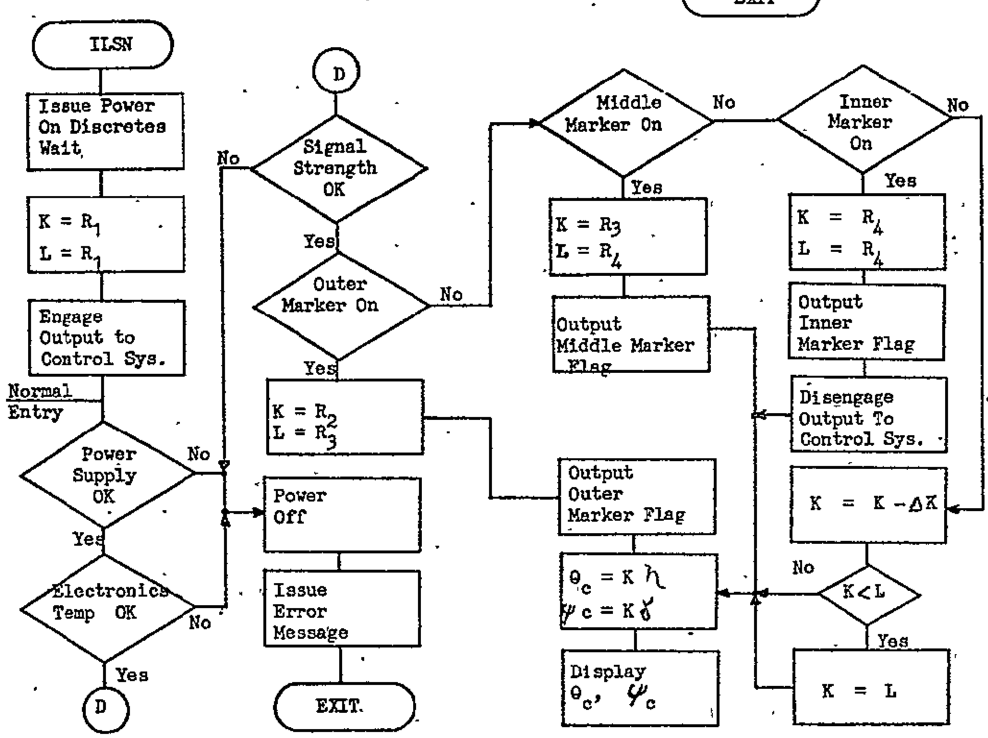


Figure 5-65. ILS Monitoring and Navigation (ILSN)

supply voltages are then tested. The ILS is commanded to test configuration number 1 and a pointer established to expected test values for test configuration number 1 outputs. The elevation angle, azimuth angle, marker beacon discretes, and signal strength discrete are then tested against expected values. A test is then made to determine if test configuration number 1 is commanded. If it is, test configuration number 2 is commanded and elevation, azimuth, marker beacon discretes and the signal strength discrete tested against configuration number 2 expected values. The electronics temperature is then tested and the power turned off. If any test indicates improper operation an error message is issued and the power is turned off.

Figure 5-65 is a flow diagram of the ILS monitoring and navigation program (ILSN). The program has a normal and an initial entry point. Entering the program at the initial entry point causes the power on discretes to be issued. A wait is then programmed to allow for power transients to decay. The elevation and azimuth angles measured by the ILS are presented to the pilot on a display,and issued to the control system as pitch and yaw attitude commands. A gain is applied to these commands and in order to regulate the vehicle response this gain is decreased as the vehicle approaches the runway. The gain setting and reduction is controlled by setting the gain to a fixed value at each marker beacon crossing,and then decrementing the gain as a function of time in between the marker beacons. A lower limit on the gain value is used to keep the gain from becoming too small. The initialization program assigns an initial value to the gain, K, and the lower limit, L.

The initialization program also sets a flag indicating that ILS landing data is available to the control system.

The power supply voltages and electronics temperature are first tested upon entering the program at the initial entry point. If either of these tests fail the power is turned off, and an error message issued to the pilot and the control system. The control system will disengage the ILS steering commands when the signal strength is inadequate.

A test is then performed to determine if any marker beacon discrete is on. The gain value and gain limit value are set to the appropriate values if a marker beacon discrete is on. A flag indicating which marker beacon is being crossed is output to the pilot. If the inner marker beacon discrete is on, the attitude commands are disengaged from the control system. If no marker beacon discrete is on, the gain is decremented and tested against the limit value. If the gain is below the limit value it will be set equal to the limit value.

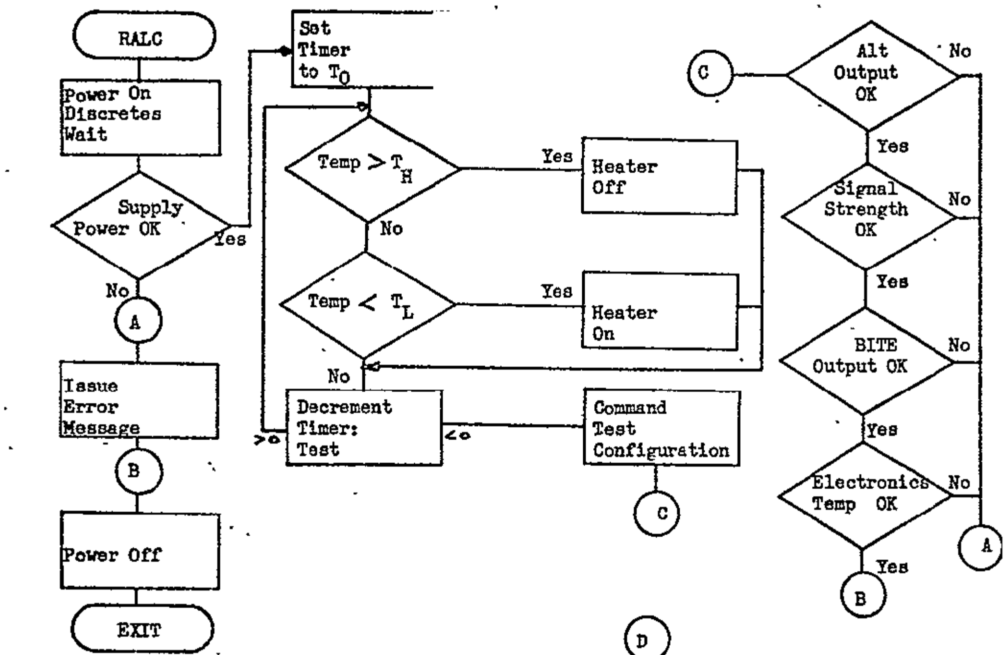
Pitch and yaw commands (θ_c and ψ_c) are then computed by multiplying the programmed gain times the sensed elevation and azimuth angles respectively. These commands are then displayed and made available to the control system.

5.4.6 Radar Sensors

The space shuttle booster contains two radar sensors, a radar altimeter and a weather radar. Two programs within the DMS control the operation of the radar altimeter. Figure 5- is a flow diagram of the radar altimeter ground checkout program (RALC). This program is run at a 1 per second iteration rate. Upon entering the program the power on discretes are issued, and a wait programmed to allow for all power transients to decay. The supply voltages are then tested. A timer is set to allow sufficient time for the temperature of the crystal controlled oscillator oven to stabilize. While the timer is running, the oven temperature is controlled by turning the heater off if the temperature is above an upper limit T_H, and on if the temperature is below a lower limit T_L. The test configuration is commanded when the timer runs out. The radar altimeter altitude, signal strength BITE output, oven temperature, and electronics temperature is tested against expected values. The power is then turned off and the program exited. In any of the tests indicate faulty operation an error message is generated and the power turned off.

Figure 5-67 is a flow diagram of the radar altimeter flight program (RALF). This program is run at a 1 per second iteration rate. The program contains an initial and a normal entry point. In the initial entry, the power is turned on and a delay programmed to allow for power transient decay. The radar altimeter is then commanded to an operational mode. A timer is set to allow sufficient time for the crystal oven to achieve operating temperature. The oven temperature is then

Figure 5-66 Radar Altimeter Ground Checkout (RALC)



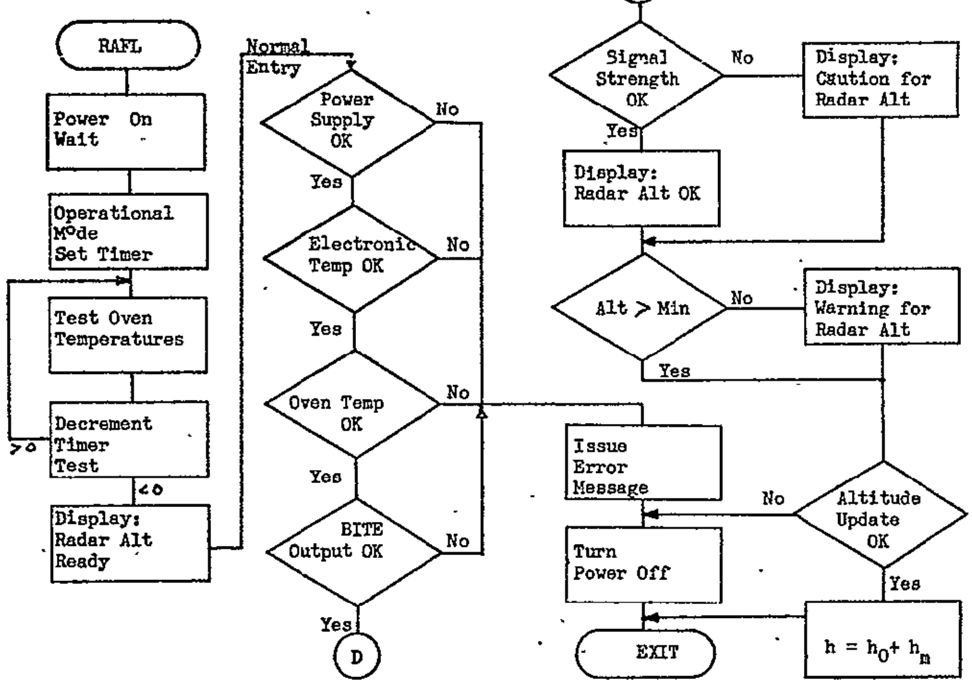


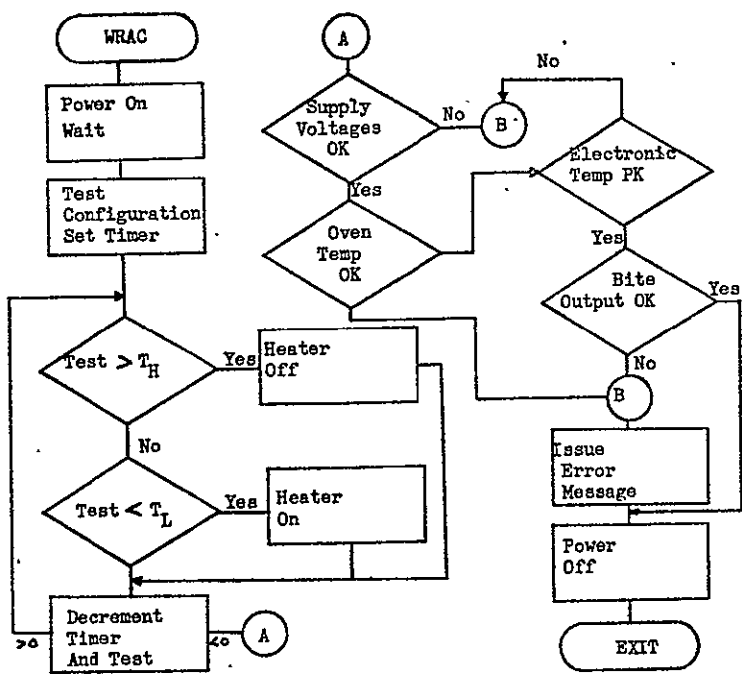
Figure 5-67 Radar Altimeter Flight Program (RAFL)

requested a navigation update using radar altimeter data. The pilot must insert through the keyboard the value of the local terrain elevation along with his update request. The altitude of the booster above sea level is then computed by adding the local terrain elevation to the radar altimeter measured altitude.

Two programs are used in conjunction with the weather radar, a ground checkout program, and an inflight monitoring program. Figure 5-68 is a flow diagram of the Weather Radar Ground Checkout Program (WRAC). This program is run at a 1 per second ineration rate. Upon entry to the program the AC and DC power on discretes are issued and a delay programmed to allow for power transients to decay. The test configuration is then commanded. A timer is set of sufficient length to allow the temperature of the oven to stabilize at its nominal value. As the time is decremented and tested, the oven temperature is controlled by turning the heater on if the temperature is below a lower limit value, and off if above an upper limit value. The supply voltage, oven temperature, electronic temperature, and BITE output are then tested. If a failure is indicated, an error message is issued. After testing the power is turned off.

Figure 5-69 is a flow diagram of the weather radar flight program (WRAF) This program is run at a 1 per second iteration rate. The program has an initial and normal entry point. Upon entering the program at the

Figure 5-68 Weather Radar Ground Checkout (WRAC)



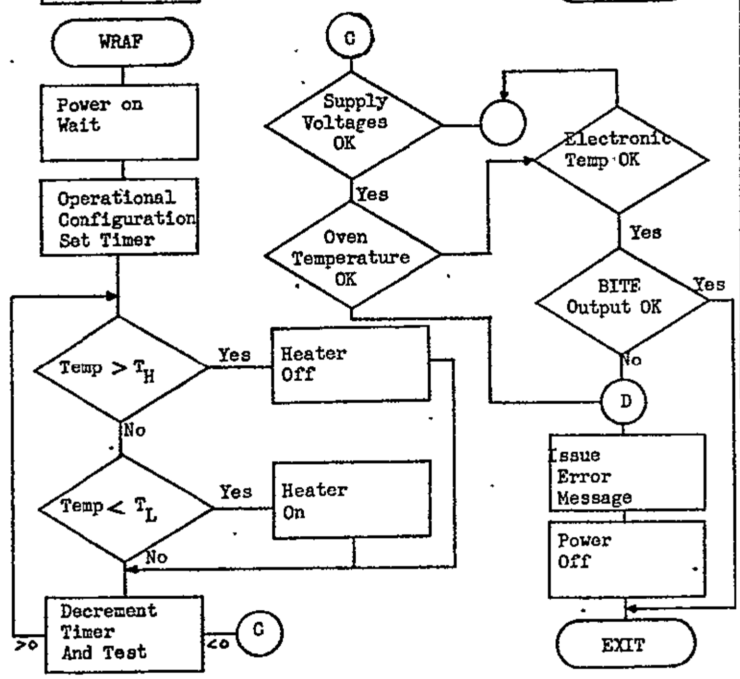


Figure 5-69 Weather Radar Flight Program(WRAF)

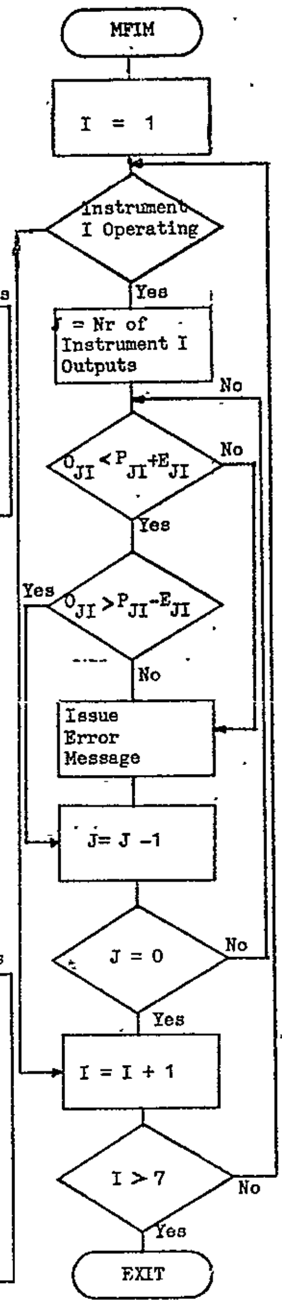


Figure 5-70 Miscellaneous Instrument Monitoring (MFIM)

initial entry point the power on discretes are issued and delay programmed to allow for power transients to decay. The operational configuration is then commanded and a timer set. Control is then transferred to the normal entry point of the program. Upon entering the program at the normal entry point, an oven temperature control cycle is performed where the heater is turned off if the oven temperature is above an upper limit, and off if below a lower limit. The timer, set by the initialization program, is then decremented and tested. Until the timer runs out only the oven temperature is performed at each entry to the program. After the timer runs out, the supply voltages, oven temperature, electronics temperature and BITE output are tested along with the performance of a temperature control cycle. If testing indicates a failure, an error message is issued and the power turned off.

5.4.7 Miscellaneous Flight Instruments

The ground checkout and inflight monitoring of the miscellaneous flight instruments shall be performed by the same program. It is assumed that the inputs from the flight instruments will be parameters already developed by the navigation system within the DMS, making comparison with existing data the only testing required. Figure 5-70 is a flow diagram of the miscellaneous flight instrument monitoring program(MFIM). This program is run at a 1 per second iteration rate. On each entry to the program a counter I is set. The counter indicates which instrument is being tested and acts as a pointer to gather the data required to test each instrument. A test of the discrete input for the I^{th} instrument is then made. This discrete is turned on whenever power is being supplied to the I^{th} instrument. If the instrument is operating a counter J is set equal to the number of outputs from the I^{th} instrument. These outputs are then tested one at a time against appropriate navigation data. The testing limits are also established for each instrument output. An error message is issued if a failure is indicated. J is then decremented and tested. When all outputs of the I^{th} instrument have been tested I is incremented and tested. When all instruments have been tested (7 instruments are assumed) the program is exited.

5.4.8 Guidance

The booster flight program guidance is divided into five parts: boost, coast, reentry, cruise, and landing.

5.4.8.1 Boost

Boost guidance starts at launch and continues through rocket engine thrust termination. The primary concern of boost guidance immediately after launch is to insure that the vehicle clears the launch tower. Immediately after launch a small pitch angle command (θ_{LTC}) is generated to tilt the vehicle away from the launch tower. After sufficient elapsed time this pitch command is removed.

The alignment of the booster on the launch pad is dictated by the physical requirements of umbilical attachments. Before the pitch over maneuver is initiated the booster must perform a roll maneuver so that the body X - Z plane becomes coincident with the desired orbital plane. The roll maneuver is initiated when the booster has reached a sufficient altitude (h_T) to be above the launch tower. The primary coordinate systems for boost guidance have their X - Z planes coincident with the desired orbit plane. The roll maneuver command angle in guidance coordinates in thus zero degrees. Prior to the roll maneuver initiation a roll command angle equal to the launch roll angle is used. This keeps the vehicle from performing any roll attitude change until roll maneuver initiation. During the first portion of the boost trajectory the major constraints upon the vehicle are imposed by aerodynamic loading. The pitch guidance during this period is an open loop function of time. During this period the guidance equations are:

$$\theta_c = \begin{cases} \sum_{i=0}^{4} F_{1i}(t-t_0)^i & \text{for } t<t_1 \\ \sum_{i=0}^{4} F_{2i}(t-t_1)^i & \text{for } t_1<t<t_2 \\ \sum_{i=0}^{4} F_{3i}(t-t_2)^i & \text{for } t_2<t<t_3 \\ \sum_{i=0}^{4} F_{4i}(t-t_3)^i & \text{for } t_3<t<t_4 \end{cases} \qquad (1)$$

The booster is equipped with multiple rocket engines. It is assumed that the mission can be completed in the event of up to four rocket engine failures. The guidance trajectory must be modified in the event of an engine failure. The trajectory modification is achieved by computing a incremented time ΔT_f in the event of a failure from the formula

$$\Delta T_f = \begin{cases} \Delta N[B_{11}+B_{12}(t-t_0)] & \text{for } t<t_1 \\ \Delta N[B_{21}+B_{22}(t-t_1)] & \text{for } t_1<t<t_2 \\ \Delta N[B_{31}+B_{32}(t-t_2)] & \text{for } t_2<t<t_3 \\ \Delta N[B_{41}+B_{42}(t-t_3)] & \text{for } t_3<t<t_4 \end{cases} \qquad (2)$$

ΔT_f is computed for each engine failure. If there is a multiple simultaneous engine failure ΔN is the number of simultaneous engine failures. The value of θ_c is held constant at its previously computed value for ΔT_f seconds after an engine failure. The values of t_1, t_2, t_3, and t_4 are increased by ΔT_f in the event of a failure.

Open loop guidance is discontinued and iterative guidance initiated upon achieving an altitude where aerodynamic effects are negligible. The iterative guidance described below is a modification of the guidance used on the Saturn V vehicle. Iterative guidance makes use of several coordinate systems. Of primary importance are:

Tangent Plane Coordinate System (X_4, Y_4, Z_4)

The tangent plane coordinate system is a right hand orthogonal space fixed system with origin at the geocentric center of the earth. The X_4 axis lies along the intersection of the desired orbital plane and the earth's equatorial plane and is positive toward the descending node of the desired orbit. The positive Z_4 axis lies in the desired orbit plane 90° downrange from the X_4 axis.

Injection Plane Coordinate System (X_v, Y_v, Z_v)

The injection plane coordinate system is a space fixed system with origin at the goecentric center of the earth. The positive X_v axis lies in the desired orbit plane at an angle $-\phi_T$ from the X_4 axis, and passes through the predicted injection point. The Z_v axis lies in the desired orbit plane 90° downrange from the X_v axis.

Navigation Coordinate System (X_s, Y_s, Z_s)

This navigation coordinate system is a space fixed coordinate system defined at time t_o by the launch site position. The time t_o occurs shortly before launch,at which time strapdown inertial navigation begins. The positive Z_s axis lies along the north polar earth axis. The positive X_s axis lies in the equatorial plane intersecting the equator at the Greenwich meridian at time t_o.

The iterative guidance equations must be initialized at time t_o. This initialization requires the establishement of the transformation martix between the navigation coordinate system and tangent plane coordinate system. This transformation matrix is given by

$$[M_{S4}] = \begin{bmatrix} \cos\lambda & \sin\lambda & 0 \\ \sin i \sin\lambda & \sin i \cos\lambda & -\cos i \\ \cos i \sin\lambda & \cos i \cos\lambda & \sin i \end{bmatrix} \qquad (3)$$

where λ is the latitude of the descending node of the desired orbit at t_o and i is the inclination angle of the desired orbit.

The boost mission from launch to the insertion of the orbiter into orbit is divided into two phases, which are the period of booster and orbiter thrusting. The iterative guidance mechanization requires the initialization of the following constants before the main guidance program can be entered.

- T_{1i} - estimation of time between end of open loop guidance and boost thrust termination
- T_{ci} - estimation of zero thrust period during separation
- T_{2i} - estimation of duration of orbiter thrusting
- V_{ex1} - estimate of booster engine exhaust velocity
- V_{ex2} - estimate of orbiter engine exhaust velocity
- τ_1 - estimate of the booster specific impulse, i.e., mass to mass flow rate ratio
- τ_2 - estimate of the orbiter specific impulse
- $(M/F)_B$ - estimate of the reciprocal of the booster acceleration
- $(M/F)_O$ - estimate of the reciprocal of the orbiter acceleration

In addition to these parameters the constants specifying the desired orbit must be inserted in the program. These values are

R_T - vector position of injection point

$\overline{V}_T$ - vector vehicle velocity at injection point

Υ_T - earth central angle between launch site and injection point

A reciprocal acceleration filter is mechanized throughout powered boost flight. The filter requires an input value of vehicle acceleration determined from the navigation system. Defining the navigation velocity output as the vector having components V_x, V_y, and V_z a characteristic velocity change is computed from

$$\delta V_c = \sqrt{(V_{x_n} - V_{x_{n-1}})^2 + (V_{y_n} - V_{y_{n-1}})^2 + (V_{z_n} - V_{z_{n-1}})^2} \tag{4}$$

The filter reciprocal acceleration is then computed from

$$\begin{aligned}(M/F)_{s_n} = {} & K_1(\Delta T/\delta V_c)_n + K_2(\Delta T/\delta V_c)_{n-1} \\ & + K_3(\Delta T/\delta V_c)_{n-2} + K_4(\Delta T/\delta V_c)_{n-3} \\ & + K_5(\Delta T/\delta V_c)_{n-4} + K_6(M/F)_{s_{n-1}} \\ & + K_7(M/F)_{s_{n-2}} + K_8(M/F)_{s_{n-3}} \\ & + K_9(M/F)_{s_{n-4}}\end{aligned} \tag{5}$$

where ΔT is the period between guidance iterations (.5 seconds).

Booster thrust termination is determined normally by a test of the characteristic velocity which is determined from

$$V_c = \sqrt{V_x^2 + V_y^2 + V_z^2} \tag{6}$$

When V_c is greater than the termination value, thrust is terminated.

The iterative guidance calculation used by the booster are performed in the following sequence:

(1) Compute booster time-to-go (T_{1i}) and specific impulse (τ_1) from

$$M_c = M_{co}\, e^{-V_c/V_{ex1}} \tag{7}$$

$$F_c = M_c / (M/F)_s \tag{8}$$

$$\delta V_{c1} = V_{c1N} - V_c \tag{9}$$

$$\tau_1 = V_{ex1}\,(M/F)_s \tag{10}$$

$$T_{1i} = \tau_1 \left(1 - e^{-\delta V_{c1}/V_{ex1}}\right) \tag{11}$$

where M_{co} is the total booster/orbiter lift off mass and V_{cin} is the total booster velocity to be gained. A backup thrust termination is performed if M_c is less than a preset value.

(2) Transform the vehicle position and velocity from navigation to target plane coordinates by

$$\overline{R}_4 = [M_{S4}]\,\overline{R}_s \tag{12}$$

$$\overline{V}_4 = [M_{S4}]\,\overline{V}_s \tag{13}$$

(3) Compute the following set of intermediate parameters

$$L_1 = V_{ex1} \ln\left(\frac{\tau_1}{\tau_1 - T_{1i}}\right) \tag{14}$$

$$J_1 = L_1 \tau_1 - V_{ex1} T_{1i} \tag{15}$$

$$S_1 = L_1 T_{1i} - J_1 \tag{16}$$

$$Q_1 = S_1 T_1 - \frac{1}{2} V_{ex1} T_{1i}^2 \tag{17}$$

$$P_1 = J_1 T_1 - \frac{1}{2} V_{ex1} T_{1i}^2 \tag{18}$$

$$U_1 = Q_1 T_1 - \frac{1}{6} V_{ex1} T_{1i}^3 \tag{19}$$

$$S_{12} = S_1 + L_1 T_{ci} \tag{20}$$

$$L_2' = V_{ex2} \ln\left(\frac{T_2}{T_2 - T_{2i}}\right) \tag{21}$$

$$L_y' = L_1 + L_2' \tag{22}$$

$$J_2' = L_2' T_2 - V_{ex2} T_{2i} \tag{23}$$

(4) Compute the predicted total time to go from

$$T_{1c} = T_{1i} + T_{ci} \tag{24}$$

$$T^* = T_{1c} + T_{2i} \tag{25}$$

(5) The current range angle (ϕ_1) is calculated from the position vector components in the target plane system by

$$\phi_i = TAN^{-1}\left(\frac{Z_4}{X_4}\right) \tag{26}$$

(6) The terminal range angle (ϕ_T) is then predicted from vehicle state variables, projected future performance, and desired terminal conditions from

$$V_T = \sqrt{V_{T_x}^2 + V_{T_y}^2 + V_{T_z}^2} \tag{27}$$

$$\delta_2 = V_c\, T^* - J_2' + L_y' T_{2i} - (K/V_{ex2})[(T_1 - T_{1i}) L_1 + (T_2 - T_{2i}) L_2'](L_y' + V_c - V_T) \tag{28}$$

$$R_T = \sqrt{R_{T_x}^2 + R_{T_y}^2 + R_{T_z}^2} \tag{29}$$

$$\phi_{iT} = \frac{\cos\theta_T}{\pi R_T}(S_{12} + \delta_2) \tag{30}$$

$$\phi_T = \phi_i + \phi_{iT} \tag{31}$$

The angle θ_T is the desired terminal path angle and must be initialized with the value of γ_T prior to launch.

(7) The components of the desired terminal position, velocity, and gravity vectors are obtained by orienting the injection system so that the terminal position vector is coincident with the X_V axis. This is done by computing:

$$X_{VT} = R_T \tag{32}$$

$$\dot{X}_{VT} = V_T \sin\theta_T \tag{33}$$

$$\dot{Z}_{VT} = V_T \cos\theta_T \tag{34}$$

$$\ddot{X}_{VgT} = -\mu / R_T^2 \tag{35}$$

$$\ddot{Z}_{VgT} = 0 \tag{36}$$

Where μ is the gravitational constant. Equations 32 through 36 represent components of vectors $\overline{R}_{VT}$, $\overline{V}_{VT}$ and $\overline{G}_{VT}$. The additional components of these vectors not included in equations 32 through 36 remain zero throughout flight.

(8) The current position, velocity, and gravitational vectors are transformed to the injection coordinate system by computing

$$[M_{V4}] = \begin{bmatrix} \cos\phi_T & 0 & \sin\phi_T \\ 0 & 1 & 0 \\ -\sin\phi_T & 0 & \cos\phi_T \end{bmatrix} \tag{37}$$

$$\bar{R}_V = [M_{V4}]\,\bar{R}_4 \tag{38}$$

$$\bar{V}_V = [M_{V4}]\,\bar{V}_4 \tag{39}$$

$$\bar{G}_V = [M_{V4}][M_{4S}]\,\bar{G}_S \tag{40}$$

The vector $\bar{G}_s$ is the gravitational vector developed in the navigation system.

(9) As estimated velocity-to-be-gained vector ($\Delta V_v'$) is computed. This is done by computing an average gravitational acceleration vector ($\bar{G}_v^*$) for the remaining flight time. The estimated velocity-to-be-gained vector is then computed by subtracting the current acquired velocity vector and the vector representing the velocity loss during the remaining time due to gravitational acceleration from the desired terminal velocity vector. The computations are:

$$\bar{G}_V^* = \frac{1}{2}\left(\bar{G}_{VT} + \bar{G}_V\right) \tag{41}$$

$$\Delta V_V' = \bar{V}_{VT} - \bar{V}_V - T^* \bar{G}_V^* \tag{42}$$

(10) An improved estimate of T^* based on the new information now available for velocity-to-be-gained is computed from

$$\Delta L_2 = \frac{1}{2}\left[\frac{(\Delta V_V')^2}{L_y'} - L_y'\right] \tag{43}$$

$$\Delta T_2' = \Delta L_2 \left(\mathcal{T}_2 - T_{2i} \right) / V_{ex2} \tag{44}$$

$$T_{2i} = T_{2i} + \Delta T_2' \tag{45}$$

$$T^* = T_{1c} + T_{2i} \tag{46}$$

(11) Equations 14 through 46 are based on a time-to-go value which is normally computed during a previous computation cycle. If the vehicle performs as predicted, then the predicted time to go and intermediate parameter values of equations 14 through 25 are very close to the corrected values of equations 43 through 46. The vehicle performance is often not this predictable, and a second iteration of equations 27 through 46 is made to improve accuracy. Before doing this it is necessary to execute:

$$L_2' = L_2' + \Delta L_2 \tag{47}$$

$$L_y' = L_y' + \Delta L_2 \tag{48}$$

$$J_2' = J_2' + T_{2i} \Delta L_2 \tag{49}$$

(12) After performing a second iteration of equations 27 through 46 an improved estimate of velocity to be gained is made based on the new T^* value and the average gravitational acceleration by computing

$$\overline{\Delta V}_v = \overline{\Delta V}_v' - \Delta T_2' \, \overline{G}_v^* \tag{50}$$

(13) Preliminary guidance commands relative to the target plane system are then calculated from:

$$\tilde{X}_y = TAN^{-1}\left(\frac{\Delta \dot{X}_v}{\Delta \dot{Z}_v}\right) \tag{51}$$

$$\tilde{X}_z = TAN^{-1} \frac{\Delta \dot{Y}_v}{\sqrt{(\Delta \dot{X}_v)^2 + (\Delta \dot{Z}_v)^2}} \tag{52}$$

(14) In order to generate guidance commands which satisfy the position constraints while meeting the necessary velocity constraints, position correction terms are added to the velocity constraint guidance commands by computing:

$$L_2 = L_2' + \Delta L_2 \tag{53}$$

$$J_2 = J_2' + T_{2i}\,\Delta L_2 \tag{54}$$

$$S_2 = L_2 T_{2i} - J_2 \tag{55}$$

$$Q_2 = S_2 T_2 - \tfrac{1}{2} V_{ex2} T_{2i}^2 \tag{56}$$

$$L_y = L_1 + L_2 \tag{57}$$

$$J_y = J_1 + J_2 + L_2 T_{1c} \tag{58}$$

$$S_y = S_{12} - J_2 + L_y T_{2i} \tag{59}$$

$$Q_y = Q_1 + Q_2 + S_2 T_{1c} + J_1 (T_{ci} + T_{2i}) \tag{60}$$

$$K_y = L_y / J_y \tag{61}$$

$$D_y = S_y - K_y Q_y \tag{62}$$

$$\Delta Y_v = Y_v + \dot{Y}_v T^* + \tfrac{1}{2} \ddot{Y}_{vg}^* T^{*2} + S_y \, SIN\, \tilde{X}_z \tag{63}$$

$$K_3 = \Delta Y_v / (D_y \, COS\, \tilde{X}_z) \tag{64}$$

$$K_4 = K_y K_3 \tag{65}$$

$$P_3 = J_2(\mathcal{T}_2 + 2T_{1c}) - \frac{1}{2} V_{ex2} T_{2i}^2 \tag{66}$$

$$U_3 = Q_2(\mathcal{T}_2 + 2T_{1c}) - \frac{1}{6} V_{ex2} T_{2i}^3 \tag{67}$$

$$L_P = L_y \cos \widetilde{X}_z \tag{68}$$

$$C_2 = \cos \widetilde{X}_z + K_3 \sin \widetilde{X}_z \tag{69}$$

$$C_4 = K_4 \sin \widetilde{X}_z \tag{70}$$

$$J_P = J_y C_2 - C_4(P_1 + P_3 + L_2 T_{1c}^2) \tag{71}$$

$$S_P = S_y C_2 - Q_y C_4 \tag{72}$$

$$Q_P = Q_Y C_2 - C_4 U_1 + U_3 + S_2 T_{1c}^2 + (T_{2i} + T_{ci}) P_1 \tag{73}$$

$$K_P = L_P / J_P \tag{74}$$

$$D_P = S_P - K_P Q_P \tag{75}$$

$$\Delta X_v = X_v - X_{vT} + \dot{X}_v T^* + \frac{1}{2} \ddot{X}_{vg}^* T^{*2} + S_P \sin \widetilde{X}_y \tag{76}$$

$$K_1 = \Delta X_v / (D_P \cos \widetilde{X}_y) \tag{77}$$

$$K_2 = K_P K_1 \tag{78}$$

(15) Velocity commands which will cause the vehicle to satisfy both the position and velocity constraints are computed in the target plane system by summing the commands needed to satisfy the constraints separately by

$$\chi_{y4} = \widetilde{X}_y - \frac{1}{\pi}(K_1 - K_2 \Delta T) - \phi_T - \frac{1}{2} \tag{79}$$

$$\chi_{z4} = \tilde{\chi}_z - \frac{1}{\pi}\left(K_3 - K_4 \Delta T\right) \tag{80}$$

(16) These commands must then be transformed to the navigation coordinate system for use in the control system as a pitch command (θ_c) and yaw command (ψ_c). This is done by computing:

$$\overline{F}_S' = \left[M_{S4}\right]^T \begin{bmatrix} COS\,\chi_{y4}\,COS\,\chi_{z4} \\ SIN\,\chi_{z4} \\ -SIN\,\chi_{y4}\,COS\,\chi_{z4} \end{bmatrix} \tag{81}$$

$$\theta_c = TAN^{-1}\left(F_{Sz}'/F_{Sx}'\right) \tag{82}$$

$$\psi_c = TAN^{-1}\left(F_{Sy}'/\sqrt{1 - F_{Sy}'^2}\right) \tag{83}$$

Once the iterative guidance mode is entered a complete guidance computation set is performed each pass through the program. The initialization of the guidance parameters prior to launch represents a major computational task. On vehicles such as the Saturn V this task is performed by a ground computer complex requiring in the vicinity of 40,000 words of memory for program and data storage. It is anticipated that for the space shuttle this program will be executed by the flight computers during the prelaunch activities. This program will be stored in the system mass memory and loaded into the flight computers for execution at the appropriate time during prelaunch activities. The total program does not have to reside in core at the same time, i.e., the program can be executed as an overlay job. For this reason the preflight computations of guidance coefficients will not influence the flight computer requirements and is thus not investigated in this study. Figure 5-71 is a flow diagram of the boost guidance flight program (BGFP). This program is run at a 2 per second iteration rate. The program has an initial and normal entry. The initial entry point includes programming to initialize all boost guidance parameters not initialized by the prelaunch guidance program. Figure 5-71 includes the flow diagram of the initialization programming. The timer used to hold constant attitude commands (ΔT_F) in the event of an engine failure is set equal to zero. N is set equal to the number of booster rocket engines. The attitude commands θ_c, ψ_c, and ϕ_c are initialized. The magnitude of the desired terminal position and velocity vectors are determined, and the transformation matrix between the navigation and target plane coordinate systems is formed. A flag used to control the execution path through the boost guidance flight program is set equal to zero. Control is then transferred to the normal entry point.

Upon entry to the program at the normal entry point, time is incremented and a multiple branch made upon the value of the program control flag. When the control flag is 0, a test is made to determine if launch time (t_0) has arrived. In not, the program is exited until the next iteration pass. If launch time has arrived, a launch command is issued and any launch time

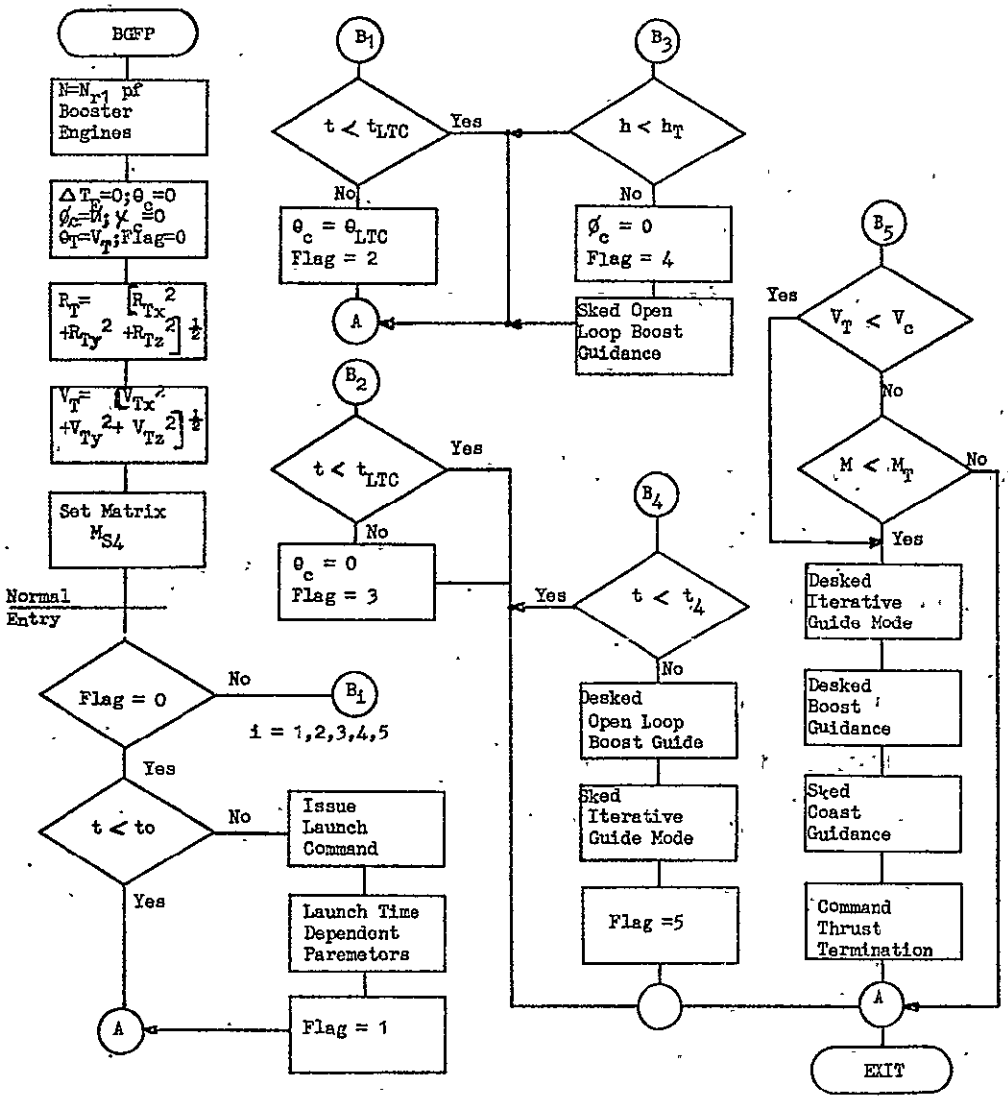


Figure 5-71 Boost Guidance Flight Program (BGFP)

dependent guidance parameters initialized. The control flag is then set equal to 1 and the program exited.

If the control flag is 1, a test to determine if the time to generate the launch tower clearance pitch command has arrived. If not, the program is exited. If the time has arrived the pitch attitude command is set equal to the constant value θ_{LTC} and the control flag set equal to 2 before the program is exited.

If the control flag is 2, a test of time is made to determine if the launch tower clearance pitch command should be removed. If the time for removal has arrived, the pitch attitude command is set equal to zero and the control flag set equal to 3.

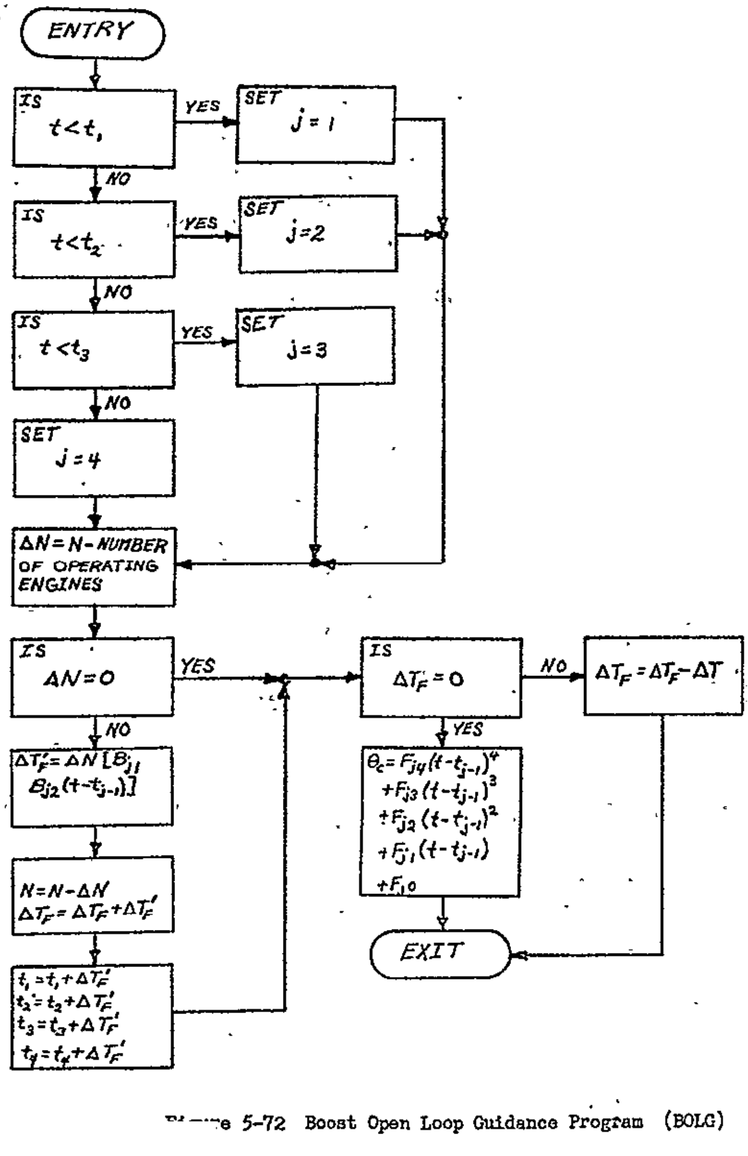
If the control flag is equal to 3 the vehicle altitude is tested to determine if sufficient altitude for pitchover has been achieved. If the altitude is sufficient, the roll attitude command is set equal to zero, the open loop boost guidance program is scheduled and the control flag is set equal to 4.

If the control flag is 4, a test is made to determine if the time to stop open loop guidance has arrived. If the time has arrived the open loop guidance program is descheduled and the boost iterative guidance program scheduled. The control flag is set equal to 5.

If the control flag is 5, tests are made to determine if the vehicle characteristic velocity has reached the terminal velocity or if the total vehicle mass has been reduced below a prestored value. If either condition is present the iterative guidance program is descheduled, coast guidance is scheduled, thrust termination is commanded and this program descheduled.

Figure 5-72 is a flow diagram of the boost open loop guidance program (BOLG). This program is run at a 2 per second iteration rate. With each entry to the program time is tested and a pointer (j) set equal to 1,2,3 or 4 dependent upon $t < t_1$, $t_1 < t < t_2$, $t_2 < t < t_3$, and $t_3 < t$ respectively. The value of ΔN is then obtained by subtracting the number of operating booster rocket engines from the number of engines operating on the previous pass through the program. The value of ΔN is then tested and if not zero, a constant attitude command period ($\Delta T_F'$) is computed. The number of failed engines is then updated and the newly computed constant attitude command period added to any remaining constant attitude command period from previous passes through the program. The time test limits t_1, t_2, t_3 and t_4 are then increased by $\Delta T_F'$.

A test of ΔT_F is then made to determine if a constant attitude command period exists. If it does, ΔT_F is decremented by ΔT and the program exited. If ΔT_F is less than zero θ_c is computed and the program exited.



F---e 5-72 Boost Open Loop Guidance Program (BOLG)

12

Figure 5-73 is a flow diagram of the Boost Iterative Guidance Program (BIGP). This program is run at a 2 per second iteration rate. The program follows the previous description of the inerative guidance equations. The numbers indicated in the flow diagram are the equation numbers of those equations which must be computed at each flow diagram step.

5.4.8.2 Coast

Coast guidance is initiated at the time of the issuance of the booster thrust termination command. After thrust termination, the booster must maintain the attitude present at the end of boost guidance until separation has occurred and the orbiter has achieved a sufficient separation distance to be clear of any booster attitude changes. The booster then assumes a pitch attitude which minimizes orbiter plume impingement while the orbiter ignites its main rocket engines and accelerates out of the area of the booster.

The booster then assumed a reentry attitude. A proper reentry attitude maintains the body y axis in the earth's tangent plane, maintains the y body axis component of the relative velocity between the booster and the atmosphere zero and maintains a constant pitch attitude with respect to the earth's local tangent plane. During coast the inertial navigation system is generating a position and velocity vector with respect to an

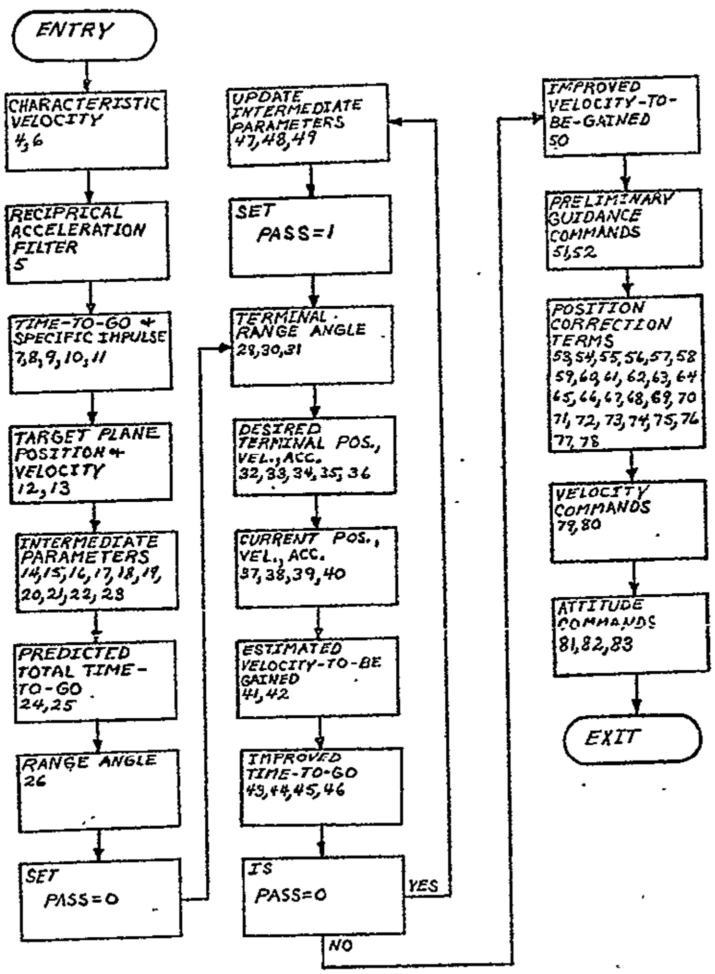


Figure 5-73 Boost Interactive Guidance Program (BIGP)

inertial coordinate system established at launch. The rotational coordinate transformation between the body axis and inertial coordinate system is also generated by the inertial navigation system. Figure 5-41 shows the inertial coordinate system used by the inertial navigation system.

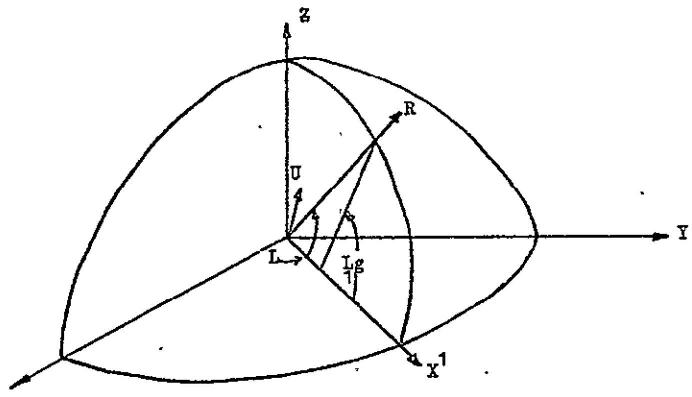


Figure 5-74 Parameters Required for Reentry Attitude

To determine reentry attitude a unit vector normal to the earth's tangent plane must be transformed to body axis. Figure 5-74 shows various parameters required to generate this unit vector. The X, Y, Z coordinate system is the inertial system used by the strapdown inertial navigation system. $\bar{R}$ is the position vector of the booster having components R_x, R_y, and R_z. The vector $\bar{R}$ is first transformed to the X', Y', Z coordinate axis resulting in the components of $\bar{R}'$ being

$$R'_x = R_x^2 + R_y^2 \qquad (1)$$
$$R'_y = 0 \qquad (2)$$
$$R'_z = R_z \qquad (3)$$

In the X', Y', Z coordinate system a unit vector normal to the local earth's tangent plane has components

$$U'_x = \cos L_g \qquad (4)$$
$$U'_y = 0 \qquad (5)$$
$$U'_z = \sin L_g \qquad (6)$$

where L_g is the geodetic latitude of the booster. Geodetic and geocentric latitude are related by

$$\tan L_g = (1-e) \tan L \qquad (7)$$

where e is the earth's ellipticity. However

$$\mathrm{TAN}\ L = R_z'/R_x' \tag{8}$$

Using trigonometric relationships equations 4,5, and 6 become

$$U_x' = R_x'/\sqrt{(1-e)^2 R_z'^2 + R_x'^2} \tag{9}$$

$$U_y' = 0 \tag{10}$$

$$U_z' = (1-e) R_z'/\sqrt{(1-e)^2 R_z'^2 + R_x'^2} \tag{11}$$

Transforming the $\bar{U}'$ vector to the inertial coordinate system yields:

$$U_x = R_x U_x'/\sqrt{R_x^2 + R_y^2} \tag{12}$$

$$U_y = R_y U_y'/\sqrt{R_x^2 + R_y^2} \tag{13}$$

$$U_z = U_z' \tag{14}$$

The computations required for obtaining the $\bar{U}$ vector can be obtained by combining equations 1,2,3,9,10, 11, 12, 13 and 14 to yield

$$U_x = R_x/\sqrt{(1-e)^2 R_z^2 + R_x^2 + R_y^2} \tag{15}$$

$$U_y = R_y/\sqrt{(1-e)^2 R_z^2 + R_x^2 + R_y^2} \tag{16}$$

$$U_z = (1-e)R_z/\sqrt{(1-e)^2 R_z^2 + R_x^2 + R_y^2} \tag{17}$$

The direction cosine matrix $[C]$ defined in section 5.1.4.1 transforms body axis vectors to the inertial coordinate system. The above unit vector transformed to body axis is

$$\overline{U}_b = [C]^T \overline{U} \tag{18}$$

The roll attitude command angle is determined from

$$\phi_c = K_1 U_{bY} + K_2 \int_0^t U_{bY}\, dt \tag{19}$$

where K_1 and K_2 are constant gain factors and the integral term is approximated by

$$\int_0^t U_{bY}\, dt \cong I_{\phi_n} = I_{\phi_{n-1}} + U_{bY} \Delta T \tag{20}$$

The strapdown inertial navigation program provides the booster velocity, $\overline{V}$ with respect to inertial space in inertial coordinates. The velocity of the air mass in inertial coordinates has components of

$$V_{w_x} = R_y \Omega \tag{21}$$

$$V_{w_y} = R_x \Omega \tag{22}$$

$$V_{w_z} = 0 \tag{23}$$

where Ω is earth rate. The relative velocity of the booster with respect to the air mass in body axis coordinates is

$$\overline{V}_{RB} = [C]^T (\overline{V} - \overline{V}_w) \tag{24}$$

The yaw attitude command is generated from

$$\psi_c = K_1 V_{RB_y} + K_2 \int_0^t V_{RB_y}\, dt \tag{25}$$

where the integral term is approximated by:

$$\int_0^t V_{RB_y}\,dt \cong I_{\psi_n} = I_{\psi_{n-1}} + V_{RB_y}\Delta T \tag{26}$$

If the desired angle between the booster X body axis and the local earth's tangent plane at reentry is θ_E then proper reentry attitude is established when

$$\theta_E = TAN^{-1}\frac{U_{bx}}{U_{bz}} \tag{27}$$

If the relationship of equation 27 is not true, then a reentry pitch attitude error exists of

$$\theta_\varepsilon = \theta_E - TAN^{-1}\frac{U_{bx}}{U_{bz}} \tag{28}$$

A pitch attitude command will be generated from this error by:

$$\theta_c = K_1\theta_\varepsilon + K_2\int_0^t \theta_\varepsilon\,dt \tag{29}$$

where the integral term is approximated by

$$\int_0^t \theta_\varepsilon\,dt \cong I_{\theta_n} = I_{\theta_{n-1}} + \theta_\varepsilon\Delta T \tag{30}$$

Figure 5-75 is a flow diagram of the coast guidance program (COGP). This program is run at a 2 per second iteration rate. The program has both an initial and normal entry point. In the initial entry a time counter is

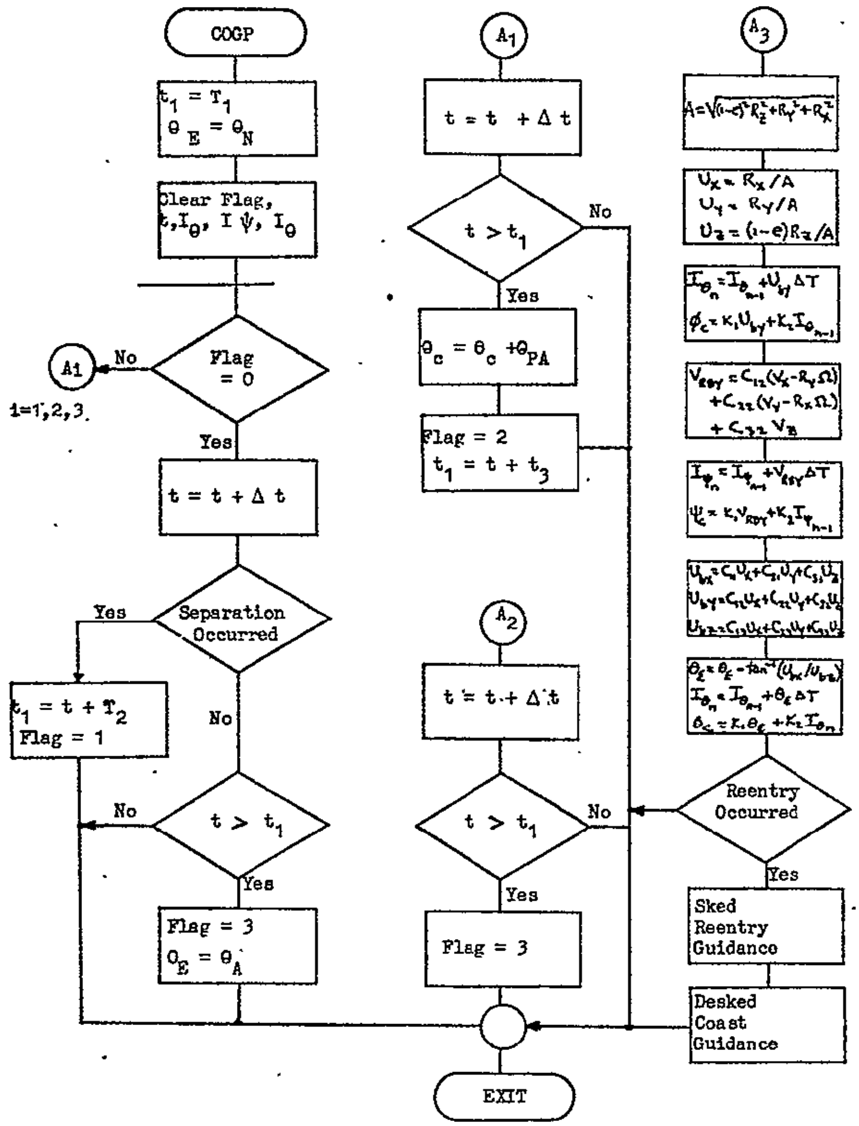


Figure 5-75 Coast Guidance (COGP)

NOT REPRODUCIBLE

set to zero, a control flag initialized to zero, an initial time limit set, the desired reentry pitch attitude established, and the three attitude command integrators initialized to zero. In the normal entry a multiple branch is performed based upon the value of the control flag.

If the control flag is zero, the value established by the initial entry program, the timer is incremented and a test performed to determine if separation has occurred. Separation is tested by a continuity test of the booster/orbiter connection. If separation has not occurred the timer is tested against the timer limit t_1. If the timer has exceeded the t_1 value, an abort situation exists in that booster/orbiter separation has apparently failed. Under this condition the control flag is set equal to 3 and the desired reentry command angle changed to an abort value. If the separation test indicated separation has occurred, the timer limit t_1 is set to the present timer value plus a constant T_2 and the control flag to 1.

If the control flag is 1, the timer is incremented and tested against the limit value. If the limit value has been exceeded, the pitch attitude command is increased by the plume impengment avoidance value, the control flag set equal to 2 and the timer limit set equal to the present time plus a constant T_3.

If the control flag is 2, the timer is incremented and tested against the time limit t_1. If the timer is greater than t_1 the control flag is set equal to 3.

If the control flag is 3, the reentry attitude commands are computed. Figure 5-75 also shows the program required to compute the reentry attitude commands. First the reentry attitude commands are computed using the equations described above. A test is then made to determine if reentry has occurred. This test is based upon vehicle skin temperature. If reentry has occurred, reentry guidance is scheduled and coast guidance descheduled.

5.4.8.3 REENTRY GUIDANCE

At the point of reentry the navigation system starts computing position and velocity of the booster in a navigation coordinate frame and establishes a coordinate transformation matrix D between the navigation system and body axis system. The navigation coordinate systems has its X, Y plane locally tangent to the earth with the X axis pointing north, the Y axis east, and the Z axis directed downward. The navigation system also generates the Euler angles between the body axis and the navigation coordinates.

Initially the primary function of reentry guidance is the maintenance of vehicle heating and loading within tolerable limtis. This initial period is termed the survival reentry phase. During this phase the vehicle's large potential and kinetic energy is converted to heat. As the vehicle velocity with respect to the air mass is reduced, it becomes more difficult to exceed heating and load limits allowing the guidance system to control the vehicle toward the desired flight path. This phase of reentry is termed the controlability reentry phase. The relationship between lift and drag forces are used to control the vehicle during reentry (see Figure 5-76).

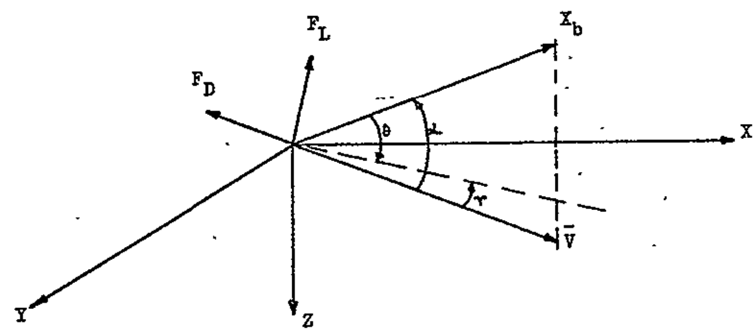


Figure 5-76 Reentry Aerodynamic Relationships

The X, Y, Z axis are the navigation coordinates and X_b is the vehicle X body axis. The flight path angle, γ ,is measured between the vehicle total velocity vector V and the navigation system X, Y coordinate plane. The angle of attack, α , is measured between the velocity vector and the

X body axis. Drag force F_D is a vector having a direction opposite to the velocity vector. Lift force F_L acts in a direction normal to the velocity vector and in the vehicle X_b, Z_b plane.

The magnitude of the lift and drag vectors are given by:

$$F_L = K_2 \rho V^2 \sin\alpha \cos\alpha \qquad (1)$$

$$F_D = \rho V^2 (K_3 + K_4 \sin^2\alpha) \qquad (2)$$

The drag force opposes the velocity and thus decreases the velocity. The lift force being normal to the velocity vector causes the velocity vector to rotate, but does not effect its length.

The heating rate of the vehicle is given by:

$$\dot{Q} = K_1 \sqrt{\rho}\, V^3 \qquad 3)$$

where $\dot{Q}$ is the heating rate per unit area.

At thrust termination the booster is in a suborbital ballistic trajectory. At reentry the vehicle has passed the apogee and thus has a positive flight path angle γ. The survival reentry phase lasts until the velocity magnitude is less than a constant V_o. The vertical plane equations governing the motion of the vehicle are:

$$\ddot{X}' = -\frac{F_D}{m} \cos\gamma + \frac{F_L}{m} \sin\gamma \cos\phi \qquad (4)$$

$$\ddot{Z} = -\frac{F_D}{m} \sin\gamma - \frac{F_L}{m} \cos\gamma \cos\phi + g \qquad (5)$$

where ϕ is the vehicle roll angle and X' is a horizontal axis along the direction of flight. If a roll angle exists the lift force will not be in the vertical plane and thus must be multiplied by cos ϕ.

Reentry guidance during the survival period employs a vehicle model. This model is integrated faster than real time from the vehicle's instantaneous position to a position where V is less than or equal to V_o. The model consists of first determining atmospheric density by

$$\rho = \rho_o e^{-Kh} \tag{6}$$

where ρ_o and K are constants and h is the vehicle altitude obtained from the navigation program. The navigation program also delivers the initial values of the velocity vector components V_x, V_y and V_z. The flight path angle γ is computed from

$$\gamma = TAN^{-1} \frac{V_z}{\sqrt{V_x^2 + V_y^2}} \tag{7}$$

and velocity magnitude from

$$V = \sqrt{V_x^2 + V_y^2 + V_z^2} \tag{8}$$

The instantaneous lift and drag forces are computed from

$$F_L = K_2 \rho V^2 \, SIN\alpha_o \, COS\alpha_o \tag{9}$$

and

$$F_D = \rho V^2 (K_3 + K_4 SIN^2\alpha_o) \tag{10}$$

K_2, K_3 and K_4 are constants and it is assumed that the vehicle will fly a constant angle of attack trajectory during the survival phase which for nominal conditions will be α_o. The acceleration on the vehicle is then computed from

$$a_x = -\frac{1}{m}\left(F_D \cos\gamma + F_L \sin\gamma\right) \tag{11}$$

$$a_z = -\frac{1}{m}\left(F_D \sin\gamma + F_L \cos\gamma\right) \tag{12}$$

where m is the nominal vehicle mass. A zero roll angle is assumed. The accelerations are then integrated using the formula

$$V'_{x_n} = V'_{x_{n-1}} + \frac{\Delta T}{2}\left(3a_{x_n} - a_{x_{n-1}}\right) \tag{13}$$

$$V'_{z_n} = V'_{z_{n-1}} + \frac{\Delta T}{2}\left(3a_{z_n} - a_{z_{n-1}}\right) \tag{14}$$

On the initial integration iteration past and present acceleration values are assumed constant. The initial integration iteration assumes

$$V'_{x_{n-1}} = \sqrt{V_x^2 + V_y^2} \tag{15}$$

$$V'_{z_{n-1}} = V_z \tag{16}$$

Flight path angle and total velocity are propagated using

$$\gamma = \tan^{-1}\frac{V'_{z_n}}{V'_{x_n}} \tag{17}$$

$$V' = \sqrt{V'^2_{x_n} + V'^2_{z_n}} \tag{18}$$

altitude is propagated using

$$h'_n = h'_{n-1} + \frac{\Delta T}{2}\left(3V'_{z_n} - V'_{z_{n-1}}\right) \tag{19}$$

On the initial integration iteration present and past velocity values are assumed equal and the altitude past value is initialized to the navigation altitude output. At each iteration the heating and loads on the vehicle are computed from

$$L = F_D \, SIN\alpha_o + F_L \, COS\alpha_o \tag{20}$$

$$\dot{T}_h = K_1 \sqrt{\rho}\, V'^3 - K_t \left(T_{n-1} - T_o\right) \tag{21}$$

$$T_n = T_{n-1} + \frac{\Delta T}{2}\left(3\dot{T}_n - \dot{T}_{n-1}\right) \tag{22}$$

For the initial integration,iteration T_{n-1} is initialized from a skin temperature sensor and it is assumed that the present and past temperature rates are the same. The maximum load and temperature point is determined as the model equations are integrated. If the maximum load or temperature exceeds a maximum limit value,the angle of attack is commanded to reduce the above limit parameter.

It is anticipated that the shuttle booster will reenter using a very large angle of attack (approximately 60^o). Temperature and loads produce opposing requirements on angle of attack. Increasing the angle of attack increases the drag and decreases the lift forces. This causes a more rapid decay in velocity decreasing the total temperature rise, however the normal force loads are increased. The maximum load value, L_m and the maximum temperature T_m determined from the model are used to determine an angle of attack command from the formula:

$$\dot{\alpha} = \begin{cases} K_T(T_m - T_{max}) & \text{for } T_m > T_{max} \\ 0 & \text{for } T_m < T_{max} \;\&\; L_m < L_{max} \\ -K_L(L_m - L_{max}) & \text{for } L_m > L_{max} \end{cases} \tag{23}$$

where T_{max} and L_{max} are constant limit values. Pitch command is then computed from

$$\theta_{c_n} = \theta_{c_{n-1}} + K_{\dot{\alpha}}\dot{\alpha}\,\Delta T \tag{24}$$

Roll and yaw command remain zero during the survival period. As soon as the total velocity from the navigation system is less than a preset value the survival phase computations are halted and controllability phase computations initiated.

The vehicle is controlled to fly over a target point which is on the desired route to the return landing field. The required range and cross range to reach the target point is first computed. The great circle arc between the vehicle present position and the desired target point is computed from

$$\theta_{AR} = COS^{-1}[SIN\,L\,SIN\,L_T + COS\,L\,COS\,L_T\,COS(\lambda - \lambda_T)] \tag{25}$$

where L and λ are the vehicle's latitude and longitude as determined by the navigation program and L_T and λ_T the target point latitude and longitude. The cross range angle θ_{cR} is computed from

$$\theta_{cR} = \sin^{-1}\left[\sin\theta_{AR}\cos\left(\psi_v + \sin^{-1}\frac{\sin(L_T - L)}{\sin\theta_{AR}}\right)\right] \tag{26}$$

where ψ_v is the horizontal heading angle from the navigation system. The required range is then estimated from

$$R_R = r_e\left(\theta_{AR} + \tfrac{1}{2}\theta_{cR}\right) \tag{27}$$

where r_e is the earth's equatorial radius.

An average geocentric radius to the vehicle's position during the remaining reentry phase is computed from

$$r_{AV} = \frac{r_e}{2}\left[2 - e\left(\sin^2 L + \sin^2 L_T\right) + h + h_T\right] \tag{28}$$

where e is the earth's ellipticity, h the vehicle's present position altitude from the navigation system and h_T the desired vehicle altitude at the target point.

A lift to drag ratio command $(L/D)_c$ is then obtained from an equilibrium glide range formula:

$$\left(\frac{L}{D}\right)_c = \frac{2R_R}{r_{AV}\left(\ln \phi_1 - \frac{b}{\gamma q}\ln Q_2\right)} \tag{29}$$

where

$$a = -\frac{\mu}{r_{AV}} + (r_{AV}\,\Omega \cos L)^2 \tag{30}$$

$$b = 2 r_{AV}\,\Omega \cos L \sin(\psi_{V_T} - \psi_V) \tag{31}$$

$$q = b^2 - 4a \tag{32}$$

$$V = \sqrt{V_x^2 + V_y^2 + V_z^2} \tag{33}$$

$$\phi_1 = \frac{V_T^2 + bV_T + a}{V^2 + bV + a} \tag{34}$$

$$\phi_2 = \frac{(2V_T + b - \sqrt{q})(2V + b + \sqrt{q})}{(2V + b - \sqrt{q})(2V_T + b + \sqrt{q})} \tag{35}$$

where μ is the gravitational constant, Ω the earth rate and ψ_{vT} the desired heading at the target point.

A command flight path angle γ_c is determined by applying the formula:

$$\gamma_c = \frac{1}{V}\left(\frac{\frac{d\rho}{dV}\cdot\frac{dV}{dt}}{\frac{d\rho}{dh}}\right) \tag{36}$$

where ρ is the atmospheric density. Equation 36 is evaluated by computing

$$\frac{d\rho}{dV} = \frac{-2B}{g_0 V (L/D)_c}\left(\frac{F_D}{V} - \frac{dF_D}{dV}\right) \tag{37}$$

$$\frac{dV}{dt} = -\frac{VF_D}{(L/D)_c} \tag{38}$$

and

$$\frac{d\rho}{dh} = -K\rho_0 e^{-Kh} \tag{39}$$

where

$$F_D = \frac{\mu}{r^2 V} - \frac{V + 2r\Omega \cos L \sin(\psi_{V_T} - \psi_V)}{r} - \frac{r\Omega^2 \cos^2 L}{V} \tag{40}$$

$$\frac{dF_D}{dV} = \tag{41}$$

and

$$r = r_e (1 - e \sin^2 L) \tag{42}$$

B is the vehicle's ballistic coefficient at the nominal angle of attack and g_o the surface gravitational acceleration.

Attitude commands are computed from the command flight path angle by computing:

$$C_V = \frac{V}{2} \left(\frac{\gamma_c - \tan^{-1} \frac{V_z}{\sqrt{V_x^2 + V_y^2}}}{\Delta T} \right) + V F_D \tag{43}$$

$$\eta = \frac{\pi}{2} - \psi_V - \sin^{-1} \frac{\sin(L_T - L)}{\sin \theta_{AR}} \tag{44}$$

$$C_H = V \left(\frac{\eta}{\Delta T} - \Omega \sin L \right) \tag{45}$$

$$L_n = \frac{V_{x_n} \cdot V_{x_{n-1}} + V_{y_n} \cdot V_{y_{n-1}} + V_{z_n} \cdot V_{z_{n-1}}}{V \Delta T} \tag{46}$$

$$\alpha_n = \tan^{-1} \frac{V_z}{\sqrt{V_x^2 + V_y^2}} - \theta \tag{47}$$

$$\dot{\alpha}_c = \frac{\left(\frac{C_V}{|C_V|} \sqrt{C_V^2 + C_H^2} - L_n \right)(\alpha_n - \alpha_{n-1})}{\Delta T (L_n - L_{n-1})} \tag{48}$$

$$\dot{\phi}_c = \frac{\tan^{-1}\left(\frac{C_H}{|C_V|} \right) - \phi}{\Delta T} \tag{49}$$

$$\theta_{c_n} = \theta_{c_{n-1}} + \dot{\alpha}_c \Delta T \tag{50}$$

$$\phi_{c_n} = \phi_{c_{n-1}} + \dot{\phi}_c \Delta T \tag{51}$$

Yaw command is maintained zero by the guidance system. The control system will automatically command a yaw rate that will generate a coordinated turn.

Figure 5-77 is a flow diagram of the reentry guidance program (REGP). This program is run at a 2 per second iteration rate. Upon entry to the program those parameters which are common to both the survival and controlability reentry phases are computed. A test of which reentry phase exists is made, based upon total vehicle velocity. In the survival phase the reentry model is integrated and $\dot{\theta}_c$ computed. It is assumed that the model equations must be computed 20 times in order to predict the vehicle maximum loading and heating during the survival phase. At the finish of each iteration through the controlability phase guidance equations a test for the end of the reentry guidance computations is made based upon vehicle velocity and altitude. As soon as the velocity or altitude drop below a preset value, the reentry guidance program is descheduled and cruise guidance scheduled.

NOT REPRODUCIBLE

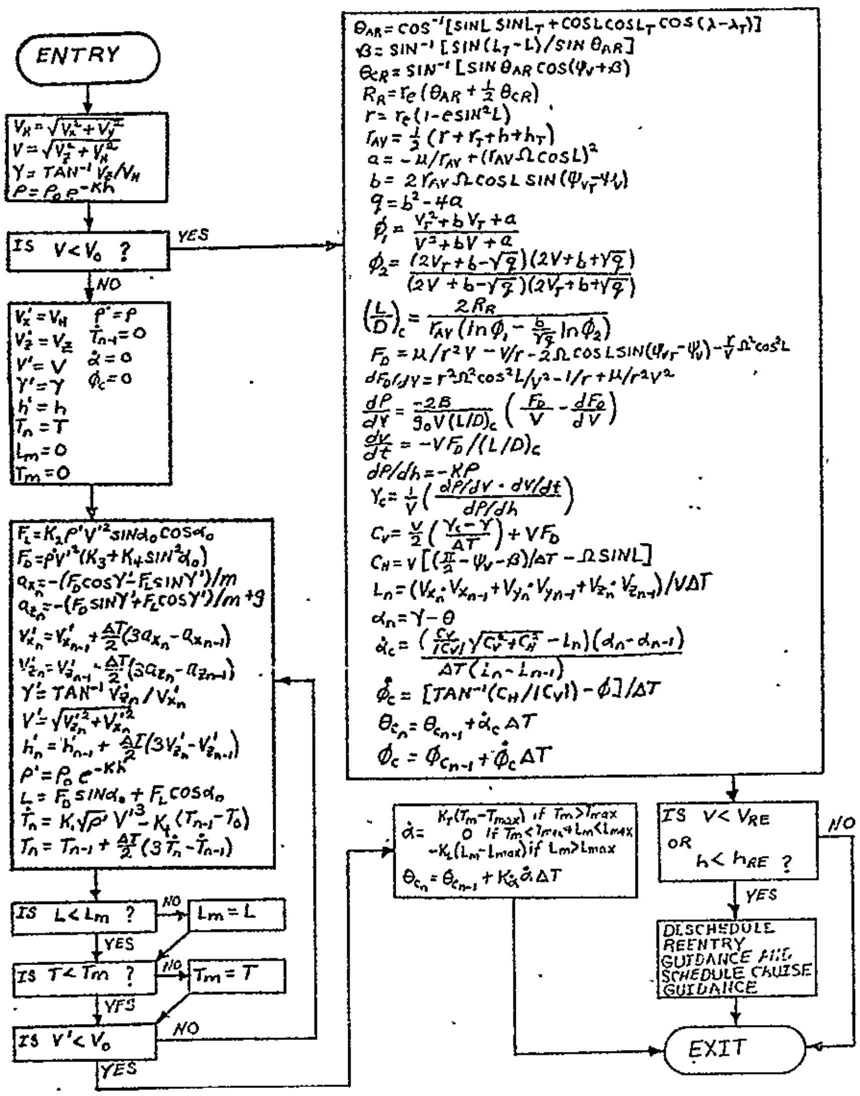


Figure 5-77 Reëntry Guidance Program (REGP)

5.4.8.4 CRUISE GUIDANCE

At the beginning of cruise the booster will be at an altitude above the normal cruise altitude and at a velocity below nominal cruise velocity. Initially the cruise engines will be deployed and at the same time a pitch down attitude command issued. The booster will dive until it reaches cruise altitude which should approximately coencide with reaching cruise velocity.

During cruise the pilot can select one of several steering modes. Within the DMS computers will be stored a table of route points. The data stored for each route point will be latitude (L_i), longitude (λ_i) and altitude (h_i) of the flight leg which should be maintained when flying to the route point. In addition to the route point table, there is also a sequence table which determines the order in which the route points should be flown.

Two automatic modes exist for flying between route points. In the first mode a steering command is generated to maintain the vehicle on the flight leg defined by two route points. In the second mode,steering commands are generated to cause the booster to fly a great circle route between its present position and the next route point. If the vehicle is at latitude - longitude coordinates L, λ flying between route points having coordinates L_1, λ_1 and L_2, λ_2 then the central angle arc distance between the vehicle and the first route point is given by

$$d_1 = \cos^{-1}[\sin L \sin L_1 + \cos L \cos L_1 \cos(\lambda_1 - \lambda)] \qquad (1)$$

and between the vehicle and the second route point given by

$$d_2 = \cos^{-1} \left[\sin L \sin L_2 + \cos L \cos L_2 \cos (\lambda - \lambda_2)\right] \qquad (2)$$

The central angle arc distance between the two route points is given by

$$d_{12} = \cos^{-1} \left[\sin L_1 \sin L_2 + \cos L_1 \cos L_2 \cos (\lambda_1 - \lambda_2)\right] \qquad (3)$$

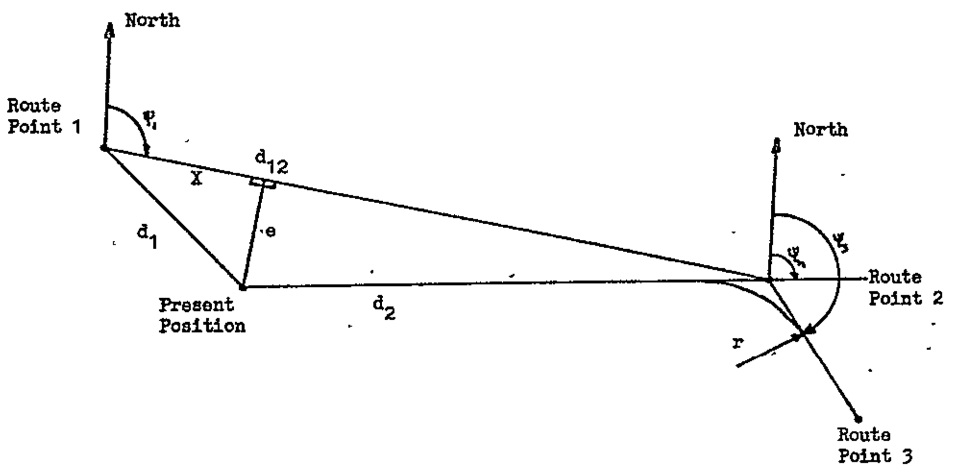


Figure 5-78 Route Point Steering Geometry

Figure 5-78 shows the geometric relationship required to generate the steering commands for route point steering. Applying the law of cosines to the spherical right triangle having sides with central angle arc lengths of X, d_1, and e yields

$$\cos d_1 = \cos x \cos e + \sin x \sin e \cos \frac{\pi}{2} \qquad (4)$$

In like manner for the triangle having sides $d_{12} - x$, d_2 and e yields

$$\cos d_2 = \cos (d_{12} - x) \cos e + \sin (d_{12} - x) \sin e \cos \frac{\pi}{2} \qquad (5)$$

Eliminating x between equations 4 and 5 and solving for e yields

$$e = \cos^{-1} \left[\frac{\sqrt{\cos^2 d_2 - 2 \cos d12 \cos d_1 \cos d_2 + \cos^2}}{\operatorname{Sin} d_{12}} \right] \qquad (6)$$

The steering command generated depends upon the magnitude of e. All major steering turns are made at a constant turning rate which produces a circular flight path which is the arc of a circle of radius r. If e is less than r/r_e a roll command is generated from

$$\phi_c = \begin{cases} -\phi_L & \text{if } K_\phi e < -\phi_L \\ K_\phi e & \text{if } -\phi_L < K_\phi e < \phi_L \\ \phi_L & \text{if } \phi_L < K_\phi e \end{cases} \tag{7}$$

If e is larger than r/r_e the vehicle will be commanded to fly normal to the desired route path until it gets within a distance of r from the desired route path. In order to determine the desired heading angle, ψ_e the following computations must be made

$$d_{12}-x = \cos^{-1}\left[\cos d_2/\cos e\right] \tag{8}$$

$$\sigma = \sin^{-1}\left[\cos L_1 \sin(\lambda_1 - \lambda_2)/\sin d_{12}\right] \tag{9}$$

$$L' = \cos^{-1}\left[\cos(d_{12}-x)\sin L_2 + \sin(d_{12}-x)\cos L_2 \cos\sigma\right] \tag{10}$$

$$\psi' = \sin^{-1}\left[\cos L_2 \sin\sigma/\sin L'\right] \tag{11}$$

$$\psi_e = \psi' - \frac{\pi}{2} \tag{12}$$

The desired heading is achieved by commanding a roll angle proportional to the error in the actual aircraft heading and the desired heading. The error is computed from

$$\psi_\varepsilon = \psi_e - \psi_h \tag{13}$$

Where ψ_h is the hroizontal heading angle from the inertail navigation system. The roll command angle is generated from

$$\phi_c = \begin{cases} -\phi_L & if \quad K_\psi \psi_\varepsilon < -\phi_L \\ K_\psi \psi_\varepsilon & if \quad -\phi_L < K_\psi \psi_\varepsilon < \phi_L \\ \phi_L & if \quad \phi_L < K_\psi \psi_\varepsilon \end{cases} \tag{14}$$

In the second automatic mode a yaw error is established from the formula

$$\psi_\varepsilon = \pi - SIN^{-1}\left[\cos L_1 \, SIN(\lambda_1 - \lambda_2) / SIN\, d_{12}\right] - \psi_h \tag{15}$$

A roll command angle is generated by equation 14 for the second automatic mode.

The pilot can select a heading reference derived from the magnetic flux gate compass or the TACAN receiver instead of the inertial navigation system. When a different reference is selected, ψ_h in equation 13 and 15 is substituted with the new reference. A new leg of the route is selected when d_2 becomes less than r

Altitude control is achieved by commanding pitch attitude as a function of the error between the vehicle's actual and desired altitudes. An altitude reference can be selected from either the inertial navigation system, the air data computer (i.e., pressure altitude), or the radar altimeter. The attitude command is determined from

$$V_{ZC} = \begin{cases} V_L & \text{if} \quad K_h(h_c - h) > V_L \\ K_h(h_c - h) & \text{if} \quad V_L > K_h(h_c - h) > -V_L \\ -V_L & \text{if} \quad -V_L > K_h(h_c - h) \end{cases} \tag{16}$$

$$\dot{\theta}_c = K_v (V_{ZC} - V_Z) \tag{17}$$

$$\theta c_n = \theta_{c_{n-1}} + \dot{\theta}_c \Delta T \tag{18}$$

In the automatic mode the commanded altitude (h_c) is found in the route point tables. Both an altitude and heading command can be selected manually in one of two different ways. A value of desired heading or altitude can be entered numerically by the pilot or the pilot can issue a command to hold the present altitude or heading.

Another cruise steering function which can be selected is to fly an automatic holding pattern. Upon selecting the holding pattern mode the booster will be commanded to fly a circular path having a radius of r_H . The circle will be tangent to the flight path at the point where the holding pattern mode was commanded. To compute the steering commands required to keep the vehicle on the holding pattern flight path it is necessary to compute the latitude and longitude of the circular flight path center. This is done by computing

$$L_P = SIN^{-1}\left[COS\, r_H\, SIN\, L + COS\, L \sqrt{SIN^2 r_H - COS^2 \psi_h}\right] \tag{19}$$

$$\lambda_P = \lambda - SIN^{-1}\left[SIN\, r_H\, COS\, \psi_h / COS\, L_P\right] \tag{20}$$

where r_H is the radius of the holding pattern circle expressed in central angle arc units, and L_p and λ_p the latitude and longitude of the center of the circle. At each iteration through the program the distance from the vehicle to the holding pattern center is computed from

$$r = \cos^{-1}\left[\sin L \sin L_p + \cos L \cos L_p \cos(\lambda - \lambda_p)\right] \qquad (21)$$

A roll rate command is then computed from

$$\dot{\phi}_c = K_{\phi_h}(r_H - r) \qquad (22)$$

A roll command is then computed from

$$\phi_{c_n} = \phi_{c_{n-1}} + \dot{\phi}_c \Delta T \qquad (23)$$

and limited by

$$\phi_c = \begin{cases} \phi_L & \text{if} \quad \phi_{c_n} > \phi_L \\ \phi_{c_n} & \text{if} \quad \phi_L > \phi_{c_n} > -\phi_L \\ -\phi_L & \text{if} \quad -\phi_L > \phi_{c_n} \end{cases} \qquad (24)$$

Figure 5-79 is a flow diagram of the cruise guidance program (CRGP). This program is run at a 2 per second iteration rate. The program has an initial and normal entry point. In the inital entry the first automatic flight leg is established. This is done by picking up the first and

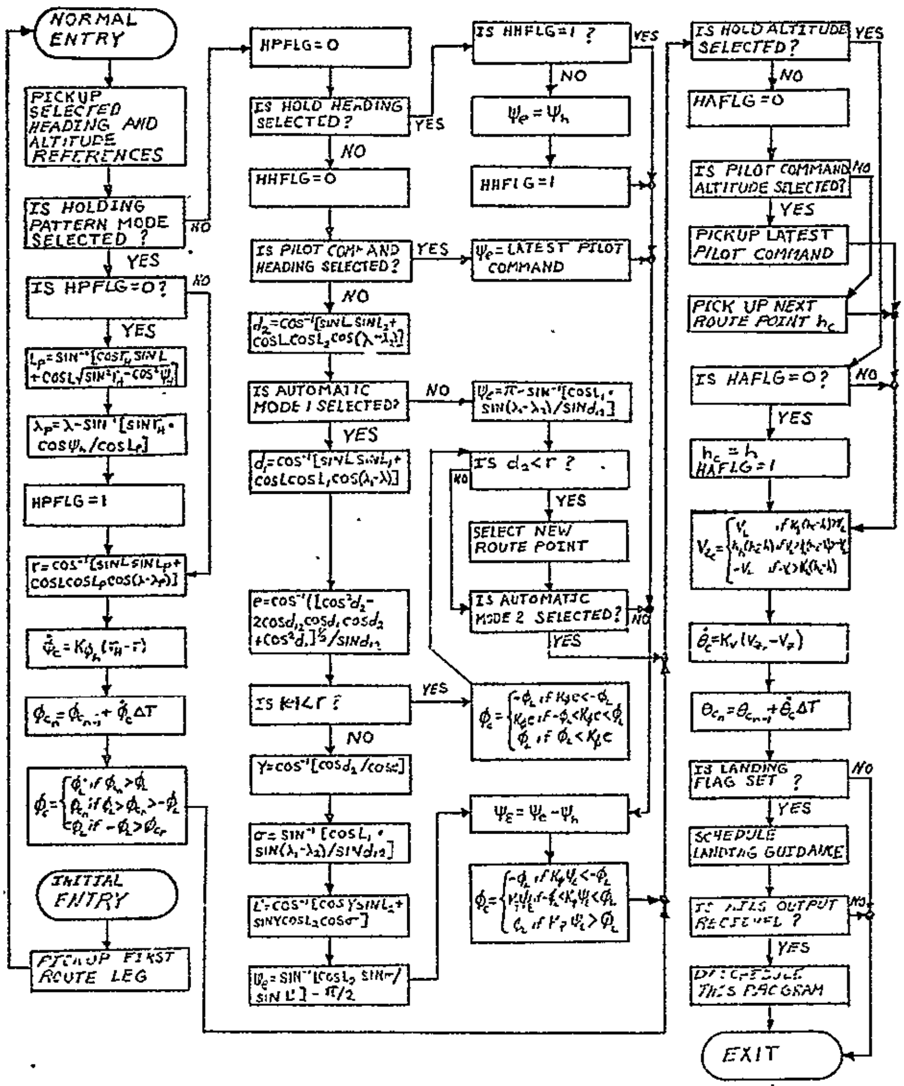


Figure 5-79 Cruise Guidance Program (CRGP) .

second route point as determined by the sequence table. The first route point will normally be the desired reentry target point. The pilot has the ability to modify both the sequence table and route point table. After establishing the first flight leg the program proceeds to the normal entry point.

At the normal entry point the heading and altitude reference value is selected. The heading references that can be selected by the pilot are:

Inertial Navigation System

Magnetic Flux Gate Compass

TACAN

The altitude references that can be selected by the pilot are:

Inertial Navigation System

Air Data Computer

Radar Altimeter

The desired reference is determined by a switch setting on the pilot's console. A test is then performed to determine if the holding pattern mode is selected. If the pilot has selected the holding pattern mode a flag (HPFLG), which is set to zero whenever the holding pattern mode is not selected, is tested. If the flag is 0 this is the first pass through the program with the holding pattern mode selected and the latitude and longitude of the circular flight path center is calculated and the flag is set equal to 1. The computation of a roll command based upon the distance from the holding pattern center is then computed for either a normal or initial holding pattern entry.

If the holding pattern mode is not selected, the holding pattern flag (HPFLG) is set equal to zero and a test made to determine if the pilot has selected a hold heading mode. If a hold heading mode is selected a flag (HHFLG), which is set equal to zero whenever the hold heading mode is not selected, is tested. If the flag is zero indicating hold heading has just been selected, the heading command is set equal to the present heading reference value and the hold heading flag set equal to 1. If the hold heading flag was 1 when tested or after being set to 1,a heading error is computed and converted to a roll guidance command.

If the hold heading is not selected, the hold heading flag is set equal to zero and a test is made to determine if the pilot has selected a pilot commanded heading. If a pilot commanded heading is selected the latest pilot heading entry is picked up and used as a heading command. A heading error is then computed and converted to a roll command by the same set of instructions used in the hold heading mode.

If a pilot command heading is not selected then an automatic heading mode must be selected. The distance to the next route point is computed and a test made to determine which automatic mode is selected. If automatic mode number 1 is selected the distance from the booster to the desired flight path is computed. This distance is then tested against a prestored constant. If the distance is less than the prestored constant, a roll command proportional to the distance from the desired flight path is computed. If the distance is greater than the prestored constant a heading command is computed which will require the booster to fly normal to, and toward the desired flight path. The heading command is used to compute a heading error and roll command by the same instructions used in the hold heading modes.

If automatic mode number 2 is selected, a heading command is computed which will cause the booster to fly toward the desired route point. A test is then performed to determine if the booster is within a distance r of the desired route point. This same test is performed on that branch of the program where automatic mode number 1 is selected and the booster is within the prestored distance from the desired flight path. If the booster is within the distance r of the next route point, a new flight leg is selected by taking the next point in the sequence table. At the time of selecting a new flight leg the flight leg distance (d_{12}) is computed. A test is then made to separate the program path for the automatic mode number 2 program. If automatic mode number 2 is selected, a heading error and roll command is computed using the same instruction set used for the hold heading program.

After computing the roll command, a test to determine if hold altitude is selected. If hold altitude is selected, a test of a flag (HAFLG), which is set equal to zero whenever hold altitude is not selected, is made. If the flag is zero the altitude command is set equal to the present altitude and the flag set equal to 1. If hold altitude is not selected the hold altitude flag is set equal to zero and a test made to determine if pilot command altitude is selected. If pilot command altitude is selected the latest entered altitude command is picked up. If pilot command altitude is not selected, an automatic altitude mode must be selected and the altitude command is obtained from the next route point in the route point tables.

A pitch command is computed from the altitude command. A test is then performed to determine if the landing mode should be entered. This test is based on either a pilot input command or a flag stored in the route point at which landing sequence should be initiated. If the landing flag is set, landing guidance is scheduled by scheduling the Advanced Instrument Landing System program. As soon as this program indicates that it is receiving inputs from the AILS receiver, the cruise guidance program is descheduled. The AILS provides guidance inputs for the remainder of the flight.

5.5 Flight Control

The primary function of the flight control system is to control the attitude of the booster. The flight control system receives attitude commands from the guidance system or pilot controls, generates attitude errors between the commanded attitude and actual attitude received from sensors, and issues commands to systems which produce torques on the vehicle which change the vehicle attitude reducing the attitude errors. In performing this function the flight control system must guarantee system stability and maintain acceptable attitude responses which maintain forces and torques on the vehicle below a maximum value where structural damage to the vehicle or physical damage to the crew would result. The commands generated by the flight control system are a highly filtered combination of command and sensor signals. In a digital computer filters are mechanized through the use of difference equations. An example of a difference equation is

$$Y_n = a_1 X_n + a_2 X_{n-1} + a_3 X_{n-2} - b_1 Y_{n-1} - b_2 Y_{n-2} \qquad (1)$$

where Y_n is the filter output and X_n is the input to the filter. The filter output is calculated at a fixed iteration rate of 1/T times per second, i.e., a solution is calculated every T seconds. If X_n is the value of the input at some time t then X_{n-1} is the value of the input at t - T, X_{n-2} at t - 2T, etc. This same relationship holds for Y_n, Y_{n-1}, Y_{n-2}, etc. The parameters a_1, a_2, a_3, b_1 and b_2 are constants which determine the filter characteristics. The computational requirements of mechanizing a digital filter is dependent upon the number of past values required by the filter. In the above example four past values (2 past values of X and 2 past values of Y) are required. In the flight control equations given below the symbol F_m will be used to represent a filter requiring m past values. Statistics on computer requirements for mechanizing a digital filter are

multiply and/or divide instructions	m + 1
all other instructions	4m + 1
total number of instructions	5m + 2
constant storage area	m + 1
variable storage area	m

The flight control equations are dependent upon the method used to produce torques upon the vehicle. Torques are produced by thrust vector control during boost, reaction jets during coast and initial reentry, and areodynamic surfaces during reentry, cruise, landing and ferry operations. During boost the flight control computational requirements in equation form are

$$\theta_E = \theta - \theta_c \qquad (2)$$

$$\psi_E = \psi - \psi_c \qquad (3)$$

$$\phi_E = \phi - \phi_c \qquad (4)$$

$$\mathcal{E}_\theta = K_{P1}\, \theta_E\, F_5 + K_{P2}\, A_Z\, F_5 + K_{P3}\, q_1 F_5 + K_{P4}\, q_2\, F_5 \qquad (5)$$

$$\mathcal{E}_\phi = K_{R1}\, \phi_E\, F_5 + K_{R2}\, p\, F_5 \qquad (6)$$

$$\varepsilon_\psi = K_{y1}\,\psi_E F_5 + K_{y2} A_y F_5 + K_{y3}\, r\, F_5 \tag{7}$$

$$\delta_{TVC1} = \varepsilon_\theta + \varepsilon_\phi \tag{8}$$

$$\delta_{TVC2} = \varepsilon_\theta - \varepsilon_\phi \tag{9}$$

$$\delta_{TVC3} = \varepsilon_\psi + \varepsilon_\phi \tag{10}$$

$$\delta_{TVC4} = \varepsilon_\psi - \varepsilon_\phi \tag{11}$$

$$\delta_E = K_{E1}\,\theta_E F_5 + K_{E2} A_Z F_5 + K_{E3}\, q_1 F_5 + K_{E4}\, q_2 F_5 \tag{12}$$

Equations 2, 3 and 4 generate pitch yaw and roll attitude errors by subtracting the guidance commands from the vehicle attitude. Equations 5, 6 and 7 are used to calculate the desired thrust vector deflections for each vehicle axis. Desired pitch deflection is computed as the proportional sum of filtered pitch attitude error, θ_E, normal acceleration, A_z, and the two pitch rate gyro outputs q_1 and q_2. Each term is filtered separately by a fifth order filter (represented in each case by F_5). The filter constants (6 for each filter) are changed four different times during the boost flight. The gains K_{p1}, K_{p2}, K_{p3} and K_{p4} are continously varied thoughout boost by interpolating a 10 point lookup table for each gain. Desired roll deflection is computed in the same manner as pitch deflection however only two terms, roll attitude error, ϕ_E, and the roll rate gyro output, p, are used. Desired yaw deflection is calculated in the same manner using three terms, yaw error, ψ_E, lateral acceleration, A_y, and yaw rate, r. Equations 8, 9, 10 and 11 are used to calculate the four engine command outputs. During the atmospheric portion of boost it is assumed that the elevons are used to supplement pitch attitude control. The control effectiveness of the elevons is proportional to dynamic pressure. By using the elevons for load relief, the control of loads on the vehicle is automatically phased in and out during high dynamic pressure regions where loads are most severe. The elevon command is calculated using the same terms used in determining E_θ with different filter coefficients and gains.

During coast and the initial reentry phase the reaction jet system is used to control the vehicle attitude. Figure 5-80 shows the assumed reaction control jet arrangement. There are 16 jets mounted in two rings, one located in the forward portion of the vehicle and the other in the aft. Each jet when activated causes torques about two vehicle axis and a linear acceleration along one of the vehicle axis. Figure 5-81 shows the angular and linear accelerations produced by each jet. The DMS computes a desired torque about each axis on each iteration pass through the program. The desired torques are based upon the magnitude of an error computation for each axis. These error equations are

$$E_\theta = K_\theta(\theta - \theta_c) + K_{\dot\theta}\, q_1 + K_q\, \sigma_{\bar\theta}\, F_1 \qquad (13)$$

$$E_\psi = K_\psi(\psi - \psi_c) + K_{\dot\psi}\, r + K_r\, \sigma_{\bar\psi}\, F_1 \qquad (14)$$

$$E_\phi = K_\phi(\phi - \phi_c) + K_{\dot\phi}\, p + K_p\, \sigma_{\bar\phi}\, F_1 . \qquad (15)$$

where θ_c, ψ_c and ϕ_c are the guidance system attitude commands, θ, ψ,

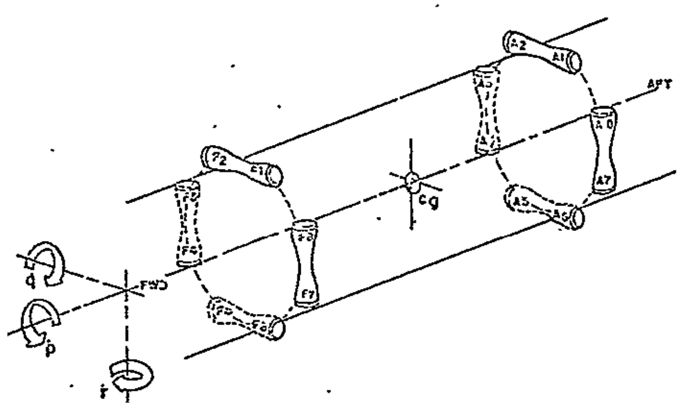


Figure 5-80 Reaction Control Jet-Arrangement

JETS	Accelerations $\dot{p}$	$\dot{q}$	$\dot{r}$	$\ddot{Y}$	$\ddot{Z}$
F1	+	0	+	+	0
F2	–	0	–	–	0
F3	+	–	0	0	+
F4	–	+	0	0	+
F5	+	0	–	–	0
F6	–	0	+	+	0
F7	+	+	0	0	–
F8	–	–	0	0	+
A1	+	0	–	+	0
A2	–	0	–	+	0
A3	+	+	0	0	+
A4	–	–	0	0	–
A5	+	0	+	–	0
A6	–	0	–	+	0
A7	+	–	0	0	–
A8	–	+	0	0	+

Figure 5-81 Accelerations Caused by Each Reaction Jet

and ϕ are the vehicle attitude reference outputs from the strapdown system, q_1, r and p the rate gyro outputs (only one of the pitch rate gyro outputs is used with the reaction jet system), and σ_θ, σ_ψ and σ_ϕ are the pilot commands. The pilot commands are filtered by a first order filter. The detent switches on the pilot's sidestick controller and rudder pedals cause K_θ, K_ψ and K_ϕ to be zeroed when they are activated.

Reaction jet produced accelerations are requested when ever the calculated error exceeds a constant value. The requested angular accelerations are determined according to:

request	$+\dot{p}$	if	$E_\phi < -\varepsilon_\phi$
request	$-\dot{p}$	if	$E_\phi > \varepsilon_\phi$
request	$+\dot{q}$	if	$E_\theta < -\varepsilon_\theta$
request	$-\dot{q}$	if	$E_\theta > \varepsilon_\theta$
request	$+\dot{r}$	if	$E_\psi < -\varepsilon_\psi$
request	$-\dot{r}$	if	$E_\psi > \varepsilon_\psi$

where ε_ϕ, ε_θ and ε_ψ are constants. The reaction jets which are activated at any time is dependent upon the combination of requests. Figure 5-82 indicates the reaction jets that are fired for each possible request combination. The selection logic which must be performed by the DMS as determined from Figure 5-82 is

$$F_1 = \dot{r}_+ \cdot (\dot{p}_0 + \dot{q}_0) + \dot{p}_+ \cdot (\dot{r}_0 + \dot{r}_+) \tag{16}$$

$$F_2 = \dot{r}_- \cdot (\dot{p}_0 + \dot{q}_0) + \dot{p}_- \cdot (\dot{r}_0 + \dot{r}_-) \tag{17}$$

$$F_3 = \dot{q}_- \cdot (\dot{p}_0 + \dot{r}_0) + \dot{p}_+ \cdot (\dot{q}_0 + \dot{q}_-) \tag{18}$$

$$F_4 = \dot{q}_+ \cdot (\dot{p}_0 + \dot{p}_- + \dot{r}_0) + \dot{p}_- \cdot \dot{q}_0 \cdot (\dot{r}_+ + \dot{r}_-) \tag{19}$$

$$F_5 = \dot{r}_- \cdot (\dot{p}_0 + \dot{q}_0) + \dot{p}_+ \cdot (\dot{r}_0 + \dot{r}_-) \tag{20}$$

$$F_6 = \dot{r}_+ \cdot (\dot{p}_0 + \dot{q}_0) + \dot{p}_- \cdot (\dot{r}_0 + \dot{r}_+) \tag{21}$$

$$F_7 = \dot{q}_+ \cdot (\dot{p}_0 + \dot{p}_+ + \dot{r}_0) + \dot{p}_+ \cdot \dot{q}_0 \cdot (\dot{r}_+ + \dot{r}_-) \tag{22}$$

$$F_8 = \dot{q}_- \cdot (\dot{p}_0 + \dot{p}_+ + \dot{r}_0) + \dot{p}_- \cdot \dot{q}_0 \cdot (\dot{r}_+ + \dot{r}_-) \tag{23}$$

$$A_1 = \dot{r}_- \cdot (\dot{p}_0 + \dot{p}_+ + \dot{q}_0) \tag{24}$$

$$A_2 = \dot{r}_+ \cdot (\dot{p}_0 + \dot{p}_- + \dot{q}_0) \tag{25}$$

$$A_3 = \dot{q}_+ \cdot (\dot{p}_0 + \dot{p}_+ + \dot{r}_0) \tag{26}$$

$$A_4 = \dot{q}_- \cdot (\dot{p}_0 + \dot{p}_- + \dot{r}_0) \tag{27}$$

$$A_5 = \dot{r}_+ \cdot (\dot{p}_0 + \dot{p}_+ + \dot{q}_0) \tag{28}$$

$$A_6 = \dot{r}_- \cdot (\dot{p}_0 + \dot{p}_- + \dot{q}_0) \tag{29}$$

$$A_7 = \dot{q}_- \cdot (\dot{p}_0 + \dot{p}_+ + \dot{r}_0) \tag{30}$$

$$A_8 = \dot{q}_+ \cdot (\dot{p}_0 + \dot{p}_- + \dot{r}_0) \tag{31}$$

REQUEST $\dot{p}$	$\dot{q}$	$\dot{r}$	JETS F1	F2	F3	F4	F5	F6	F7	F8	A1	A2	A3	A4	A5	A6	A7	A8
0	0	0																
0	0	–		X			X				X					X		
0	0	+	X					X				X			X			
0	–	0			X					X				X			X	
0	–	–		X	X		X			X	X			X		X	X	
0	–	+	X		X			X		X		X		X	X		X	
0	+	0				X			X				X					X
0	+	–		X		X	X		X		X		X			X		X
0	+	+	X			X		X	X			X	X		X			X
–	0	0		X				X										
–	0	–		X		X	X			X	X					X		
–	0	+	X			X		X		X	X					X		
–	–	0		X	X			X		X				X			X	
–	–	–		X						X				X		X		
–	–	+						X		X		X		X				
–	+	0		X		X		X	X				X					X
–	+	–		X		X										X		X
–	+	+				X		X				X						X
+	0	0	X				X											
+	0	–		X	X		X		X		X					X		
+	0	+	X		X			X	X			X			X			
+	–	0	X		X		X			X				X			X	
+	–	–			X		X				X						X	
+	–	+	X		X										X		X	
+	+	0	X			X	X		X				X					X
+	+	–					X		X		X		X					
+	+	+	X						X				X		X			

Figure 5-82 Jet Selection per Acceleration Request

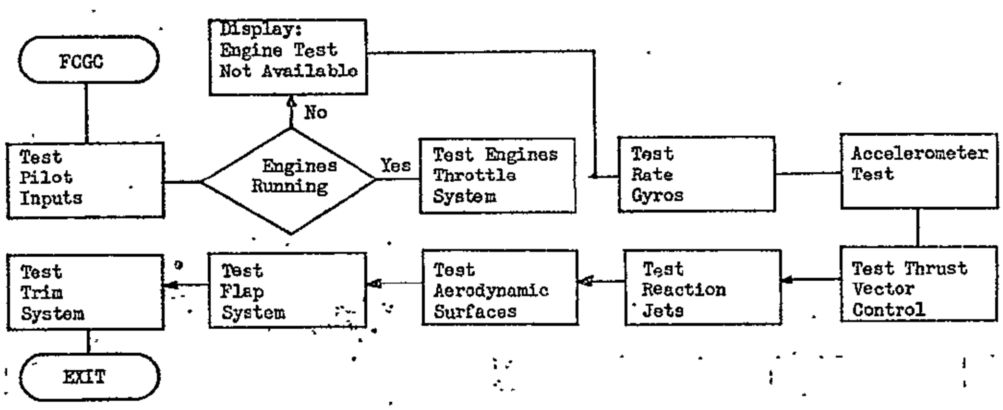


Figure 5-83 Flight Control System Ground Checkout (FCGC)

The reaction jet system remains effective until dynamic pressure, Q, exceeds a preset value, Q_{max}, at which time the reaction jet system is shut down.

The aerodynamic surface control system is activated at the same time that the reaction jet system is turned on, i.e., at the end of powered boost. The control system errors for each axis for the aerodynamic control inputs for pitch and roll are given by the formulas:

$$A_\theta = K_\theta (\theta - \theta_c) + K_{\dot\theta} q_1 + K_G \sigma_\theta F_1 \qquad (32)$$

$$A_\phi = K_\phi (\phi - \phi_c) + K_{\dot\phi} p + \sigma_\phi F_1 \qquad (33)$$

The control system gains K_θ and K_ϕ are set equal to zero if the pitch or roll pilot command values are greater than the detent position. The gain K_θ is derived from a three dimensional table lookup procedure as a function of α, angle of attack, M, Mach number and Q, dynamic pressure. The gain K_G is constant until dynamic pressure is greater than Q_{max} and then is determined as a function of 1/V where V is the vehicle total velocity.

The yaw aerodynamic error is determined from

$$A_\psi = K_{R1} r F_1 + X - K_{cF} p + \sigma_\phi F_2 \qquad (34)$$

if Mach is less than 4 and for Mach greater than 4 is is computed from

$$A_\psi = K_{R2} r + X - K_{CF} p + \sigma_\phi F_2 \tag{35}$$

The gains K_{R2} and K_{R1} are determined from a three dimensional table lookup procedure as a function of α, M, and Q. The gain K_{CF} is computed as a function of α.

The parameter X is equal to the side slip angle β when Q is less than Q_{max} and equal to lateral acceleration when Q is greater than Q_{max}. The commands to the elevons are computed from the filtered aerodynamic errors determined from the formulas

$$\delta_\theta = K_p E_\theta F_9 \tag{36}$$

$$\delta_\phi = K_R E_\phi F_6 \tag{37}$$

$$\delta_R = K_y (E_\psi - \sigma_\psi) F_5 \tag{38}$$

$$\delta_{ER} = \delta_\theta + \delta_\phi \tag{39}$$

$$\delta_{EL} = \delta_\theta - \delta_\phi \tag{40}$$

The gains K_p, K_R and K_y are each computed from a table lookup as functions of M, α and Q. The magnitudes of δ_θ, δ_ϕ, δ_R and σ_ψ are each limited in absolute magnitude before being used.

Included in the flight control system is automatic throttle control of the cruise air breathing engines. The throttle position is computed from

$$\delta_{TC} \doteq K_1 \delta_F + K_2 V_R + K_V (V_R - V) + (V_R - V) F_1 + \theta F_2 \qquad (41)$$

where δ_F is the throttle feedback position and V_R the commanded reference velocity. The term $(V_R - V)\ F_1$ is limited before being applied in the above equation.

The flight control function is accomplished through the use of several programs in the DMS. Figure 5-83 is a flow diagram of the flight control ground checkout program (FCGC). The program sequentially tests all of the flight control systems. Upon entering the program the pilot and copilot sidestick controller and rudder pedal functions are tested. Figure 5-84 is a detailed flow diagram of the testing required for this function. By use of the display system the DMS makes a request for the station select switch to be placed in the pilot position. A flag is then set to zero for use in the program for selection of pilot inputs and a wait programmed to allow for the crew to respond to the request. Upon completion of the wait the station select switch is tested and a request issued to place the sidestick controller and rudders in the zero position. Another wait is then programmed to allow the crew to respond and all inputs from the sidestick controller and rudders tested for proper outputs. The detent switches are then tested and should be off. A

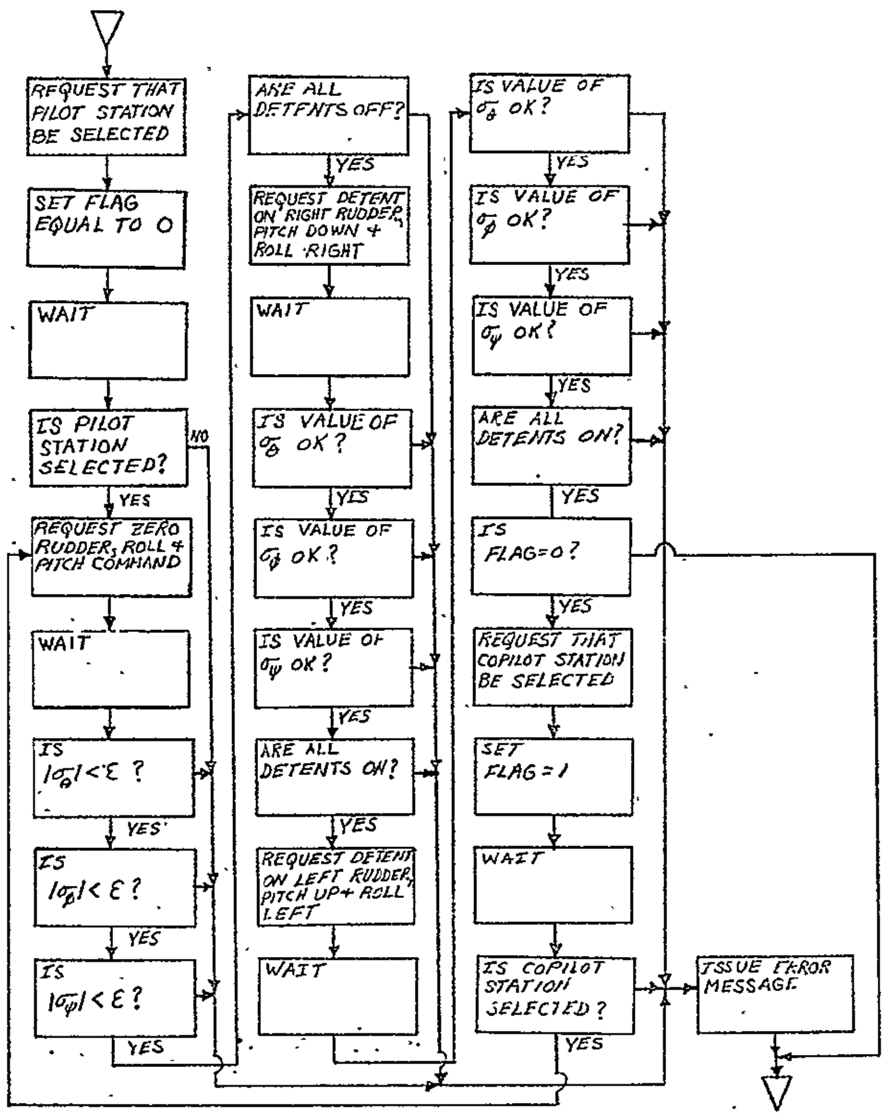


Figure 5-84 Pilot/Copilot Controls Testing

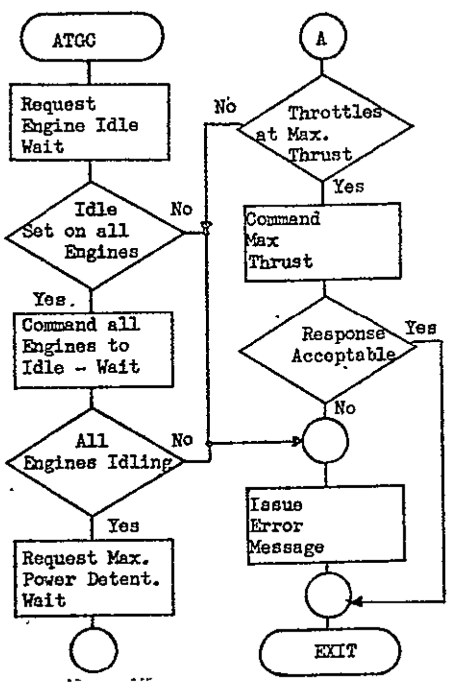
request is then issued to place the rudders and sidestick controller into the detent position for right rudder, pitch down and roll right. A wait is programmed to allow for the crew to respond and then all inputs including detent switches tested. A request of detent position for left rudder, pitch up and left roll is then made, a wait programmed and then all inputs tested for proper value. The flag is then tested and if zero a request for the station switch to be placed into the copilot position is made. The flag is then set to one and a wait programmed. The station switch is then tested for copilot position and the program transferred to the controls testing programming for tests of the copilot station. With the flag equal to one at the flag test, the control section of the flight control ground checkout program is exited. If any test fails an error message is displayed.

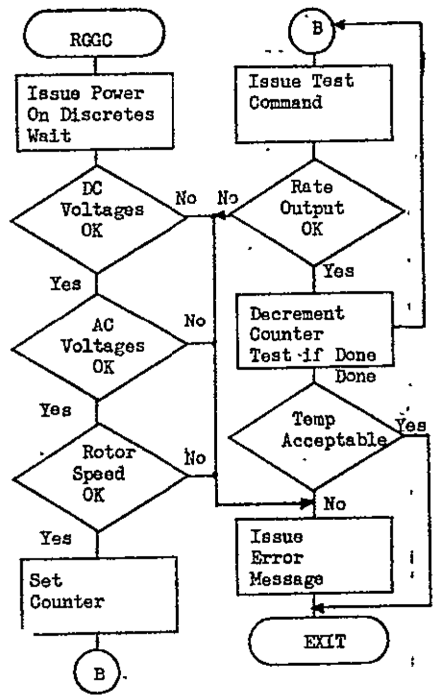
Upon completion of the testing of the pilot and copilot controls a test is made to determine if the cruise air breathing engines are operating. If they are not operating the test of the autothrottle interface must be bypassed. If they are operating the testing shown in Figure 5-85 is conducted. First a display request is made to place the throttles for all six engines in the idle detent position. A wait is then programmed to allow sufficient time for the crew to respond to the request. A test is then made to determine if the throttles have been placed in detent. A test of both the detent discretes and throttle position is made. A command of idle thrust is then issued to each engine and a wait programmed to allow for the engines to respond. Each engine response signal is then tested to determine if each engine is idling. A display request is then made for the crew to place the throttles in the maximum power position. A wait is then programmed to allow for crew response and then the throttle positions and detents are tested for each engine. The engines are then commanded to maximum thrust and their response compared with expected response values stored in the computer. An error message is issued if any test fails.

Referring to Figure 5-83 the next test performed is a test of the four rate gyros. Each rate gyro is tested by the procedure shown in the flow diagram of Figure 5-86. The testing on all four rate gyros is performed simultaneously. The power on discretes for both the AC and DC power are issued and a wait programmed to allow for power transients to decay and rotor speed to build up to its operating value. The AC and DC supply voltages and the rotor speed are then tested.
A counter is then initialized to count the seven test command outputs. A loop is then executed where each test configuration is commanded and the resultant rate gyro output tested for being within desired limit values. Upon completion of the test configuration testing the temperature of the rate gyro electronics is tested.

The next flight control ground checkout tests to be performed is the checkout of the linear accelerometers and the main engines thrust vector

Figure 5-85 Auto Throttle Ground Checkout(ATGC) Figure 5-86 Rate Gyro Ground Checkout(RGGC)





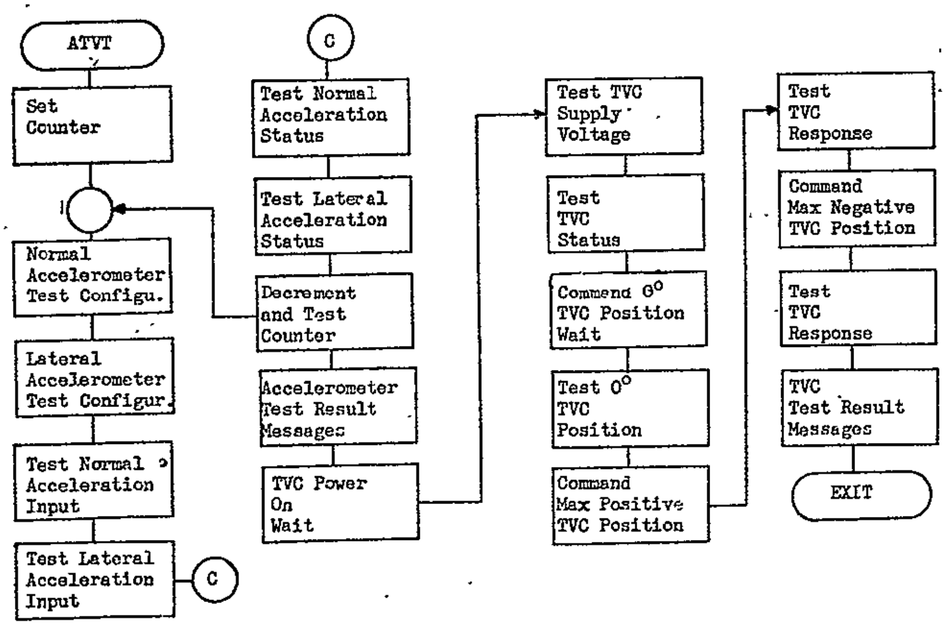


Figure 5-87 Accelerometer and Thrust Vector Control System Testing(ATVT)

control servos. A detailed flow diagram of this testing is shown in Figure 5-87 . A counter is set which initializes a loop for testing both accelerometers in each of three test configurations. With each pass through the testing loop both the normal and lateral accelerometers are commanded to a test configuration and their outputs and status tested. The thrust vector control (TVC) servos are tested by first issuing a power on discrete to each of the twelve TVC systems and waiting for power turn on transients to decay. The DC supply voltages to each servo and the status indications from each servo are then tested. All twelve servos are then commanded to a zero position which requires the issuance of four command words. A wait is then programmed to allow the servos to achieve zero position. The position of each of the twelve servos is then tested to determine if its position is within a minimum range about zero degrees. All servos are then commanded to a maximum positive deflection. Their response is tested by comparing their sampled instantaneous positions with prestored limits. All servos are then commanded to their maximum negative position and their response tested. If any errors are encountered an error message is issued.

The remaining preflight flight control ground checkout includes the testing of the reaction jet system, the servos for the aerodynamic control surfaces, the flaps system and the trim system. Figure 5-88 is a detailed flow diagram of the program required for this testing. All of the reaction

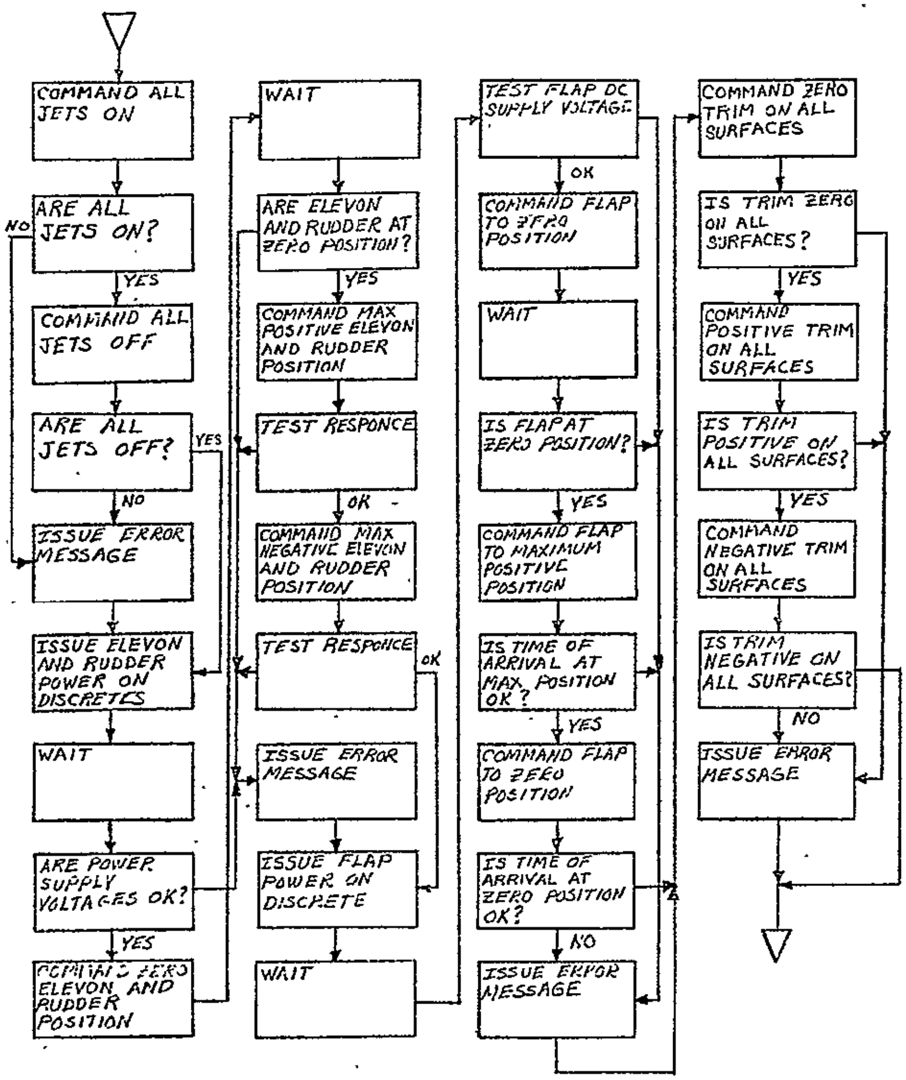


Figure 5- 88 Reaction Jet and Aerodynamic Controls Checkout

jets are commanded on and a test made of their returned status to determine that they are on. All reaction jets are then commanded off and tested for being off. A power on discrete is then issued to the elevons and rudder servo systems. A wait is then programmed to allow for power transients to decay and then the supply voltages tested. Both elevons and the rudder are commanded to zero position and a wait programmed to allow the servos to respond to the command. Their positions are then tested for being at zero within a preset limit. The elevon and rudder servos are then commanded to their maximum positive position and their output compared with prestored values as they respond to the command. The servos are then commanded to their maximum negative position and their response tested. A power on discrete is issued to the flap system and a wait programmed to allow for power transients to decay. The flaps are then commanded to zero degrees and a wait programmed to allow the flaps to respond. The flaps are then commanded to full deflection and a test performed to determine that the time required to reach full deflection is within preset limits. The flaps are then commanded to zero deflection and their response time again tested. The trim system is tested by commanding all surfaces to zero trim, then positive trim and then negative trim and testing for the correct return status for each case.

During boost there are three programs which perform all of the flight control computations. One program performs all of the slow iteration rate

PRECEDING PAGE BLANK NOT FILMED

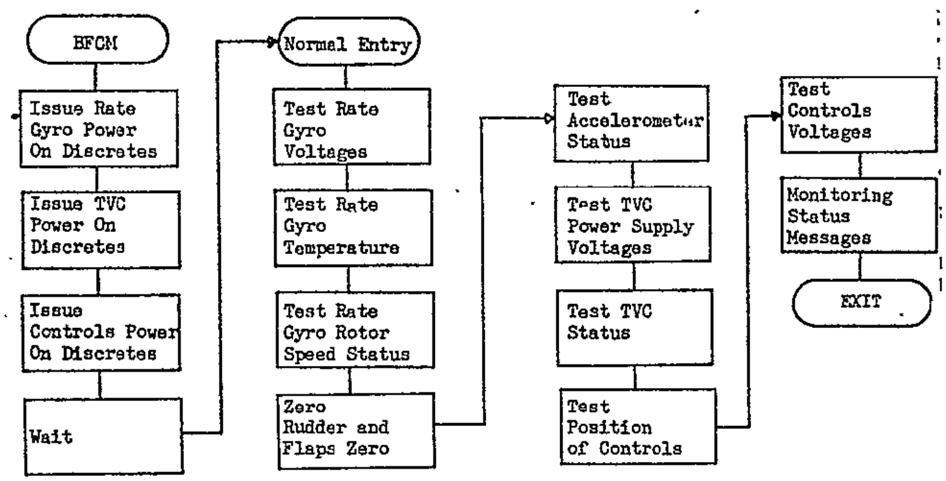


Figure 5-89 Boost Flight Control Monitoring Program (BFCM)

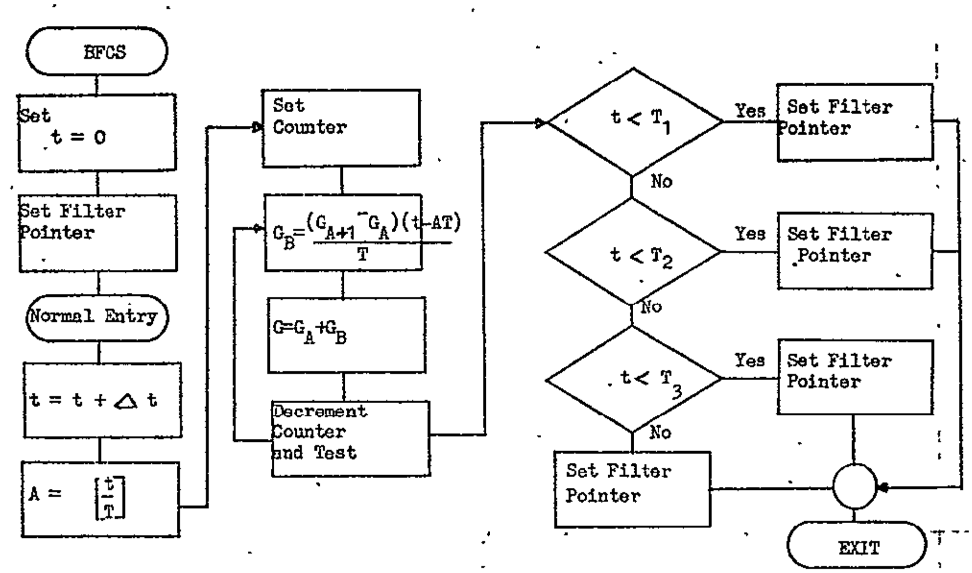


Figure 5-90 Boost Flight Control Gain and Filter Scheduling (BFCS)

flight control functions require that 13 gains be continously varied as a function of time with each gain being constructed by interpolation between values in a 10 point table lookup array. There exists a different table lookup array for each of the 13 gains. Generation of each gain value is achieved by the process of first computing the integer A which is the largest integer contained in the ratio t/T where T is the time increment between table values. Each gain is then computed by applying the formula

$$G = G_A + \frac{(G_{A+1} - G_A)}{T}(t - AT) \tag{42}$$

where G_A and G_{A+1} are the $A^{\underline{th}}$ and $(A+1)^{\underline{th}}$ table entry. This formula is computed once for each gain value. Testing is then performed to determine if t is between 0 and T_1, T_1 and T_2, or T_2 and T_3 or greater than T_3 and a pointer set for the filter coefficients dependent upon the region in which t appears. The gain values and filter pointer are passed to the Main Boost Flight Control Program.

Figure 5-91 is a flow diagram of the Main Boost Flight Control Program (MBFC). The program first tests the thrust vector control servos to determine if they have properly responded to the previous command outputs and then performs the boost flight control computations previously described. Errors in TVC position cause an error message to be displayed and an abort to be initiated if multiple failures are indicated.

Several programs are used by the flight control system during coast, reentry, cruise and landing. Figure 5-92 is a flow diagram of the Reaction Jet Control Program (RJCP). The program computes reaction jet pitch, roll and yaw error from the rate gyro inputs, the attitude data provided from the strapdown system, the attitude command from the guidance program and filtered pilot inputs received from a lower iteration rate control program. The attitude errors are tested against limit values to determine attitude commands. The attitude commands are used in the jet selection logic to generate individual jet commands. Before the jet commands are issued a test is made to determine if the previous response of the reaction jets was as commanded. An error message is generated if any reaction jet failure is indicated. This program is run at a 32 per second iteration rate.

Figure 5-93 is a flow diagram of the Aerodynamic Control Surface Program (ACSP). This program is run at a 32 per second iteration rate. The program computes pitch and roll errors as a function of pitch and roll rate, pitch and roll attitude reference, pitch and roll attitude command inputs. The pilot command filtering is done by a slower iteration rate flight control program. Yaw error is computed by one of two different formulas, dependent upon the value of Mach number. Yaw error is a function of yaw rate, roll rate, filtered pilot roll input command and either side slip angle or lateral acceleration dependent upon aerodynamic pressure. Aerodynamic surface commands are then generated by filtering and limiting and then subtracting directly from yaw error. Right and left elevon commands

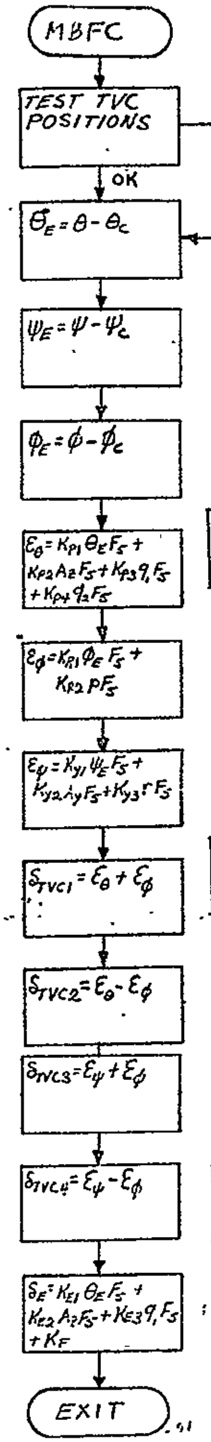


Figure 5- 91 Main
Boost Flight Control Program (MBFC)

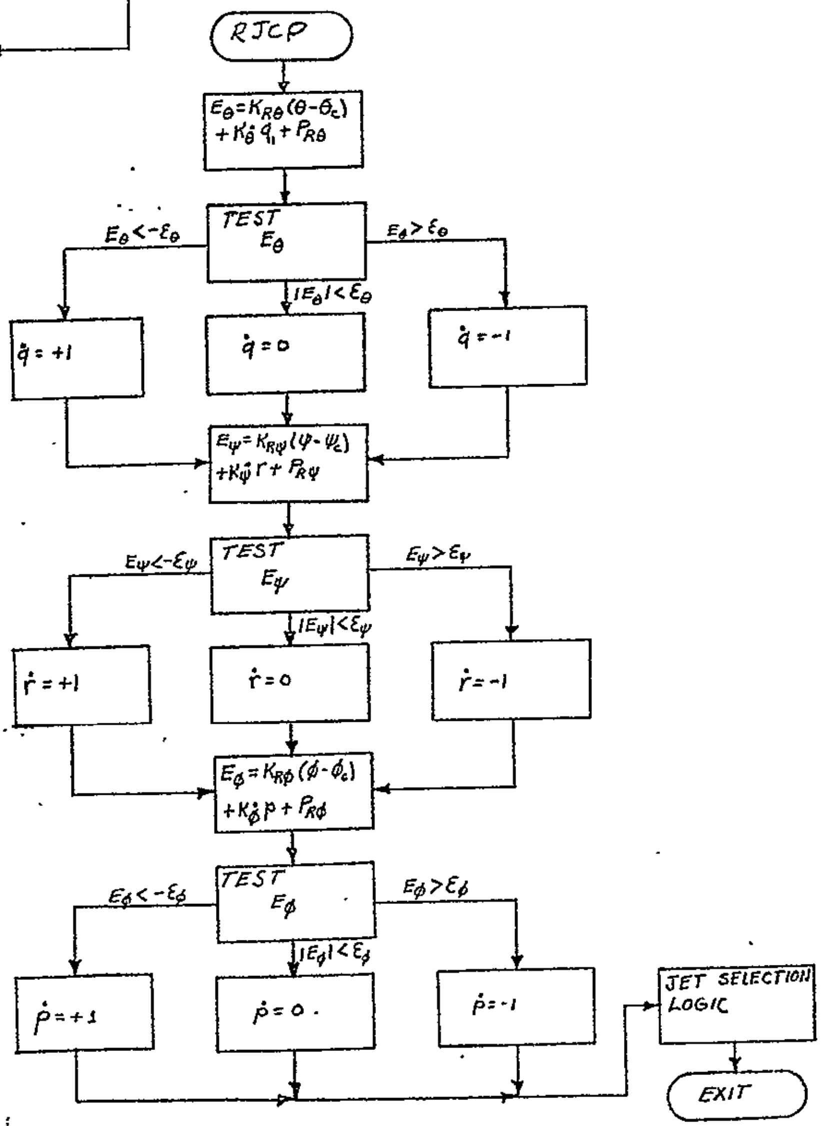


Figure 5- 92 Reaction Jet Control Program (RJCP)

are formed from pitch plus roll and pitch minus roll commands. The major control gains used in the computations are functions of sensed aerodynamic parameters and are generated by lower iteration rate flight control programs.

Figure 5-94 is a flow diagram of the Pilot Command Filtering and Gain Generation Program (CFGG). This program is run at a 16 per second iteration rate. Dynamic pressure is tested and if below a preset value pitch, roll and yaw reaction jet position gains are set to pre-stored values, pilot inputs are filtered, the yaw error X term set equal to the side slip angle, and the K_G gain set equal to a constant value. If Q is greater than Q_{max} the reaction jet control program is descheduled, X_G is computed as a curve fit function of 1/V and the yaw error X term set equal to lateral acceleration. Aerodynamic control system gains and filtered pilot command outputs for the aerodynamic control system are then formed. The pilot command detent switches are then tested and position gains set equal to zero for both the reaction jet and aerodynamic control systems if the pilot command input is above the detent value.

Figure 5-95 is a flow diagram of the Throttle, Trim, and Flap Control Program (TTFC). This program is run at an 8 per second iteration rate. The pilot selects either manual or automatic throttle control which the program tests and either sets the engine thrust command to the throttle

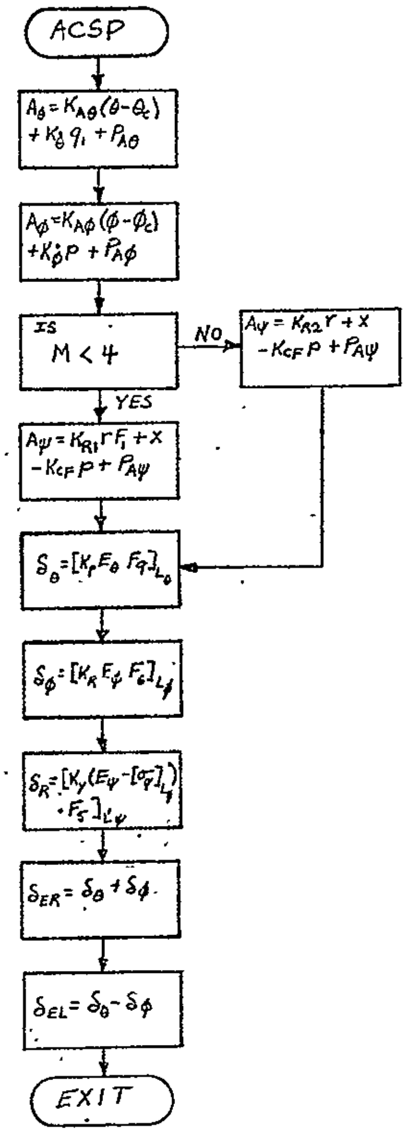


Figure 5- 93 Aerodynamic Control Surface Program (ACSP)

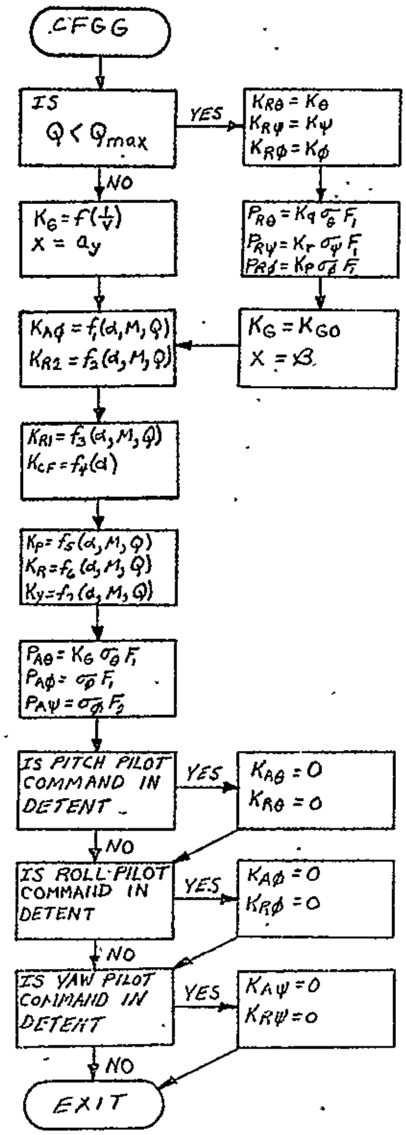
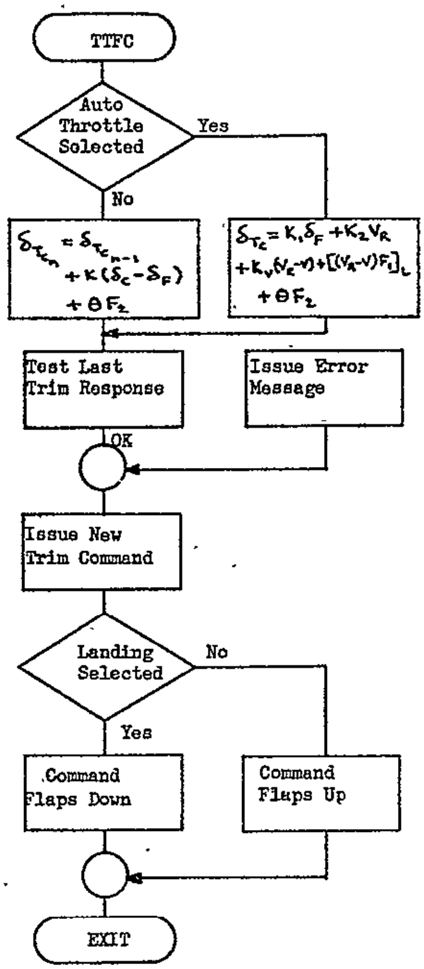


Figure 5- 94 Pilot Command Filtering and Gain Generation Program (CFGG)

position or the computed automatic value. The automatic value is a computed velocity command system compensated with vehicle attitude. The trim system is tested to determine if the system has responded to previous commands. The pilot trim input is then tested and the trim system activated according to the commanded input. The landing system is then interrogated and the flaps commanded down if the appropriate landing condition exists.

Figure 5- 96 is a flow diagram of the Cruise Flight Control Monitoring Program (CFCM). This program is run at a 1 per second iteration rate. This program tests the rate gyro power supply voltages, rate gyro temperatures, the rate gyro rotor status, the lateral accelerometer status, the elevon and rudder position, the elevon and rudder power supply voltages, the flap position, and the flap power supply voltage issuing an error message upon the detection of out-of-tolerance indication.



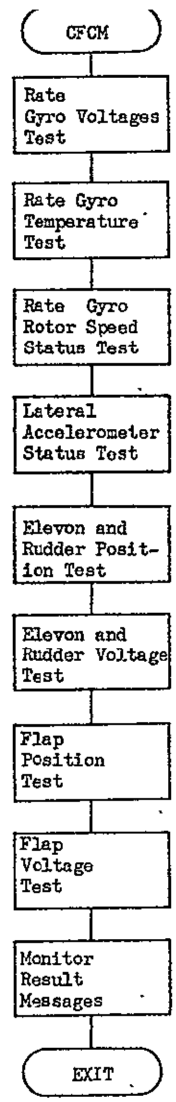


Figure 5-95 Throttle Trim and Flap Control(TTFC)

Figure 5-96 Cruise Flight Control Monitoring (CFCM)

5.6 COMMUNICATIONS

5.6.1 VOICE COMMUNICATIONS

Voice Communications Executive (VCEX) routines are used to supervise voice communication unit usage and provide information required by the Voice Communications Equipment Tester (VCET) routine in performing its functions of testing, initiating, and recording test results and status data. Figure 5-97 is a flow diagram of VCEX.

VCEX is entered under the following situations:

- During the prelaunch checkout
- When a change in voice communication mode is required
- When a request for a status display is made by the flight crew
- When a request for a Voice Frequency Select and Display (VFSD) is made by the flight crew
- During inflight testing

VCEX answers these entries by adjusting the bit patterns of the request and frequency link words. During the prelaunch check-out, all units are tested and an initial status display is formed for the flight crew evaluation. Units scheduled for use during the boost mode are designated by setting the appropriate bits of the request word.

A change in voice unit usage may be required due to a change in mission phase or to failure of a unit in use. For a scheduled change, VCEX has access to a mission phase/communications mode plan which

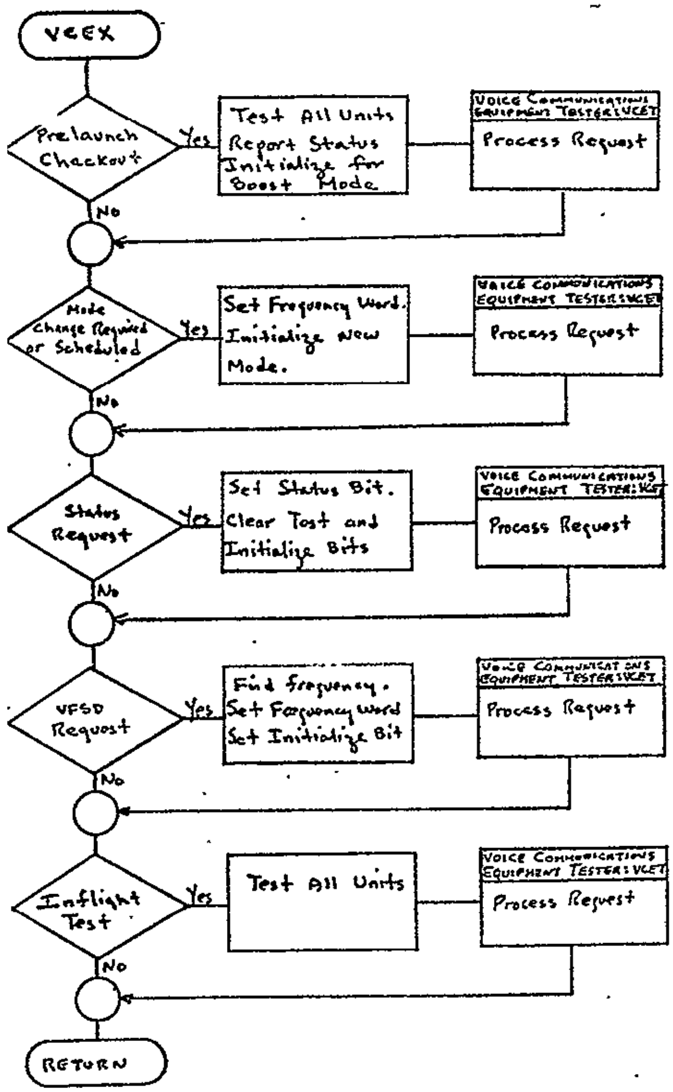


Figure 5-97 Voice Communications Executive

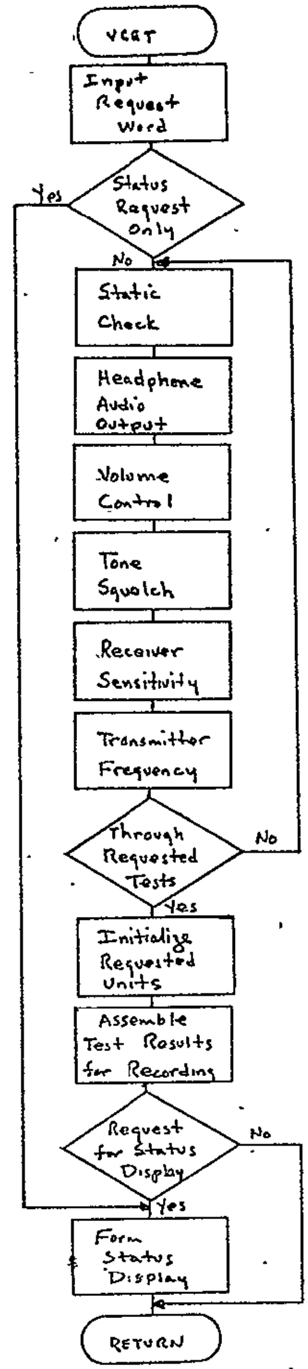


Figure 5-98 Voice Communications Equipment Tester

designates primary and alternate units to be used. VCEX sets the request and frequency link codes as required to establish the schedule. For degraded mode, VCEX examines the status table prepared by VCET, and assigns available units for use.

At any time during the flight, the crew may request a status report through the use of the function keys of the display keyboard. VCEX answers this request by setting the status report bit of the request word and clearing the remainder of the word. VCET then assembles the latest status information and forwards it to the display file. Another function key request activates the VFSD routine. The crew positions the cursor symbol over a geographic location and makes a request to set and display the frequency of the station located at the cursor point. VFSD determines the latitude and longitude of the point and forwards the information to VCEX. VCEX examines the prestored frequency table, correlates the position information with frequency and station name data, and assembles the request words for use by VCET. Name and frequency characters are displayed on the assigned CRT and the appropriate voice unit is switched to the requested station.

Inflight tests are scheduled periodically and VCEX is entered during this time in order to test voice communications equipment. VCEX sets all bits of the test portion of the request word and leaves the initialization bits in their present status so that current communications is not disrupted. VCET then processes this request, reporting any change in status.

Voice Communications Equipment Tester (VCET) procedures are used to test voice communications units, to initialize units according to requests and to assemble test and status data for recording and display. Figure 5-98 is a flow chart of VCET.

VCET is entered from VCEX when action on test, initialization, or status display requests are required. VCEX provides a variable which has bits set in accordance with unit testing or initialization requirements, and status reporting desires. This variable is also used as a link to a table established by VCEX for frequency settings of the units to be activated. VCET first checks to see if status reporting is the only service required, and, if this is the case, assembles the data on the status of voice communication units, and provides a display format for this status reporting. VCET checks switch and frequency settings. Data for units turned on are obtained from prestored tables or from maintenance records updated by VCET. The COMM/SEL switch position indicates which of the two settings for each unit is available for use. VCET then moves the information to the specified display file area.

The variable input from VCEX specifies the units to be tested. VCET masks this word and sets up a loop for conducting the testing operations. Static checks are performed initially to verify resistance measurements, circuit continuity, and circuit impedance at signal inputs. The static checks and the functional tests make use of inputs from the onboard checkout system and prestored limit criteria and parameters. The head phone audio output functional capability is evaluated for a specified amplitude and frequency using prestored modulation index values. Volume control is checked by having the knob set to a maximum position. The audio output amplitude is then measured and evaluated for a specified minimum at the maximum volume setting. Tone squelch operation will mute an otherwise active receiver audio output. The receiver RF carrier must contain both tone and voice modulation components to enable the audio output. The tone squelch function is tested for its capability to sense and react to the tone component at each of two separate RF carrier frequencies. The receiver output is monitored for the presence and then absence of audio signal when the tone modulation is respectively present and then absent from the receiver carrier.

Receiver sensitivity tests are made at the maximum and minimum frequencies of the unit being tested, as well as four other frequencies chosen at random across the units operating band. Transmitter frequencies accuracy and power output checks are also made at these same frequencies. All these tests are conducted without flight crew intervention. Following completion of the tests for particular unit, VCET checks to see if other units are required to be tested. The procedures are repeated if additional checks are required.

When the tests are completed, VCET assembles the test results for maintenance and post flight diagnostic purposes. If a status report is requested, the test data is also used to provide qualitative unit status information and to pin-point LRU's inoperative or in marginal condition. VCET checks the initialization portion of the request word. The units specified are activated and set to the frequencies designated by a table updated by VCEX. Other units not required are deactivated and the information for the voice communication status report is revised.

5.6.2 Command Communications

<u>Up Link Executive</u> (UPEX) procedures perform general supervision of the initialization and testing of command communications equipment, and the processing and execution of messages transmitted to the booster. Figure 5.99 presents a normal mission command communications mode schedule and a flow diagram of UPEX. UPEX is entered when a message is being transmitted from a ground station, a change in mission phase occurs, or a request for equipment testing is sent by the systems executive

A command decoder interrupt is generated by the uplink equipment when a message is received. UPEX recognizes this interrupt and calls on the Up Link Command Processor (UPCP) to process and verify the information. If the message meets the specified criteria, UPCP returns control to UPEX providing information on the type of command received. The command execution procedure is dependent on the booster flight mode. If the booster is unmanned, UPEX prepares the commands for execution by addressing the appropriate subsystem and putting the command information on the data bus. If the booster is manned, UPEX first checks to see if crew execution is desired or required. For both automatic or manual modes, UPEX provides the Displays and Controls routines with the information necessary for message display and reprogammable switch operation.

If a change in mission phase or a request for equipment test is received, the Up Link Communications Initializor (UPCI) routine is entered. UPCI performs functions associated with the communications mode schedule.

FLIGHT MODE \ MISSION PHASE	PRELAUNCH	BOOST	STAGING	COAST	REENTRY	TRANSITION	CRUISE	APPROACH AND LANDING	PREFERRY	FERRY
MANNED	TEST ALL MODES	6	6	6	6	6	7	7	TEST ALL MODES	7
UNMANNED	TEST MODES 1,3,5	5	5	5	5	5	3	3		

COMMAND COMMUNICATION MODE SCHEDULE

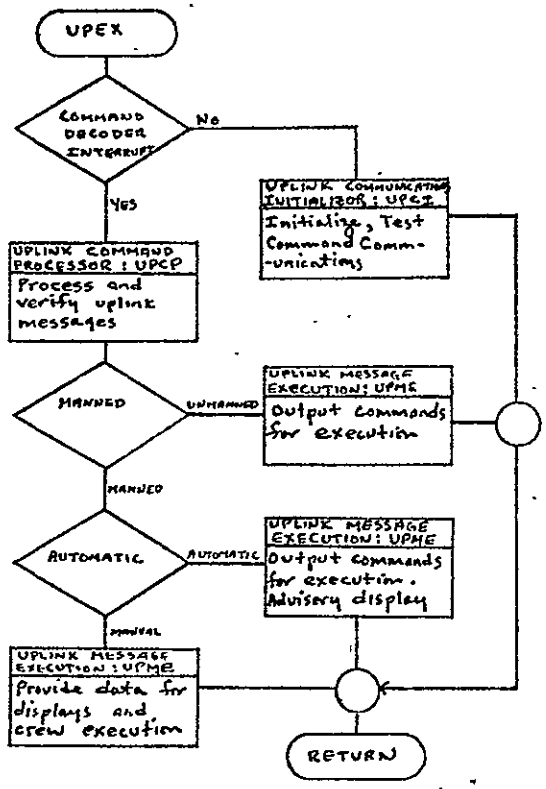


Figure 5.99 Uplink Executive

REAL TIME COMMANDS (RTC)	
Bits	Identification
1-3	Vehicle Address
4-6	System Address
7-12	RTC Instruction

GUIDANCE AND NAVIGATION DATA (GND)	
Bits	Identification
1-3	Vehicle Address
4-6	System Address
7	Parity
8-12	Symbol
13-17	Symbol Complement
18-22	Symbol Repeat

STORED PROGRAM COMMAND (SPC)	
Bits	Identification
1-3	Vehicle Address
4-6	System Address
7-14	Message Control
15-22	Information
23-36	Complement

CENTRAL TIMING EQUIPMENT (CTE)	
Bits	Identification
1-3	Vehicle Address
4-6	System Address
7-12	Seconds
13-18	Minutes
19-24	Hours
25-30	Days

Figure 5-100 Command Word Formats

After initialization and testing equipment, UPCI prepares status messages for crew display interpretation and forwards maintenance data for recording or telemetering.

<u>Up Link Command Processor (UPCP)</u> procedures are used to interpret, process and verify uplink messages. UPCP formats status and error messages for telemetry and provides information to UPEX for completion of the requested command. Figure 5-100 are examples of formats specified for command words. Verification of uplink messages is dependent upon information being received in the exact format required for a particular command word. Figure 5-101 is a general flow chart of UPCP. UPCP is entered from UPEX when a command decoder interrupt occurs. UPEX also has responsibility for maintaining or recovering uplink synchronization.

UPCP first validates the vehicle and system address bits against its list of acceptable ones. UPCP branches to verify the particular command word format being sent on the uplink. Figure 5-101 shows the flow for stored program (SPC) verification. (The requirements for real-time command, guidance and navigation data, and central timing verification are similar to those shown for SPC). Up link messages may require more than one 30-bit word. Figure 5-102 lists some possible uplink message types. The first word in each message is a mode command whose information bits refer to the desired mode to be executed. UPCP tests to determine if the mode word is being received. (Mode or data words are distinguished by logical combinations of two of the message control bits). Tests are then made on the format of the mode word. If any of the tests fail, a coded error message is formulated for interpretation by the ground station. Figure 5-103

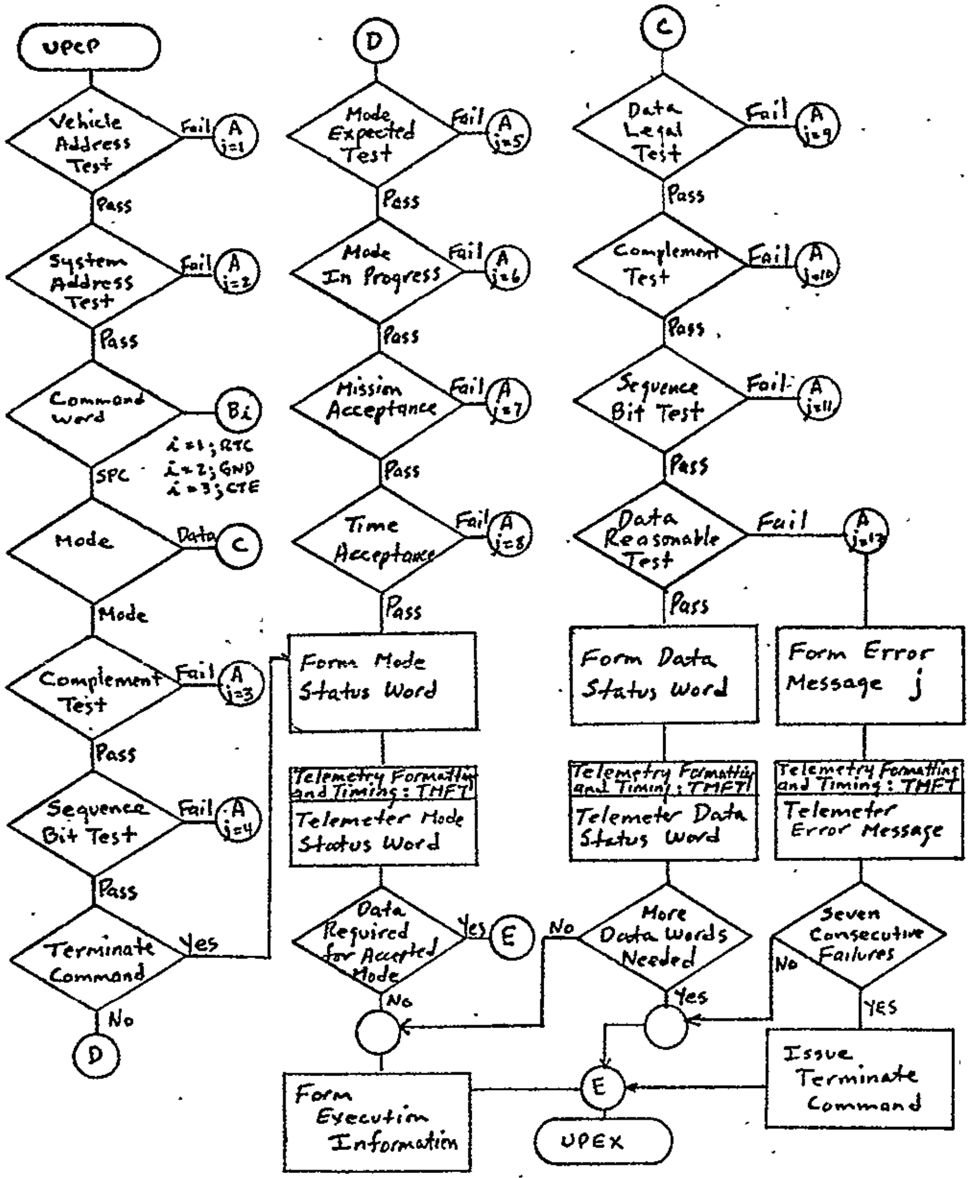


Figure 5-101 Up Link Command Processor

Figure 5-102Mode Command Description and
Data Word Requirements

Mode Command	Name	Description	Data Words Required
1	Terminate	Terminate routines whose data word requirements have not been met. Terminate memory dump if in progress Prepare for new mode.	0
2	Memory Dump	Telemeter the contents of the memory modules specified by the data words.	2
3	Single Memory Location	Telemeter the contents of one specified DMS memory location.	3
4	Mission Time-line Update	Increments or decrements starting times for future mission phases.	1
5	Inhibit Event k	Stop performance of the maneuver specified by the mode command.	0
6	Update Event k	Increment or decrement initiation time for event k.	1
7	Execute Event k	Perform computations for and execute maneuver specified by mode command.	0
8	Alternate Sequence m	Add or omit functions in the preplanned mission schedule as specified by the alternate schedule.	0
9	Switch Selector	Activate or deactivate the switches specified by the data command format.	2

1	Bits 1-3 not a legal vehicle address
2	Bits 4-6 not a legal system address
3	Information bits 15-22 are not the complement of bits 23-30
4	Sequence bit (7) incorrect. Should be 0 for mode command
5	Mode command received when a data word is required to complete the requirements of a previous mode command
6	Another uplink routine is presently being processed
7	The mode command is not defined for the mission
8	The mode command is not acceptable at the time it is received (e.g., event to be modified has occurred)
9	Data command received when a mode command is expected
10	Data bits 15-22 are not the complement of bits 23-30
11	Sequence bit (7) does not meet requirements. 1 for odd-numbered data words, 0 for even-numbered data words
12	Data does not meet reasonableness criteria

Figure 5-103 Command Communications Error Messages

lists the error messages assumed for UPCP validation of SPC messages. UPCP maintains a count of consecutive failures. An exit will be made to UPEX to wait for the next word. Upon return to UPCP, processing is continued until the number of consecutive failures reaches a prespecified number. If this occurs, UPCP originates a terminate command to inform the ground station of communication status.

UPCP first tests if bits 23 to 30 are the 1's complement of the mode command information bits. If this test fails, UPCP does not attempt error correction. Processing of the message is discontinued and the appropriate error message is formed and an exit is made to wait for repetition One of the message control bits is called the sequence bit. This bit must be a 0 for the mode command word.

UPCP next checks to see if a terminate command has been received. If this command is received, other verification tests are bypassed and the ground station is informed of its acceptance. As the terminate command does not require data words, an exit is made to UPEX, where the appropriate action is initiated. If the mode command is not the terminate command, the checks continue with the mode expected test. For each mode, UPCP checks to see if the currect number of data words are received. If a new mode word is received before all the data word requirements are met, then the mode expected test checks to see if the mode is defined for the current flight. The time acceptance test verifies that time criteria are satisfied. For example, if an event to be cancelled has already occurred, the time acceptance test fails.

After all tests have passed, UPCP forms a mode status word consisting of the information bits received plus code bits.

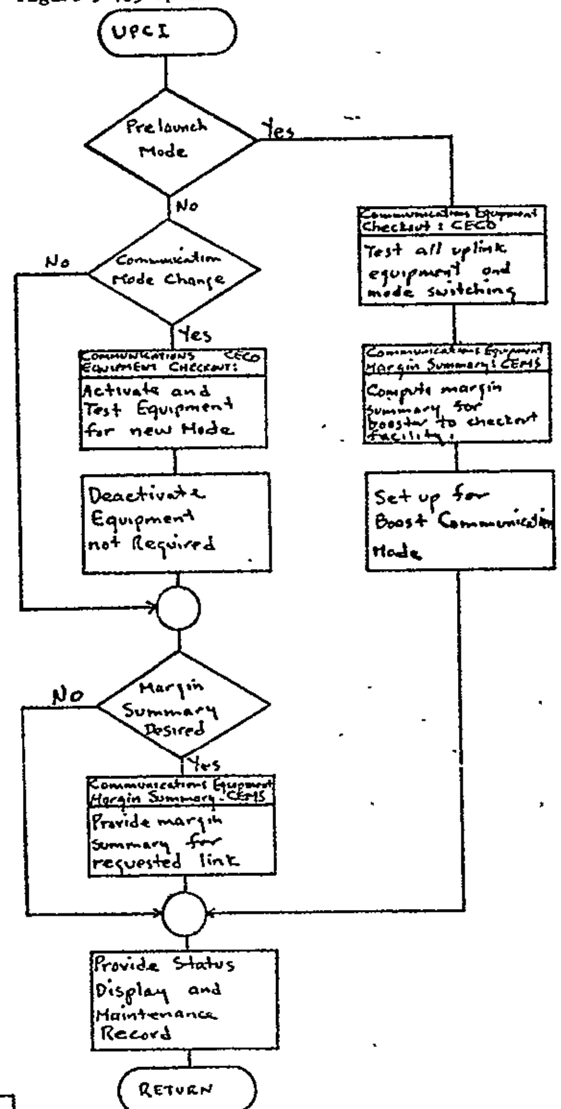
The Telemetry Formatting and Timing (TMFT) routine is then used to transmit the mode acceptance to the ground station. UPCP checks the mode and data word requirements table to determine the number of data words associated with the accepted mode. If no data words are required, then the information, required by UPEX for mode execution, is assembled. If data words are required, an exit is made to UPEX to wait for the next word.

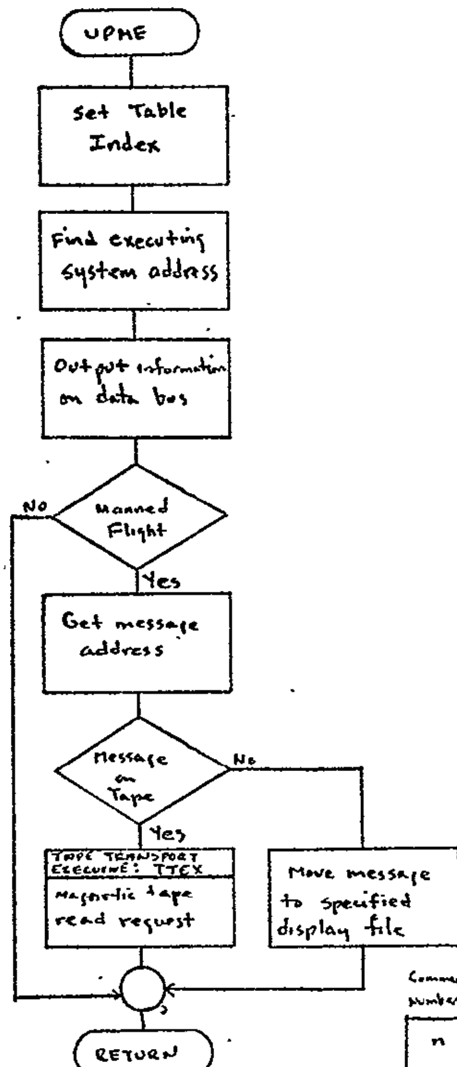
Testing of data words is similar to that performed for mode words. The message control bits inform UPCP that a data word is to be processed. A data legal test checks to see if a data word is expected. If all data word requirements for the current mode have been met, then the next word should be another mode command. The complement test is the same as that conducted for the mode command. For a data word, the sequence bit should be a 1 for odd-numbered words and a 0 for even-numbered words. Finally, reasonableness tests are performed on the data. These tests are primarily logical ones (e.g., time changes illogical, etc.). When the data word passes all the tests, a data status word is formed and telemetered. UPCP repeats this loop if more data words are required. When all the specified number of data words are received, UPEX is provided information necessary to initiate execution of the command message.

Up Link Message Execution (UPME) procedures are used to provide a flow of information to support command messages. Following verification of the uplink messages, UPCP forwards the mode command number and other information to UPME. UPME uses this command number to set a table index as shown in Figure 5-104 , a general flow chart of UPME. A prestored table contains all necessary data for linking the command number with the address of the subsystem responsible for completion of the command. For both automatic and manual modes, the table lists what type of media is used for storing messages, what the address or record number of the message is, and to what display file storage area the message is to be moved.

Using this table index, UPME finds the address of the system scheduled to perform the requested operation. The information necessary for the command is then outputted to the proper subsystem. If the flight is unmanned or the command does not require assistance from or notification to the flight crew, all action necessary for command execution is conducted at the scheduled time. For a manned flight, crew intervention may be required or desired. In this case, UPME returns to the table and determines the source and destination for the display message associated with the command. If the message is on tape, UPME forms a read request for use by the Tape Transport Executive (TTEX) and then exits to this routine. If the message is stored in memory, it is obtained and moved to the designated display file memory area.

Figure 5-105 Up Link Communications Initializor





Command Number	Executing System	Manned Mode	Media	Message Address	Message Destination
n	DMS	Auto	Mem	Address	HSD
	DCS	Manual	Tape	Record #	MFD
n+1	. . . .	. .	. .	. .	. .

Figure 5-104 Up Link Message Execution

Up Link Communications Initializor (UPCI) routines are used to initialize equipment required for the current communications mode, to test and validate this equipment, and to provide status displays for crew evaluation and maintenance records for post flight diagnostics. Figure 5-105 is a general flow chart of UPCI.

UPCI is entered during the prelaunch checkout, whenever a change in communication mode occurs, or when a request for circuit margin calculations or communication subsystem status display is initiated. During the prelaunch checkout, all the uplink equipment is tested and the capability to switch from one communication mode to another is verified. Figure 4-34 lists the various uplink S-band modes of operation and provides information on the equipment required, the modulation techniques used, and the frequencies of the subcarriers. The Communications Equipment Checkout (CECO) routine uses this data as it cycles through the switching from one mode to the next. As crew interface is required for some of the modes, check lists are also employed using tape handler and display subsystem capabilities. Following the completion of CECO functions, the Communication Equipment Margin Summary (CEMS) routine is entered. CEMS established a link with the spacecraft checkout facility or adjacent MSFN facility. Then using standard range equations, the circuit performance margin is computed. CEMS analyzes the results, and prepares a summary for display and recording. UPCI then establishes the uplink equipment for the boost phase communication mode.

Communication mode changes may be required following a change in mission phase, through a request from a ground station or by equipment failure necessitating a degraded mode. For any of these cases, UPCI is entered with the appropriate flags and variables so that the required switching can be performed. CECO activates and tests equipment required for the new mode and assembles status display information. Equipment not required for the new communication mode is shut down.

A circuit performance margin summary is performed when communications are initiated with a different ground link, when requested by the flight crew, or scheduled as part of inflight testing. CEMS is entered with the information required to specify the link for which the summary is desired. CEMS then obtains the parameters required for its calculations and assembles the results in summary form. Prior to exiting, UPCI updates the equipment status and maintenance history files and makes them available for display and recording.

Communications Equipment Margin Summary (CEMS) routines are used to compute circuit performance margins for radio links, to analyze the results of its calculations, and to assemble the results in summary format for display or post flight diagnostics. Figure 5-106 is a flow chart of CEMS.

Parameters required for margin calculations are obtained from prestored tables, CECO, or other subsystem routines such as navigation or onboard

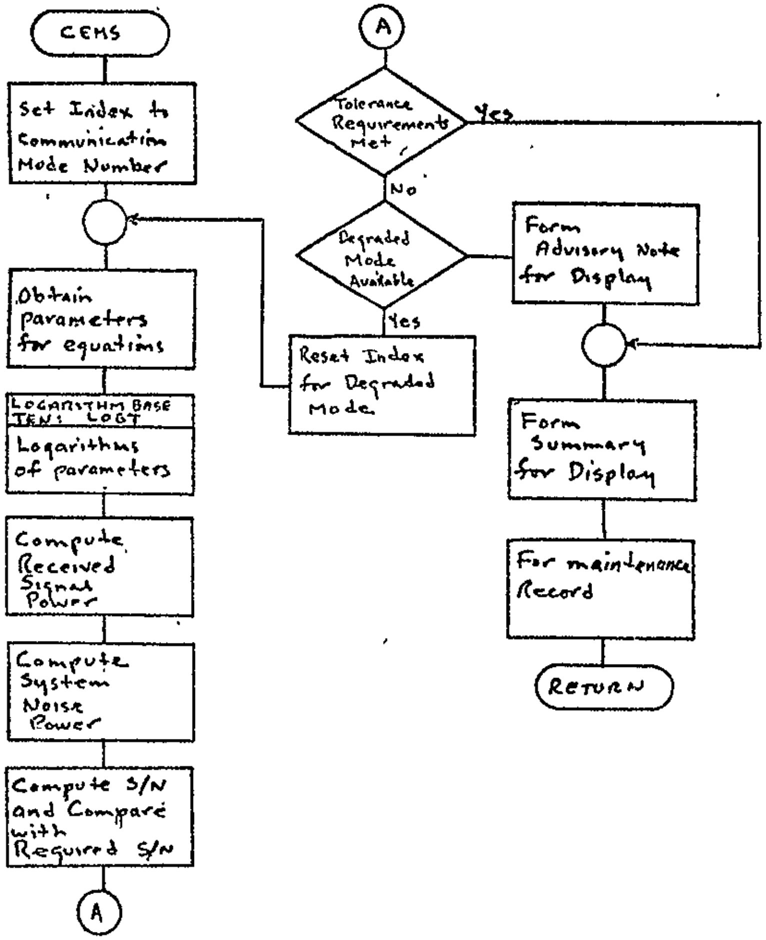


Figure 5-106. Communications Equipment Margin Summary

Mode Description - Uplink Mode 6

Transmitting System	Nominal	+Tol	-Tol
1) RF Power			
2) Circuit Loss			
3) Antenna Gain			
4) Pointing Loss			
Path Parameters			
5) Slant Range			
6) Carrier Frequency			
7) Atmospheric Absorption			
8) Space Loss			
Receiving System			
9) Antenna Gain			
10) Polarization Loss			
11) Circuit Loss			
12) System Noise Temp			
13) Noise Power			
Total Received Power			
Actual S/N			
Required S/N			

Figure 5-107 Margin Calculations Summary Display

checkout. The computation of the circuit performance margin is carried out in two essentially independent calculations: (1) calculation of received signal power, and (2) calculation of noise power.

(1) Calculation of received signal power is performed using the equation:

$$P_r = 10 \log P_t + 10 \log G_t + 10 \log G_r - 10 \log L - 20 \log R - 20 \log f - 37.8$$

where

P_r = received signal power in -dBW

P_t = transmitter power output in dBW

G_t = transmitter antenna gain in db

G_r = receiver antenna gain in db

L = system losses in decibels (antenna pointing, antenna polarization, transmission line losses, power divider losses, etc

$20 \log R + 20 \log f + 37.8$ = space loss factor in decibels, where

R = range in nautical miles

f = carrier frequency in megahertz

(2) Calculation of noise power uses the equation:

$$\text{noise (dBW)} = -228.6 + 10 \log T_s + 10 \log B$$

where

T_s = system temperature ($^{\circ}$K)

B = noise bandwidth (hertz)

The system noise temperature is computed from the equation:

$$T_s = \frac{T_a}{L} + (1 - \frac{1}{L})\ 290^\circ + T_r$$

where,

T_a = antenna temperature ($^\circ$K)

T_r = receiver temperature ($^\circ$K)

L = circuit losses from antenna terminal to receiver input in ratio form

T_a is obtained from prestored tables for systems likely to be linked to the booster. T_r is the noise generated within the receiver itself and is specified by the noise figure (N_f) given in decibels. T_r is computed from the equation:

$$T_r = (N_f - 1)\ 290^\circ$$

Using the above equations an actual S/N (signal to noise) ratio is computed. Associated with each link is a required S/N ratio for receiver threshold, high intelligibility, low error rate or other performance criteria. Tolerances are also specified for each performance factor. CEMS compares computed and required ratios to determine if system performance is within allowable tolerances. The computations are then summarized in a format similar to Figure 5-107 and made available for display. The results are also formatted for recording.

5.6.3 TELEMETRY

Telemetry programs are required to control the formatting and timing of down link data. Associated with this requirement are data compression routines and communication mode control. Test and calibration routines check the operability and accuracy of telemetry subsystem equipment. (Telemetry programs associated with up link routines, such as verification loop and test request, are described in Section 5.6.2 .

It is anticipated that telemetry requirements will be a function of mission phases and flight mode (manned or unmanned). For each mission phase a down link communicatiom mode is specified. This mode establishes a telemetry bit transmission rate. This bit rate along with requirements for outputting data at varying sample rates will dictate the TLM format used. Finally, a TLM list organizes the data and fits it into the proper time slots. Fig 5-108 shows a possible relationship between these considerations. Normally, telemetry output requirements will be less for the manned flight than for the unmanned flight, but the type data required will be similar. This implies that a reduced telemetry list can be used during most of the mission phases of manned flights. The contents of the telemetry list primarily depend on mission phases. Three possible arrangements of telemetry list requirements are: 1, used in boost and staging phases; 2, used in the coast, reentry, and transistion to aerodynamic flight phases; and 3, used in ferry flight, cruise, and approach and landing phases. Associated with each of these lists is a reduced list. Telemetry programs select a subset of a telemetry list if transmitted in a reduced state.

Figure 5-108 Telemetry and Mission Phases for Manned and Unmanned Booster

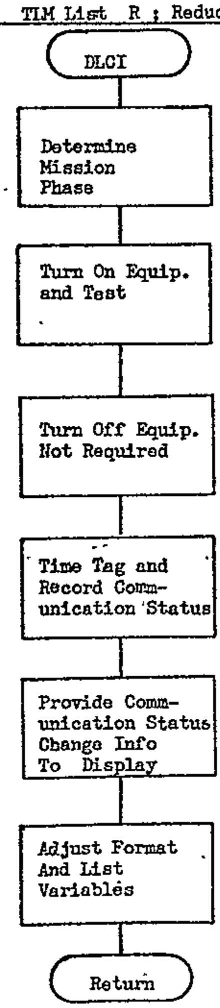
	Mission Phase	Boost	Staging	Coast	Reentry	Transistion	Cruise	Approach	Ferry
Manned	TLM Comm Mode	3	3	3	3	3	4	4	4
	TLM Format	4	4	4	4	4	1	1	1
	TLM List	1R	1R	2R	2R	2R	3R	3R	3R
Unmanned	TLM Comm Mode	2*	2*	2*	2*	2*	1*	1*	1*
	TLM Format	3	3	3	3	3	2	2	2
	TLM List	1	1	2	2	2	3	3	3

* Less Voice

TLM Comm Modes (1 thru 9) defined in Figure 441

TLM Formats (1, Model; 2, Gemini; 3, Apollo High Rate; 4, Apollo Low Rate).

TLM List R ; Reduced



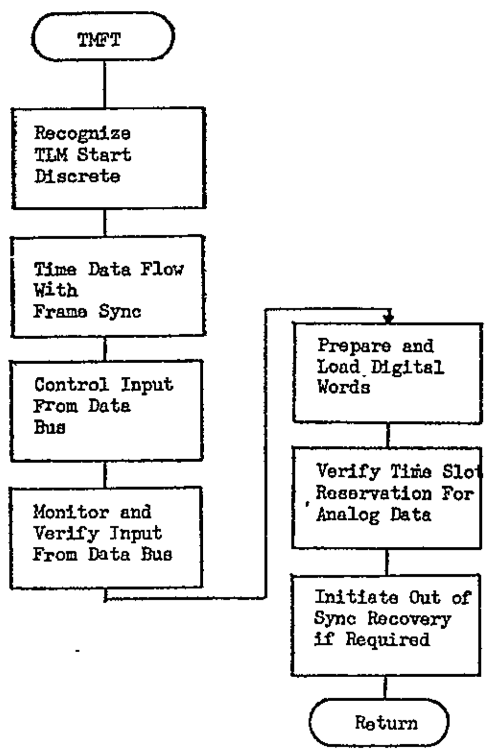


Figure 5-110 Telemetry Format and Timing

Figure 5-109 Down Link Communications Initializor

Program DLCI (Down Link Communications Initializor) is used to initialize and establish the requirements of Figure 5-108 , telemetry and mission phases for manned and unmanned booster. The functions of the routine are:

- Determine, or receive notification of, a change in mission phase.
- Activate and test equipment called for by new down link communication mode.
- Shut down equipment no longer required.
- Update communication status for display and maintenance record.
- Adjust indices, counters and other variables for associated telemetry format and telemetry list.

Figure 5-109 shows the general flow of DLCI procedures. A change in mission phase can be determined by the occurrence of a particular event (boost and orbiter separation, etc.) or by a point in the mission time line. When a mission phase change occurs, DLCI is entered from the EXEC. DLCI determines which mission phase is current from a variable set by EXEC, calls up the matching communication mode and telemetry format,and lists numbers from a table. An additional table contains information on communication equipment associated with each down link mode. If a change in equipment status is required, the new equipment is turned on and the appropriate test subroutine is entered. Equipment not required is shut down. The results of any tests performed and an indication of change in the status of communication equipment is prepared for maintenance history tape recording. The appropriate information is also prepared to notify the flight crew of changes made. Variables associated with different telemetry formats and lists are initialized. These variables, consisting of appropriate flags, counters, indicators, addresses and other parameters, are used by the telemetry formatting and timing (TMFT) routine to control and monitor the output of information via the telemetry equipment.

Program TMFT (Telemetry Formatting and Timing), using the information provided by DLCI, supervises the flow of the various data types into the telemetry subsystem. (It is assumed that some data will bypass the data bus - e.g., computer memory dumps, flight qualification test points, etc.) The functions of TMFT are:

- Establish operating procedures as required by telemetry format and list, and initiate flow of data in response to start and synchronization signals.
- Monitor and control data flow from data bus to PCM output register.
- Load and update parallel digital words for sequencing to telemetry buffer.
- Verify time slot reservation for analog data.
- Monitor synchronization signals and initiate out of sync recovery procedures if required.
- Recognize telemetry stop discretes and inform interested systems of end of transmission.

Figure 5-110 is a general flow chart of TMFT. Although control of the various types of data is shown sequentially, logic, as dictated by format and list requirements, will manage interleaving of data.

Program TMFT, following initialization by DLCI, waits for the occurrence of a telemetry start discrete. It then initiates action required to ensure that data is ready for transmission at the time of frame sync discretes, and that a steady bit flow is maintained. Data from the data bus consists of address, parity,sync, and information bits. TMFT verifies that the data is intended for transmission, checks the information and parity bits (by a COUNT ONES subroutine) to ensure that errors did not occur in data bus transmission, and tests synchronization with the established time standards.

If digital data (computer words) are to be transferred directly to the telemetry equipment, a telemetry buffer (memory area) is loaded with the appropriate parameters. TMFT provides packing and formatting functions. These words will be shifted into the telemetry output register at a fixed rate (e.g., 1600 microseconds for one 32-bit word at 51.2 KBPS rate). TMFT reloads the buffer as required for repeating of the transfer process. Care must be taken in loading the buffer so that telemetry words will not be loaded into an area which is currently being transferred. In addition, it is necessary to maintain synchronization between the transfer cycles and the telemetry cycle.

If analog data, which has not been previously converted and put on the data bus, is to be telemetered, then TMFT verifies its timing and synchronization to ensure there is no interference by other data being transmitted.

Sync bits are tested at specified intervals. If these discretes do not satisfy their required condition at a particular time, then an out of sync condition exists and TMFT will initiate recovery procedures. This primarily results in a delay of valid telemetry data until the proper discrete settings and time considerations are met.

Telemetry is terminated either by completion of a predetermined amount of data transmissions or by the receipt of a telemetry stop discrete indicating that receiving station is out of range. TMFT sets a flag to indicate end of transmission and then waits for start of next down link output.

Program DCPX is an executive routine to handle requests for a particular computational algorithm for data compression requirements. Prior to putting data onto the data bus for telemetry transmission, the routine computing, testing, or inputting the data to be transmitted requests the appropriate data compression routine. The function of DCPX is to initialize and switch to the requested routine, and to insert the parity bit on the compressed data prior to return to the requesting routine.

Program DCBP provides a bit packing of discrete, bi-level, or event information into a single status word. Its function is to determine the answer to the UP/DOWN, ON/OFF, or YES/NO condition, and put a "1" or "0" in the appropriate status word position.

Program DCDB provides debiasing for a specified signal. If the signal is expected to have a small dynamic range with a large magnitude, it may be advantageous to subtract a bias value from each sample and transmit the deviation from this bias value. An example of the use of DCDB is in the monitoring of a power supply voltage where its value is expected to have small variations about some RMS value (e.g., 28 volts). It will required a fewer number of bits to represent the signal if 28 volts is subtracted from each sample. The function of DCDB is to perform the computation

$$T_n = Y_n - K_y$$

where,

Y_n is the actual value of the n-th sample

K_y is the bias constant for the signal Y

T_n is the transmitted value for the sample

Program DCDC uses a difference coding algorithm for data compression. Instead of transmitting the value of the sample, the difference between successive samples is transmitted. This transmission of first order differences is similar to DCDB in its application to samples having a small dynamic range or to signals which are relatively smooth. Neither DCDB or DCDC introduce errors by their compression algorithm. The function of DCDC is to perform the computation

$$T_n = Y_n - Y_{n-1}$$

where

Y_n is the value of the sample at time n

Y_n is the value of the sample at time n-1

T_n is the transmitted value.

Program DCZP uses a zero order polynomial predictor algorithm. DCZP transmits the difference between samples if the differences exceed some preset value. If no value of the difference is transmitted at time t, the value of the sample is assumed to be the same as at t-1. The function of DCZP is to perform the comparison

$$| Y_n - Y_{n-1} | > K$$

and indicate to the requesting routine the result of the computation.

Program DCZI uses a zero order polynomial interpolator algorithm. DCZI is similar to DCZP with the difference being that instead of predicting succeeding values from past values, successive data points are examined and a horizontal line fitted to as many consecutive points as possible without creating errors

---

NOTE: Telemetry data compression may be done external to the DMS with special purpose hardware.

greater than the established criteria. DCZI performs the comparisons and computations necessary to determine the value to be transmitted, and to adjust the upper and lower bounds about the horizontal line. Given that

Y_i is the current sample

K_u is the upper bound

K_e is the lower bound

A is the aperture width

Y_t is the transmitted difference

then

$$Y_t = \frac{K_u - K_e}{2} - Y_{t-1}$$

if

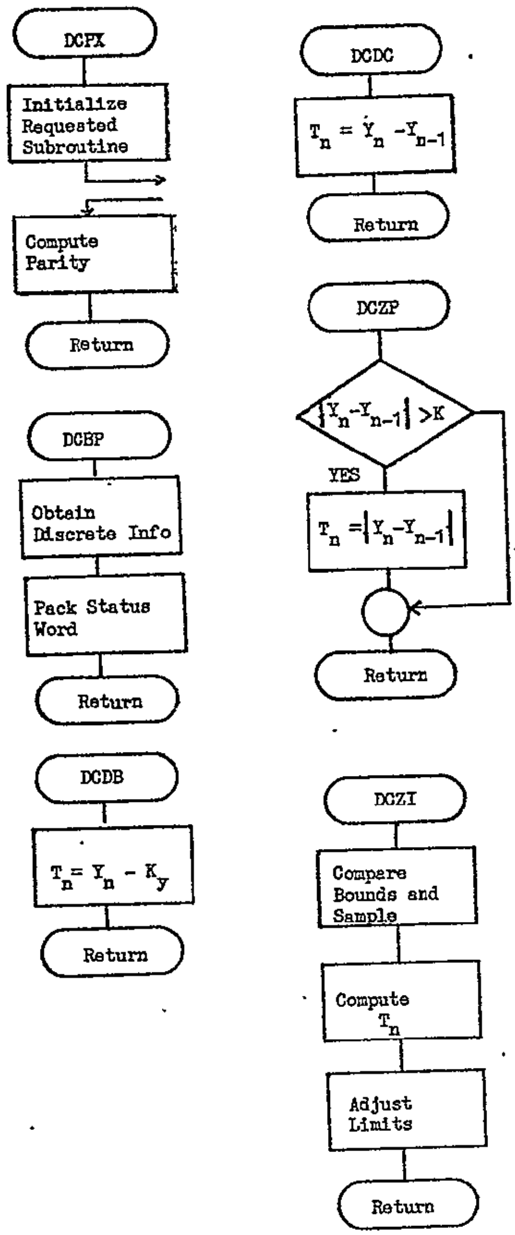
$$Y_i > K_e - 2A$$

or

$$Y_i < K_u + 2A$$

If these inequalities are not satisfied no adjusted transmitted value is required. The bounds are adjusted (to Y_i) if the change in sample values is greater than the error bound.

Fig 5-111 is a flow diagram of the above data compression routines. Due to the short mission time and no flight crew experiment requirements, it is assumed that more sophisticated data reduction techniques, such as higher order predictors or interpolations, statistical algorithms and complete data reduction schemes, will not be required. In addition, as there will not be significant time intervals in which the booster will be out of line of sight to a ground telemetry facility, there is no requirement for data compression in the processing and storing of data for later transmission.



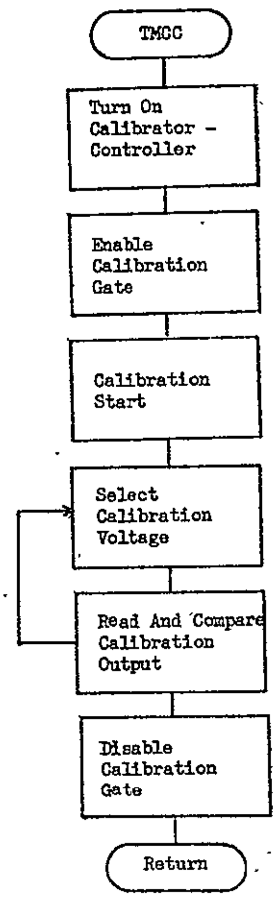


Figure 5-112 Telemetry Calibration

Figure 5-111 Data Compression

Program TMCC controls the Calibrator-Controller assembly and interprets the results of tests. The functions of TMCC are

- Output discretes to control the assembly
- Read calibration outputs and compare with prestored values for each calibration step
- Return assembly to operational mode

TMCC starts by supplying 28VDC power to the Calibrator-Controller assembly Then an Enable/Disable switch is set to the Enable position which permits data flow through the calibration gates. (Disable position is for use by normal operational telemetry data). The calibration start command resets the calibration voltage to the first of six steps. The output of the calibration gate is then read and the results noted (A message - Telemetry out of Calibration - is displayed if bits other then the least significant bit disagree with the prestored value). TMCC then steps through the other five steps (using Calibrator - Controller Advance discrete). When the test is completed, the calibration gate is disabled and the assembly turned off. Figure 5-112 shows the general flow of TMCC.

Booster telemetry operations are verified with the help of the ground based facilities with which it is communicating. A test pattern word can be included in down link transmissions. Also the verification loop, test request and margin calculation routines described in Section 5.6.2. are an indication of satisfactory telemetry operations.

5.6.4 Recording

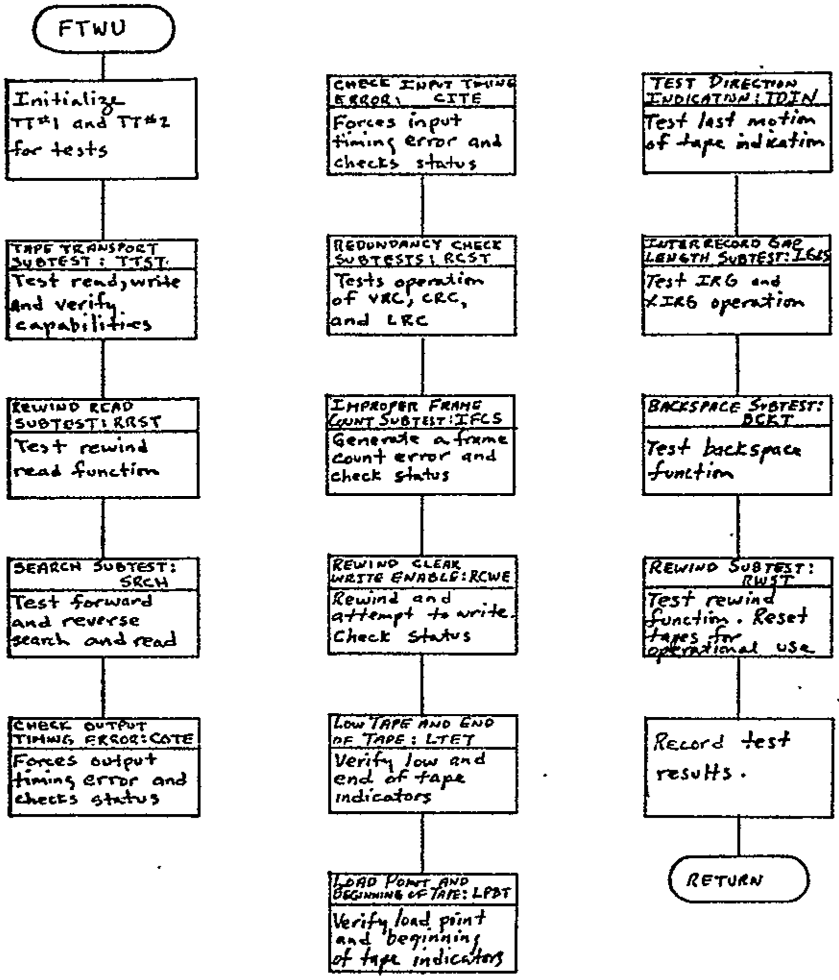
Recording programs are required to perform preflight tests of write and read units and to provide operational control and monitoring of these units in flight. The basic software method for interpreting and controlling the magnetic tape units are through the status and function words. A preselected tape transport assignment is required for program organization. The following assignment is used for this study:

- TT #1 - primary: record telemetry, FAA, and maintenance data
  secondary: record flight qualification data
- TT #2 - primary: record flight qualification data
  secondary: record telemetry, FAA and maintenance data
- TT #3 - read checklists
- TT #4 - read copy of checklists
- Voice recorder - record voice

Functional Tests of Write Units (FTWU) routines are used to provide a complete verification of the capabilities and characteristics of the write units (TT #1 and TT #2) of the magnetic tape subsystem (MTS). FTWU cycles through various subtests, controls the display of error messages if required, resets and initializes the tapes for operational use, and records the results of the functional tests.

Figure 5-113 is a flow chart of FTWU. FTWU first sets up for the subtests by initializing TT #1 and TT #2. A request for transport status is issued, and the status word received is examined by FTWU. If the

Figure 5-113 Functional Test of Write Units



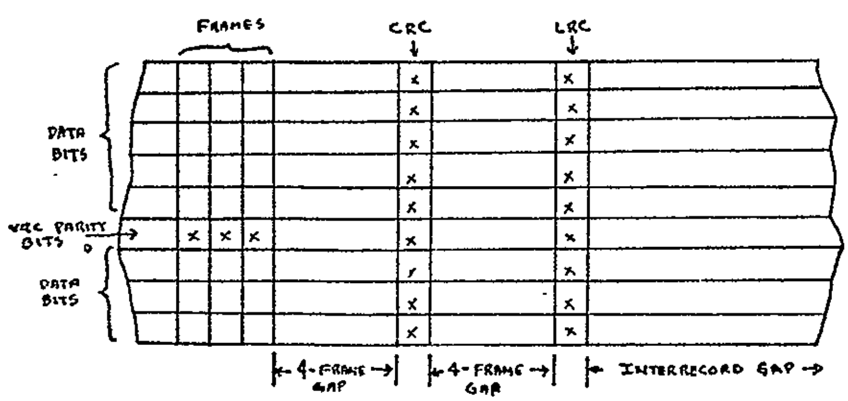


Figure5-114 9-Channel Magnetic Tape Configuration transports are ready, the first subtest is started. If a transport is not ready, an advisory message is displayed on the CRT assigned for magnetic tape subsystem testing. As FTWU cycles through the tests, it maintains a table of results for recording at the conclusion of the tests. If test criteria are not satisfied during each subtest evaluation, an exit is made to FTWU for error message display and advisory cues to the flight crew for recovery. When discrepancies have been corrected, FTWU will continue the tests.

The Tape Transport Subtest (TTST) routine tests the formatting capabilities of the selected transport by cycling the tape unit through the writing, reading and verifying a record for all legitimate combinations of density and modulus. Records are verified by comparing data read back from the MTS with data sent to the MTS. TTST will inform FTWU which data, if any, fails to meet the comparison test. The repeat and rewind function word bits are then set so that the Rewind Read Subtest(RRST) routine can be conducted. The returned status word is analyzed and the record which is read is verified. The Search Subtest (SRCH) routine is then used to test the forward and backward search and read functions. Identifier words are transmitted to the MTS control unit. The record, when found, is read into memory and verified.

The Check Output Timing Error (COTE) routine forces a condition where an output timing error should be noted by the MTS control. After sending a write instruction, COTE goes into a test MTS busy loop instead of giving data required by the write instruction. When the MTS is no longer busy, the status word is examined for the output timing error discrete. If found, the test is successful. The input timing error test is conducted in a similar manner by the Check Input Timing Error (CITE) routine. The Improper Frame Count Subtest (IFCS) routine is used to verify the operation of the frame count error discrete. The frame count error is generated by writing a fixed length record in one modulus and reading this record in another modulus so that the record is incomplete. IFCS then repeats the reading operation using the same modulus as during the writing of the record. This will verify that the frame count error was generated by an incomplete record and not by conditions, such as a bad spot on the tape, which would cause characters to be lost.

The Redundancy Check Subtests (RCST) routine verifies the operation of the Vertical Redundancy Check (VRC), Cyclic Redundancy Check (CRC) and Longitudinal Redundancy Check (LRC) information placed on the tape by the MTS control unit. Figure 5-114 shows the configuration of check and data bits on the 9-channel magnetic tape. The VRC is an odd parity bit added to each tape frame. The CRC is computed by the MTS control unit during writing and recorded at the end of each tape record preceding the longitudinal check frame. The nine bits of the CRC character are generated by an algorithm involving exclusive or addition, shifting and inversion of predesignated bits. The LRC is added to insure that the number of one bits in each channel is even.

This frame is generated by the tape control during writing and checked on reading. RCST creates conditions in which redundancy check errors will exist, verifies the proper status word response, and then uses its recovery subroutines to correct the bits in error.

The Rewind Clear Write Enable (RCWE) routine rewinds the tape to the load point and then attempts to write a record. As this is an improper condition, RCWE tests the status word for the proper setting of both the improper condition and the no write enable discretes. Low tape and end of tape conditions are evaluated by the LTET routine. LTET also ensures that the improper condition, forward command at end of tape, is recognized by the control mechanism. Similar procedures are evaluated by the Load Point and Beginning of Tape (LPBT) routine. An attempt is made to backspace over the load point. In this situation both the load point and improper condition bits should be set in the status word. The Test Direction Indication (TDIN) routine tests the last motion of tape bit of the status word. A record is written on tape and the status word is checked for proper indication of forward motion of tape. TDIN then backspaces over the record and tests the status word for backward motion indication. The Interrecord Gap Length Subtest (IGLS) routine tests the operation of writing both the XIRG and IRG, and recognition of the gaps by the status words. A check of the gap lengths is made by IGLS by a time counting procedure which recognizes first and last characters of a sequence of records, and reads through the interrecord gap to maintain tape speed. The backspace function is evaluated by the Backspace Subtest (BCKT) routine, and the Rewind Subtest (RWST) routine tests the rewind function and resets the tapes to the position required for the start of operational recording. FTWU then completes the assembly of data concerning the functional testing of the write units and writes a record on the assigned tape. The data recorded consists of status word configuration after each test, operation timing information, and other information of use during post flight diagnostic analysis.

<u>Functional Tests of Read Units (FTRU)</u> routines are used to verify the check list and special symbology records contained in the tapes of TT #3 and #4, and to check the operational characteristics and capabilities of these units.

Figure 5-115 is a general flow diagram of FTRU. The Tape Transport Read Initialize (TTRI) routine readies the transports for testing. Power is turned on, blower operation is monitored, and the over temperature alarm is tested. Status words are examined to verify that the no write enable and transport ready bits are set. TTRI then initializes indices, flags and counters used in the test loop.

FTRU verifies records by searching for, reading into a preassigned storage area, and comparing word for word the information contained on the two tapes. As one tape is a copy of the other, corresponding words in the two storage areas should be the same. An additional check is made (to ensure that both words from tape do not have the same error)

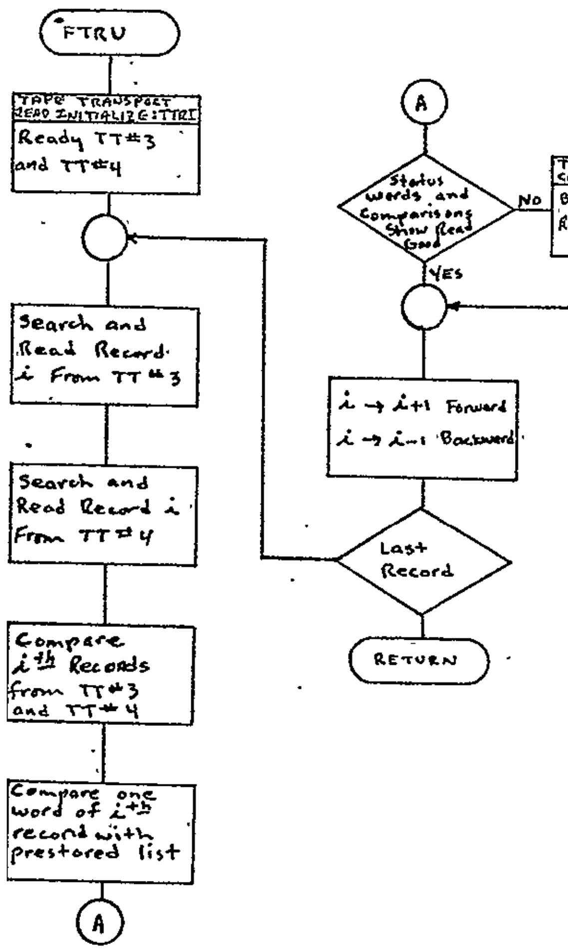


Figure 5-115 Functional Test of Read Units

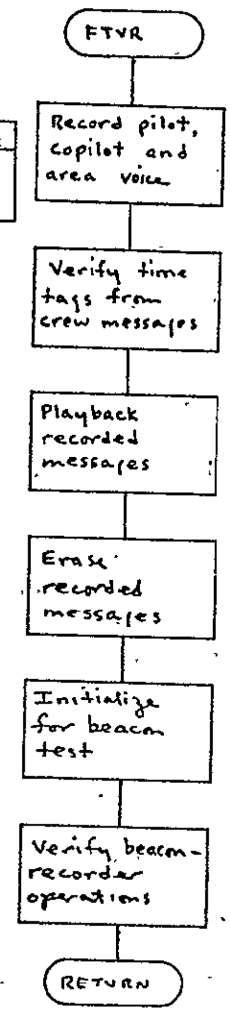


Figure 5-116 Functional Test of Voice Recorder

by a comparison test with a prestored table. This table contains one word from each record (the word has been preselected randomly and verified). If comparison tests and status words indicate that the read operation has been performed successfully and that stored information is accurate, FTRU then repeats the procedure for the next record. FTRU proceeds through the records first in a search and forward read operation. When all records have been checked in this manner, the operation is repeated using search and backward read instructions.

The Tape Read Error Correction (TREC) routine is entered if an indicatio of character error is received. During a tape read operation, the CRC frame is computed again and compared to the CRC frame recorded when the check list records were written. The VRC and LRC characters which were written are also checked during the reading process. Under control of TREC, the error correction capability is designed to correct almost any pattern of erroneous bits along a single channel within any record. Joint use of the CRC and VRC can locate the channel in error when an erroneous record is read. Error correction is performed by the MTS control unit when TREC backspaces and rereads the erroneous record. When the VRC determines during the reread that a frame is in error, the bit in that frame in the channel that contained errors during the original read is corrected. TREC then rejoins the verification process under FTRU control.

Functional Test of Voice Recorder (FTVR) procedures are designed to test and verify the operational requirements of the voice recorder equipment. The functions of FTVR are to:

- Verify the operation of the record, playback and erase features of the voice recorder
- Test the data time tag function of microphone closure
- Test the voice and flight data recorder operations associated with the recovery beacon

Figure 5-116 is a general flow diagram of FTVR. FTVR works in conjunction with the preflight checklist being displayed on the assigned CRT. The displayed check list provides cues to the flight crew for verifying the voice recorder functions. The crew operates the reprogrammable switches associated with the test to initialize the voice recorder. Pilot, copilot and area microphones are checked by having a message from each recorded on the tape. At the same time the operation of the microphone closure discretes is verified. The tape is then reversed and the messages played back. If the crew is satisfied with the quality of the recording, the tape is again reversed and the erase feature is tested.

To test the recording requirements associated with beacon operations, the voice recorder is turned OFF, and a simulated excessive acceleration data word is placed on the data bus. FTVR verifies that the voice recorder is turned ON following receipt of this over limits information and that the magnetic tape unit associated with flight data recording is activated and has recorded the required flight data.

Tape Transport Executive (TTEX) procedures provide supervision of the magnetic tape read and write units. The timing of the data to be recorded can be prescheduled as a function of mission phase requirements and known flight qualification, telemetry and maintenance requirements. Additional recording may be required in response to uplink requests or emergency situations. These additional requirements will fit into available time slots or take precedence over current tape usage. Record lengths for each recording requirement are determined by the parameter to be recorded, its sample rate and number of bits. For example, a record of 120 8-bit characters each 6 seconds would satisfy the flight data requirements of Figure 4-41. The display file data on tape transports 3 and 4 are arranged in normal mission sequential order to minimize tape access time. Figure 5-117 is a general flow chart of TTEX. The Write Flag is set if writing of data is required. If this flag is set, TTEX continues input functions associated with the data bus interface. A variable word associated with the record requirements is formed and an exit is made to the Operational Use of Write Units (OUWU) routine for further processing. OUWU returns to TTEX where the Read Busy flag is checked. This flag is set when the tape units are supplying data from the read transports. If the flag is not busy, the condition of the Magnetic Tape subsystem (MTS) is verified. This test checks blower operation, transport temperatures, and status words for conditions which might require reassignment of the read or write units. If reassignment is necessary, TTEX revises the lists it maintains and sets appropriate flags for use by other tape handling routines. TTEX recognizes read requests from the Displays and Controls programs, and, after checking

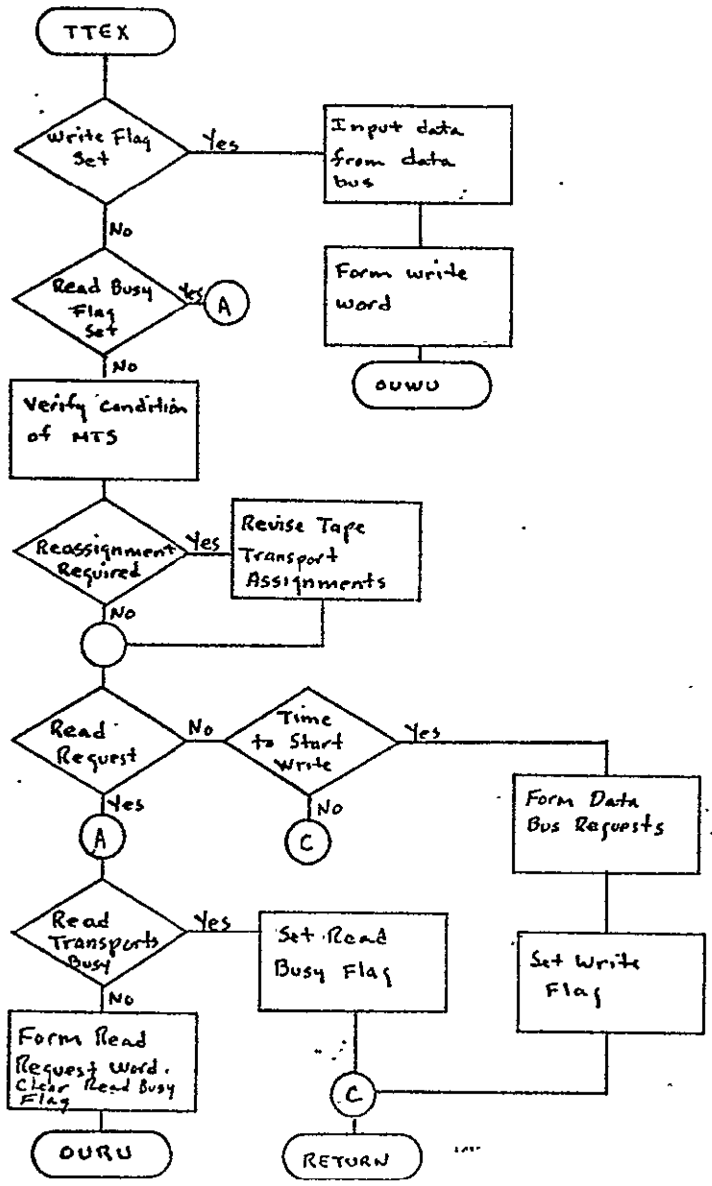


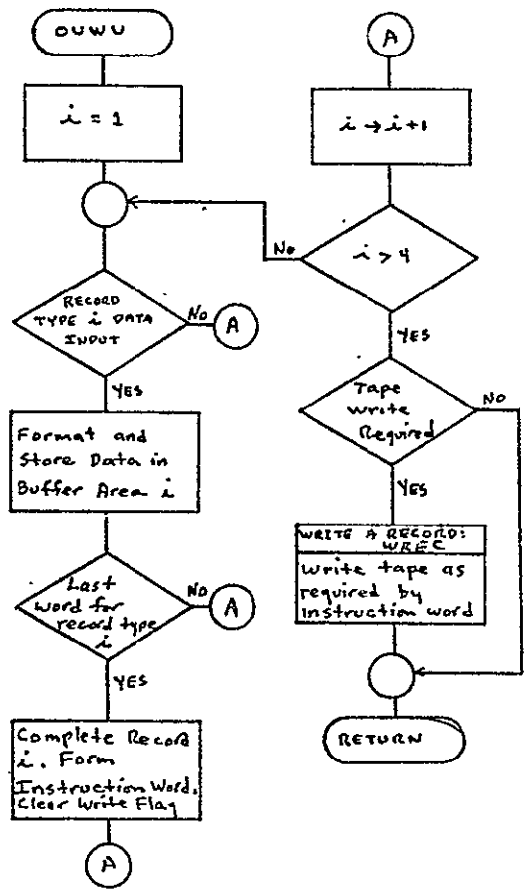
Figure 5–117 Tape Transport Executive

to determine if the transports are not currently busy reading, will form a read request word containing a record number and usage information. TTEX then exits to Operational Use of Read Units (OURU) where the reading operation is performed. OURU returns to TTEX where a schedule of writing requirements is checked. If it is time to start a new write record, TTEX formulates requests for the parameters to be included in the record. After setting the Write Flag, TTEX exits until the next assigned entry time.

<u>Operational Use of Write Units</u> (OUWU) procedures are used to record scheduled and requested data and to monitor the status of the assigned tape units. The functions of OUWU are to:

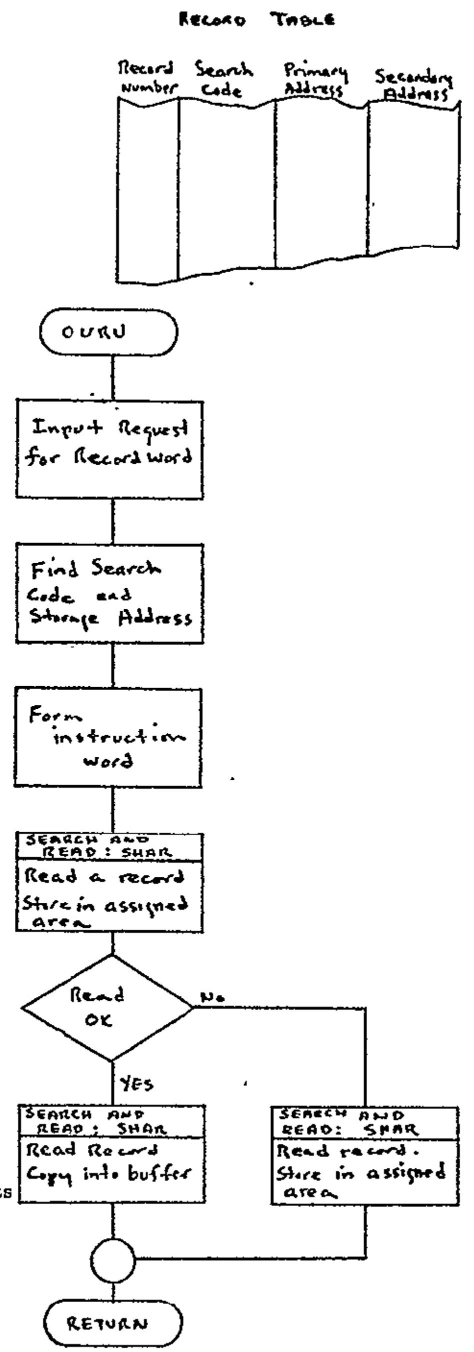
- Format and store flight qualification, maintenance, telemetry and flight data in their assigned buffer areas
- Provide identifier codes, time tags and other data as required to complete a record
- Write a record at specified time intervals on the designated tape

Figure 5-118 is a general flow diagram of OUWU. The Tape Transport Executive (TTEX) interfaces with the DMS and the data bus for required inputs. At each entrance to OUWU, TTEX advises OUWU on the number of the inputs destined for each record. OUWU formats the data (removes addressing or parity bits and forms into 8-bit bytes) and stores the data into the buffer area associated with each data type. If the current inputs complete the data words required for a record, OUWU will add forward and backward search words, time tag, and other information words required to assist in the reading and identification of tape records during post flight analysis. OUWU then forms the instruction word (specifying format and transport address) for



-119

Figure 5-118 Operational Use of Write Units



-119

Figure 5-119 Operational Use of Read Units writing. After repeating these steps for other records being formed, the Write a Record (WREC) subroutine is entered. WREC performs the output of the instruction word, monitoring of the status word, and reporting of results of the write operation.

Operational Use of Read Units (OURU) routines are used to input check list or display symbology characters into the specified display file memory area and to monitor the status of the assigned tape units. Figure 5-119 is a general flow chart of OURU. OURU operates in conjunction with Displays and Controls routines to provide display characters. OURU receives a request for a specific record number. A table look up is made to find the identifier code and display file address for the requested record. OURU maintains information on the current position of the read tapes, so that data from the record table is sufficient to determine if a search and read forward or backward is required. (The tapes have been written in a sequential manner for a normal mission so that a minimum of time is required for access to the record. However, records may have to be taken out of sequence or repeated if events, determined by the systems executive, require changes in procedures). The instruction word is formed and the Search and Read (SHAR) routine is entered. SHAR performs the functions of issuing the read commands, monitoring the status words from the MTS control unit, and reporting the results of the search and read operation. If the read operation was successful, , the copy of the display file data is read into the tape buffer in order to put the tape transports at the same positions and to provide the copy for verification purposes as desired by Displays and Controls routines. The display file data may be required for use if the read operation on TT#3 was unsuccessful. In this case, the instruction word is reformed to call on TT#4 to provide the data for use by the assigned CRT. Prior to exiting, OURU updates its data on tape positioning and tape transport status.

5.6.5 Beacon Subsystems

Beacon Executive (BCNX) routines supervise the testing, initializing and usage of beacon subsystem units. Figure 5-120 is a general flow chart of BCNX. BCNX calls on routines for testing the radar beacon (ATC transponder), PRN (pseudo random noise) ranging subsystem, C-band radar, and recovery beacon units during prelaunch or inflight testing. Communication mode changes may require reassignment of C or S-band equipment, and operational requirements may ask for activation and usage of the radar or recovery beacons. In these cases, BCNX recognizes the requests generated by other programs and enters the appropriate program.

When BCNX is entered during the prelaunch mode, BCNX initiates the cycling through the Radar Beacon Test Initializer (RBTI), the PRN Ranging Test Initialize (PRTI), the C-Band Radar Test Initialize (CRTI), and the Recovery Beacon Test (RBNT) routines. These routines test the various beacon components, establish the condition dictated by communication mode schedules, and report the equipment status to BCNX. These routines are also entered if inflight testing is desired. Changes in S-band communication modes scheduled as a function of mission phase or required by equipment degraded operation may cause a reassignment of PRN ranging. PRTI is entered to activate or deactive units as required and to test the equipment being activated. If a shift in C-band radar usage is also scheduled, CRTI performs the necessary functions.

BCNX is alerted when an air traffic control (ATC) station requests position or identification data. BCNX calls on the Radar Beacon Response (RBNR) routine to input necessary data from other programs, and to verify and

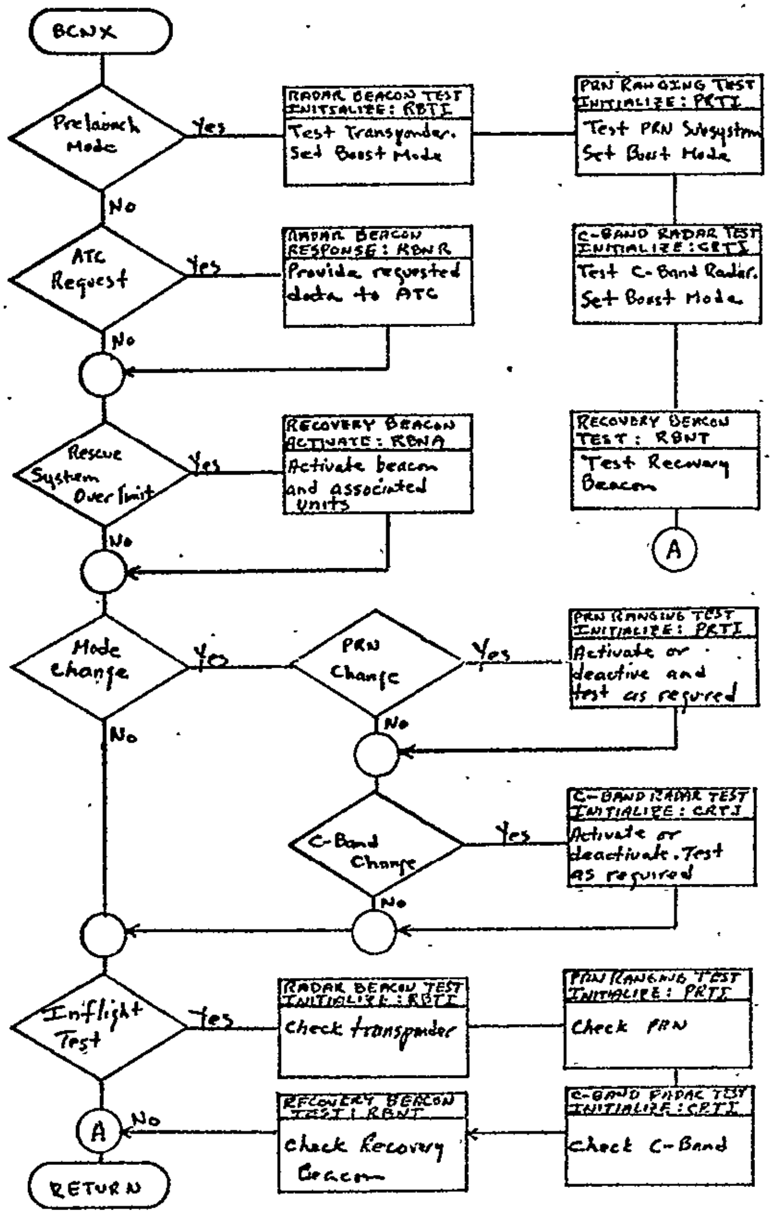


Figure 5-120 Beacon Executive

format the information for output by the ATC transponder. BCNX also monitors the recovery beacon subsystem, and, if emergency situations require, calls on the Recovery Beacon Activate (RBNA) routine to turn on and verify the recovery beacon.

Radar Beacon Test Initialize (RBTI) procedures are used to verify beacon voltage levels, switch and adjustment settings, frequency and power output parameters, and code response operation; to initialize the ATC transponder units as required by the mission schedule; and to provide status information to CRT's for flight crew use, and test results for post flight diagnostics. Figure 5-121 is s flow diagram of RBTI.

On entry to RBTI during the prelaunch checkout, initialization procedures are performed. These procedures consist of starting the equipment and performing a routine check in order to determine if gross discrepancies such as blower motor failure, power failure, or circuit breaker openings have occurred. When the equipment is ready, the functional and adjustment checks are started. Power and heater voltages are first checked against their allowable tolerances. If these measurements are out of limits, the Radar Beacon Status Message (RBSM) routine is entered to prepare an appropriate status message. RBSM uses a prestored message table and information form the calling routine to fill in the blanks of the message.

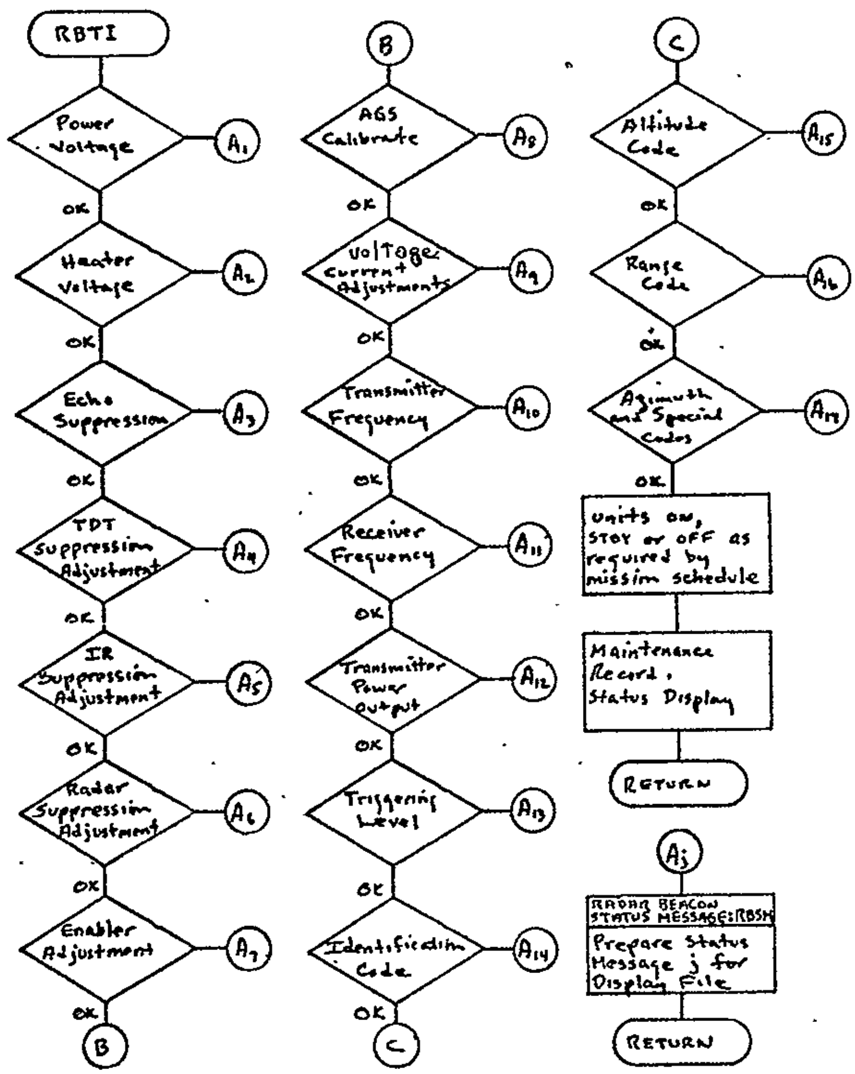


Figure 5-121 Radar Beacon Test Initialize

When all tests are completed, RBSM assembles all the status messages and informs the Displays and Controls subsystem that the data is ready for display and flight crew analysis.

Some transponder functions require the presetting or adjustment of various circuits. The echo suppression switch is set to ON in areas where echo interference is more troublesome than pulse-type jamming, and OFF for the reverse condition. This setting is preselected to provide the most consistently satisfactory operation of the equipment. The transmitter dead time (TDT) circuit disables the transponder set for 225 or 500 microseconds after the transmission of each reply pulse. This dead time limits the maximum repetition rate of the transponder set to either 4000 or 2000 pulses per second. The IR suppression circuits share the same time constant components supplying pulses of equal durations. IR suppression prevents mutual interference between transponder sets. The radar suppression circuit prevents other booster radar from interfering with transponder set operation. When a radar is connected to the radar suppression input the transponder is insensitive for about 18 microseconds at the start of each radar transmission. The transponder set may be rendered insensitive except when enabling pulses from a separate receiving system are present. RBTI checks this enabler link if this optional circuitry is used. An automatic gain stabilization (AGS) circuit performs an anti-jamming function if extremely high pulse recurrence frequencies are received. AGS can be turned off, permitting an increase in noise level and thereby serving as an operational check of the equipment. RBTI gathers measurements and settings of these special functions, compares them with their preassigned requirements, and calls on RBSM to assemble advisory messages, if required.

RBTI then cycles through the voltage and current measurements available from the voltage current selector switch. All the measurements are compared against their upper and lower limits. Receiver frequency, and transmitter frequency and power output are checked using data from the onboard checkout equipment. The normal triggering level is the minimum peak voltage required to cause full firing of the transponder set. The triggering level is checked at 1010 and 1030 megahertz to insure that full firing can occur at both ends of the receiver band. RBTI next checks the code functions of the transponder. The preassigned identification code is tested to verify that the associated pulses are activated. Two other codes are also checked to ensure the capability to switch to other identification codes if requested by ATC ground stations. Altitude, range and azimuth codes are checked by inputting a known value into the transponder and verifying that the expected (according to prestored tables) pulses are activated. Following the completion of the radar beacon tests, RBTI completes the table of test results for recording purposes. RBTI then requests RBSM to move the status record to the assigned display file area.

Radar Beacon Response (RBNR) procedures are used to input, format and store navigation data in the proper transponder buffer, and to verify correct radar beacon operation following ground radar triggering of transponder circuits. Figure 5-122 is a flow chart of RBNR.

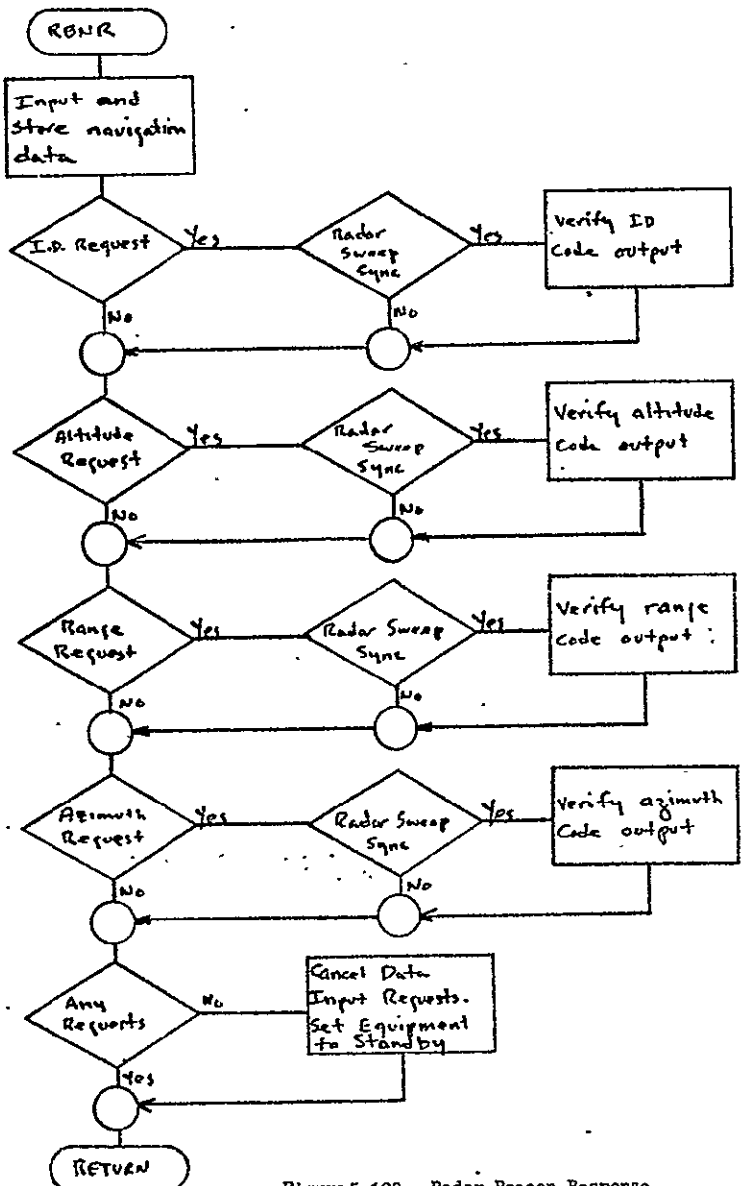


Figure 5-122 Radar Beacon Response

RBNR is entered from BCNX when a request for radar beacon information is detected. BCNX then sets a flag so that entry to RBNR continues at one second intervals. When there are no longer any requests, RBNR clears this flag, discontinues requests for navigation data inputs, and sets the radar beacon equipment in a standby condition. Upon entry to RBNR, data bus input functions are performed. The information from the navigation system is decoded and address, parity and insignificant bits are masked out. The data is then stored in the transponder output registers reserved for each particular request. At the next ground radar sweep, the data is transmitted by the transponder circuitry. RBNR then verifies the operation of the equipment. The identification code is verified by checking the setting of the pilot's control unit against the transponder pulse output. Verification of the navigation data outputs are performed by matching the transmission code outputs with the data received from the navigation subsystem. A table lookup on the output code (tables correspond to that shown for the altitude transmission code) is performed and compared with the inputs to ensure that outputs are within the code resolutions.

5.7 OPERATIONS MANAGEMENT

5.7.1 Data Management Computer

Data Management Computer Executive (DMCX) routines are used to checkout, monitor, and isolate faults for the data management subsystem's digital computer; to provide recovery procedures in the event of failures; and to reconfigure digital computer equipment as required to maintain optimal computational and redundancy capabilities. Figure 5-123 is a flow chart of DMCX.

DMCX is entered when a request for computer self-check is made, when periodic monitoring is scheduled, when a fault is detected, or when reconfiguration of the computer subsystem units is required. The self-check test is made during the prelaunch checkout and inflight if requested or scheduled. As part of the prelaunch preparations, the computer units are powered up and made ready for testing. Then the Digital Computer Command Test (DCCT) routine is entered. DCCT cycles through various subtests which verify the complete repertoire of command and arithmetic instructions. The Memory and Register Test (MART) routine verifies input and output operations. If faults are detected by these test routines, the Digital Computer Fault Isolation (DCFI) routine is entered and the failure symptom catalog is examined to determine the appropriate action.

DMCX responds to program fault interrupts such as computer power failures, illegal codes and addresses, and computational overflows. DCFI attempts to isolate the fault and then the Program Recovery (PRGY) routine is entered to take the appropriate remedial action.

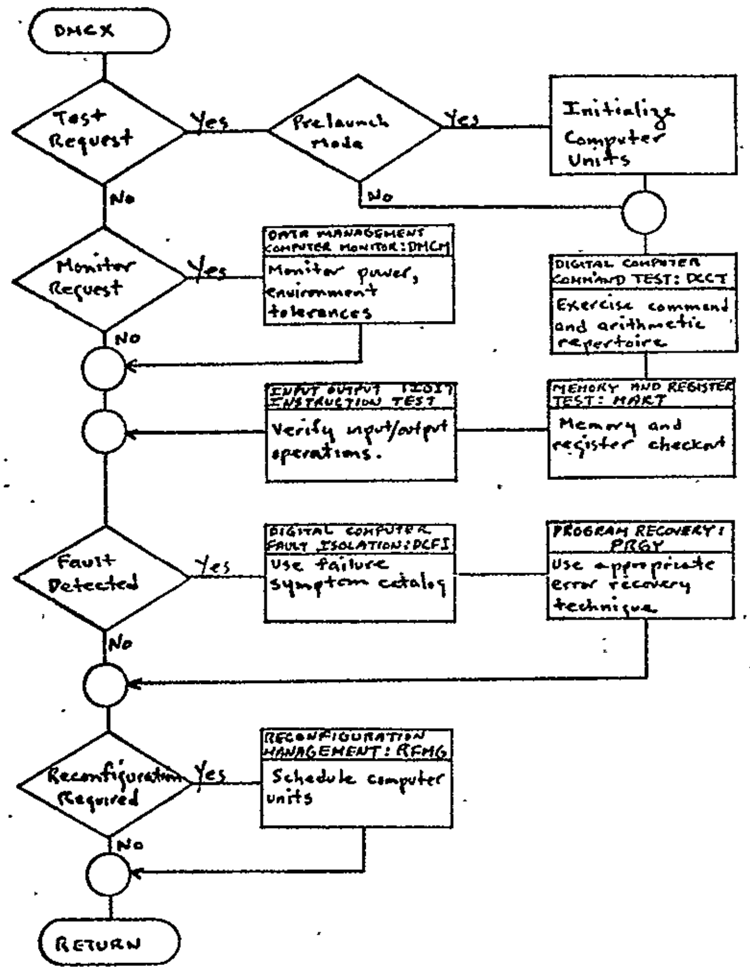


Figure 5-123 Data Management Computer Executive

If a central processor, memory unit, or Input/Output controller fails, the Reconfiguration Management (RFMG) routine performs the functions of scheduling the various computer units so that computational efficiency and redundancy requirements are maintained. RFMG provides necessary information to maintain data flow between the booster subsytems. The Data Management Computer Monitor (DMCM) routine provides monitoring capability for functional and environmental parameters essential for status reporting. DMCM is entered periodically under executive control,at the request of the flight crew for status information, or from the onboard checkout subsystem following an out of tolerance measurement.

Digital Computer Command Test (DCCT) procedures are used to test the control and arithmetic sections of the computers. Figure 5-124 is a flow chart of DCCT. Tests performed by DCCT are organized into subtests which involve similar instruction types. These subtests check the legal combinations allowed by the instruction word format (function codes, index designators, jump indicators, and operand addresses). If errors are encountered during the tests, a table is assembled for use by DCFI in isolating the cause. The final subtest of DCCT checks the operation of program faults and interrupts to ensure entry to DCFI and PRGY in order to recover from operational program errors.

Subtest 1 checks instructions concerned with data transfers to and from storage. These include instructions such as:

- Store A ; Store Q ; Store A Complement
- Enter A ; Enter Y-A ; Enter Y+A

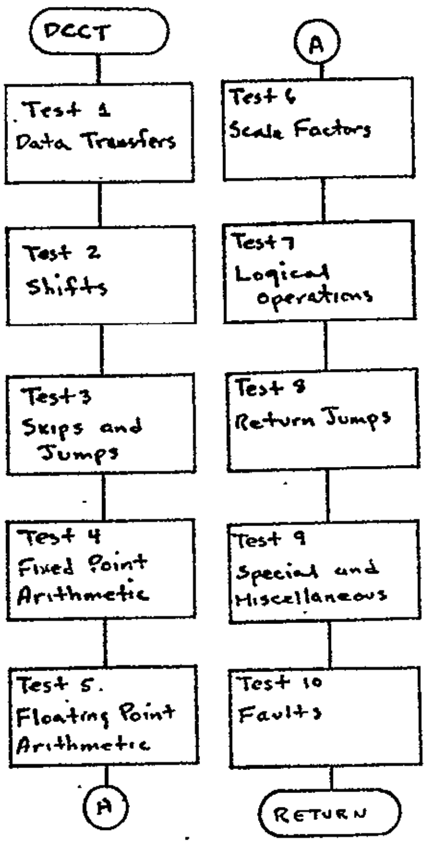


Figure 5-124 Digital Computer Command Test

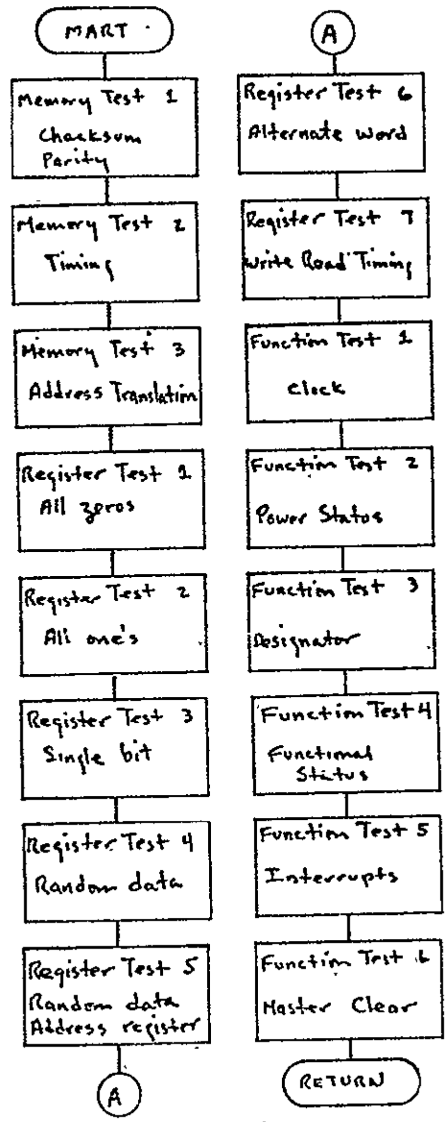


Figure 5-125 Memory and Register Test

Subtest 1 uses preinitialized memory areas, cycles through its assigned instructions (in more than one way), and arrives at final results which can be compared with predetermined values for deciding whether an error has occurred. Similar schemes are used for the other instruction repertoire subtests.

Subtest 2 tests the various shift instructions including operations such as:

- Shift Left Circular ; Shift Left Circular Double
- Shift Right Fill Zeros ; Shift Right Fill Sign

Skips and jumps are checked in subtest 3. Typical instructions in this group are:

- Unconditional jump ; unconditional skip
- Jump A positive (negative, zero)
- Jump equal (greater than, less than)
- Jump within (outside) limits
- Skip B zero (negative, equal)

Both of these subtests exercise all the function codes of their groups together with the indexing, jump, and operand designator capabilities. The end results of the tests determine if an error has been detected.

Subtests 4 and 5 are the primary ones used for evaluating arithmetic operations. In addition to standard add, subtract, multiply and divide instructions, fixed point arithmetic operations may include other functions such as

- Replace add (subtract)
- Partial add (halves or thirds of registers)
- Double precision add (subtract)

Floating point arithmetic includes add, subtract, multiply and divide with optional rounding and double precision operations. DCCT validates the precise functioning of all arithmetic instructions in the repertoire.

Subtest 6 verifies scale factor instructions. The single scale factor instruction normalizes the accumulator register and stores the shift count in an index register. The Double scale factor instruction performs the same function for a double length register. DCCT cycles through a table of known bit patterns, and verifies the shift counts generated.

Subtest 7 is used to check the instructions which perform logical operations such as:

- Selective set (clear, substitute, complement)
- Logical AND (Exclusive OR, inclusive OR)
- Replace selective set (clear, substitute, complement)

Return jumps, both conditional and unconditional, are validated by subtest 8. Subtest 9 checks special or miscellaneous instructions not included in the other subtests. Typical instructions consist of:

- Square root
- Masked search for equal (not equal, greater, less than)
- Count Ones; Test parity (odd, even)

Following completion of subtest 9, DCCT assembles the results of all the tests. If an error has occurred, the fault detected flag is set and DCFI is entered when memory and input/output tests are finished.

Subtest is used to check the functioning of program recovery aids following illegal operations such as the use of illegal codes or addresses, and computational faults such as arithmetic overflow. DCCT verifies that the appropriate subroutines of Program Recovery (PRGY) are entered and that returns are made at the proper points following recovery.

Memory and Register Test (MART) procedures are used to verify memory performance, and to test fault indicators and control registers. Figure 5-125 is a flow chart of MART. The tests conducted by MART are designed to provide the information necessary for the Digital Computer Fault Isolation (DCFI) routine to determine the cause of computer malfunctions that directly affect memory operation.

Memory tests are performed to verify core storage and associated circuitry. Test 1 does a search instruction which reads every core address. If a parity error is encountered, MART records the address at which it occurs. Test 1 also check sums programs storage areas to provide a high confidence in storage reliability. Test 2 provides data to isolate malfunctions in main memory timing and enabling circuits. Test 3 verifies the address selectors in main memory.

Control register tests are designed to exercise arithmetic, index, program control, and special registers under worst case conditions which may cause failures in marginal control memory areas. This section of MART is organized in subtests for different bit patterns. Test 1 checks the capabilities of the control memory to retain an all zeros pattern. Test 2 performs the same check using an all one's pattern. Test 3 checks the control memories ability to retain a single bit shifted through all the registers' bit positions. For test 4, random data is generated with check sums used to verify the data. Using this same random data, test 5 does a sum check by converging from either exterme in addresses, thus exercising the address register and associated line drivers. Test 6 uses an alternate pattern of zeros and ones in exercising the control memory. A checkout of critical write-in and read-out timing functions is performed by test 7. MART then assembles test data for use by DCFI in analyzing the condition of register enable, inhibit driver, sense amplifier and other memory circuits.

Some registers are reserved for special functions. These registers are tested to ensure that the assigned functions are performed. Function test 1 verifies the operation of the central processor clock register. A typical clock register is decremented at a rate of 1024 counts per second with an accuracy of 2 counts during a 10 second period. The clock may be deactivated by setting the most significant bit negative. Upon cycling through zero, an interrupt is generated so that precise timing of computer programs is possible. Function test 2 examines the power status register. To onboard checkout system and BITE units provide the information to set the bits of this register. Figure 5-126 shows a typical format used. Test 2 cycles through the possible combinations to ensure the capability of the power status register to perform its functions. Test 3 exercises designator storage word registers. These special registers provide necessary information such as interrupt status codes, entrance and return addresses and indirect addressing data. Test 4 verifies the operation of the

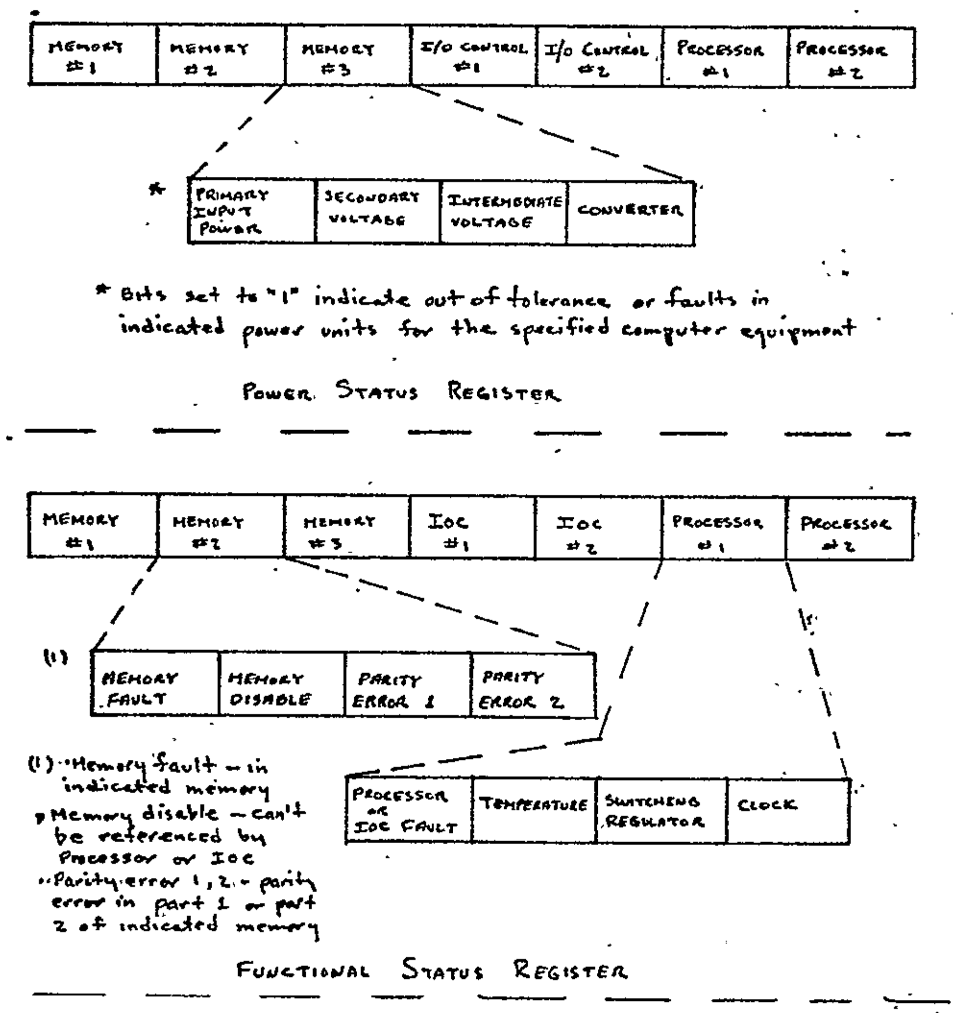


Figure 5-126 Status Registers

functional status register. This register, like the power status register, has bits set to indicate various out of tolerance or error
- 121
conditions. Figure 5-126 shows a typical format used for this function.

Function 5 verifies the correct operation of all interrupts which have not been tested by other DMCX routines. Interrupts used vary in number and characteristics depending on the computer subsystem. Typically, the interrupts are organized in four classes.

- Fault and hardware interrupts - such as the power tolerance fault
- Program error interrupts - such as illegal instructions
- Input/output interrupts - such as external function monitors
- Executive interrupts - which turns control over to the executive program

Test 5 checks the interrupt processing scheme designed for the data management computer, including the proper activation of various registers and the setting of interrupt lockouts if specified. Test 6 checks the operation of the master clear feature. This signal normally clears hardware and program fault indicators and assists in program recovery techniques. Test 6 verifies that the appropriate bits are cleared and that the specified registers are activated.

Input Output Instruction Test (IOIT) procedures are used to checkout input/output (IO) instructions, and to verify IO data paths and control circuits used for communications with the central processors (CP) and peripheral equipment. (Other IO functions are tested by additional checkout routines - for example, the man-machine interface is verified by the Display Tester (DITR) routine). Figure 5-127 is a flow chart of IOIT.

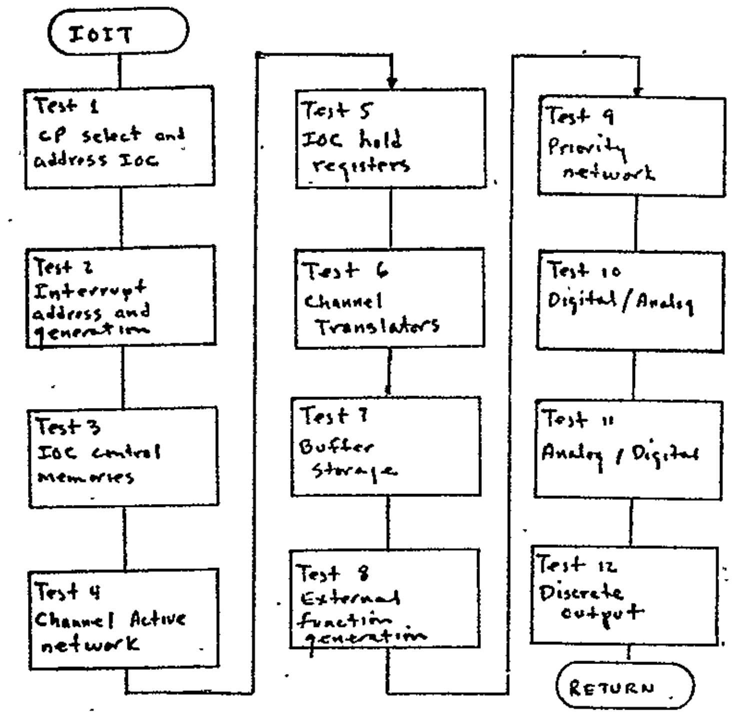
Typical input and output instructions include

- Enable (disable) interrupts
- Set (clear) discretes
- Jump on channel busy (not busy)
- Read monitor clock
- Initiate external function buffer
- Terminate data buffer

Control signals between IO controllers, and central processors or peripheral equipment commonly use a technique described in Figure 5-128. The tests performed by IOIT verify the IO instructions and communications procedures used.

Test 1 checks the ability of any of the central processors to select any of the IO controllers in the data management computer subsystem. Cross data paths between the CP's and the IOC's, the circuitry associated with status and assignment registers, and function code translation logic is verified. Test 2 performs a checkout of the interrupt generation and sense circuitry. The ability of the interrupt registers to hold

Figure 5-127 Input Output Instruction Test



	Signal	From	Meaning
Input Channel	Input Request	Peripheral	Data word on input lines ready for acceptance
	Input Acknowledge	IOC	Data word on input lines has been sampled
	External Interrupt	Peripheral	Interrupt code word on the input lines ready for acceptance
Output Channel	Output Request	Peripheral	Ready to accept a data word
	Output Acknowledge	IOC	Data word is on output lines ready to be sampled
	External Function	IOC	External function message is on output lines ready to be sampled
	External Function Request	Peripheral	Ready to accept an external function message

Figure 5-128 Input Output Control Signals Description

all necessary data configurations is tested. Interrupt instructions are exercised and verified. Test 3 checks and verifies the operation of the circuitry associated with the control memories in each of the IO controllers. The ability of the memories to be set to specific data configurations, register translation logic, and associated control memory logical enables and timing are tested. Test 4 checks the channel active network. Instructions requiring a test of the channel active condition are evaluated. Test 5 examines the function hold registers of the IO controllers, checking bank selection circuitry and IO parity networks. Test 6 is concerned with validating function enable circuitry and function termination logic, and the instructions used to activate the applicable networks.

Test 7 examines the circuitry and instructions associated with data transfer via an IO controller. This test results in a verification of buffer storage, acknowledge timing, request sense logic, input data amplifiers, and output data amplifiers. The buffer comparator is also checked using specific data combinations for proper termination control enables. Test 8 is concerned with the setting and clearing of interrupt lockout logic, the ability of the IO controllers to transfer interrupts to the proper central processor, the generation of external functions, and the timing of interrupt acknowledges. Test 9 checks and verifies the operation of the circuitry associated with priority networks. IOIT cycles through the list of interrupts for the computer system, and determines that priority requirements are satisfied. These requirements normally consist of interrupts listed in priority order, interrupts which are locked out under some conditions, interrupts which are never locked out, and other interrupt processing conditions. The remainder of IOIT tests are concerned with signal conditioning and interface verification. Test 10 checks digital to analog conversion techniques. Test 11 verfies the operation and accuracy of analog to digital procedures. Test 12 tests discrete output logic and circuitry. Following completion of its tests, IOIT assembles the results and forwards the information to DCFI for fault analysis.

Digital Computer Fault Isolation (DCFI) procedures are used to provide information for updating computer status, performing onboard repairs, or reconfiguring computer subystem units. Figure 5-129 is a flow chart of DCFI. Prestored tables used by DCFI, and display input generated by DCFI are also shown in Figure 5-129.

DCFI is entered if failures are detected by the computer test routines, DCCT, MART, or IOIT. These routines input variables to DCFI specifying which test failed and on which central processor, IO controller, or memory unit the fault occurred. Using this information, DCFI can initialize table search counters and indices, and enter the fault catalog table to determine the corrective action required.

The fault isolation catalog is a prestored table which is developed through various diagnostic and fault isolation techniques. Simulation methods are commonly used to generate a catalog. The logic equations which define the equipment are adjusted to simulated failures and the effects of the failures are noted. Functional level checking is also simulated in a manner similar to that done at the logic level. The register, gates, and transmission paths of a device are anticipated, the sequence of operations necessary to use these elements is established, and the effects of a failure are predicted.

For a short mission such as anticipated for the booster, a complete fault catalog is not required. The table will provide sufficient information to replace some of the more critical replacement cards in the computer subsystem. Other table information will indicate that the use

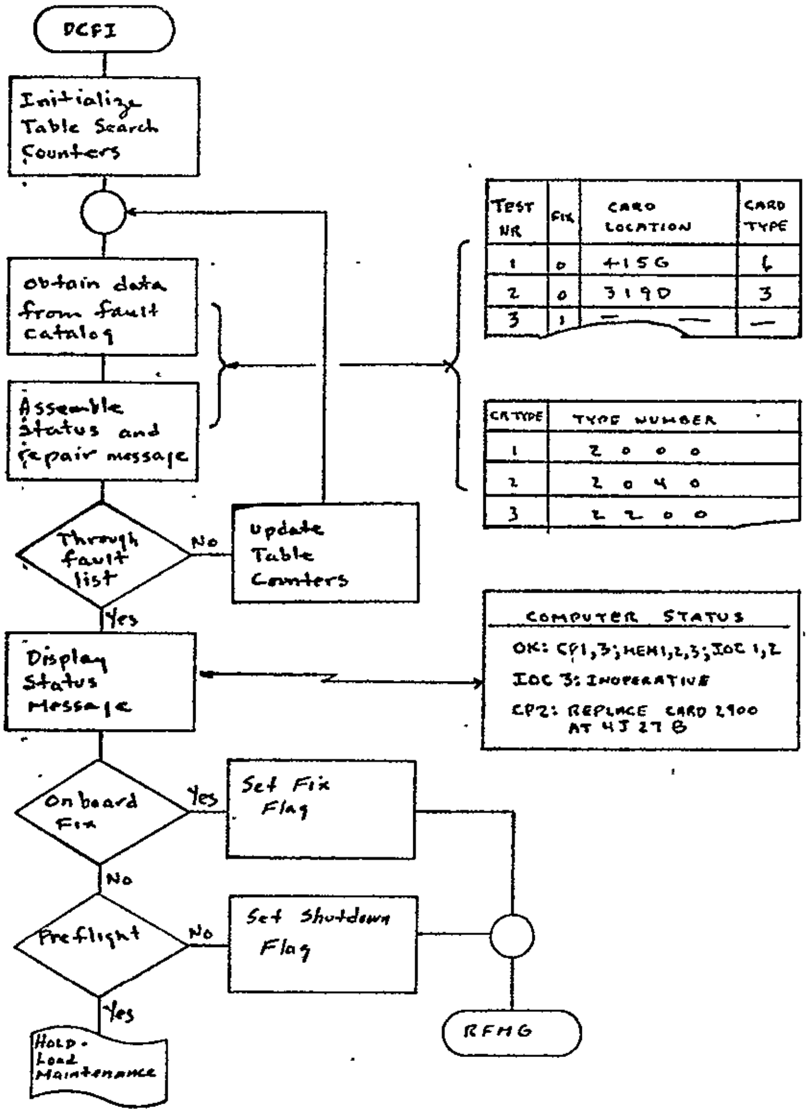


Figure 5-129 Digital Computer Fault Isolation

of additional test procedures or equipment is required to pinpoint the discrepancy. If this type failure is detected during the prelaunch checkout, a hold in the mission may be ordered to perform additional maintenance procedures.

As DCFI cycles through the catalog, it assembles the data gathered into a format suitable for status display. When all the faults reported by the test routines have been diagnosed, DCFI informs the display subsystem that the status and repair message is ready. If an onboard fix can be made by a card replacement, then the flight crew informs the computer subsystem (via the alphanumeric keyboard) whether an attempt to perform the card replacement operation is desired. If a fix is desired, DCFI sets a variable containing all necessary information and exits to the Reconfiguration Management (RFMG) routine. RFMG then monitors the fix until the applicable unit is either back in operation or declared inoperative. For unmanned flight, an onboard fix is not possible, so that in this case shutdown procedures for the failing unit is indicated. In addition for failures during manned flight where no card replacement fix is specified, shutdown procedures are initiated. DCFI again provides a data word for use by RFMG in reconfiguring to a degraded mode of operation.

Program Recovery (PRGY) procedures are used to provide techniques for continuing program operation in the event of various hardware or software faults. Figure 5-130 is a flow chart of PRGY.

PRGY is entered in response to faults or abnormal interrupts requiring software recovery techniques. PRGY is primarily concerned with recognizing the fault, taking any immediate action necessary, setting up an appropriate variable so that tests required can be conducted, and setting a flag so that RFMG can monitor the testing or supervise the shutdown of the unit at fault.

If a primary input power failure is sensed to be imminent, a hardware interrupt will cause a jump to PRGY. Typically, this interrupt occurs a minimum of 250 microseconds prior to power failure. During this time PRGY will store essential information which will be of value to RFMG in the event of the return of normal power to the unit. PRGY saves the contents of arithmetic and index registers, the time of entry into PRGY, the address at which the imminent power failure was detected, and any other vital information to the routine that was interrupted. PRGY determines which data is vital by means of a table correlating interrupt addresses and storage locations. PRGY then sets a power failure flag which is used by RFMG in reconfiguring the computer subsystem.

Abnormal internal interrupts are generated by the IO controllers (IOC) when various timing, buffer initiation, or memory reference criteria are not satisfied. Associated with each of these interrupts is a list of possible causes. Using this list PRGY forms a test request word for the indicated IOC. PRGY then sets a flag for use by RFMG in evaluating the

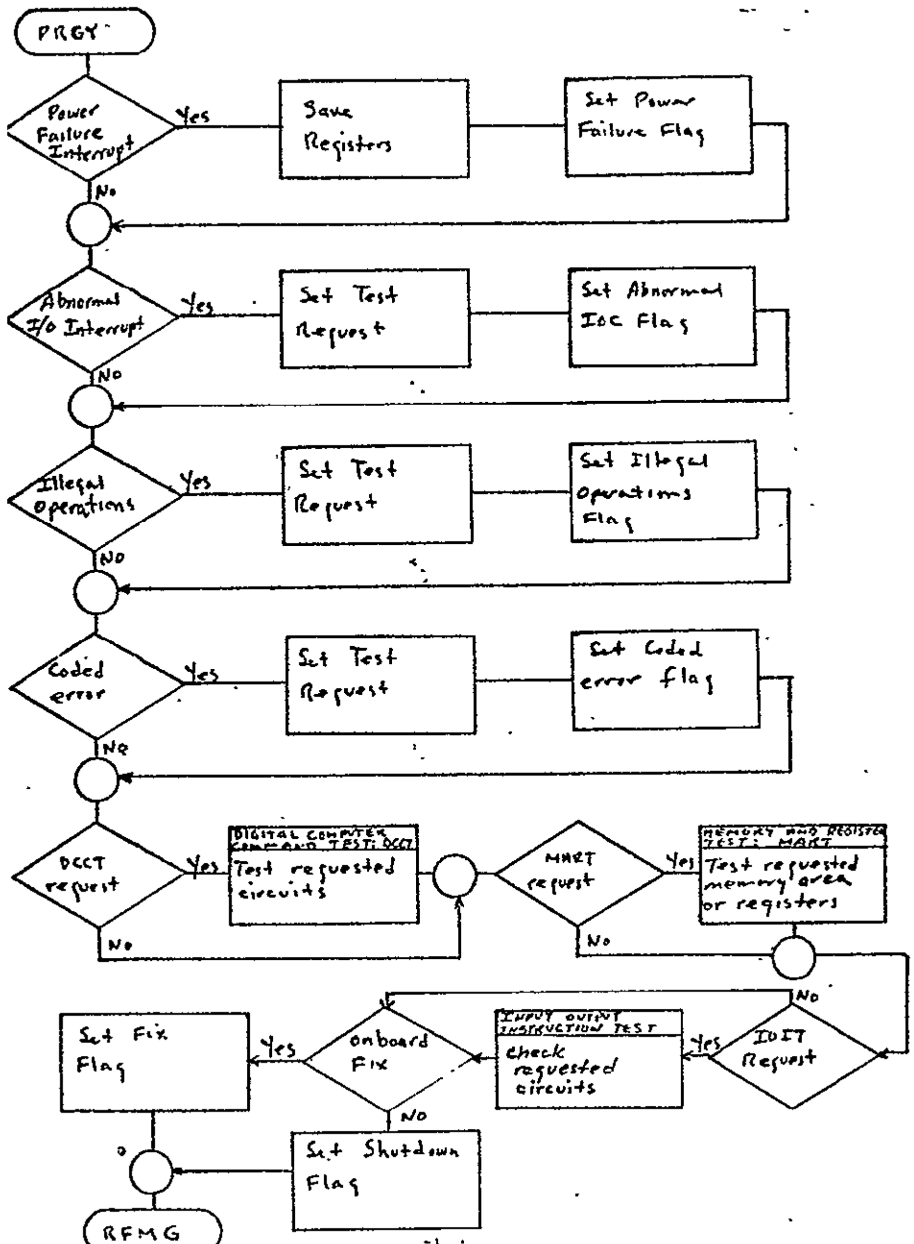


Figure 5-130 Program Recovery

computer subsystem. Similarly, the encounter of an illegal code or an illegal address in a routine indicates a probable memory failure as it is assumed that the software has been extensively debugged prior to operational use. PRGY is entered if illegal operations are detected. A test request word is formed and an illegal operations flag is set for RFMG monitoring.

Various coding techniques can be used to assist in determining subsystem discrepancies. Common methods include setting a time limit on computational loops, or magnitude limits on input data. Discrepancies of these types may indicate computer or other subsystem failures. PRGY sets up a test request and an error flag for use by RFMG. PRGY then goes to any of the test routines requested, determines if an onboard fix is possible and sets the appropriate flags for RFMG.

Reconfiguration Management (RFMG) procedures are used to supervise the computer subsystem configuration. Figure 5-131 is a flow chart of RFMG. RFMG receives information on the status of central processors, memories, IO controllers, interface units, and data transmission systems from other DMCX routines. Using this information, RFMG updates a table which maintains a status record and history of abnormal conditions or power failure for each unit.

If a power failure occurred, PRGY was entered where a flag was set and data was saved. RFMG checks this flag and, if set, the input power status to the specified unit is monitored. If power returns to normal, the unit involved is put back into the system unless RFMG's table shows that the power failure has been a recurring one. If the unit is acceptable after a power failure, a request for retesting of the unit is formed, the power failure flag is cleared, RFMG table is updated, and the data saved by PRGY is restored. On the next cycle through DMCX, the unit is retested and if the test is passed the output of the unit is considered valid. If the power failure is a recurrent one, or if the power does not return to normal within a prescribed time interval, the unit is taken out of the system. The shutdown flag for the unit involved is set and its power failure flag is cleared.

RFMG monitors attempts to repair computer units for which onboard card replacements are provided. Other DMCX routines have determined that an onboard fix is possible, have set the unit's fix flag and informed the crew by way of a CRT of the steps necessary to repair the unit. When the

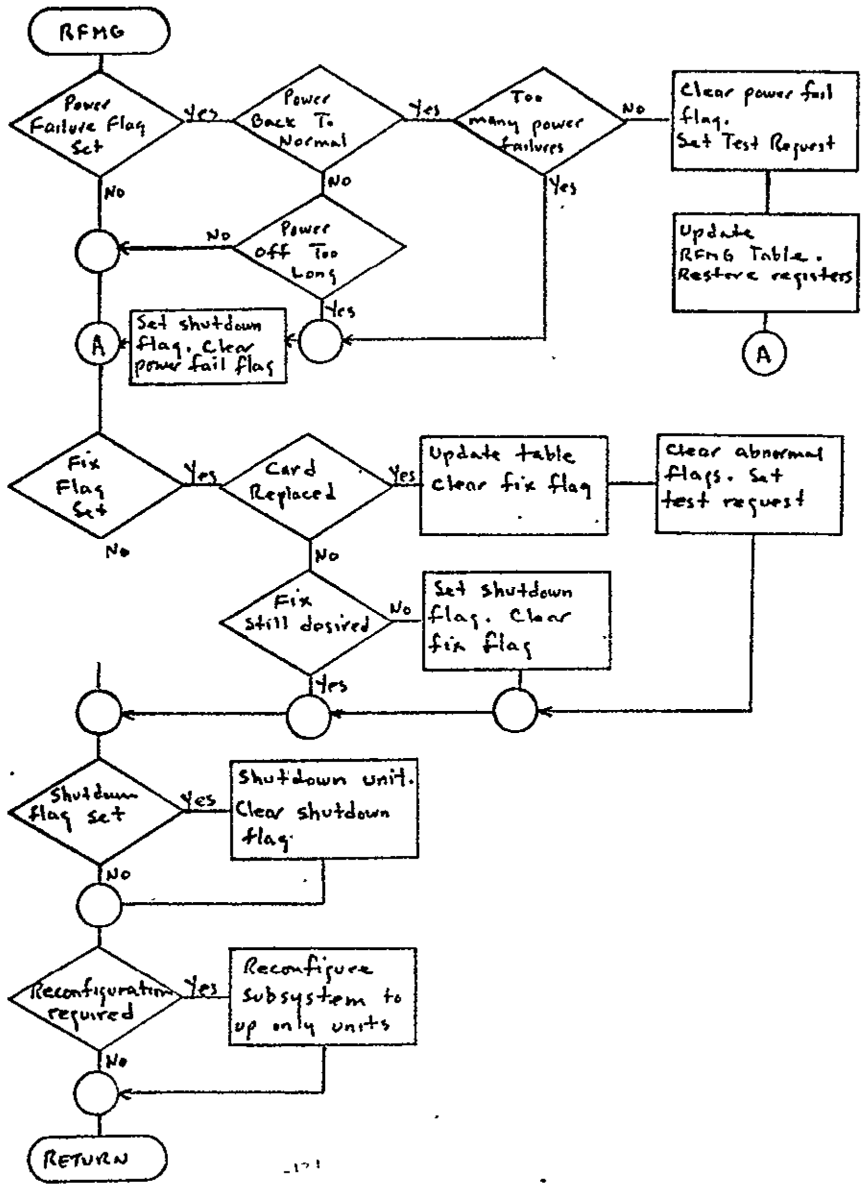


Figure 5-131 Reconfiguration Management

card is replaced, the crew informs the system by means of the alphanumeric keyboard. RFMG recognizes this action by updating its table, clearing various flags, and requesting a retest of the unit which has been put back into the system. If, due to time or other constraints, a fix of a particular unit is not desired, the crew activates the appropriate alphanumeric key. RFMG will then remove the unit from the system by setting the shutdown flag and clearing the fix flag. The RFMG table will maintain information of the repairable unit for status reporting or future emergency use.

RFMG performs shutdown operations for a unit if its associated shutdown flag is set. Reconfiguration consists either of adding units to, or removing units from the computer subsystem. Units which have recovered from a power failure, or have had a successful card replacement made are added to the system. Units which have been shutdown, have had a power failure, or are in the process of being repaired are removed from the system. RFMG monitors status to ensure that only units providing valid data are included in the current reconfigured system.

5.7.2 Displays and Controls

Displays and Controls Executive (DCEX). The displays and controls executive program uses the information on data content, display formats, display control, special processing functions, interface data formats, and manual methods and procedures presented in section 4.7 to supervise the displays and controls subsystem tasks. The functions of DCEX are to:

- Process requests for display functions and information
- Process DMS input data
- Provide data storage and retrieval capabilities.
- Provide means for controlling the distribution, updating, testing and scheduling of displays
- Provide display formats to meet display requirements
- Present the display output in a complete, accurate and easily interpreted form

Figure 5-132 relates these general requirements to the specific tasks anticipated for the booster, and Figure 5-133 presents a flow diagram of the Displays and Controls Subsystems Executive program.

DCEX is entered every 31.25 milliseconds (at refresh rate). When the displays and controls subsystem is first powered up, the Display Control (DTRL) routine initializes the displays to ensure their readiness for service A flag is set when the equipment is ready for use, so that subsequent program flow will be through the Input Processing (IPRC) routine. IPRC examines the matrix of current display requirements, obtains the desired information from the DMS, verifies its reasonableness, and then converts the data to

Executive Routines	Program Purpose	Subroutines
Request Processing	This function responds to standard or special display requests. A standard display is requested by a simple push button. Special display requests provide greater flexibility in selecting display content, formats and controls. With preprocessing and automatic mode, displays are routinely generated and updated.	. Emergency and Abort Procedures . Phase Change Display Initialization . Uplink Data Display . Alphanumeric Keyboard Code Processor . Data Entry Keyboard Codes
Data Storage and Retrieval	This function provides means for locating and retrieving various types of information from diverse locations in the data base. When any program or display request requires certain types of data, the criteria defining the data desired is provided. The data retrieval processes will then provide data by means of indexing, linking, and searching.	. Emergency Procedures Call Up . Check List Call Up . Parameter Table Look up . Filmslide Manager
Input Processing	This function performs the tasks necessary to ensure that the DMS information is compatible with display system requirements. These tasks include verification, unpacking and reformatting.	. Input Data Management . Display Data Preprocessor . Data Unpack and Verify . Display Data Code Conversion Input Buffer Storage
Display Control	This function deals mainly with the distribution, updating and priorities of display data. Testing and rescheduling, if required, are additional requirements. Regeneration of displays is initiated at the specified refresh rate. Display records are set up and maintained in the display file while it is in view.	. Timeline Management . Display Initializor . Display Application Scheduler . Time Display Update . Display Tester . Display Activate and Position
Display Formatting	This function determines the position, form and coding of information in a display. Combinations of symbolic, graphical and alphanumeric forms provide display formats in various predetermined position. When updating information is received the form and location of the data is determined and the record in the display file is modified	. Vector Generator . Conic Generator . Cursor Retrieval . Bar Chart Construction . Trend Analysis Chart . Display Scaling . Display Translation and Rotation . Display Perspective . Symbol Positioning
Display Output	This function performs the detailed steps which expands the data from the compact form in which they were stored and processed to the level of detail necessary to complete the display Initiates and controls the transmission of display to the display equipment.	. Scan Converter Management . Vertical Tape Controller . Electronic-Moving Bargraph Application . Flight Controller Management . Reprogrammable Switch Application . Circuit Breaker Monitor Application . Navigation Map Orientation . Refresh Memory Area Designator

Figure 5-132 Displays and Controls Functions, and Subroutines

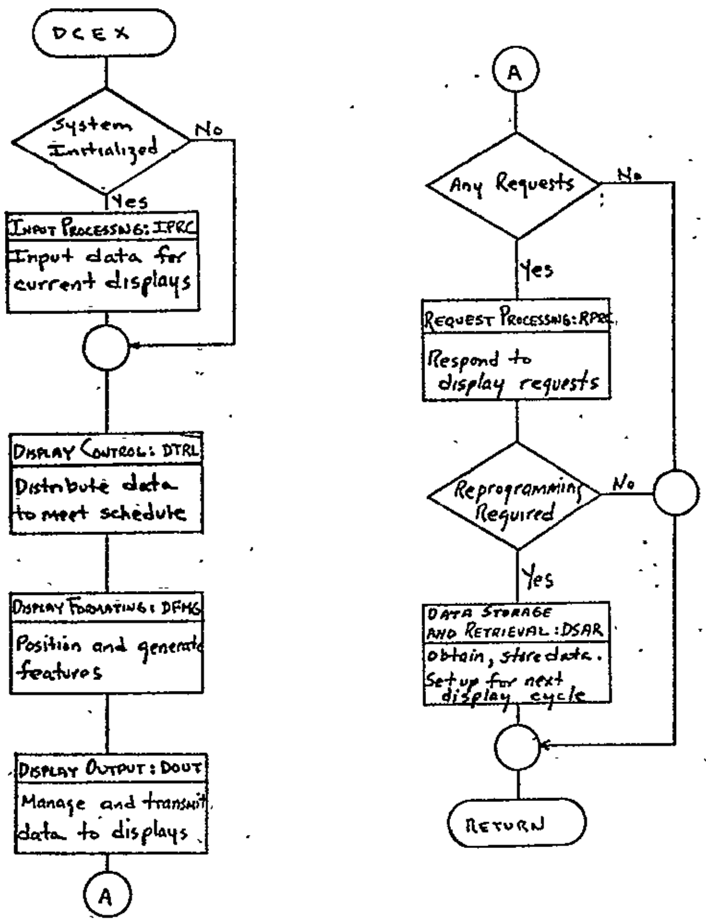
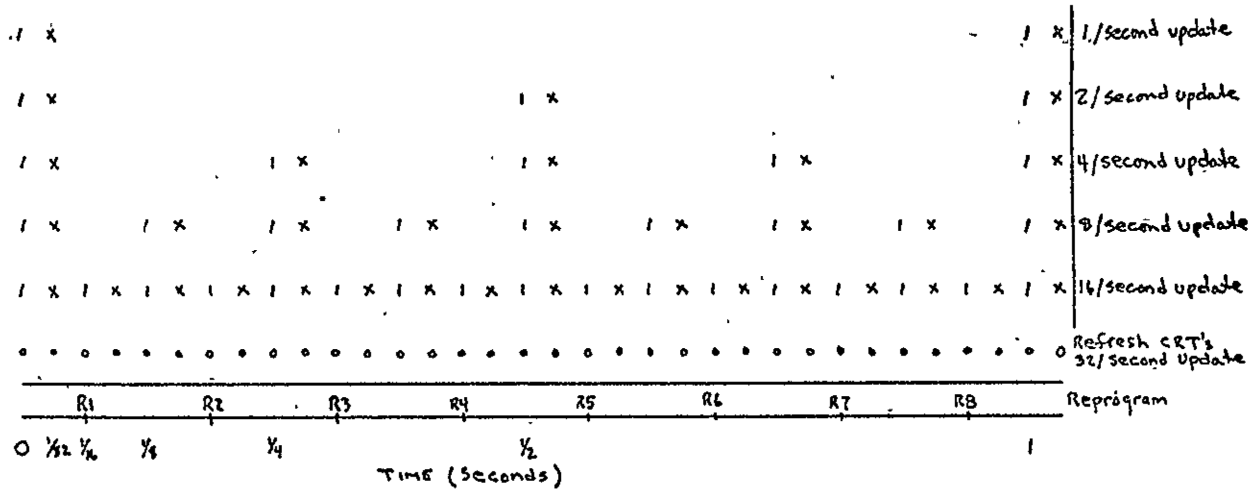


Figure 5-133. Displays and Controls Subsystem Executive

the format required for the current display file. DTRL then distributes this information in accordance with the timeline (Figure 5-134) and the current display usage schedule. Display commands, consisting of bit fields, are formulated into optimum word structures, and are used to initiate and control the various operations shown in Figure 5-135 These words are then formed into messages and transmissions. Display commands for functions, such as time displays, which do not require additional processing are completed by DTRL. The Display Formatting (DFMG) routine performs the major tasks in positioning the alphanumeric, graphic and symbolic features of the displays. The Display Output (DOUT) routine then performs any additional details required to ensure the generation of the complete and accurate display.

Flight crew operations at the alphanumeric or data entry keyboards result in interrupts which are recognized by the Request Processing (RPRC) routine. RPRC performs software tasks as necessary to satisfy these requests. In addition, the system Executive program informs DCEX when phase changes or abnormal conditions, requiring display usage revisions, occur. RPRC initiates the action necessary to answer these alerts. The Data Storage and Retrieval (DSAR) routine assists by obtaining data contained in mass storage units, filmslides, or memory tables. DSAR also sets up the frame work for display usage schedule plans and data parameter matrices.

Display Control (DTRL) The display control (DTRL) routine is concerned with the distribution, updating, testing and scheduling of the displays. The functions of DTRL are:



UPDATE GROUPS

/	EADI 1, HSD1, HUD1, MFD
X	EADI 2, HSD 2, HUD 2, VSI

REPROGRAMMING GROUPS

R1	EADI 1	R5	EADI 2
R2	HSD 1	R6	HSD 2
R3	HUD 1	R7	HUD 2
R4	MFD	R8	VSI, Reprogram Switches

Figure 5-134 Display Control Timeline

FUNCTION	OPTIONS	BITS
Beam Control	ON/OFF	1
Mode	Graphic/Alphanumeric	1
Blink	Start/Stop	1
Character Size	Basic/Large	1
Character Orientation	Normal/90° Left	1
Character Spacing	Normal/Large	1
Character Brightness	4 Levels	2
Character Color	White/Red/Yellow	2
Beam Position	X - Coordinate	10
Beam Position	Y - Coordinate	10
Line Widths	Normal/Emphasize	1
Vector Brightness	4 Levels	2
Vector Color	White/Red/Yellow	2
Pages in Checklist	Maximum 8 Pages	3
Start Of Message	Digital Code	4
End Of Message	Digital Code	4
End of Transmission	Digital Code	4
Frame Sync	Digital Code	4
Display Identification Number	7 CRT, 8 VSI, 8 Ded.	5
EADI Mode	8 Options/Phase	3
HUD Mode	8 Options/Phase	3
Vertical Tape Control	Direction and Dist.	5
Scan Converter Control	ON/OFF/Select	2
HUD Airspeed Reference	Set HUD reference	4
ILS Path Select	Shallow/Normal/Steep	2
Film Transport Control	ON/OFF/Forward/Reverse	2
Map Orient	North/Course	2
Filmslide Number	100 Maximum	7
Filmslide Course Control	360° Max to 1°	10
Dedicated Instrument Value	Variable	7
VSI Parameter Value	Variable plus Control	11
Controller Positions	Variable	7
A/N Keyboard Filmslide Group	2/Phase	1
A/N Keyboard Control	Lock/Unlock	1
A/N Line Number	1 to 52	7
A/N Column Number	1 to 74	8
A/N Character	43 ASCII Codes	7
A/N Functions	32 ASCII/Phase	7
A/N Control and Editing	12 ASCII Codes	7
Data Entry Keyboard Controls	5 Discretes	5
Data Entry Functions	1 to 15	4

Figure 5-135 Display Commands

- Perform display warm up and initial activation
- Provide tests to ensure display readiness and accuracy
- Maintain, revise and manage the display usage plan
- Update the current display file as required by the timeline schedule
- Control, activate or position display devices
- Maintain the GMT and mission time displays

The flow chart for DTRL is shown in Fig. 5-136 DTRL is initially entered early in the preflight check sequence. As the displays are a primary tool for the booster checkout, the displays themselves are first initialized and tested. The Display Initializor (DINR) protects the CRT circuitry during its initial activation, and then presents a test pattern on each CRT display. The test pattern can be manually adjusted and focused by the flight crew as desired. The other display and control equipment are set to their preflight settings. The Display Tester (DITR) is then entered and each displays operational capability is verified. Special test patterns verify the persistence, capacity and accuracy of each CRT. Other tests check the electronic-moving bargraphs, the dedicated instruments, keyboards and film transport units. DITR may also be entered during the operational mission if degraded display performance is detected, or if a request for display testing is made by manual operation of a keyboard function key. If DITR determines that a display is not operating properly, the Display Application Scheduler (DASC) routine is entered and the master display usage plan is revised. DASC informs the flight crew of revisions in display usage, records the discrepancy for post-flight maintenance analysis,

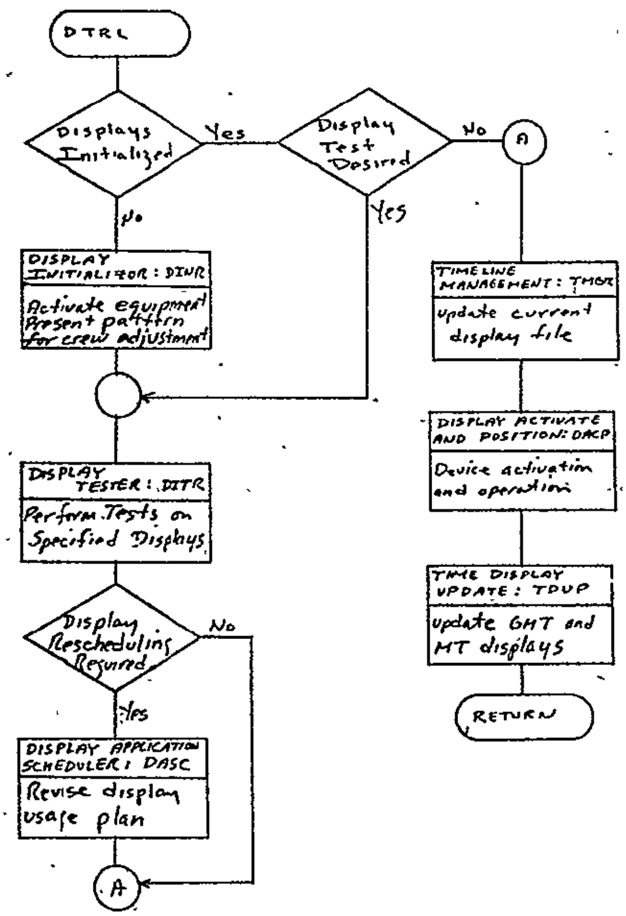


Figure 5-136 Display Control Flow Chart

and performs shut down or degraded mode procedures on the effected equipment.

The timeline management (TMGR) routine is responsible for the timing requirements shown in Figure 5-134. If reprogramming of the displays is required because of a phase change or equipment malfunction, the change in each display task is accomplished in the allocated 125 millisecond time slot for each reprogramming group of displays. The displays are also divided into update groups so that the programming tasks are more evenly divided across the timeline. TMGR then takes the data from the input buffer memory area (data preprocessed by the Input Processing routine) and moves it into the display file area reserved for each display. More detailed processing and combination of data is then performed by the Display Formatting and the Display Output routines.

If reprogramming calls for the activation or the assignment of new tasks to a particular display, the Display Activate and Position (DACP) routine performs the necessary tasks. These include functions such as centering of CRT beams, and initial positioning of vertical tapes and map course settings. DACP sets, clears, and maintains a record of discretes pertaining to the displays and controls subsystems. The time display is a function which does not require additional processing prior to display generation. If the GMT or mission time displays have been requested, the Time Display Update (TDUP) routine uses the time count data maintained by TMGR to revise the messages to generate new time display output. If a revised time signal is received from a ground-based station, this information is received by TDUP and used to reset the appropriate display.

Request Processing (RPRC) routines respond to manual interrupts generated by the flight crew at the alphanumeric or data entry keyboards, and to flags or signals which indicate a change or special application for the displays. The functions of RPRC include:

- Processing alphanumeric (A/N) keyboard function code interrupts
- Processing A/N keyboard alphabetical and numerical inputs
- Processing A/N keyboard editing and cursor control requests
- Processing data entry keyboard function code interrupts
- Initializing display rescheduling in response to change of phase
- Establishing priority and performing initializing in response to an emergency or abort procedures alarm
- Perform interpretation and initialization for accepting and presenting uplink command and informative data

Figure 5-137 is a general flow diagram of RPRC. If a dangerous or potentially dangerous situation is encountered, through BITE or system diagnostic routines, an alarm flag is set by the system Executive and the Emergency and Abort Procedures (EAAP) routine is entered. EAAP first provides an aural alert signal and then displays a short notice which indicates the general nature of the emergency. The appropriate check lists are then obtained, and emergency procedures are initiated. The execution of these procedures may require crew/keyboard interface. In this case, RPRC continues with the processing of requests. If the emergency is of such a nature that time requires automation of recovery procedures, or abort procedures are required, then other display requests are ignored until the emergency has passed.

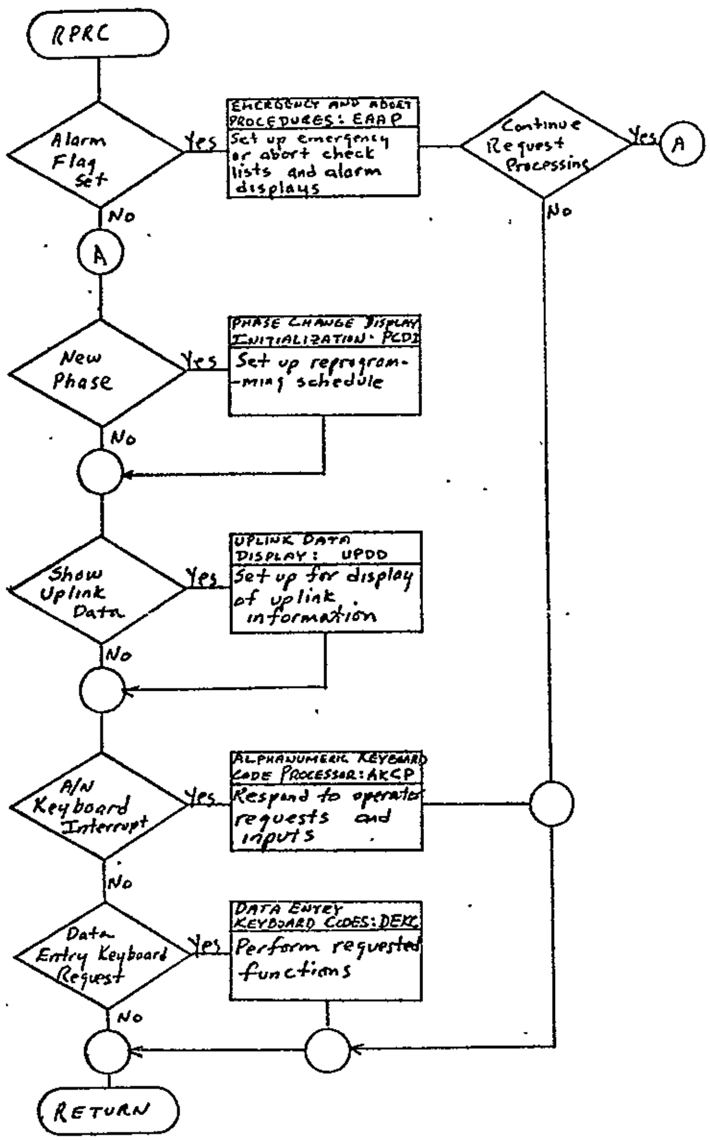


Figure 5-137 Request Processing Flow Diagram

The system Executive program maintains a record of events and when a new mission phase begins a variable is set. The Phase Change Display Initialization (PCDI) routine recognizes the change in phase and sets appropriate flags and indices so that the Display Control routine can perform the required reprogramming. PCDI is aware of any emergency under EAAP's supervision so that initialization procedures may be revised as required.

Commands and information sent by ground stations are processed by uplink routines. The Uplink Data Display (UPDD) routine is informed when messages are received. UPDD matches a list of anticipated messages with its associated alphanumerics and displays the information. Normally, these messages are presented on a reserved portion of a particular display. For example, if the booster has been ordered to hold prior to a landing approach, the position of the booster in the holding pattern is shown on the Horizontal Situation Display (HSD). When the ground station pattern departure message is received, UPDD will perform the set up steps required to display the alphanumerics "LEAVE HOLDING PATTERN AT 072618 GMT" adjacent to the holding pattern outline. Similar procedures are initiated by UPDD for other message types.

A display keyboard is provided as the primary means for flight crew communication with the displays and controls subsystem. The Alphanumeric Keyboard Code Processor (AKCP) routine answers the request generated by manual operation of the keys. AKCP has three subroutines which handle the function, alphanumeric, or control and editing keys respectively. The primary group for operational use are the function keys which initiate a particular program applicable to the displays when a key is depressed

AKCP recognizes the code assocated with the button, provides a brief message to indicate which function is to be activated, and, when this message is verified and accepted by the crew, activates the designated program. The alphanumeric editing and cursor control keys are used, primarily, by programming and maintenance personnel for on-line debugging and diagnostic purposes. Some limited use of these keys is anticipated for the operational booster. For example, the cursor control can be used to point to one of multiple choice options presented by a check list, and query or function key completion routines will require the use of alphanumeric characters.

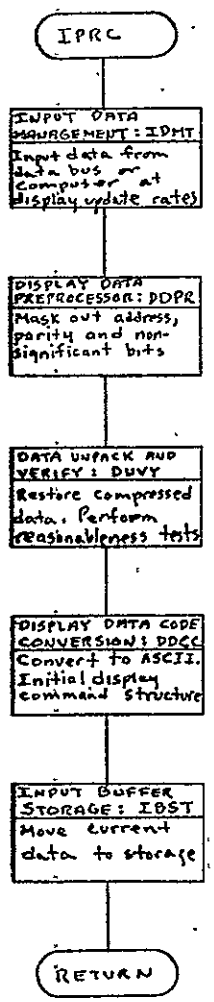
The data entry keyboard is used to enter data into, obtain data from, and request tests of the DMS computers. The Data Entry Keyboard Codes (DEKC) routine associates the function key operated by the pilot or copilot with a particular program. If the program requires manual entry of parameter values, DEKC will initiate the display of cues, accept the numerical data and convert it to the format required for storage in computer memory. For all programs initiated by keyboard request, DEKC provides any necessary display cues and provides a summary of program results.

Input Processing (IPRC) routines interface with the data bus and computer subsystems to provide display data as required to generate current updated displays. The functions of IPRC are to:

- Input data from the data bus or computer at the specified update rates
- Perform initial preprocessing of the input data
- Verify the reasonableness of data and perform unpacking tasks
- Complete conversion to display codes and provide information for display command structures
- Move the converted data to an input buffer storage area for use by other DCEX routines

Figure 5-138 is a general flow diagram of IPRC routines. The Input Data Management (IDMT) performs the input control tasks required in interfacing with the data bus and computer subsystem. IPRC outputs requests for data, and verifies address, parity and synchronization bits prior to accepting input information. IPRC uses the information contained in the current display usage plan to maintain a matrix containing source of input data and update rates. Input timing is controlled through counters and indices which match the specified update rates and groups as shown in the Display Control Timeline.

The input data then goes through various steps to prepare it for display generation. In the first step, the address, parity and sync bits are masked out of the data word. If the data word contains more bits than required for display resolution, the appropriate number of least significant bits are also removed. Some data to be displayed may be transmitted on the



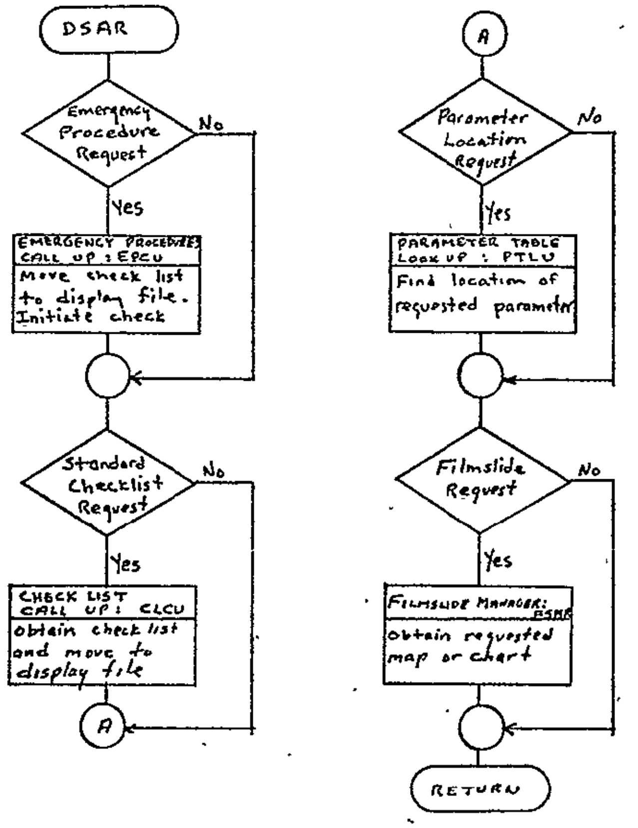


Figure 5-139 Data Storage and Retrieval

Figure 5-138
Input Processing

data bus or be available in computer memory in a packed or data compressed format. The Data Unpack and Verify (DUVY) routine maintains a list of these data types with an associated switch table to the appropriate unpacking procedure. These procedures are normally the reverse of the operations described for the data compression routines of section 5.1.6.3. For example, if the transmitted value of the sample is

$$T_n = Y_n - K_y$$

where

Y_n is the actual value of the n-th sample and, K_y is the bias constant for the signal Y then, the data word for display purposes is

$$Y_n = T_n + K_y$$

The value of the bias constant can be obtained from the tables maintained by the data compression routines. DUVY also conducts reasonableness tests as an aid in reducing the possibility of displaying erroneous information. Code conversion is performed by the Display Data Code Conversion (DDCC) routine. DDCC converts, for example, octal data or 2-out-of-5 frequency codes to the ASCII code used for display generation. DDCC will also add various information bits which can be used by other DCEX routines to construct the display command framework. For example, if the reasonableness tests of DUVY show that a parameter to be displayed is in a caution area, bits indicating blink yellow can be included in the data word. The final step is to store the data word in its assigned memory location. This function is performed by the Input Buffer Storage (IBST) routine which maintains a current table for display data storage.

Data Storage and Retrieval (DSAR) routines are used to obtain information contained in the data base, computer memory, and special peripheral equipment. The functions of DSAR are to:

- Provide emergency check lists and initiate the check procedures
- Provide standard check lists and initiate the check procedures
- Maintain tables of parameter locations and provide directory service
- Provide map, procedural, and chart filmslides on request

Fig 5-139 is a general flow chart of the Data Storage and Retrieval (DSAR) routines. Priority is given to requests for emergency procedure lists. The Emergency Procedures Call Up (EPCU) routine is requested by EAAP to provide a specific list to be used in resolving the emergency. The emergency procedure mass storage file is searched for the requested list. When it is obtained it is moved into the display file for the CRT assigned to present the procedures. Verification tests are used to ensure that no errors occur in the movement of the list. If errors occur, a copy of the list is obtained from the mass storage area. The title and first step of the procedures are then moved to the display refresh memory unit of the assigned CRT. Table length counters and indices are set for the use of EAAP in continuing emergency check procedures.

Standard checklists are normally required at the start of each new mission phase to initialize a new booster configuration and to prepare for operational modes. The Check List Call Up (CLCU) routine is flagged by PCDI to obtain a specific list. Checklists may also be required following keyboard function key operation. CLCU uses the information contained in calling routine information words to search for the required list on the magnetic tape storage units. The check lists normally contain more words then a CRT display refresh memory can hold. CLCU will move the first page of the checklist into the display file of the CRT assigned to present the checklist and will also move the title and first step of the check procedures to the appropriate display refresh memory unit. CLCU maintains a record of the page numbers in each checklist. CLCU is informed as each page is executed and, if additional pages exist, the next page is loaded into the display file.

Some data entry keyboard functions result in the loading of computer memory locations with numerical data entered by the flight crew. The Parameter Table Lookup (PTLU) routine assists in the execution of these functions by providing memory address and other information required to complete the data entry functions.

Filmslides are used to provide the basic framework for navigation maps, departure and approach procedures, charts and graphs. Additional alpha-numerics and graphics are added by other DCEX routines to complete the display. As in the case of prestored checklists, the use of filmslides will reduce the requirement for bulky maps, handbooks, and operational manuals aboard the booster. The Filmslide Manager (FSMR) routine maintains a list of filmslide numbers and contents. When a request for a filmslide is received this list is searched and the appropriate film transport unit is operated to show the requested slide.

Display Formatting (DFMG) routines are used to construct and position CRT display features prior to display generation. The functions of DFMG are:

- Generate coordinates for vector and lines
- Generate parameters for specifying arcs, circles or ellipses
- Obtain cursor position information and perform functions using this data
- Construct and maintain bar charts displayed on CRT's
- Construct and maintain trend analysis diagrams
- Provide scaling to optimize presentation of display features
- Provide two dimensional perspective to maintain pictorial realism
- Position alphanumeric, warning or other special symbols as required to complete the display

DFMG routines are concerned with the tasks required to maintain the CRT application programs associated with the display usage plan, and to maintain any special symbology required as a result of manual input requests or alert condition requirements. The Display Output (DOUT) routines then complete the CRT display, if required, in addition to completing the displays using the dedicated instruments.

Fig 5-140 is a general flow chart of the Display Formatting (DFMG) routines. The Vector Generator (VGTR) routine provides the starting point (X_1, Y_1) and the stopping point (X_2, Y_2) for lines and vectors. Vectors are used for horizon lines, runway centerlines, heading and velocity vectors, and other applications specified for the CRT application programs.

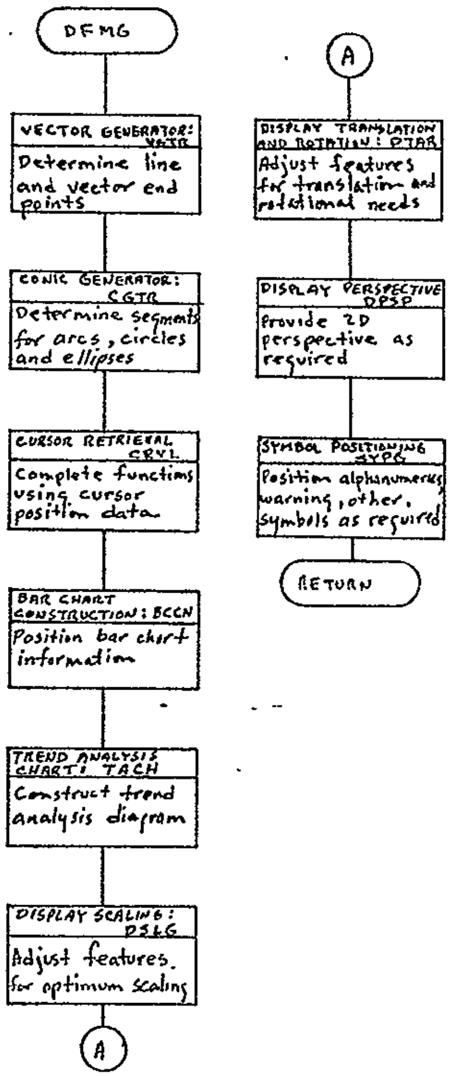


Figure 5-140 Display Formatting

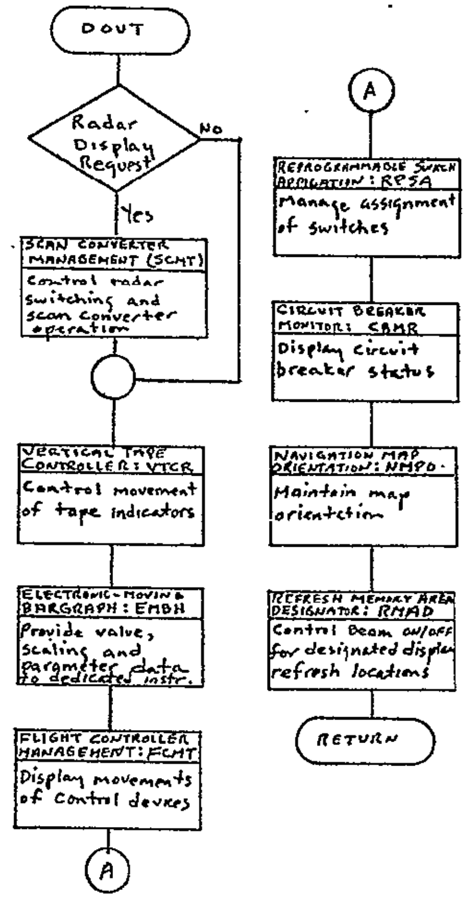


Figure 5-141 Display Output

NOT REPRODUCIBLE

Algorithms using system inputs are required to generate each vector. For example, a runway centerline is computed from distance to the field, aircraft heading, map scaling, and runway length information. The Conic Generator (CGTR) provides arcs, circles and ellipses for CRT application programs. For a circle, CGTR obtains the center point and radius information from prespecified application program requirements. For example, the flight director circle for approach to a landing may be specified to have a radius $\frac{1}{2}$ the runway width, and a center at the point where the commanded heading intersects the ground. CGTR then constructs a circle from line segments using an iterative technique to advance starting and stopping points. The number of line segments per octant is a function of the radius of the circle and is designed to give a smooth curve appearance on the CRT. Ellipses are similarly constructed from information obtained on the lengths of the major and minor axis and the positioning requirements for the conic. Arcs may be specified for some application programs. For example, half circles are drawn for holding pattern magnification. For this task the center point, radius, start and stop point information can be derived from prestored information on the holding pattern characteristics. VGTR can be used to complete the pattern by constructing the parallel lines of the pattern.

The cursor of the display keyboard can be used to supply raster points to the console memory. Using this capability the flight crew can request various information from the system. The Cursor Retrieval (CRVL) routine maintains the display of the requested information. For example, the crew may position the cursor at a point on the HSD map and request estimated time of arrival. CRVL will compute the requested information from current data on ground speed, position and heading, and then position the data at the optimum place on the assigned display. CRVL then updates the display until the request is terminated.

In addition to the electronic moving bargraphs which are dedicated to various parameters, the CRT's can be used to generate bar charts from the parameter values available to the displays and controls subsystem. The Bar Chart Construction (BCCN) routine determines the position of the end points of the bars. Bar charts are used in conjunction with filmslides which contain legend, abseissa, ordinates and other fixed information. BCCN maintains a record of parameters required for each chart along with a listing of initial points for each bar. The CRT has 1024 raster points in the vertical (Y) and horizontal (X) directions. If the largest parameter value exceeds 1024, then the Display Scaling (DSLG) is called upon to rescale the bar chart. When the four coordinates are found for each parameter, BCCN sets up the display command word for line generation.

Trend analysis charts provide predication and extrapolation information to the flight crew. For example, the Trend Analysis Chart (TACH) routine uses fuel consumption data to predict times at which fuel reserves are sufficient to reach an alternate. TACH updates the end point on the charts curve as the current sample is received, refreshes the past history, and predicts future values using dashed vectors. As in BCCN, fixed legends are shown using filmslides.

The Display Scaling (DSLG) provides assistance to BCCN for bar chart scaling, adjusts features on navigation maps as required by map scaling, and performs magnification of patterns and features when requested. The bar chart scaling ensures that parameter values do not exceed the available 1024 raster points. Rescaling is performed by assigning the largest parameter a value near the maximum of 1024 and then provides a proportional value for the other parameters being displayed. For map scaling, DSLG maintains a list of scalings used for each map in the filmslide library. Each adjustable parameter to be displayed on the associated CRT is checked against this list and its relative value maintained by DSLG. Magnification of any feature on a display may be requested by the crew through the use of the cursor to point to the area to be magnified following activation of the appropriate function key on the display keyboard. DSLG reads the cursor points using the CRVL routine, tests whether the requested magnification (by optional powers of 2) will overflow the display, and repositions the display if required. The Display Translation and Rotation (DTAR) routine is used to assist DSLG in display repositioning. For example, if the crew is ordered into a holding pattern, a request for magnification of the pattern may be inputted. The pattern is then translated so that the center of the pattern is at the display center. Rotation is accomplished so that magnetic north or heading of the inbound leg (at the crew's option) is at the top of the display. DTAR is also used to maintain the attitude displays.

Some special displays, such as runway outlines, use two dimensional perspective to enhance the realism of the display. The Display Perspective (DPSP) routine uses information on runway lengths and widths, altitude and distance to the field in order to adjust the coordinates of the end points of the runway edges.

As a final step in completing the display commands for CRT generation, special alphanumerics or graphics are added by the Symbol Positioning (SYPG) routine. Display commands for generating some special symbols are prestored. SYPG generates the position commands for these display features (in general, these are unusual shapes which would required excessive software to generate). SYPG maintains a record of the relative positions of alphanumeric data and graphical information and generates the appropriate position commands.

Display Output (DOUT) routines are used to complete CRT displays prior to regeneration at each refresh period, to perform special functions, and to complete dedicated instruments prior to each updating period. The functions of DOUT are to:

- Provide control of the radar switching and scan converter units when radar displays are requested
- Control and maintain the operation of vertical tape indicators used with CRT displays
- Update electronic-moving bargraphs and other dedicated instruments
- Display movements of control devices
- Manage the assignments of reprogrammable switches
- Monitor the circuit breaker status program and provide recovery information
- Maintain the orientation of navigation map filmslides
- Designate the areas of the CRT display refresh memories to be regenerated and provide beam control commands

Fig 5-141 is a general flow diagram of DOUT. The scan converter management (SCMT) routine is entered if a doppler or weather radar display is desired. SCMT manages the superimposition of the designated radar picture on the specified CRT by performing switching, and monitoring tasks. Vertical tape indicators are associated with the CRT's in the display of parameters such as altitude and airspeed. The Vertical Tape Controller (VTCR) routine obtains the current parameter value, and compares it to the current tape setting. If the change in the parameter is greater than the resolution of the tape, then VTCR activates the tape in the appropriate direction to put the new tape value next to the fixed pointer. VTCR then clears the tape

INCREASE or DECREASE discretes until the next update period. The computer driven electroluminescent vertical scale indicator (VSI) is an bargraph whose indicators are controlled and updated by the Electronic-moving Bargraph (EMBH) routine. EMBH maintains a record of the current assignments of the VSI's, obtains the parameter values from the display file memory, and formats this data to meet the VSI requirements. EMBH also performs similar functions for instruments which are dedicated to individual parameters throughout the mission. The display of flight control devices require special indicators. Control of these indicators is the function of the Flight Controller Management (FCMT) routine. FCMT also initiates any aural or visual warning signal required.

The Reprogrammable Switch Application (RPSA) routine correlates the operations of the alphanumeric (A/N) keyboard and the CRT's involved in reprogrammable switch functioning. RPSA controls filmslide usage on the function keys of the A/N keyboard, verifies the sequence of operations of push button switches and provides display information to the crew for normal and recovery functions. The Circuit Breaker Monitor (CBMR) routine provides the flight crew with status information. Onboard checkout routines will notify CBMR when a circuit breaker is tripped. CBMR then activates an aural alert signal, prepares the alphanumeric message associated with the particular circuit breaker that has opened, and moves the message into the display file and display refresh memory area reserved for caution messages. When the circuit breaker is pushed, CBMR clears the message. CBMR also provides status messages of all system circuit breakers upon flight crew request. Navigation maps may be oriented with magnetic north or booster magnetic heading at the top of the filmslide. The Navigation Map Orientation (NMPO) routine tests the option selected by the crew, and uses the magnetic heading information in the display file to control the film transport's rotation mechanism.

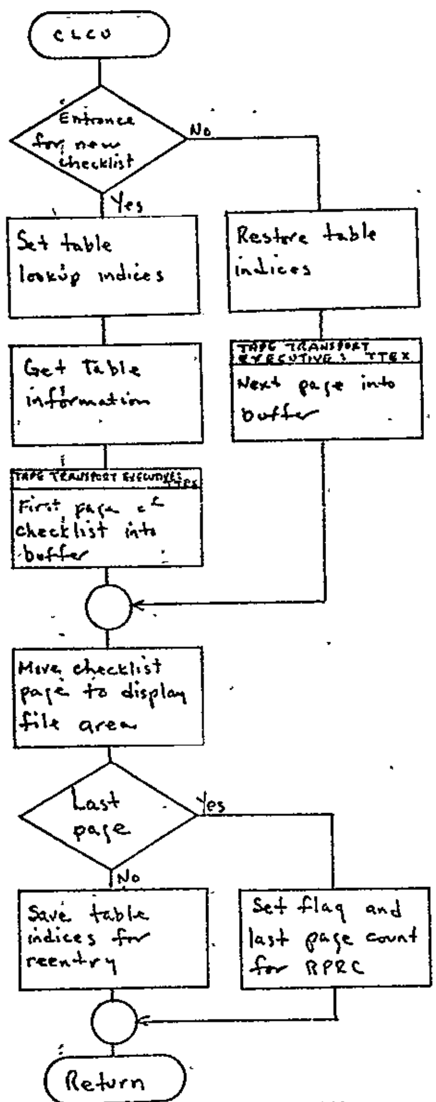
To save time involved in the refresh memory operation only those portions of memory which are to be displayed are refreshed. The Refresh Memory Area Designator (RMAD) routine maintains a record of the areas of each CRT which are to be regenerated at each refresh period. RMAD sets up jump instructions and controls CRT beam operation so that only the required areas are refreshed. For example, in a checklist operation only the title line and the current check line are required to be displayed.

Check List Call Up (CLCU) procedures are used to obtain the normal procedure check lists on request of RPRC. Where a new checklist is required, RPRC initiates the request to CLCU by providing a checklist number. CLCU sets table lookup indices and obtains search code, page and line information from a prestored table. TTEX is entered with the search code request and the first page of the checklist is read into the magnetic tape buffer storage area.

The checklist page is then moved into the designated display file area where RPRC will supervise the presentation and execution of the checklist steps. When a page is completed a return is made to CLCU to provide the next page. On this entrance, CLCU restores link table indices and repeats the operations with TTEX and the display file area. A test is made to determine if the last page of the checklist has been processed. If more pages are required, CLCU saves necessary data and exits for RPRC action. When the last page has been obtained, CLCU provides RPRC with line count information and completes its functions until the next request. Figure 5-142 is a flowchart of CLCU, and also provides information on normal procedure checklist requirements and link table format.

Emergency Procedures Call Up (EPCU) routine is used to :

- Find a requested emergency checklist
- Verify and move the checklist to the designated display files and CRT's
- Provide information to EAAP for use in the supervision of checklist execution.



	Procedure	Memory (32 bit)
1	Prelaunch	7500
2	Separation	250
3	Reentry	350
4	Rocket Fuel Dump	120
5	Ferry Preflight	5040
6	Ferry Starting	275
7	Before Takeoff	240
8	Takeoff and Climb	180
9	Cruise	50
10	Descent	150
11	Approach	125
12	Landing	90
13	After Landing	80
14	Shutdown and Parking	150

	Search Code	Nr. of Pages	Nr. of lines on last page
1			
2			
3	SRTY	2	100
4			
5	5FPF	40	40
6			
7			

Figure 5 -142 Checklist Call Up

EPCU is entered initially when a need for an emergency checklist is determined by the system's executive program. To provide redundancy and confidence in the checklist accuracy two copies of each list are stored in memory and an additional copy is available on magnetic tape if required to resolve conflicts between the copies in memory. When EPCU is first entered the emergency number is provided and EPCU conducts its initialization procedures. The table lookup index is set and checklist word and line item counters are initialized. Entering the table, the checklist starting addresses are obtained. The tape search code is also obtained for later search procedures if required. Figure 5-143 illustrates these initialization procedures. Using the address information obtained from the link table, the first word, O_1, is obtained from its prestored memory area, and the copy of this word, C_1, is obtained from its storage area (which is a known increment from the storage area of the original checklist). These two words (the first four characters of the checklist title line) are compared and, if they are the same; O_1 is used in assembling the title line. If they do not compare, a third copy of the checklist is obtained from the mass storage on magnetic tape by calling on TTEX using the search code found in the link table. This copy is then used as a majority voter in resolving conflicts between the checklists stored in memory.

After each word is read, compared and assembled as part of the current checklist line item, a test is made to determine if the word being processed

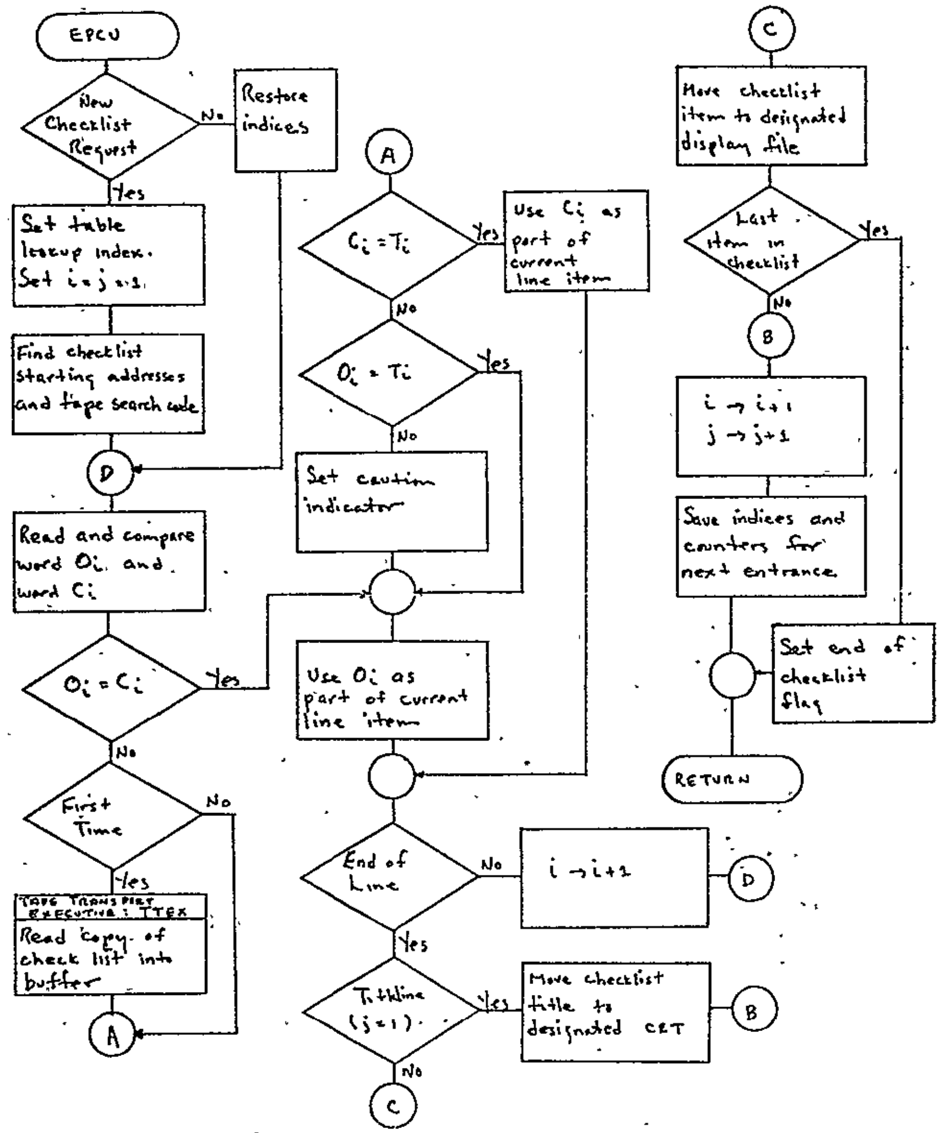
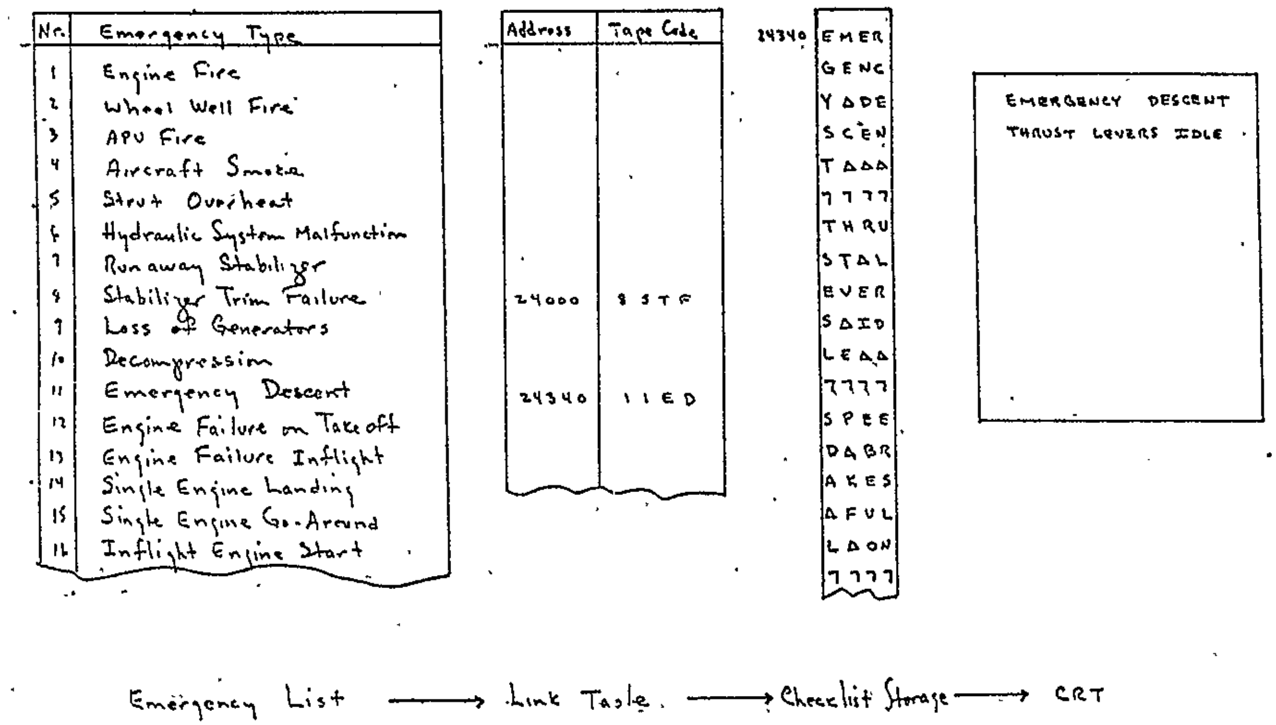


Figure 5-143. Emergency Procedures Call Up.

is an end of line code (the 7777 of Figure 5-144). If it is not the end of a line, the word counter is incremented, the next word in the checklist is obtained and the process is repeated. When the end of line code is encountered, the line item is moved so that it can be displayed under EAAP control. If the item is the emergency checklist title line, it is moved to the designated CRT. EAAP clears the remainder of the display and waits for the checklist lines form EPCU. As each checklist item is completed, EPCU sends it to the designated display file. When the required action has been completed by the flight crew, EAAP reenters EPCU with a request for the next item in the checklist.. EPCU restores its word and line indices and assembles the next line. A test is made to determine if the current line being processed is in the last one in the checklist. If it is, an end of checklist flag is set for EAAP's information and EPCU has completed its functions for the current request.



Figuer 5-144 Emergency Checklist Processing

5.7.3 Executive Program

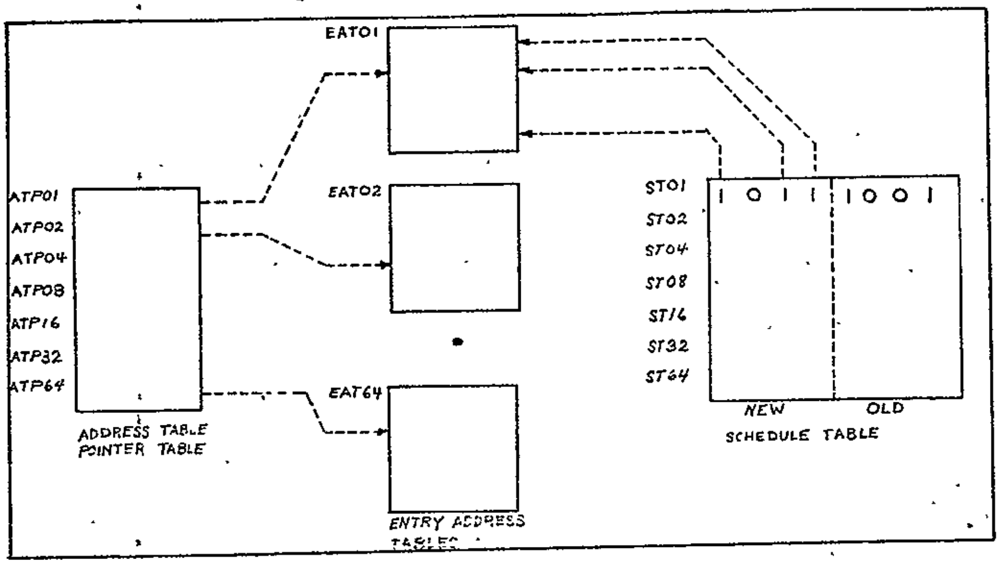
Each computer contains an executive program which controls the sequence and timing of the execution of all other programs in the computer except interrupt routines. Executive programs, in general, reduce computer efficiency since they require the execution of instructions which are non-productive to the primary computer tasks. In an avionics application the computational requirements upon the computer are well specified which allows a minimal executive program to be used. Each program required by the avionics system is executed at its specified rate which is either 1, 2, 4, 8, 16, 32, or 64 times per second. Thus each program can be classified into one of 7 groups associated with its execution iteration rate.

The assumed executive program contains several tables used in the control of the program execution flow. These tables are:

<u>Entry Address Tables</u> There is an entry address table for each rate group. Each table contains the entry addresses of all of the programs in the computer to be executed at the associated execution rate. The first location in each entry address table is designated EAT01, EAT02, EAT04, etc., for the rate groups 1, 2, 4, etc., respectively.

<u>Address Table Pointer Table</u> This is a table with 7 entries, one for each rate group. Each entry is a single word containing the address of the word in its associate entry address table which contains the entry address of the program presently being executed in that rate group. Each entry in the address Table Pointer Table is given its own designator, ATP01, ATP02, ATP04, etc. Figure 5-145 shows the

Figure 5-145 Main Executive Tables




```
        DIS                 DISABLE INTERUPT SYSTEM
        STX   TEMP,1
        LDX   IRRGA,1
        SAB   0,1
        LDA   TEMP
        STA   2,1
        EXA   2             STORE ALL REGISTERS
        STA   3,1
          :
        EXA   7
        STA   8,1
        LDA   PLUS2         SET STATUS WORD
        STA   9,1
        LDA   SW64
        TNZ                 RESTART 64/SEC RATE GROUP
        LDB   MINUS
        STB   SW64
        LDA   TIME          INCREMENT TIMER
        ADD   PLUS1
        STA   TIME
        SRC   1
        TRN   EXEC
        SRC   1
        TRN   PW32          TEST FOR WHICH RATE GROUPS TO RESTART
        SRC   1
        TRN   PW16
          :
        SRC   1
        TRN   PW02
PW01    LDA   SW01
        TNZ
        STB   SW01
PW02    LDA   SW02          RESTART DESIRED RATE GROUPS
        TNZ
        STB   SW02
          :
PW32    LDA   SW32
        TNZ
        STA   SW32
        TRA   EXEC
```

Figure 5-146 Executive Interrupt Routine relationship between the Address Table Pointer Table and the Entry Address Tables.

Schedule Table The schedule table has seven entries, one for each rate group. Each entry consists of two words. There is one for one correspondence between the words in the Entry Address Table, and the bits in the first word of the schedule table for each rate group. The most significant bit in the first word in the schedule table is associated with the last entry in the entry address table for the corresponding rate group. If the bit is a "1" the program with the corresponding entry address is scheduled, and if "0" not scheduled. The second word in each Schedule Table entry is the value that the first word had on the previous iteration through the same rate group program.

Temporary Storage Schedule Table During execution of the programs in a rate group the data within the schedule table is shifted. The Temporary Storage Schedule Table is used to store the shifted results.

Rate Group Status Table The rate group status table contains one word for each rate group. Each word contains the execution status of its rate group in the following manner:

Status Word Negative	- rate group has not yet been run
Status Word Zero	- rate group has been completed
Status Word + 1	- rate group is being run and is uninterrupted
Status Word + 2	- rate group is being run and is interrupted

Register Storage Tables The Executive program has an associate Executive interrupt routine which establishes the timing of all programs based upon interrupts derived from a clock signal. The interrupt routine must have available a storage area for each rate group in which to store all computer registers when an interrupt occurs. The status word for each rate group will be stored in the Register Storage Table for that rate group, therefore the Rate Group Status Table is a part of the Register Storage Table.

There are two programs used to perform the executive function, an interrupt routine which is executed every time a clock interrupt occurs, and the main executive program. A transfer to the main executive program occurs at the completion of each scheduled computer program and from the interrupt routine. The interrupt routine is shown in detail in Figure 5-146. Upon entering the interrupt program the interrupt system is temporarily disabled. The point at which the interrupt system is reenabled is dependent upon the relationship between this and other computer interrupts. All of the compute registers are then stored in main memory. The area in main memory used to store the computer registers is dependent upon the rate group that was being executed when the interrupt occurred. The main executive routine stores the beginning address of the memory area to be used in a location available to the interrupt routine (this location is given the designation IRRGA in the example). In the example shown in Figure 5-146, the registers are stored by first temporarily storing the contents of index register 1. Index register 1 is then loaded with the main memory storage area location. The accumulator (A) and lower accumulator (B) is then stored into the first two locations of the main memory area by a double length store command. The temporarily stored contents of index register 1 are then loaded into the accumulator and stored into the main memory area. Each additional index register is then transferred to the accumulator and from the accumulator to the main memory storage area. Seven index registers are assumed in the example. The status word for the interrupted rate group is then set to a value of plus two. In the example the main memory storage area for each rate group is ten words and contains the following stored data.

1. accumulator
2. lower accumulator
3. index register 1
4. index register 2
5. index register 3
6. index register 4
7. index register 5
8. index register 6
9. index register 7
10. status word

The fastest rate group is 64/sec and the interrupt occurs at 64 times per second. Thus all of the 64/sec programs that are scheduled must be executed each time the interrupt occurs. The main executive program causes all the programs in a rate group to be exeucted once if the status word is negative. Upon completing the execution of all the programs in a rate group the main executive program stores a zero in the status word for that rate group. The 64/sec rate group status word is first tested for zero. If the status word is not zero, a computer timing overload has occurred and appropriate action such as issuing error messages, descheduling now critical tasks, etc., must be taken. If the status word is zero, indicating normal program operation, it is loaded with a minus value causing the 64/sec rate group to be rerun. The main program timer is then incremented by 1. A typical binary counter incremented by 1 every 64th of a second and the rate groups that should be scheduled are

0000001	64
0000010	64,32
0000011	64
0000100	64,32,16
0000101	64
etc	

From this example it can be seen that the 64/sec rate group is scheduled every time, the 32/sec rate group is scheduled whenever the least significant bit is zero, the 16/sec rate group whenever the last two least significant bits are zero, etc. The example also shows that whenever a rate group is scheduled all other faster iteration rate groups are also scheduled. The timer is used to determine which rate groups are scheduled by executing a series of short right cycle shifts and testing the accumulator sign bit after each shift. Each rate group is rescheduled by testing its status word to determine if a computer timing overload has occurred and then loading the status word with a negative number.

The time and storage required by the executive interrupt routine is dependent upon the number of registers to be stored, the number of rate groups to be run, and computer instruction repertoire. The example program requires the execution of 2201 instructions each second and 58 memory locations for instruction storage not including the error routines which are executed in the event of a computer timing overload.

The main executive program determines the order in which all scheduled programs are run. The fastest rate group programs are always first. Each scheduled program returns to the executive upon completion of its tasks. Figure 5-147 shows example coding of a main executive program. Upon entering the program, the interrupt system is disabled. The 64/sec rate group status word (SW64) is then tested with one of four possible branches taken dependent upon the value of SW64 being negative, zero, plus one, or plus two. If the value is zero the 32/sec rate group status word (SW32) is tested in the same manner. This process of testing status words continues until a non-zero status word is encountered. The status words are tested in the order of fastest to slowest rate group. If all status words are zero the example program contains programming to place the computer in a wait mode. This programming, shown with an entry address of

```
EXEC   DIS
       LDA  SW64
       TRZ  TST32
       TRN  NB64          TEST 64/SEC
       SUB  ONE           STATUS
       TRZ  OB64
       TRA  TB64

TST32  LDA  SW32
       TRZ  TST16
       TRN  NB32          TEST 32/SEC
       SUB  ONE           STATUS
       TRZ  OB32
       TRA  TB32
         :
TST01  LDA  SW01
       TRZ  TST00
       TRN  NB01          TEST 1/SEC
       SUB  ONE           STATUS
       TRZ  OB01
       TRA  TB01

TST00  ENI                WAIT
       TRA  TST00+1

NB32   LAB  ST32
       STA  ST32+1        INITIALIZE
       LDX  AFAT32,2      RATE GROUP
       TRA  LE32

OB32   LDX  ATP32,2       PICKUP SCHEDULE WORD AND
       LAB  STP32         ADDRESS TABLE POINTER
LP32   DEC  X2            TEST FOR NEXT
       LLS  1             SCHEDULED PROGRAM
LE32   TRP  LP32
       STX  ATP32,2       STORE SCHEDULE WORD AND
       SAB  STP32         ADDRESS TABLE POINTER
       LDA  MA32          SET REGISTER STORAGE AREA
       STA  IRRGA
       LDA  ONE           SET STATUS WORD
       STA  SW32
       ENI
       TRA* 0,2

END32  LDA  ZERO          FINAL RATE GROUP
       STA  SW32          PROGRAM
       TRA  EXEC

TB32   LDA  MA32          SET REGISTER
       STA  IRRGA         STORAGE AREA
       LDA  ONE           SET STATUS WORD
       STA  SW32
       LAB  MS32
       LDX  MS32+2,1
       LDX  MS32+3,2      LOAD REGISTERS
         :
       LDX  MS32+8,7
       ENI
       TRA  0,1
```

Figure 5-147 Main Executive Program

TSTOO, first enables the interrupt system and then executes a transfer instruction, transferring to itself, which places the computer in a one word loop waiting for the next timing interrupt. In the actual flight program an additional rate group will probably be defined. This is a background rate group which is executed whenever all other rate group programs have been completed. If a background rate group exists the computer is never placed in a wait mode.

The example program of Figure 5-147 shows only that programming required by the 32/sec rate group when the status word is non-zero. The programming for the other rate groups is identical except for the data addresses. If the status word is negative then none of the programs in the rate group have yet been run. The new and old schedule table entries are loaded into the accumulator and lower accumulator with a double length load instruction. The new schedule value in the accumulator is then stored into the old value position in the schedule table to update the old value for the next time the rate group is to be run. Index register 2 is then loaded with the address of the last word in the entry address table for the rate group being processed. A transfer to the loop test for determining which program is to be run next is then executed.

If the status word is plus one, the rate group is in the process of being scheduled and has not been interrupted in the middle of executing a program. The address table pointer is loaded into index register 2. Note that the negative status word branch loaded the initial address table pointer value into index register 2. The partially processed schedule word and its old value is loaded into the accumulator and lower accumulator from the temporary storage schedule table by a double length load instruction. Index register 2 is then decremented by one and the combined accumulator/lower accumulator shifted left one bit position. The most significant bit of the accumulator is then tested. If the bit is a one, the associated program is scheduled and if zero, the associated program is not scheduled. If the bit is zero, a transfer back to the decrement index instruction is executed. A three instruction loop of decrement index,shift and test is thus performed until a one bit is encountered in the most significant bit of the accumulator. The program branch taken with a negative status word enters this loop at the test instruction after initializing the accumulator/lower accumulator and index register. Upon encountering a one in the accumulator most significant bit position, the index register is stored back into the address table pointer table and the accumulator and lower accumulator into the temporary storage table. The address of the area in which the interrupt routine stores the computer registers for this rate group is stored into the proper interrupt routine word. The status word for this rate group is then set to plus one, the interrupt system enabled , and a transfer to the beginning location of the scheduled program is executed. In the example this transfer is an indirect transfer on index register 2.

Each rate group has a special program which is always scheduled as the last program to be executed in the rate group. This program when executed stores a zero into the status word for that rate group signaling the executive program that the rate group is completed until it is recalled by the interrupt routine.

If the status word is plus two, the rate group was interrupted during its last execution. If this is the case, the computer register storage area is set for the proper rate group. The status word is set to plus one, the computer registers loaded from the register storage area, the interrupt system enabled, and a transfer to the interrupt return address executed. In the example it is assumed that the return address is left in index register number 1 when an interrupt occurs.

The loading and storage required by the executive program is dependent upon the total number of programs in the computer and the number of programs scheduled at any one time. The example program described above made no provision for the number of programs in any rate group being greater than the number of bits in a word. To provide for more programs per rate group than bits per word requires that the programs in each rate group be divided into subgroups with each subgroup containing the number of programs equal to the number of bits in a word. The last program executed in each subgroup reinitializes the address table pointer and temporary storage schedule table for the next subgroup. Defining the parameters T_{64}, T_{32} ..... and T_1 to be the total number of programs in each rate group and N_{64}, N_{32} ··· and N_1 the number of programs scheduled at a particular time in each rate group excluding the special end programs, then the number of short instructions executed per second by the executive is approximately

$$4385 + 1024N_{64} + 576N_{32} + 320N_{16} + 176N_8 + 96N_4 + 52N_2 + 29N_1$$
$$+ 192T_{64} + 96T_{32} + 48T_{16}\ 24T_8 + 12T_4 + 6T_2 + 3T_1$$

Combining the overall requirements of both the main executive program and the executive interrupt program yields

Execution Cycles per second

$$6586+1024N_{64}+576N_{32}+320N_{16}+176N_8+96N_4+52N_2+29N_1$$
$$+192T_{64}+96T_{32}+48T+24T_8+12T_4+6T_2+3T_1$$

Instruction Storage Area

341 + Overload error routines $\approx$ 550 locations

Constant Storage Area

25 + T

Variable Storage Area

108

5.7.4 Subroutines

The mathematical subroutines of sine, cosine, tangent, arcsine, arcosine, arctangent, exponential, logarithm and matrix multiply are required by the DMS programs. These functions are generated by polynominal approximation to least significant bit accuracy. For all programs except navigation and guidance the required accuracy is 16 binary digits; for navigation and guidance 32 binary digits are required. The computational requirements for each function are given below.

Sine and Cosine

The sine and cosine are generated by a double entry subroutine by applying the formula

$$COS(X) = SIN\left(\frac{\pi}{2} - X\right) \tag{1}$$

The input argument to the subroutine will be scaled in semicircles at binary zero which automatically restricts the range of the input argument in radian measure to be between $-\pi$ and $+\pi$. Upon entry to the cosine subroutine the argument is subtracted from $\frac{1}{2}$ (i.e., $\frac{\pi}{2}$ radians scaled in semicircles at binary zero) reducing the cosine generation to that of generating the sine function. In generating the sine functio[n] the input argument range is first restricted to positive arguments by use of the identity

$$SIN(-X) = -SIN(X) \tag{2}$$

The output argument is then generated by

$$\gamma = \begin{cases} a_3X^6 + a_2X^4 + a_1X^2 + a_0 & \text{if } 0 \leq X \leq \frac{\pi}{2} \\ b_3(X-\frac{\pi}{2})^6 + b_2(X-\frac{\pi}{2})^4 + b_1(X-\frac{\pi}{2})^2 + b_0 & \text{if } \frac{\pi}{2} < X \leq \pi \end{cases} \tag{3}$$

for 16 binary bit accuracy. For 32 bit accuracy an additional 2 terms must be added to each polynominal of equation 3. The average sine and cosine subroutine characteristics are

	Single Precision	Double Precision
Total Instructions	28	32
Short Constants	11	2
Long Constants	0	13
Short Variable	3	0
Long Variable	0	3
Fast Normal Path Instructions	26	28
Slow Normal Path Instructions	4	6

Tangent

Input arguments to the tangent subroutine will be restricted to lie between $-\pi$ and $+\pi$ radians. Within the subroutine an argument range reduction will be made by employing the identities

$$TAN(-X) = -TAN(X) \tag{4}$$

$$TAN\left(\frac{\pi}{2}+X\right) = -\frac{1}{TAN(X)} \tag{5}$$

$$TAN(2X) = \frac{2\,TAN(X)}{1 - TAN^2(X)} \tag{6}$$

Equation 4 is used to restrict the argument to between 0 and π radians, equation 5 to between 0 and $\frac{\pi}{2}$ radians, and equation 6 to between 0 and $\frac{\pi}{4}$ radians. The tangent is then developed from the formula

$$TAN(X) = X\left(a_4 X^8 + a_3 X^6 + a_2 X^4 + a_1 X^2 + a_0\right) \tag{7}$$

for16 bit accuracy.For 32 bit accuracy four more terms must be added to equation 7. The computer requirements for tangent become

	Single Precision	Double Precision
Total Instructions	51	59
Short Constants	8	0
Long Constants	0	12
Short Variable	3	0
Long Variable	0	3
Fast Normal Path Instructions	37	41
Slow Normal Path Instructions	8	12

Arctangent

The arctangent subroutine has two input arguments, X and Y. The subroutine generates TAN^{-1} (Y/X). Within the subroutine an argument Z is constructed from

$$Z = \begin{cases} X/Y & \text{if } |X|<|Y| \\ Y/X & \text{if } |Y|<|X| \end{cases} \qquad (8)$$

This limits the value of Z to the range $-1 \leq Z \leq 1$. The subroutine output is determined from

$$\tan^{-1}(Y/X) = \begin{cases} \pm \tan^{-1} Z & \text{if } |X|>|Y| \\ \pm\left(\frac{\pi}{2} - \tan^{-1} Z\right) & \text{if } |Y|>|X| \end{cases} \qquad (9)$$

with the appropriate sign chosen dependent upon the signs of X and Y. The value of $\tan^{-1} Z$ is evaluated from

$$\tan^{-1} Z \doteq Z \; \frac{a_2 Z^4 + a_1 Z^2 + a_0}{b_2 Z^4 + b_1 Z^2 + b_0} \qquad (10)$$

for 16 bit accuracy and

$$\tan^{-1} Z = Z \; \frac{a_4 Z^8 + a_3 Z^6 + a_2 Z^4 + a_1 Z^2 + a_0}{b_3 Z^6 + b_2 Z^4 + b_1 Z^2 + b_0} \qquad (11)$$

for 32 bit accuracy.

The computer requirements for the arctangent subroutine are .

	Single Precision	Double Precision
Total Instructions	46	52
Short Constants	7	0
Long Constants	0	10

	Single Precision	Double Precision
Short Variable	6	1
Long Variable	0	5
Fast Normal Path Instructions	31	34
Long Normal Path Instructions	8	11

Square Root

The square root subroutine consists of three parts, range reduction, initial estimate, and an accurate convergence. Upon entry to the subroutine the input argument, X, is tested for zero and if zero an immediate zero square root value is returned. If X is not zero its range is reduced by a normalizing process. A binary number scaled at binary zero having 2 n + K leading binary zeros can be represented by

$$X = X_0 \, 2^{-(2n+K)} \tag{12}$$

where $\frac{1}{2} \leq X_0 < 1$ and K is either 0 or 1.

The $\sqrt{X}$ is then given by

$$\sqrt{X} = \sqrt{X_0 2^{-K}} \cdot 2^{-n} \tag{13}$$

The normalization process determines the value of n and

$$X_1 = X_0 2^{-K} \tag{14}$$

An initial estimate of the value of $\sqrt{X_1}$ is obtained from

$$Y_n = a_2 X_1^2 + a_1 X_1 + a_0 \tag{15}$$

This initial estimate is improved to 16 bit accuracy by performing a single Newton - Raphson iteration of the form

$$Y_{n+1} = \frac{1}{2}\left(\frac{X_1}{Y_n} + Y_n\right) \tag{16}$$

By applying a second Newton - Rophson iteration, 32 bit accuracy is obtained. The final result is obtained by shifting y right n places. The computer requirements for the square root are:

	Single Precision	Double Precision
Total Instructions	29	33
Short Constants	4	4
Long Constants	0	0
Short Variables	3	1
Long Variables	0	2
Fast Normal Path Instructions	22	25
Slow Normal Path Instructions	4	5

Arcsine

The arcsine subroutine applies the formula

$$SIN^{-1}(X) = TAN^{-1}\left(\frac{X}{\sqrt{1-X^2}}\right) \tag{17}$$

which is solved by using both the square root and arctangent subroutines. The total requirements of the arcsine subroutine are then

	Single Precision	Double Precision
Total Instructions	11	11
Short Constants	1	0
Long Constants	0	1
Short Variables	1	0
Long Variables	0	1
Fast Normal Path Instructions	63	69
Slow Normal Path Instructions	13	17

The instructions required in executing the arctangent and square root subroutines have been added to the Normal Path instructions.

Exponential

The exponential subroutine first restricts the input argument to the range 0 to 1 by use of the formula

$$e^{-x} = \frac{1}{e^x} \tag{18}$$

The exponential function to 16 bit accuracy is determined from

$$e^x = a_4 x^4 + a_3 x^3 + a_2 x^2 + a_1 x + a_0 \tag{19}$$

For 32 bit accuracy three more terms must be added. The computer requirements for the exponential function are

	Single Precision	Double Precision
Total Instructions	18	24
Short Constants	5	0
Long Constants	0	8
Short Variables	2	1
Long Variables	0	1
Fast Normal Path Instructions	14	17
Slow Normal Path Instructions	4	7

Logarithm

The logarithm subroutine generates an output y for the input argument X where

$$Y = \log_e X \tag{20}$$

For an output to exist X must be greater than 0. For a 16 bit word the smallest possible value of X which is greater than 0 is 2^{-15} and for a 32 bit word it is 2^{-31}. Thus if $2^{-15} \leq X < 1$ then $-10.40 < Y < 0$ and if $2^{-31} \leq X < 1$ then $-21.49 < Y < 0$. The 16 bit accuracy output will be scaled at binary 4 and the 32 bit output at binary 5.

Representing X as

$$X = X_o \cdot 2^{-n} \quad \text{where } \tfrac{1}{2} \leq X_o < 1 \tag{21}$$

allows y to be written as

$$Y = \log_e(X) = \log_e(X_o \cdot 2^{-n}) = \log_e(X_o) - n\log_e(2) \tag{22}$$

The range of Xo is further restricted by defining

$$X_1 = \begin{cases} X_o\sqrt{2} & \text{if } \tfrac{1}{2} \leq X_o < \tfrac{1}{\sqrt{2}} \\ X_o & \text{if } \tfrac{1}{\sqrt{2}} \leq X_o < 1 \end{cases} \tag{23}$$

making

$$\log_e X_o = \begin{cases} \log_e X_1 - \log_e\sqrt{2} & \text{if } \tfrac{1}{2} \leq X_o < \tfrac{1}{\sqrt{2}} \\ \log_e X_1 & \text{if } \tfrac{1}{\sqrt{2}} \leq X_o < 1 \end{cases} \tag{24}$$

an intermediate value is formed from

$$Z = \frac{X_1 - 1}{X_1 + 1} \tag{25}$$

For 16 bit accuracy

$$\log_e X_1 = a_1 Z^2 + a_o \tag{26}$$

and for 32 bit accuracy

$$\log_e X_1 = a_3 Z^6 + a_2 Z^4 + a_1 Z^2 + a_o \tag{27}$$

The computer requirements of the logarithm subroutine is then

	Single Precision	Double Precision
Total Instructions	37	41
Short Constants	7	1
Long Constants	0	8
Short Variables	3	1
Long Variables	0	2
Fast Normal Path Instructions	28	30
Slow Normal Path Instructions	6	8

<u>Matrix Multiply</u>

The matrix multiply subroutine is used only by the 32 bit accuracy programs. The subroutine can be written either for rapid execution or for minimum storage requirements. Because the routine is used by a 64 per second rate group program, the sizing estimates are based upon a minimum execution time program. These estimates are:

	Double Precision
Total Instructions	102
Short Constants	0
Long Constants	0
Short Variables	0
Long Variables	0
Fast Normal Path Instructions	75
Slow Normal Path Instructions	27

Summary

A summary of the memory storage requirements of each subroutine package is:

	Single Precision	Double Precision
Total Instructions	220	54
Short Constants	43	7
Long Constants	0	52
Short Variables	21	4
Long Variables	0	17

A summary of the required execution times is given in Figure 5-148

	SINGLE PRECISION			DOUBLE PRECISION	
	FAST	SLOW		FAST	SLOW
SIN & COS	26	4		28	6
TAN	37	8		41	12
SIN^{-1} & COS^{-1}	63	13		69	17
TAN^{-1}	31	8		34	11
EXP	14	4		17	7
LOG	28	6		30	8
$\sqrt{\ }$	22	4		25	5
MATRIX MULTIPLY				75	27

Figure 5-148 Subroutine Execution Times

6.0 DATA MANAGEMENT SYSTEM EVALUATION TOOLS

This section presents details of two techniques used to evaluate the data management system. The first tool is the Evaluation of Shuttle Computational and Processing Events (ESCAPE) program. The significant results of this program were presented in Volume I. This section gives information of value to a user of the program. Included are program variable descriptions, card deck characteristics and instructions for initializing and running the program. Program flow charts, listing, and printouts of sample runs are provided.

The second evaluation tool uses standard reliability techniques to compare the three DMS computer configurations basic to this study.

6.1 EVALUATION OF SHUTTLE COMPUTATIONAL AND PROCESSING EVENTS (ESCAPE) PROGRAM

6.1.1 Program Objectives

The aims of the simulation program described below may be summarized as follows:

- To generate processor and data network loading time lines
- To simplify the task of estimating the effects on processor and I/O loading of changes in program modules and I/O characteristics
- To allow an easy comparison of configurations of the DMS with different numbers of processors
- To permit easy collection of data on processor and I/O loading with various program module arrangements within each processor configuration
- To determine the processor capabilities necessary to handle the loads for various configurations
- To determine the I/O network capacity to handle the loads required f(each configuration
- To determine the DMS system required to handle different program implementations.

These objectives will be discussed below.

As the system evolves, new program modules and data points will appear and others will disappear. To keep up to date on the performance required to meet these needs, the changes are inserted into the program initialization deck and a re-estimation of processor capabilities and network loadings is made. There may also be some further system experimentation.

Each of the configurations can be tested to determine how much processor and how capable a data transfer scheme are required to handle the tasks imposed upon them by the mission. This will be done for a given program and sensor implementation.

Within a given configuration there is another degree of freedom - the arrangement of programs. Program modules may be moved around in a attempt to achieve minimal CPU and network loadings. The current implementation of the program requires that the analyst change a few processor assignment data cards to effect this change.

Each of these steps assumed a fixed processor capability, which is specified in the program as the number of processor cycles required for the fast and slow instructions, program parameters which have been developed elsewhere in this document, and which include instructions executed in math routines. These processor characteristics may also be changed to reflect advances in technology or to permit the evaluation of different processors.

The utilization of the tool may follow some such scheme. A program and data description are placed on data cards. For this realization a processor configuration is chosen and the simulations of several program arrangements are made. This step may be done in one computer run by stacking varous arrangement decks. A new processor configuration may be chosen and the iteration repeated. The object is to give the analyst enough data to look at to make a good decision concerning the number of processors and the type of data transfer scheme - data bus, hardwire etc - most appropriate. The total process or some part of it may be repeated for various program and data point changes and for various processor instruction characteristics.

The intention is that the program remain a tool throughout the period of development of the computational algorithms and the vehicle configuration.

It should be mentioned that the program may be described as a static simulation of the system. The term static implies that the dynamic aspects of system operation such as memory access conflicts and data flow have been replaced by constant descriptors which may represent average, worst-case, etc estimates. The dynamic aspects of system performance only become visible after further decisions have been made in the system realization, such as processor instructions and channel characteristics, data transfer network and so on.

In the listing included here a worst-case assumption has been made concerning memory conflicts - each I/O transfer represents a cycle steal from the processor.

The program has been written in Fortran V, an enhancement of Fortran IV available on the Univac 1108.

6.1.2 Program Description

The structure of the ESCAPE program has been designed to achieve the objectives stated above. The program has been designed as a series of concentric loops. The outer loop is the configuration loop. The user specifies a program arrangement within a set of processors (from two to six processors are currently available) and a simulation is run. The user may then re-arrange the program modules via input cards in the original run deck and re-execute the simulation. We see that the outer loop consists of the configuration changes (changes in the number of processors) and the inner loop consists of the n program rearrangements ($m \leq 99$) within a given processor configuration. All of these changes are effected through the run deck.

The innermost loop contains the simulation portion of the program. This section is event oriented, with the program module start and stop times forming the event list items. At each time on the event list the processors' loadings are calculated by summing the fast and slow instructions executed by each active program module and the I/O transfer rates of these programs. The processors' loadings are expressed in number of processor busy cycles, including memory conflicts, and the inter-processor I/O rates.

The outputs include the processor and program summaries at each event time and the processor summary table and maxima, which are printed after all events have occurred. The latter two summaries are printed once for each program arrangement.

6.1.3 Initialization Section

The initialization section is used for three purposes: 1. First pass through the program. 2. For each program re-arrangement. 3. For each new processor configuration.

1. First pass

- The first card read specifies the number of configurations which are to be simulated in the run, the number of program arrangements for each of these processor configurations and the number of processor cycles for the fast and slow instructions. The data on this card determines the number of passes through the initialization section since a new initialization is necessary for each configuration and each re-arrangement. The last two fields on the card indicate the processor characteristics to be simulated throughout the run. This card is unique to each run and is read on first pass only.

- The REPORT card(s) indicate whether or not the program report is to be printed and, if so, which programs are to be reported on. This card(s) appears for each new configuration, allowing the user to vary his selection during the run.

- The processor assignment cards appear next. First there is a processor number card with the format given in section 6.1.5 This card is followed by one or more arrangement cards listing the program modules assigned to processor 1. Then comes the processor 2 card and its arrangement cards, etc.

- The program module data cards come next. These cards are read once per simulation run and contain data which generally does not change very frequently - is invariant for a number of runs. The fields contain the four-letter program mnemonic (listed elsewhere), the number of slow instructions executed per program cycle, the number of fast instructions per program cycle, the program frequency in cps, the start and stop times of the module and the number of data cells - number of cells not accounted for by fields two and three. Note that the numbers in fields two and three may be the average number of instructions executed per pass, worst-case estimates, etc. The data cell number should be adjusted to account for program sizing.

- The I/O data cards are read last. These cards are like the program data cards in that they are read just once per simulation and should be relatively slowly varying for a vehicle configuration. The fields include a six letter mnemonic for the data point, the data frequency in cps, the data word length in bits, the data source module and the destination module(s). The latter mnemonics are the four-letter program module mnemonics.

The program module and I/O data cards will form a rather sizable deck - a thousand cards or so. At a later date, when the system is more accurately known, this data may be put in the program in DATA statements to obviate the execution of Fortran READ statements. This scheme will be particularly useful when executing the program using an object deck.

2. <u>Program rearrangement</u>

For each configuration there may be one or more program arrangements used. To simulate this situation cards similar to the processor arrangement cards are read. The difference is that only those programs being moved need be listed after their new processor. If the rearrangement only involves transfers to one processor then only that processor's cards need be included. In any case these cards are read at the time of a new simulation i.e. the input deck is not read together but rather in batches, after the events list is exhausted and as directed by the configuration card.

3. <u>Configuration Change</u>

For each configuration specified by field one of the configuration card there is a report card(s) and a complete set of processor arrangement cards. A complete set means that each program module in the program data deck should be listed under one (and only one) processor card.

Another function of the initialization is the establishment of the events list TEVENT(I). The program module switch-on and switch-off times are placed, non-redundantly, on the events list in increasing time order. This task must be performed for each program arrangement simulated since the same set of events occur for each arrangement and in the execution of the program the events list gets wiped out.

There are at present several limitations on system initialization which should be mentioned. These limitations were made to conserve space in the program and can be changed without major programming effort.

One limitation is on the number of destination modules of a piece of data. At present the maximum number of destination modules is nine, which represents the number of available fields on the I/O data cards. As the system determination evolves this limitation will be tested.

Another limitation on the system is the number of processors which are permitted. The arrays PROCS and PROCSM have been set up to allow space for six processors. This constraint can be changed if there is a need to do so.

Those arrays whose dimensions have been set using PARAMETER statements should be noted. The use of parameters was to minimize re-punching which will be necessitated when the number of programs, number of data points, number of events etc. are better known. As the system evolves these parameters will probably change. The description of the arrays in Figure 6-1 indicates the interpretation of these parameters.

Another important point is the treatment of the external data sources. To take advantage of the symmetry between the transfer of data between program modules and between program modules and external devices the following convention has been established - one (or more) program modules representing the external data sources are assigned to an additional processor. These modules do not execute instructions, all they do is transfer data. In the examples in the output section it is seen that all I/O data from external devices comes from (or goes to) a single module called EXTF. This module has been assigned to the last (pseudo) processor and is scheduled at time zero and runs to the end of the simulation since data from the external devices is generally present when a program module switches on. This convention means that a one central processor system will be simulated using two processors etc.

It is worth noting that this convention concerning external devices allows the generation of data concerning specific data types by assigning to them their own module names and switching times. Normally, however, the external data sources are treated as being lumped in a single module which operates during the whole mission.

Arrays

IODATA(I) - A linear array holding I/O data

Cell 1 - Pointer to next data set

Cell 2 - Six letter data point mnemonic e.g. ABCDEF

Cell 3 - Frequency of data - transfers/sec.

Cell 4 - Data length per transfer, in bits

Cell 5 - Four letter origin program module mnemonic e.g. ABCD

Cell 6 - Four letter destination program module mnemonic e.g. EFGH

. K destination modules

. .

. .

. .

Cell 6+K - Pointer to next data point

etc.

ITEM(I) - Linear array used for temporary storage

ITEMP(I) - Linear array used for temporary storage
Equivalenced to TEMP(I).

NOPROC(I) - Contains a alphabetic form of processor numbers i.e
NOPROC(1)= ONE, NOPROC(2)= TWO, etc.

NP(I) - Linear array used to contain status indicator for programs
involved in I/O data transfers.

NRRNG(I) - Number of arrangements for each configuration. NRRNG(1)
contains number of program arrangements for processor
configuration 1, etc.

OPTP(I,J) - A two dimensional array. Each row corresponds to a
processor. The columns are as follows:

Column 1 - Processor number

Column 2 - Maximum input rate for that processor

Column 3 - Time of occurrence of maximum

Column 4 - Maximum output rate for that processor

Column 5 - Time of occurrence

Column 6 - Maximum number of busy cycles

Column 7 - Time of occurrence

TEM(I) - Array used for temporary storage
Equivalenced to PROCSM.

TEMP(I) - Array used for temporary storage. Equivalenced to ITEMP(I)

TEVENT(I) - Array used for event times.
TEVENT(1) < TEVENT(2) < -----

Figure 6-1 ESCAPE Program Variables

PROCS(I,J) - Two dimensional array containing processor information. Each row corresponds to a different processor. The columns are as follows:

Column 1 - Processor number
Column 2 - Total number of slow instructions executed
Column 3 - Total number of fast instruction executed
Column 4 - Number of bits per second input from processor 1
Column 5 - Number of bits per second output to processor 1
Column 6 -15 - Similar to 4 and 5 for processors 2 through 6
Column 16 - Number of I/O memory cycle steals
Column 17 - Total number of processor busy cycles
Column 18 - Number of cells occupied

PROCSM(I,J,K) - Three dimensional array. First index corresponds to processor. Second index corresponds to the event times. The columns are:

Column 1 - Time
Column 2 - Number of busy cycles for processor I
Column 3-14 - Correspond to columns 4-15 of PROCS(I,J)

PRODAT(I,J) - Two dimensional array containing program module information. Each row corresponds to a program module. Each column is as follows:

Column 1 - Four letter program module mnemonic e.g., ABCD, left justified
Column 2 - Activity flag. 1.= ON, 0.= OFF
Column 3 - Processor number that program is assigned to.
Column 4 - Number of slow instructions executed
Column 5 - Number of fast instructions executed
Column 6 - Frequency of Execution
Column 7 - Program start time
Column 8 - Program stop time
Column 9 - Number of data cells of program

PROGS(I,J) - Two dimensional array containing information for the optional program reports.

Column 1 - Four letter program module mnemonic, left justified
Column 2 - Processor number program is assigned to
Column 3 - Number of processor cycles used for instructions
Column 4 - Number of processor cycles used for I/O transfers
Column 5 - Sum of 3 and 4
Column 6 - Output bitrate
Column 7 - Input bitrate

Figure 6-1 (Continued)

Important Program Scalar Variables

ARRANG	Current arrangement number
CONFIG	Current configuration number
FASTCY	Number of processor cycles for a fast instruction. If this field is blank on configuration card the hardwired value is used.
NARRNG	Number of program arrangements for current configuration
NCONFG	Number of processor configurations to be simulated – an input parameter.
NDATAP	Number of data points in input card deck.
NLIST	Number of event times currently in TEVENT(I)
NOPT	Number of programs to be reported on.
NPROC	Number of processors in current configuration.
NPROG	Number of programs in input deck.
NTIME	Current index setting for second dimension of PROCSM (I,J,K).
SLOWCY	Number of processor cycles for a slow instruction (see FASTCY).
T	Current simulation time. This is successively set to each item in TEVENT(I).

Program Parameters

KOPT	Number of rows in PROGS - maximum number of program options.
KPROC	Number of rows in PROCS.
KPROG	Number of rows in PRODAT - maximum number of programs allowed in input deck.
NDATA	Length of array IODATA. This constrains number of data points allowed in input deck.
NEVENT	Length of array TEVENT(I) - maximum number of events allowed.
NPCCOL	Number of columns in PROCS.
NPCOL	Number of columns in PRODAT
NSUM	Size of the second dimension of PROCSM. This should be equal to NEVENT, since for each event a row of PROCSM is written
NSMCOL	Size of third dimension of PROCSM.
OPTC	Number of columns in PROGS.

Figure 6-1 (Continued)

6.1.4 Execution Section

The first task performed by the program is calculating and printing the number of cells occupied for each processor and the listing of the program modules stored in each processor memory. This summary is illustrated in the section on output formats.

The flight-time is set to the top time on the events list and the list is pushed-up one storage location.

The program list is scanned to determine the number of fast and slow instructions executed per processor at the flight time being simulated. This requires including only those programs which are currently active. The array NP(I) is used for temporary storage of the program statuses.

Next the I/O data points are scanned. The object is to determine the I/O transfers between each pair of processors. To achieve this determination, it must be checked that the origin program module is currently active and that the destination module(s) are also active. Also, redundancies must be filtered out. These redundancies include the case in which origin and destination modules are in the same processor or two or more destination modules are in the same processor. These data transfers are not included in the processor summaries since they do not add to the I/O network loading and the processor channel loading.

Next the processor busy cycles are calculated, assuming that each I/O transfer involves a cycle steal by the I/O unit. This assumption is a worst case one and could easily be modified using a (constant) conflict ratio as a multiplier. A more adequate simulation requires the inclusion of more detailed program descriptions and the use of dynamic simulation models.

The arrays PROCS and PROCSM are used for storing the processor information. PROCS is written over at each time on the events list TEVENT, but PROCSM is used to store data throughout the time-of-flight of a specific program arrangement.

The next phase of the program is collecting data for the optional report. Each program module to be reported on uses a row in the array PROGS for storing its current operational data. The calculation of a program's data rates depends on whether or not the programs with which it communicates are operational at each specific time point. The coding in this section is primarily concerned with this factor.

The last operational section of the program which lies within the time-step loop is determining whether new processor maxima have been achieved and, if so, updating the array OPTP(I,J) which stores this data. This information is used for a summary at the end of the simulation of an arrangement.

At this point the processor and program data is printed out. Examples of these formats are given in the output section of this description. In the processor summary are included the input and output bitrates, in bits per second, for each processor and the total number of processor cycles used. The program summary includes input and output bitrates, and number of processor cycles used and the percentage of the processor load which this program accounts for.

Currently, these reports are generated at each event time. However, a minor change in coding would allow each of them to be printed on a switched or time basis.

When the events list is exhausted i.e. the largest time on the program data cards has been reached, several more reports are printed out. One report is a time-tabular summary of the processor loading and the other report indicate the maximum bitrates and busy cycles during the run and the times at which these maxima occurred.

The program branches back to the initialization section at this point if either another program arrangement has been scheduled or another processor configuration has been scheduled. Various arrays are initialized to zero before branching back to initialization. For a new program arrangement only the program assignment change cards need be read. For a new configuration, the report card(s) and a totally new set of program assignment cards are read. There is currently no provision for changing program or I/O data characteristics after the original initialization, since the object of the program is to evaluate processor and I/O loads for a specific set of program modules and data points.

It should be noted that the executive load will depend on the number of processors, so this data should be added into the numbers generated by the simulation program.

6.1.5 Input Deck Description

The types of cards used by the program are: (1) Configuration Card (2) Report Selection Cards (3) Processor Number Cards (4) Program Arrangement Cards (5) Program Module Data Cards (6) Input/Output Data Cards.

Of these cards, the latter two categories constitute a description of the system which will be changing relatively slowly - the programs used and the data points - so that only the first four categories will be of primary concern in generating a run deck.

It should be noted that in changing the number of cards in (5) or (6) a check should be made on the dimensions of various arrays in the program, in particular the arrays IODATA(I) and PRODAT(I,J), to make sure that the limits of these arrays have not beenoverrun.

Configuration Card

This card contains the number of configurations to be simulated (up to a maximum of ten), the number of program arrangements for each of these configurations, the number of processor cycles used by a fast instruction and the number of processor cycles used by a slow instruction.

The format is:

Column	Format	Contents
1-2	XX	Number of configurations (right-justified)
4-5	XX	Number of arrangements for configuration 1
7-8	XX	Same, for configuration 2
⋮	⋮	⋮
31-32	XX	Same, for configuration 10
35-40	XXX.XX	Number of processor cycles for a fast instruction
43-48	XXX.XX	Number of processor cycles for a slow instruction

Report Cards

The first field of the first card indicates whether or not the optional program report is desired and, if it is desired, the names of the program modules to be reported on.

The format is:

Column	Format	Contents
1-4	YES or NO (starting in column 1)	Whether report is desired or not
If field one is YES		
6-9	ABCD	Program module mnemonic
11-14	EFGH	Same
.	.	.
.	.	.
.	.	.

If more than 15 programs are to be reported on card two and others have the same format, with the first mnemonic starting in column 1. The second and following cards have 16 program fields rather than 15. The first pair of blank columns terminates processing of a card.

Processor Number Cards

This card contains the mnemonic PROC and the processor number field. These cards should be in serial order and each card must be followed by the program arrangement cards for that processor.

Column	Format	Contents
1-4	PROC	Designates a processor card
7-10	XXX.	Processor number. Use floating point integer e.g. 1., 2., 3. etc.

Program Arrangement Cards

Contain the mnemonics for the program modules assigned to the last-encountered processor card. The fields are four letter program mnemonics followed by a comma if there are more module names. When a blank is encountered after a comma the processing of that card terminates and the next card is read. A blank after a module name terminates reading of these cards. The following card must either be another PROC card or an END card (starting in column 1), designating the end of the processor arrangement cards.

Column	Format	Contents
1-5	ABCD(,) (comma indicates more programs to come)	Four letter program module mnemonic
6-10	EFGH(,)	Same
.	.	.
.	.	.
.	.	.

Program Module Cards

These cards contain the program module mnemonic, the number of slow instructions executed by the program, the number of fast instructions executed by the program, the frequency of execution in cps, the program module start-up time, the program module stop time and the number of data cells occupied. The last card in the set is an END card. END begins in column 1.

Column	Format	Contents
1-4	ABCD (left justified)	Program module mnemonic
6-16	XXXXX.XXX	Number of slow instructions executed per cycle
18-28	Same	Numberof fast instructions per cycle
30-40	Same	Number of program executions per second
42-52	Same	Start time of program module
54-64	Same	Stop time of program module
66-76	Same	Number of data cells occupied by program

I/O Data Cards

These cards contain the six letter data point mnemonic, the data frequency in transfers per second, the number of bits per transfer, the source program mnemonic and the destination program mnemonics. The program labels are all four letters, as seen elsewhere. The last card in the set is an END card. END begins in column 1.

Column	Format	Contents
1-6	ABCDEF	Six letter data point mnemonic
9-19	XXX XXX.XXXX	Frequency of data transfer
22-27	XXXXX.	Number of bits per transfer
30-33	ABCD	Four letter origin program module mnemonic
36-39	EFGH	Four letter destination modules mnemonic
41-44	IJKL	Same
.	.	.
.	.	.
.	.	.

The first blank field of four characters terminates the card. Fields separated by a single space.

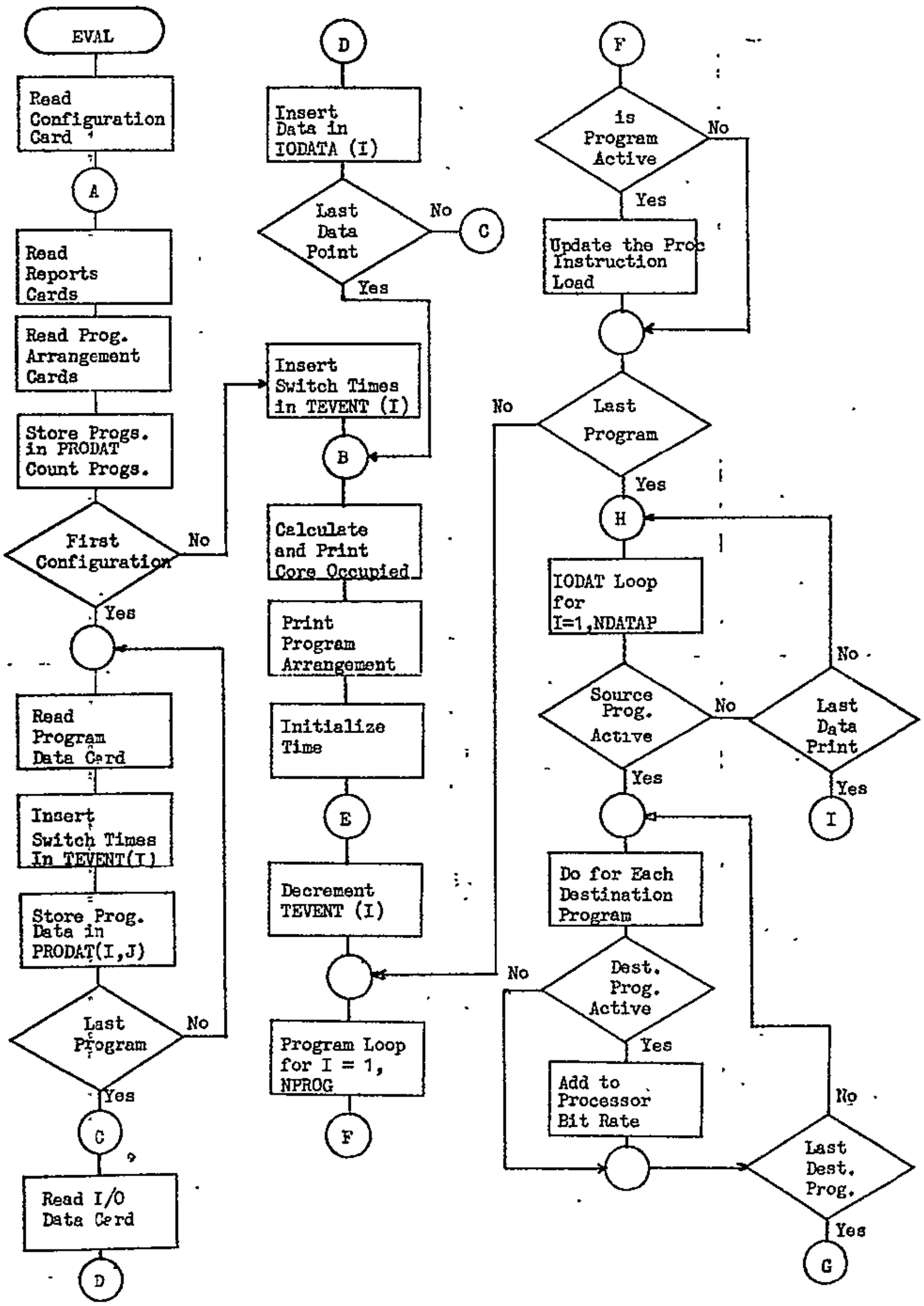


Figure 6-2 Program ESCAPE

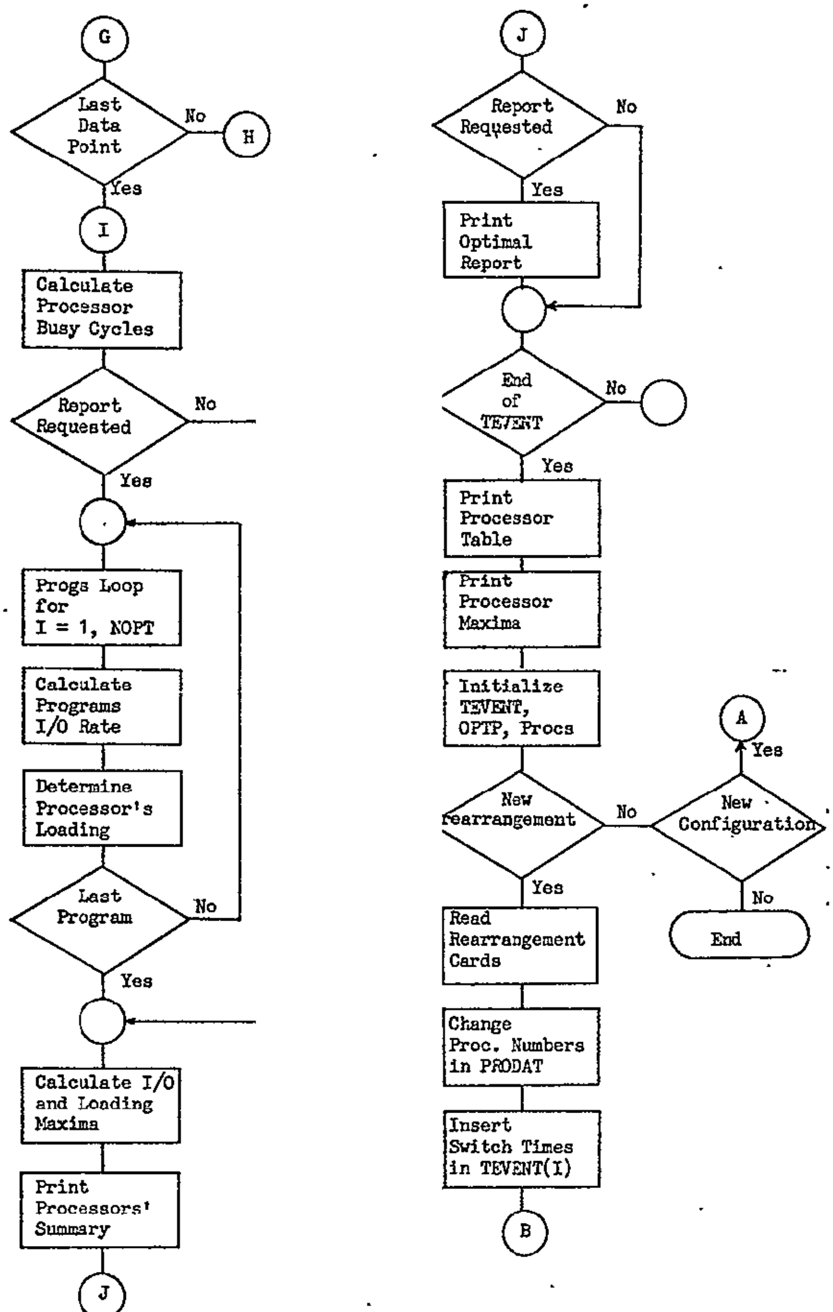


Figure 6-2 (Continued)

```
      INTEGER REPORT ,BLANK,END,YES
      INTEGER  CONFIG, ARRANG
      REAL IODATA,NOPROC,NP
      PARAMETER  CARD=5 , PRINT =6
      PARAMETER  KPROG =  20 , NPCOL = 9 , NDATA = 100,     KPROC =10,
     *  NPCCOL =18,    KOPT = 10 , NEVENT = 20 , OPTC = 7 ,
     * LPROG = KPROG* NPCOL ,   LPROC = KPROC* NPCCOL , LOPT = KOPT*OPTC
      PARAMETER NSUM=30, NSMCOL=14
      DIMENSION  ITEM(10)
      DIMENSION  PROCSM(6,NSUM,NSMCOL)
      DIMENSION  TEM(2)
      EQUIVALENCE  (TEM(1), PROCSM(1,1,1))
      DIMENSION  TEMP(40), ITEMP(40)
      EQUIVALENCE (TEMP(1), ITEMP(1))
      DIMENSION  PRODAT(KPROG, NPCOL),   IODATA(NDATA) ,
     *   PROCS(KPROC, NPCCOL),PROGS(KOPT,OPTC),TEVENT( NEVENT)
      DIMENSION  OPTP(KPROC, 7) , NP(10) ,NOPROC(10)
      DIMENSION  NRRNG(10)
      DATA  CONFIG /0/, ARRANG/0/ , YES/3HYES/,NO/2HNO/
      DATA COMMA/1H, /,BLANK/6H       /,PROC/4HPROC/,END/3HEND/
      DATA   (NOPROC(I),I=1,10)/3HONE , 3HTWO , 5HTHREE ,  4HFOUR ,
     *  4HFIVE ,  3HSIX , 5HSEVEN , 5HEIGHT , 4HNINE  ,  3HTEN /
      DATA NPROG/0/, NPROC/0/, NOPT/0/
      DATA FASTCY/2./,SLOWCY/10./
      DATA NLIST/1/, NDATAP/0/
C     INITIALIZE  THESE  ARRAYS  TO  ZERO
      DATA ((PRODAT(L,M),L=1,KPROG),M=1,NPCOL)/LPROG*0./ ,
     * ((PROCS(L,M),L=1,KPROC),M=1,NPCCOL)/LPROC*0./,
     *  ((PROGS(L,M),L=1,KOPT),M=1,OPTC)/LOPT*0./ ,
     * (IODATA(L),L=1,NDATA)/NDATA*0./
      TEVENT(1) = 99999.
      NTIME =0
C     NUMBER   OF   COMPUTER   CONFIGURATIONS   SIMULATED
C     NUMBER   OF   PROGRAM   ARRANGEMENTS  FOR  THIS  CONFIGURATION
      READ(CARD, 9000) NCONFG,(NRRNG(K),K=1,10),TEMP(1),TEMP(2)
      WRITE(PRINT,9002)
      WRITE(PRINT, 9001)  NCONFG,(NRRNG(K),K=1,10), TEMP(1),TEMP(2)
      IF(TEMP(1)) 3,3
      FASTCY = TEMP(1)
    3 IF(TEMP(2)) 5,5
      SLOWCY = TEMP(2)
    5 CONTINUE
   10 CONFIG = CONFIG + 1
      NARRNG = NRRNG(CONFIG)
      READ(CARD,9020) REPORT,(ITEMP(K),K=1,15)
      WRITE(PRINT, 9021)  REPORT,(ITEMP(K),K=1,15)
      IF(REPORT .EQ. NO) GO TO 70
      NOPT =0
      DO 30 I=1,15
      IF(ITEMP(I).EQ.BLANK) GO TO 55
      NOPT = NOPT +1
      PROGS(NOPT,1) = TEMP(I)
   30 CONTINUE
   40 CONTINUE
      READ(CARD,9020) (ITEMP(K),K=1,16)
      WRITE(PRINT,9021) (ITEMP(K),K=1,16)
      DO 50I=1,16
      IF(ITEMP(I).EQ.BLANK) GO TO 55
C    NOPT MUST  BE  LESS  THAN  OR  EQUAL  TO  KOPT,
C   THE  DIMENSION  ON  PROGS
      NOPT = NOPT +1
      PROGS(NOPT,1) = TEMP(I)
```

3

Figure 6-3 Listing of Program ESCAPE

```
   50 CONTINUE
      GO TO 40
   55 CONTINUE
   70 ARRANG = ARRANG + 1
      IF(ARRANG .EQ. 1) GO TO 150
C     CHANGE   OF   PROGRAM   ARRANGEMENT
   80 READ(CARD , 9040 )  TEMP(1), PROCNO
      WRITE(PRINT,9041) TEMP(1),PROCNO
      IF(ITEMP(1) .EQ. END ) GO TO 130
   90 READ(CARD,9030) (TEMP(N),N=1,32)
      WRITE(PRINT,9031) (TEMP(N),N=1,32)
      DO 120 I=1,16
      IF(ITEMP(2*I-1) .NE. BLANK) GO TO 100
      IF(ITEMP(2*I-2) .EQ. BLANK) GO TO 80
      GO TO 90
  100 DO 110 K=1,NPROG
      IF(TEMP(2*I-1) -PRODAT(K,1)) 110,,110
      PRODAT(K,3) = PROCNO
      GO TO 115
  110 CONTINUE
  115 IF(ITEMP(2*I) .EQ. BLANK) GO TO 80
  120 CONTINUE
C   END  OF  REARRANGEMENT  CARDS -  GO  TO  START  OF  COMPUTATION
  130 NLIST = 1
      DO 145 I=1,NPROG
      DO 140 J=7,8
      IF(NLIST .GT.1) GO TO 131
      TEVENT(1)=PRODAT(I,J)
      GO TO 137
  131 CONTINUE
      DO 135 K=1,NLIST
C    NO   REDUNDANT   ITEMS   ON   LIST
      IF(PRODAT(I,J)-TEVENT(K)) ,140,135
      DO 132 L=K,NLIST
      M=L-K
  132 TEVENT(NLIST+1-M)= TEVENT(NLIST-M)
      TEVENT(K) = PRODAT(I,J)
      GO TO 137
  135 CONTINUE
      TEVENT(NLIST+1) = PRODAT(I,J)
  137 CONTINUE
      NLIST = NLIST + 1
  140 CONTINUE
  145 CONTINUE
      GO TO 350
  150 CONTINUE
      K=0
  155 READ(CARD, 9040) TEMP(1),PROCNO
      WRITE(PRINT, 9041)  TEMP(1),PROCNO
      IF(ITEMP(1) .EQ.END ) GO TO 230
      LL=PROCNO
      PROCS(LL,1) = PROCNO
      NPROC = 1+NPROC
  160 READ(CARD,9030) (TEMP(N),N=1,32)
      WRITE(PRINT,9031) (TEMP(N),N=1,32)
      DO 220  I=1,16
      IF(ITEMP(2*I-1) .NE. BLANK) GO TO 200
      IF(ITEMP(2*I-2) .EQ. BLANK) GO TO 155
      GO TO 160
  200 K=K+1
      IF(CONFIG .LE. 1)  GO TO 210
```

Figure 6-3 (Continued)

```
      DO 205 JJ=1,KPROG
      IF(TEMP(2*I-1) - PRODAT(JJ,1)) ,205,,205
      PRODAT(JJ,3)=PROCNO
      GO TO 215
  205 CONTINUE
  210 CONTINUE
      PRODAT(K,1) = TEMP(2*I-1)
      PRODAT(K,3) = PROCNO
  215 CONTINUE
      IF(ITEMP(2*I) .EQ. BLANK) GO TO 155
  220 CONTINUE
  230 CONTINUE
  240 DO 245 I=1,KPROG
      IF(PRODAT(I,1)) 245,,245
      NPROG = I-1
      GO TO 250
  245 CONTINUE
      NPROG = KPROG
  250 CONTINUE
      IF(CONFIG . GT. 1) GO TO 130
  255 CONTINUE
      READ(CARD,9050)  (TEMP(I),I=1,7)
      WRITE(PRINT, 9051)  (TEMP(I),I=1,7)
      IF(ITEMP(1) .EQ. END) GO TO 295
      DO 275  K=5,6
      IF(NLIST .GT. 1 ) GO TO 257
      TEVENT(1)=TEMP(K)
      GO TO 272
  257 CONTINUE
      DO 270  I=1,NLIST
C     NO   REDUNDANT  ITEMS  ON   LIST
      IF(TEMP(K)-TEVENT(I)) ,275,270
      DO 260 L=I,NLIST
      J = L - I
  260 TEVENT(NLIST+1-J) = TEVENT(NLIST -J)
      TEVENT(I) = TEMP(K)
      GO TO 272
  270 CONTINUE
      TEVENT(NLIST+1) = TEMP(K)
  272 CONTINUE
      NLIST =  NLIST + 1
  275 CONTINUE
      DO 290 I=1,NPROG
      IF(TEMP(1)-PRODAT(I,1)) 290,,290
      DO 280 J=2,7
  280 PRODAT(I,J+2) = TEMP(J)
  290 CONTINUE
      GO TO 255
  295 CONTINUE
      L=1
  300 READ(CARD,9060) (TEMP(I),I=1,13)
      WRITE(PRINT, 9061) (TEMP(I),I=1,13)
      IF(ITEMP(1) .EQ. END ) GO TO 350
      DO 320 I=5,13
      IF(ITEMP(I) .NE. BLANK) GO TO 320
      K=I
      GO TO 330
  320 CONTINUE
  330 CONTINUE
      NDATAP = NDATAP + 1
      IODATA(L)=K+1
```

Figure 6-3 (Continued)

```
      DO 340 I=1,K
      IODATA(L+I) = TEMP(I)
  340 CONTINUE
      L=L+K+1
      GO TO 300
  350 IF(REPORT .EQ. NO)  GO TO 380
      DO 370  I=1,NOPT
      DO 365  K=1,NPROG
      IF(PRODAT(K,1)-PROGS(I,1)) 365,,365
C     STORE  PROCESSOR  NUMBER  IN  PROGS  COLUMN 2
      PROGS(I,2) = PRODAT(K,3)
      GO TO 370
  365 CONTINUE
  370 CONTINUE
  380 DO 400 I=1,NPROG
      K =  PRODAT(I,3)
C     TABULATE  NUMBER  OF  CELLS  OCCUPIED   FOR  THIS  PROGRAM
C     ARRANGEMENT
      PROCS(K,18)=PROCS(K,18)+PRODAT(I,4)+PRODAT(I,5)+PRODAT(I,9)
  400 CONTINUE
      WRITE(PRINT,9065)
      DO 420 I=1,NPROC
      WRITE(PRINT, 9070 ) NOPROC(I)  , PROCS(I, 18),NOPROC(CONFIG),
     *  NOPROC(ARRANG)
      KK=0
      DO 410 J=1,NPROG
      L =PRODAT(J,3)
      IF(L.NE. I) GO TO 410
      KK = KK + 1
      TEM(KK) = PRODAT(J,1)
  410 CONTINUE
      WRITE(PRINT,9080) NOPROC(I)
      WRITE(PRINT,9090)  (TEM(K),K=1,KK)
  420 CONTINUE
      WRITE(PRINT, 9110)  NOPROC(CONFIG),  NOPROC(ARRANG)
C     BEGIN  . EVENT   CALCULATIONS
  500 CONTINUE
      T = TEVENT(1)
      NLIST1 = NLIST -1
      DO 520 I=1,NLIST1
      TEVENT(I) = TEVENT(I+1)
  520 CONTINUE
      NTIME = NTIME +1
C     DECREMENT  THE  EVENTS  LIST
      NLIST = NLIST-1
      DO 600 I=1,NPROG
      IF(T.LT.PRODAT(I,7)) GO TO 600
      IF(T.GE.PRODAT(I,8)) GO TO 590
C     SET ACTIVITY  FLAG
      PRODAT(I,2) = 1.
C     CONTAINS   PROCESSOR   NUMBER
      K = PRODAT(I,3)
      PROCS(K,2) = PROCS(K,2) +  PRODAT(I,4)*PRODAT(I,8)
C     CALCULATE  NUMBER  OF  INSTRUCTIONS  PER  SECOND  BEING
C     EXECUTED   BY  EACH   PROCESSOR
      PROCS(K,3) = PROCS(K,3) +  PRODAT(I,5)*PRODAT(I,8)
C     RESET  ACTIVITY  FLAG
      GO  TO  600
  590 PRODAT(I,2) = 0.
  600 CONTINUE
C     DO . DATA TRANSFER   CALCULATIONS  FOR  EACH  DATA  POINT
```

Figure 6-3 (Continued)

```
      K=1
      DO 800  I =1,NDATAP
       KK=0
      J = IODATA(K)
      BITRAT = IODATA(K+2)*IODATA(K+3)
      J4 = J -4
       DO  700 L=1,J4
      DO 620  LL=1,NPROG
      IF(IODATA(K+L+3)-PRODAT(LL,1)) 620,,620
C     CHECK  MODULE  LIST
C     CHECK  MODULE  ACTIVITY
      IF(PRODAT(LL,2)-1.) 630,,630
      LLL = LL
      GO TO 640
  620 CONTINUE
  630 NP(L) = 0.
      IF(L.EQ.1) GO TO 740
      GO TO 700
C     ENSURE  THAT PROCESSOR BITRATES ARE COUNTED  ONLY  ONCE
  640 IF(L .EQ. 1) GO TO 660
       L1 = L-1
      DO 650 M=1,L1
      IF(NP(M)-PRODAT(LLL,3)) 650,,650
      NP(L) = 0.
      GO TO 700
  650 CONTINUE
  660 NP(L) = PRODAT(LLL,3)
  700 CONTINUE
      IF(NP(1)) ,800
      J4 = J-4
      DO 720 L=2,J4
      IF(NP(L)) ,720
      IF(NP(L)-NP(1)) ,720
      KK=KK+1
      INP = NP(1)
      INPP = 2*NP(L)+3
      PROCS(INP,INPP) = PROCS(INP,INPP) + BITRAT
      IF(KK. GT.1) GO TO 710
      PROCS(INP,17)=PROCS(INP,17) + IODATA(K+2)
  710 CONTINUE
      INP=NP(L)
      INPP = 2*NP(1) + 2
C     ADD  BITRAT TO  THE  OUTPUT AND INPUT  PROCESSOR  SUMS
      PROCS(INP,INPP) = PROCS(INP,INPP) + BITRAT
      PROCS(INP,17)=PROCS(INP,17) + IODATA(K+2)
  720 CONTINUE
  740 CONTINUE
      K = K+J
  800 CONTINUE
      DO 830  I=1,NPROC
      PROCS(I,17)  = PROCS(I,2)*SLOWCY + PROCS(I,3)*FASTCY
     *               + PROCS(I,17)
      K = PROCS(I,1)
      PROCSM(K,NTIME,1)=T
      PROCSM(K,NTIME,2) =PROCS(I,17)
      DO 820 J=3,NSMCOL
      PROCSM(K,NTIME,J)= PROCS(I,J+1)
  820 CONTINUE
  830 CONTINUE
      IF(REPORT .EQ. NO) GO TO 1000
      DO 970 I=1,NOPT
```

Figure 6-3 (Continued)

```
      DO 840M=1,NPROG
      IF(PROGS(I,1)-PRODAT(M,1)) 840,,840
      IF(PRODAT(M,2)-1.) 970,,970
      GO TO 845
  840 CONTINUE
  845 CONTINUE
      K=1
      DO 950 J=1,NDATAP
      IF(PROGS(I,1) - IODATA(K+4)) ,890
      KK = IODATA(K)
      KK5 = KK-5
      DO  850 L=1,KK5
C     DOES  THIS  REPRESENT  A  TRANSFER  OF  DATA?
      IF(PROGS(I,1) - IODATA(K+L+4)) ,860
  850 CONTINUE
      GO TO 950
  860 DO 865 M=1,NPROG
      IF(PRODAT(M,1)-IODATA(K+4)) 865,,865
      IF(PRODAT(M,2)-1.) 950,,950
      IF(PRODAT(M,3)-PROGS(I,2))  870,,870
  865 CONTINUE
      GO TO 950
  870 PROGS(I,6) = IODATA(K+2)*IODATA(K+3) + PROGS(I,6)
      PROGS(I,4)=PROGS(I,4)+IODATA(K+2)
      PROGS(I,5) = IODATA(K+2) + PROGS(I,5)
C.    CALCULATE  DATA  INPUT RATE IN  BPS
C     CALCULATE  NUMBER  OF  CYCLE  STEALS  FOR  INPUT   DATA  TRANSFER
      GO TO 950
  890 CONTINUE
      KK=IODATA(K)
      KK5=KK-5
      DO 910 L=1,KK5
      DO 900 M=1,NPROG
      IF(PRODAT(M,1)-IODATA(K+L+4)) 900,,900
      IF(PRODAT(M,2)-1.) 910,,910
      MM = M
      GO TO 905
  900 CONTINUE
  905 CONTINUE
      IF(PRODAT(MM,3)-PROGS(I,2)) 920,,920
  910 CONTINUE
      GO TO 950
  920 PROGS(I,7) = PROGS(I,7) + IODATA(K+2)*IODATA(K+3)
      PROGS(I,4)=PROGS(I,4)+IODATA(K+2)
      PROGS(I,5) = PROGS(I,5) + IODATA(K+2)
  950 K = K + KK
      DO 960 N=1,NPROG
      IF(PRODAT(N,1)-PROGS(I,1)) 960,,960
      PROGS(I,3)=PRODAT(N,4)*PRODAT(N,6)*SLOWCY
     *     +PRODAT(N,5)*PRODAT(N,6)*FASTCY
      PROGS(I,5)=PROGS(I,3)+PROGS(I,5)
      GO TO  970
  960 CONTINUE
  970 CONTINUE
 1000 CONTINUE
C     CALCULATION  OF  MAXIMUM  I/O BITRATES  AND BUSY CYCLES FOR  EACH
C     PROCESSOR
      DO 1030 I=1,NPROC
      TEMP(1) = PROCS(I,4)+PROCS(I,6) + PROCS(I,8) + PROCS(I,10)
     *        +PROCS(I,12)+ PROCS(I,14)
      IF(TEMP(1) .LE. OPTP(I,2)) GO TO 1010
      OPTP(I,2) = TEMP(1)
      OPTP(I,3) = T
 1010 CONTINUE
```

Figure 6-3 (Continued)

```
      TEMP(1) = PROCS(I,5) + PROCS(I,7) + PROCS(I,9) +  PROCS(I,11)
     *        + PROCS(I,13) + PROCS(I,15)
      IF(OPTP(I,4) .GE.TEMP(1)) GO TO 1020
      OPTP(1,4) = TEMP(1)
      OPTP(1,5) = T
 1020 IF(OPTP(1,6).GE.PROCS(I,17)) GO TO 1030
      OPTP(1,6) = PROCS(1,17)
      OPTP(1,7) = T
 1030 CONTINUE
C     PROCESSOR   SUMMARY
      WRITE(PRINT, 9200) T
C     SKIP  THE  PROCESSOR  NUMBER  FOR  WHICH . SUMMARY  IS  WRITTEN
      DO 1100  I=1,NPROC
C     NOPROC  CONTAINS FIELDATA  OF  EACH  PROCESSOR  NUMBER .
      WRITE(PRINT, 9220) NOPROC(I)
      DO 1080 M=1,2
      NPROC1 = NPROC -1
      DO 1060 J=1,NPROC1
      K=J
      IF( K .GE. I ) K = K+1
      TEMP(2*J-1) = NOPROC(K)
      TEMP(2*J)   = PROCS(I ,2*K+M+1)
 1060 CONTINUE
      NPROC1 = 2*(NPROC-1)
      IF( M .EQ. 2) GO TO 1070
      WRITE(PRINT,9240) (TEMP(L),L=1,NPROC1)
      GO TO 1080
 1070 WRITE(PRINT, 9260) (TEMP(L),L=1,NPROC1)
 1080 CONTINUE
      WRITE(PRINT , 9280 ) PROCS(I, 17)
 1100 CONTINUE
      IF(REPORT .EQ. NO) GO TO 1250
C     OPTIONAL  REPORT(S)  GENERATED   HERE
      WRITE(PRINT, 9300 )
      DO 1200 I=1,NOPT
      K = PROGS(I,2)
C     CALCULATE  PERCENTAGE OF PROCESSOR CYCLES  USED BY THIS PROGRAM
      TEMP(1) = 100.* PROGS(I,5)/ PROCS(K,17)
      WRITE(PRINT,9320) PROGS(I,1), PROGS(I,7)
      WRITE(PRINT,9330) PROGS(I,1), PROGS(I,6)
      WRITE(PRINT, 9340 ) PROGS(I,1) , PROGS(I,5)
      WRITE(PRINT, 9350 ) PROGS(I,1) , TEMP(1)
 1200 CONTINUE
      DO 1220 I=1,NOPT
      DO 1210 J=3,OPTC
      PROGS(I,J)=0.
 1210 CONTINUE
 1220 CONTINUE
 1250 CONTINUE
      L=NPCCOL-1
      DO 1270 I=1,NPROC
      DO 1260 J=2,L
      PROCS(I,J)=0.
 1260 CONTINUE
 1270 CONTINUE
 1600 IF(T .LT. TEVENT(2) ) GO TO 500
      WRITE(PRINT, 9400)
      DO 1620 L=1,NPROC
      J=PROCS(L,1)
      NPROC1=NPROC-1
      ITEMP(1)=6HTIME
      ITEMP(2)=6HCPS
```

Figure 6-3 (Continued)

```
      DO 1612 K=1,NPROC1
      KK=K
       IF(KK.GE.J)  KK=K+1
      K6=6*K-5
      ITEMP(K6)=6HIN FRO
      ITEMP(K6+1)= 6HM
      ITEMP(K6+2)=KK
      ITEMP(K6+3)=6HOUT TO
      ITEMP(K6+4)=6H
      ITEMP(K6+5)=KK
      ITEM(K) = KK
 1612 CONTINUE
      WRITE(PRINT, 9410) NOPROC(J)
      NPROC1=6*NPROC1+2
      WRITE(PRINT,9420) (ITEMP(K),K=1,NPROC1)
      DO 1616 I=1,NTIME
      TEMP(1)=PROCSM(J,I,1)
      TEMP(2)=PROCSM(J,I,2)
      NPROC1=NPROC-1
      DO 1614 K=1,NPROC1
      KK = ITEM(K)
      TEMP(2*K+1)=PROCSM(J,I,2*KK+1)
      TEMP(2*K+2)=PROCSM(J,I,2*KK+2)
 1614 CONTINUE
      NPROC1=2*NPROC
      WRITE(PRINT,9430) (TEMP(K),K=1,NPROC1)
 1616 CONTINUE
 1620 CONTINUE
      NTIME =0
      WRITE(PRINT, 9100)
      DO 1610  I=1,NPROC
      WRITE(PRINT,9120) NOPROC(I), (OPTP(I,N),N=2,7,1)
 1610 CONTINUE
      WRITE(PRINT,9999)
C     FOR  A  NEW  PROGRAM  ARRANGEMENT  ZERO  OUT  THE  ARRAYS  TEVENT,
C     AND  PROCS.
      DO  1640 I=2,NLIST
      TEVENT(I) = 0.
 1640 CONTINUE
      NLIST =1
      DO 1660  I=1,NPROC
      DO 1650 J=2,NPCCOL
      PROCS(I,J) = 0.
 1650 CONTINUE
 1660 CONTINUE
      TEVENT(1) = 99999.
      DO 1664 I=1,NPROC
      DO 1662 J=2,7
      OPTP(I,J)=0.
 1662 CONTINUE
 1664 CONTINUE
      IF(ARRANG .LT. NARRNG ) GO TO 70
      ARRANG =0
      DO  1670 I=1,NPROC
      PROCS(I,1) =0.
 1670 CONTINUE
      NPROC =0
      DO 1690 J=1,OPTC
      DO 1680 I=1,KOPT
C     WASH  OUT  PROGS  TO  PREPARE  FOR  NEW  REPORTS, IF  REQUESTED,
C     FOR  THE  NEW  COMPUTER  CONFIGURATION
      PROGS(I,J) =0.
 1680 CONTINUE
 1690 CONTINUE
```

Figure 6-3 (Continued)

```
      IF(CONFIG .LT. NCONFG ) GO TO 5
 9000 FORMAT(I2,10(1X,I2 ),2(2X,F6.2))
 9001 FORMAT( 10X, I2 ,10(1X,I2), 2(2X,F6.2) )
 9002 FORMAT(55X, 20H*** INPUT  CARDS ***  //)
 9020 FORMAT(16(A4,1X))
 9021 FORMAT(10X, 16(A4,1X) )
 9030 FORMAT(16(A4,A1))
 9031 FORMAT(10X,16(A4,A1))
 9040 FORMAT(A4,2X, F4.0)
 9041 FORMAT(10X, A4, 2X, F4.0 )
 9050 FORMAT(A4,1X,6(F11.3,1X))
 9051 FORMAT(10X, A4, 1X, 6(F11.3,1X))
 9060 FORMAT(A6,2X,F11.4,2X,F6.0,2X,A4,2X,9(A4,1X))
 9061 FORMAT(10X,A6, 2X,F11.4,2X,F6.0,2X,A4,2X,9(A4,1X))
 9065 FORMAT(1H1,48X ,33H***  INITIALIZATION  SUMMARY ***    /)
 9070 FORMAT(1H0, 13X, 19HPROCESSOR  NUMBER    , A5, 7H  HAS    ,F10.0,
     * 39H  CELLS  OCCUPIED  FOR  CONFIGURATION  ,A5 ,15H  ARRANGEMENT
     *  ,  A5//)
 9080 FORMAT(1H0,43X,19HPROCESSOR  NUMBER   ,A5,21H  CONTAINS  PROGRAMS
     *    /)
 9090 FORMAT(10X,20(A4,2X))
 9100 FORMAT(1H1,// 48X,34H******  PROCESSOR  MAXIMA  ******   //)
 9110 FORMAT(1H0,27X , 40HA  SUMMARY  FOR  CONFIGURATION  NUMBER  , A5,
     * 29H  AND  PROGRAM  ARRANGEMENT  , A5 )
 9120 FORMAT(1H0,55X,17HPROCESSOR  NUMBER   ,2X,A6  //
     *32X,28HMAXIMUM  INPUT  BITRATE  OF   ,F9.0,21H  OCCURRED  AT  TIME
     *  ,F9.0 //32X,29HMAXIMUM  OUTPUT  BITRATE  OF   ,F9.0,21H  OCCURRE
     *D  AT  TIME    ,F9.0     //32X,
     *35HMAXIMUM  NUMBER  OF  BUSY  CYCLES,  ,F10.0,26H  CPS,  OCCURRED
     * AT  TIME    ,F9.0)
 9200 FORMAT(1H0, 40X,42HPROCESSOR  SUMMARY  AT  TIME-OF-FLIGHT  =  ,
     *  F10.0 )
 9220 FORMAT(1H0 ,56X, 11HPROCESSOR  , A6 )
 9240 FORMAT(1H  ,15X,15HINPUT FROM(BPS)   ,5X,5(A6,1X,F9.0,2X)  /
     * 35X, 5(A6,1X, F9.0,2X))
 9260 FORMAT(1H  ,15X,15HOUTPUT TO(BPS)   ,5X,5(A6,1X,F9.0,2X)  /
     * 35X, 5(A6,1X, F9.0,2X))
 9280 FORMAT(1H ,15X, 36HTOTAL  NUMBER  OF  BUSY  CYCLES  =  ,  F10.0 )
 9300 FORMAT(1H0 ,44X , 44H***** OPTIONAL  PROGRAM  DATA  SUMMARY *****)
 9320 FORMAT(1H0, 40X, 9HPROGRAM  , A6 ,22HOUTPUT  BITRATE(BPS) =,F10.0)
 9330 FORMAT(1H , 40X, 9HPROGRAM  , A6 ,22HINPUT   BITRATE(BPS) =,F10.0)
 9340 FORMAT(1H , 40X, 9HPROGRAM  , A6 ,38HNUMBER  OF  PROCESSOR  CYCLES
     *  USED = , F10.0 )
 9350 FORMAT(1H  ,40X ,9HPROGRAM  , A6 ,48HPERCENTAGE  OF  PROCESSOR  BU
     *SY  CYCLES  USED =  ,  F10.1  )
 9400 FORMAT(1H1,49X,34H*** PROCESSOR  FLIGHT  SUMMARY ***    //)
 9410 FORMAT(54X,19HPROCESSOR  NUMBER   ,A5 /)
 9420 FORMAT(8X,A4,6X,A5,4X,10(A6,A2,I1,1X))
 9430 FORMAT(5X,12(F8.0,2X))
 9999 FORMAT(1H1)
      END

COMPILATION:        NO  DIAGNOSTICS.
```

Figure 6-3 (Continued)

```
*** INPUT CARDS **

2  2  1  0  0  0  0  0  0  0  0     .00     .00      CONFIGURATION CARD
YES  AAAA AAAB AAAH                                   REPORT CARD
PROC    1.
AAAA,AAAB,AAAC,AAAD,AAAE,AAAF
PROC    2.
AAAG,AAAH,AAAI,AAAJ               PROGRAM ARRANGEMENT CARDS
PROC    3.   EXTERNAL
EXTF         DATA SOURCE                                        PROGRAM
END                                                             MODULE
                                                                CARDS
AAAA     10.000     10.000     2.000      .000     20.000     10.000
AAAB     10.000     10.000     2.000      .000     20.000     20.000
AAAC     10.000     10.000     2.000      .000     20.000     30.000
AAAD     10.000     10.000     2.000     3.000     15.000     40.000
AAAE     10.000     10.000     2.000     3.000     17.000     50.000
AAAF     10.000     10.000     3.000     3.000     19.000     60.000
AAAG     10.000     10.000     3.000     5.000     22.000     70.000
AAAH     10.000     10.000     3.000     5.000     24.000     80.000
AAAI     10.000     10.000     3.000     5.000     26.000     90.000
AAAJ     10.000     10.000     3.000     5.000     28.000    100.000
EXTF       .000       .000      .000      .000     28.000       .000
END
AAAAAA     10.0000     5.  EXTF   AAAC AAAG
AAAAAB     20.0000    10.  EXTF   AAAA AAAB
AAAAAC     30.0000    15.  EXTF   AAAC AAAG
AAAAAD     40.0000    20.  EXTF   AAAD                  I/O DATA CARDS
AAAAAE     50.0000    20.  AAAA   AAAB AAAD AAAF
AAAAAF     60.0000    20.  AAAI   AAAH AAAE
END
```

Figure 6-4 Printout of Sample First Pass Cards

```
PROC    1.
AAAA,AAAB,AAAC,AAAD,AAAE,AAAI
PROC    2.
AAAF,AAAG,AAAH,AAAJ
PROC    3.
EXTF
END
```

Figure 6-5

Printout of Sample Rearrangement Deck

```
YES  AAAC AAAD
PROC    1.
AAAA,AAAC,AAAE
PROC    2.
AAAB,AAAD
PROC    3.
AAAF,AAAG,AAAH,AAAI,AAAJ
PROC    4.
EXTF
END
```

Figure 6-6 Printout of Sample Configuration Deck

```
                    PROCESSOR  SUMMARY  AT  TIME-OF-FLIGHT  =         5.

                                   PROCESSOR  ONE
INPUT FROM(BPS)     TWO        1200.  THREE      1500.
OUTPUT TO(BPS)      TWO           0.  THREE         0.
TOTAL  NUMBER  OF  BUSY  CYCLES  =       1720.

                                   PROCESSOR  TWO
INPUT FROM(BPS)     ONE           0.  THREE       500.
OUTPUT TO(BPS)      ONE        1200.  THREE         0.
TOTAL  NUMBER  OF  BUSY  CYCLES  =       1540.

                                   PROCESSOR  THREE
INPUT FROM(BPS)     ONE           0.  TWO           0.
OUTPUT TO(BPS)      ONE        1500.  TWO         500.
TOTAL  NUMBER  OF  BUSY  CYCLES  =        100.

                       ***** OPTIONAL  PROGRAM  DATA  SUMMARY *****

                    PROGRAM  AAAA  OUTPUT  BITRATE(BPS) =        0.
                    PROGRAM  AAAA  INPUT   BITRATE(BPS) =      200.
                    PROGRAM  AAAA  NUMBER  OF  PROCESSOR  CYCLES  USED =       260.
                    PROGRAM  AAAA  PERCENTAGE  OF  PROCESSOR  BUSY  CYCLES  USED =       15.1

                    PROGRAM  AAAB  OUTPUT  BITRATE(BPS) =        0.
                    PROGRAM  AAAB  INPUT   BITRATE(BPS) =      200.
                    PROGRAM  AAAB  NUMBER  OF  PROCESSOR  CYCLES  USED =       260.
                    PROGRAM  AAAB  PERCENTAGE  OF  PROCESSOR  BUSY  CYCLES  USED =       15.1

                    PROGRAM  AAAH  OUTPUT  BITRATE(BPS) =        0.
                    PROGRAM  AAAH  INPUT   BITRATE(BPS) =        0.
                    PROGRAM  AAAH  NUMBER  OF  PROCESSOR  CYCLES  USED =       360.
                    PROGRAM  AAAH  PERCENTAGE  OF  PROCESSOR  BUSY  CYCLES  USED =       23.4
```

Figure 6-7 Event Sample Printout

Figure 6-8 Initialization Printout

```
*** INITIALIZATION SUMMARY ***

PROCESSOR NUMBER ONE   HAS      330. CELLS OCCUPIED FOR CONFIGURATION ONE   ARRANGEMENT ONE

                    PROCESSOR NUMBER ONE   CONTAINS PROGRAMS
AAAA AAAB AAAC AAAD AAAE AAAF
PROCESSOR NUMBER TWO   HAS      420. CELLS OCCUPIED FOR CONFIGURATION ONE   ARRANGEMENT ONE

                    PROCESSOR NUMBER TWO   CONTAINS PROGRAMS
AAAG AAAH AAAI AAAJ
PROCESSOR NUMBER THREE HAS        0. CELLS OCCUPIED FOR CONFIGURATION ONE   ARRANGEMENT ONE

                    PROCESSOR NUMBER THREE CONTAINS PROGRAMS
EXIT
```

```
****** PROCESSOR MAXIMA ******

                 PROCESSOR NUMBER ONE
MAXIMUM INPUT BITRATE OF     2700. OCCURRED AT TIME        5.
MAXIMUM OUTPUT BITRATE OF       0. OCCURRED AT TIME        0.
MAXIMUM NUMBER OF BUSY CYCLES,   1720. CPS, OCCURRED AT TIME     5.
                 PROCESSOR NUMBER TWO
MAXIMUM INPUT BITRATE OF      500. OCCURRED AT TIME        5.
MAXIMUM OUTPUT BITRATE OF    1200. OCCURRED AT TIME        5.
MAXIMUM NUMBER OF BUSY CYCLES,   1540. CPS, OCCURRED AT TIME     5.
                 PROCESSOR NUMBER THREE
MAXIMUM INPUT BITRATE OF        0. OCCURRED AT TIME        0.
MAXIMUM OUTPUT BITRATE OF    2000. OCCURRED AT TIME        5.
MAXIMUM NUMBER OF BUSY CYCLES,    100. CPS, OCCURRED AT TIME     3.
```

Figure 6-9 Sample Processor Maxima Printout

*** PROCESSOR FLIGHT SUMMARY ***

PROCESSOR NUMBER ONE

TIME	CPS	IN FROM 2	OUT TO 2	IN FROM 3	OUT TO 3
0.	780.	0.	0.	700.	0.
0.	780.	0.	0.	700.	0.
3.	1800.	0.	0.	1500.	0.
5.	1720.	1200.	0.	1500.	0.
15.	1440.	1200.	0.	700.	0.
17.	1140.	0.	0.	700.	0.
19.	780.	0.	0.	700.	0.
20.	0.	0.	0.	0.	0.
22.	0.	0.	0.	0.	0.
24.	0.	0.	0.	0.	0.
26.	0.	0.	0.	0.	0.
28.	0.	0.	0.	0.	0.

PROCESSOR NUMBER TWO

TIME	CPS	IN FROM 1	OUT TO 1	IN FROM 3	OUT TO 3
0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.
3.	0.	0.	0.	0.	0.
5.	1540.	0.	1200.	500.	0.
15.	1540.	0.	1200.	500.	0.
17.	1480.	0.	0.	500.	0.
19.	1480.	0.	0.	500.	0.
20.	1480.	0.	0.	500.	0.
22.	1060.	0.	0.	0.	0.
24.	720.	0.	0.	0.	0.
26.	360.	0.	0.	0.	0.
28.	0.	0.	0.	0.	0.

PROCESSOR NUMBER THREE

TIME	CPS	IN FROM 1	OUT TO 1	IN FROM 2	OUT TO 2
0.	60.	0.	700.	0.	0.
0.	60.	0.	700.	0.	0.
3.	100.	0.	1500.	0.	0.
5.	100.	0.	1500.	0.	500.
15.	60.	0.	700.	0.	500.
17.	60.	0.	700.	0.	500.
19.	60.	0.	700.	0.	500.
20.	40.	0.	0.	0.	500.
22.	0.	0.	0.	0.	0.
24.	0.	0.	0.	0.	0.
26.	0.	0.	0.	0.	0.
28.	0.	0.	0.	0.	0.

Figure 6-10 Sample Processor Summary Printout

6.2 CONFIGURATION RELIABILITY EVALUATION

In this section the probability of failure, the reliability and the mean time to failure (MTTF) of the three configurations described in Volume I are examined. The reliability of the data transfer network is not considered. The discussion will be limited to the major processors. The reliability of a component will be assumed to follow a negative exponential distribution, which is generally regarded as a worst case situation after burn-in of the components. This distribution is

$$R(t) = e^{-\lambda t}$$

The probability of failure is given by

$$Q(t) = 1 - R(t) = 1 - e^{-\lambda t}$$

The negative exponential distribution describes equipment with constant failure rate, which usually applies to electronic components after burn-in.

For a series connection of n pieces of equipment we have the product rule for reliability

$$R_T(t) = \prod_{i=1}^{n} R_i(t) = e^{-\sum_{i=1}^{n} \lambda_i t} \qquad (1)$$

and for probability of failures we have

$$Q_T(t) = 1 - R_T(t) = 1 - \prod_{i=1}^{n} R_i(t) = 1 - \prod_{i=1}^{n} (1 - Q_i(t)) \qquad (2)$$

As more elements are added in series the reliability decreases.

For a parallel connection of n elements we have the following expressions:

$$Q_T(t) = \prod_{i=1}^{n} Q_i(t) = \prod_{i=1}^{n} (1 - R_i(t)) = \prod_{i=1}^{n} (1 - e^{-\lambda_i t}) \qquad (3)$$

and

$$R_T(t) = 1 - Q_T(t) = 1 - \prod_{i=1}^{n} (1 - e^{-\lambda_i t}) \qquad (4)$$

In this case, as more elements are added in parallel the probability of failure decreases and therefore the reliability increases.

The last quantity is the mean time to failure (MTTF). This is given by the integral of the reliability:

$$M_T = \int_0^\infty R(t)\, dt$$

For negative exponential we have

$$M_T = \int_0^\infty e^{-\lambda t} dt = -\frac{1}{\lambda} e^{-\lambda t} \Big]_0^\infty = \frac{1}{\lambda}$$

6.2.1 CENTRALIZED LINKED UNIT PROCESSORS

The centralized linked unit processor system shown in Figure 6-12, has four computers in parallel. The reliability of the series connection of a processor, memory and I/O unit is given by:

$$R(t) = e^{-\lambda_P t} e^{-\lambda_M t} e^{-\lambda_I t} = e^{-(\lambda_P+\lambda_M+\lambda_I)t} \tag{5}$$

Using the expression (4) for a parallel connection of these computers we get

$$Q_T(t) = \left[1 - e^{-(\lambda_P+\lambda_M+\lambda_I)t}\right]^4 \tag{6}$$

and therefore

$$R_T(t) = 1 - Q_T(t) = 1 - \left[1 - e^{-(\lambda_P+\lambda_M+\lambda_I)t}\right]^4 \tag{7}$$

Write $\lambda = \lambda_P + \lambda_M + \lambda_I$. Then equation (7) may be expanded using the binomial expansion to give:

$$R_T(t) = 1 - \left[1 - e^{-\lambda t}\right]^4 = 1 - \sum_{k=0}^{4} \binom{4}{k}(-1)^k e^{-k\lambda t}$$

$$= \sum_{k=1}^{4} \binom{4}{k}(-1)^{k+1} e^{-k\lambda t} \tag{8}$$

The MTTF can be calculated quite readily from equation (8) to get:

$$\begin{aligned} M_T &= \int_0^\infty R_T(t)\,dt \\ &= \int_0^\infty \left[\sum_{k=1}^{4} \binom{4}{k}(-1)^{k+1} e^{-k\lambda t}\right] dt \\ &= \sum_{k=1}^{4} \binom{4}{k}(-1)^{k+1} \left[\int e^{-k\lambda t}\,dt\right] \\ &= \sum \binom{4}{k}(-1)^{k+1} \frac{1}{k\lambda} = \frac{25}{12\lambda} = \frac{25}{12(\lambda_P+\lambda_M+\lambda_I)} \end{aligned} \tag{9}$$

The four-fold redundancy has enhanced the MTTF by a factor of $\simeq 2$, since for the single processor:

$$R(t) = e^{-(\lambda_P+\lambda_M+\lambda_I)}$$

$$M_T = \int e^{-(\lambda_P+\lambda_M+\lambda_I)} dt = \frac{1}{\lambda_P+\lambda_M+\lambda_I}$$

6.2.2 CENTRALIZED MULTIPROCESSOR SYSTEM

In this system the processors are four-fold redundant and are connected via switching units to the memories and I/O units, each of which are also four-fold redundant. Thus the redundancy is at the module rather than the computer level. The switches are ignored in the reliability calculations. This assumption is probably less valid than in the preceding case since there are now three switching elements in series. These elements will have to be considered when a detailed trade off is performed on the competing systems.

The reliability can be written as a product of the reliabilities of the series connection of four-fold redundant processors, memories and I/O units.

$$
\begin{aligned}
R'_T &= \{1-[1-e^{-\lambda_p t}]\}\{1-[1-e^{-\lambda_M t}]\}\{1-[1-e^{-\lambda_I t}]\} \\
&= \sum_{j=1}^{4}\binom{4}{j}(-1)^{j+1}e^{-j\lambda_p t}\sum_{k=1}^{4}\binom{4}{k}(-1)^{k+1}e^{-k\lambda_M t}\sum_{\ell=1}^{4}\binom{4}{\ell}(-1)^{\ell+1}e^{-\ell\lambda_I t} \\
&= \sum_{j=1}^{4}\sum_{k=1}^{4}\sum_{\ell=1}^{4}\binom{4}{j}\binom{4}{k}\binom{4}{\ell}(-1)^{j+k+\ell+1}\, e^{-(j\lambda_p+k\lambda_M+\ell\lambda_I)t}
\end{aligned}
\tag{10}
$$

The MTTF is given by

$$
\begin{aligned}
M'_T &= \int_0^\infty \sum_{j=1}^{4}\sum_{k=1}^{4}\sum_{\ell=1}^{4}\binom{4}{j}\binom{4}{k}\binom{4}{\ell}(-1)^{j+k+\ell+1}e^{-(j\lambda_p+k\lambda_M+\ell\lambda_I)t}\,dt \\
&= \sum_{j=1}^{4}\sum_{k=1}^{4}\sum_{\ell=1}^{4}\binom{4}{j}\binom{4}{k}\binom{4}{\ell}(-1)^{j+k+\ell+1}\frac{1}{j\lambda_p+k\lambda_M+\ell\lambda_I}
\end{aligned}
\tag{11}
$$

6.3.3 DECENTRALIZED LINKED UNITPROCESSORS

This system resembles the centralized linked unit processors. The reliability of the system will be calculated for n processors. All processors are assumed identical, with each computer connected in four-fold redundancy. Setting $\lambda=\lambda_P+\lambda_M+\lambda_I$ gives:

$$
\begin{aligned}
R''_T &= [1-(1-e^{-\lambda t})^4]^n \\
&= \left(\sum_{j_1=1}^{4}\binom{4}{j_1}(-1)^{j_1+1}e^{-j_1\lambda t}\right)\cdots\left(\sum_{j_n=1}^{4}\binom{4}{j_n}(-1)^{j_n+1}e^{-j_n\lambda t}\right) \\
&= \sum_{j_1=1}^{4}\cdots\sum_{j_n=1}^{4}\binom{4}{j_1}\binom{4}{j_2}\cdots\binom{4}{j_n}(-1)^{\sum_{i=1}^{n}(j_i+1)}\, e^{-\lambda t\left(\sum_{i=1}^{n} j_i\right)}
\end{aligned}
\tag{12}
$$

NOT REPRODUCIBLE

The MTTF is given by:

$$M_T'' = \sum_{j_1=1}^{4} \cdots \sum_{j_m=1}^{4} \binom{4}{j_1} \cdots \binom{4}{j_m} (-1)^{\sum_{i=1}^{m}(j_i+1)} \int_0^{\infty} e^{-\lambda t\left(\sum_1^m j_i\right)} dt \qquad (13)$$

$$= \sum_{j_1=1}^{4} \cdots \sum_{j_m=1}^{4} \binom{4}{j_1} \cdots \binom{4}{j_m} (-1)^{\sum_1^m (j_i+1)} \left[\frac{1}{\left(\sum_1^m j_i\right) \lambda} \right]$$

6.2.4 COMPARISON OF THE RELIABILITIES

(a) The configurations are first compared for the special case in which the processor, memory and I/O units all have the same MTTF. For the centralized systems let $\lambda_1 = \lambda_P = \lambda_M = \lambda_I$, and for the decentralized case let $\lambda_2 = \lambda_P = \lambda_M = \lambda_I$. The centralized systems are assumed to use the same computers (approximately), whereas the decentralized system will probably use less complex equipment. The decentralized system will be assumed to consist of three computers for this comparison.

For the centralized unit processor case:

$$M_T = \frac{25}{12(\lambda_P + \lambda_M + \lambda_I)} = \frac{25}{36\lambda_1} \tag{14}$$

For the centralized multiprocessor system:

$$M_T' = \sum_{j=1}^{4} \sum_{k=1}^{4} \sum_{l=1}^{4} \binom{4}{j}\binom{4}{k}\binom{4}{l}(-1)^{j+k+l+1} \frac{1}{(j+k+l)\lambda_1} \tag{15}$$

$$\cong \frac{1.2}{\lambda_1}$$

For the decentralized case:

$$M_T'' = \sum_{j=1}^{4} \sum_{k=1}^{4} \sum_{l=1}^{4} \binom{4}{j}\binom{4}{k}\binom{4}{l}(-1)^{j+k+l+1} \frac{1}{3\lambda_2(j+k+l)}. \tag{16}$$

$$\cong \frac{1.2}{3\lambda_2}$$

The ratio of the MTTF's for the centralized systems is

$$\frac{M_T'}{M_T} = \frac{1.2/\lambda_1}{25/36\lambda_1} \cong 1.7 \tag{17}$$

so the multiprocessor configuration has an MTTF about 1.7 times that of the centralized unit processor system. The ratio

$$\frac{M_T''}{M_T'} \cong \frac{1.2/3\lambda_2}{1.2/\lambda_1} = \frac{\lambda_1}{3\lambda_2} \tag{18}$$

indicates that for $\lambda_1 > 3\lambda_2$ the MTTF of the decentralized system is longer than that of the centralized multiprocessor system.

(b) Another limiting case is that in which one component amongst processor, memory and I/O unit is far less reliable than the other two. This is most easily achieved by going back to the expressions for the total reliability for each configuration. For the centralized systems let λ_1 stand for the largest time constant and in the decentralized system denote this by λ_2. Equation (8) becomes

$$R_T(t) = \sum_{k=1}^{4} \binom{4}{k}(-1)^{k+1} e^{-k\lambda_1 t} \tag{19}$$

NOT REPRODUCIBLE

and

$$M_T = \sum_{k=1}^{4} \binom{4}{k} (-1)^{k+1} \frac{1}{k\lambda_1} = \frac{25}{12\lambda_1} \tag{20}$$

Equation (10) becomes

$$R_T' = \sum_{j=1}^{4} \binom{4}{j} (-1)^{j+1} e^{-j\lambda_1 t} \tag{21}$$

Which leads to

$$M_T' = \frac{25}{12\lambda_1} = M_T \tag{22}$$

Similarly for the decentralized system, by equation (12) using a three processor system,

$$R_T'' = \left[1 - (1 - e^{-\lambda_2 t})^4\right]^3 \tag{23}$$

$$M_T'' = \sum_{j=1}^{4} \sum_{k=1}^{4} \sum_{\ell=1}^{4} \binom{4}{j}\binom{4}{k}\binom{4}{\ell} (-1)^{j+k+\ell+1} \frac{1}{\lambda_2 (j+k+\ell)} \cong \frac{1.2}{\lambda_2} \tag{24}$$

The two centralized systems have approximately the same MTTF, which corresponds to the MTTF of the weakest link. The four-fold redundancy has doubled the MTTF of this element.

For the decentralized system with three computers the MTTF of the system is only 1.2 times that of its weakest unit.

6.3 CONFIGURATION MECHANIZATIONS AND REDUNDANCY CRITERIA

In generating a DMS configuration for the space shuttle booster two primary criteria must be met:

1. The configuration must be capable of performing the required computational tasks.
2. Mission success must be capable of being achieved after 2 failures and crew safety preserved after 3 failures, i.e., a Fail Operational Fail Operational - Fail Safe (FOFOFS) redundancy criteria.

Required computational capability can be expressed in terms of memory size, memory speed, and processor speed. The most difficult requirement to incorporate in the configuration is the redundancy requirement. For the purpose of discussing some of the implications of the redundancy requirement a configuration as shown in Figure 6-11 will be considered and some of the short-comings of this configuration discussed. This example configuration consists of four linked unit computers tied through switches to four data buses with all computers functioning. All switches are open except those indicated by an X. If there is a failure the failed computer will be switched from its associated data bus and one of the other operating computers required to supply the failed computers data bus in addition to its own.

To mechanize such a configuration a method of recognizing failures must be incorporated. Methods of failure detection can be catagorized as:

1. Each computer is capable of determining if a failure has occurred within itself and generating an output signal indicating a failure.
2. The computers are capable of determining if any of the other computers have failed and generating a special output signal indicating which other computer or computers has failed.

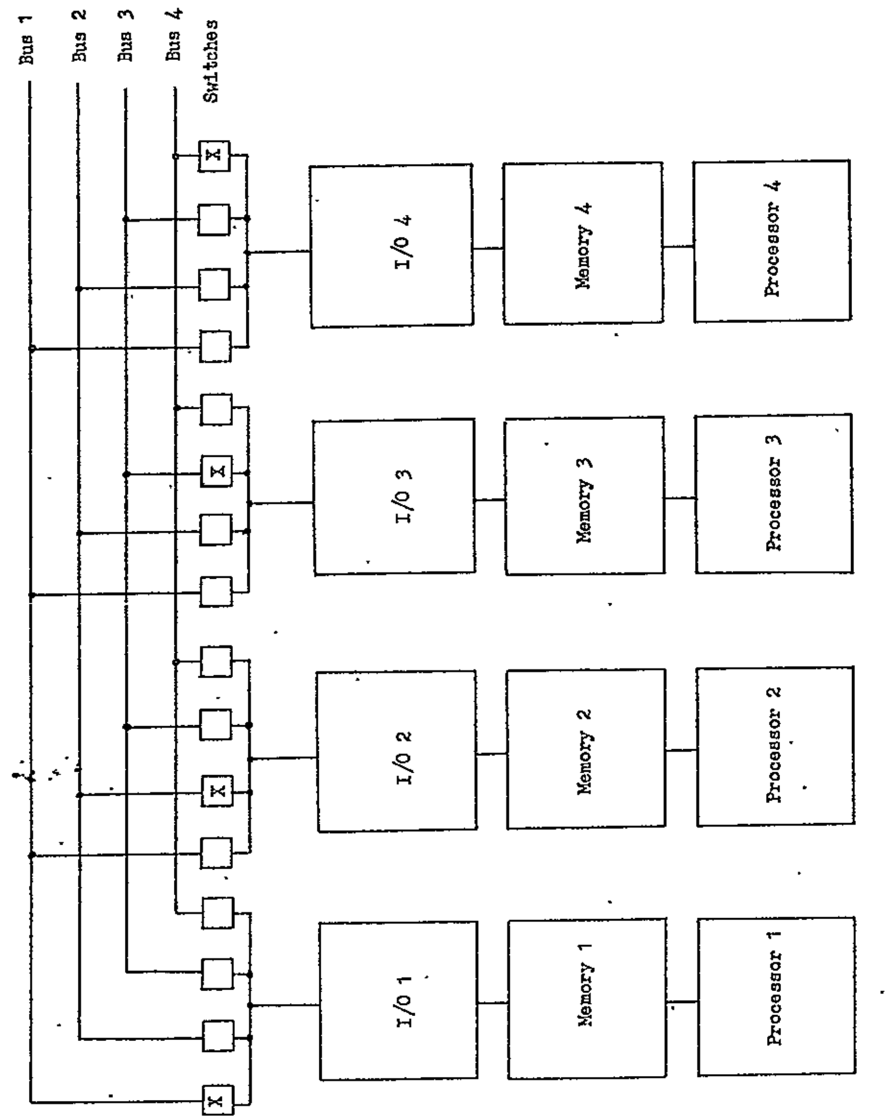


Figure 6-11 Example Configuration

3. Hardware external to the computers monitor the computers operation and determine if and which computers have failed.

4. A combination of the above methods.

Before the method of failure detection and method of implementation can be determined the required response time to a failure must be known. There are some failures which must be recognized immediately and other failures which would have minimal influence upon the mission operation if they were not immediately recognized. A failure which causes premature issuance of the booster/orbiter separation command is an example of a failure which must be recognized and corrected immediately.

There are several methods available for mechanizing the first type of failure detection method where each computer is capable of recognizing a failure within itself. The method which is most straight forward and provides the highest confidence in the detection of all failures is a dual redundant configuration for each computer where a failure is indicated whenever a difference occurs in the outputs of the redundant computers. It is possible to reduce the amount of hardware required of a dual redundant mechanization by replacing some functions with simpler error detecting devices. This would include such items as the addition of parity to the memory system and the mechanization of error detecting algorithms in the processor and I/O units. Another method commonly employed is to use software selftest routines in the computer. In software selftest all computer outputs are calculated and the selftest is performed. Selftest involves executing a known program with known input parameters where the program has been written so that a failure in any computer device will generate a selftest error. Upon successful completion of selftest the previously calculated computer outputs are issued.

In order to test the entire computer including the I/O section and to guard the system against a runaway computer several safeguards would have to be incorporated. During selftest the I/O equipment would have to be disconnected from the data bus and connected to itself, i.e.; the computer output connected to the computer input. This would allow the selftest program to test the computer I/O section. The computer would also be provided with a special register for protection against a runaway computer. This register is loaded with a number at each successful completion of the selftest program. In between loadings the register is counted down and a computer failed indication given if the register ever reaches zero. Thus if selftest is not completed at its planned rate a failure indication is given. Each of the above proposed schemes have certain disadvantages which are:

1. <u>Dual Redundant</u> This method of mechanization is the most complex thus having the highest cost, lowest reliability, greatest weight, and greatest power consumption. The occurrence of a comparison logic failure disabling the issuance of fault indications followed by a computer failure could produce bad data on the data bus.
2. <u>Built in Test Equipment</u> In this method a failure in the built in test equipment followed by a computer failure could produce bad data on the data bus.
3. <u>Software Self Test</u> This method is incapable of recognizing all types of failures. An intermittent error occuring during the output generation portion of the program would possibly not be detected with selftest. It is also generally impossible to design a selftest program that is capable of detecting all possible computer failures and a program which would detect the majority of failures would require a considerable portion of the total computer speed capacity.

Employment of the failure detection method where each computer is tested by the other computers can be rejected rather rapidly as a candidate for the space shuttle booster application. If a single computer is given the capability of shutting down another, then a failure due to a bad computer could possibly cause the shut down of a good computer. To guard against this possibility more than one functioning computer would have to vote against a failed computer before the failed computer could be disabled. To achieve the voting with software in the computers would require excessive quantities of data transferred from one computer to another and would generate numerous insolvable timing problems if failures must be detected rapidly. The external logic required to achieve voting between computers is simple and has immediate response. Systems employing such logic are classified under the third category described below.

One of the most common techniques considered for redundancy applications is majority voting. A majority voter is a device which receives inputs from multiple sources and generates an output equal to the value found on the majority of the inputs. One of the requirements imposed by simple majority voting is that the outputs of the computers voted upon must be identical which means that the computers must be synchronized. For the space shuttle booster application, computational capabilities, must be preserved after three failures have occurred. In the most straight forward mechanization of majority voting for this application, seven computers would be required to guarantee a majority of correct computer outputs after three failures. If an assumption is made concerning the way in which failures occur this number can be reduced to five. The assumption made is that identical failures will not occur simultaneously. If this assumption can be made then several mechanizations using five computers are available for achieving an operating system after three failures. The simplest mechanization would have all five computers operating but only three would have their outputs connected to a majority voter with the majority voter output routed to all four data bus systems. If a failure occurred in any one of the three computers driving the majority voter, it would be removed from the majority voter input and replaced by one of the computers whose output was not being used. Failure would be assumed with two computers voting against the third. After the second failure of a computer into the majority voter system the remaining unconnected computer would be substituted for the second failed computer. After the third failure no further substitutions are made. The majority voter would have to be quadruply mechanized. This system will not be acceptable if a strict interpretation of the redundancy requirement is used because it does not allow for the occurrence of identical simultaneous failures. From a reliability standpoint, the probability of simultaneous multiple failures will be much much less than the probability of non-simultantous multiple failures. Simultaneous is defined to mean during the same computer cycle time.

The candidate redundancy configurations are then defined as:

1. <u>Dual redundant configuration</u> - this is the system where each compuer indicates its own failure by having a dual redundant computer mechanization.
2. <u>Built in test equipment configuration</u> - this is the same type of system as the dual redundant configuration except the second computer is replaced by parity and built in test equipment wherever possible.
3. <u>Majority voting system</u> - this is the last system described where majority voters are mechanized to distinguish a failure.

The equipment required to mechanize a system which will survive 3 failures is either 4 dual redundant systems, 4 built in test equipment (BITE) systems, or a majority voting system with 5 computers. To mechanize each system requires

Dual Redundant	8 computers + failure switching
BITE	4 computers + 4 BITE units + failure switching
Voting	5 computers + voters + failure switching

The amount of hardware required to mechanize failure switching and voters is so very much smaller than a computer that it can be neglected in comparing the systems. Basing the comparison on the amount of hardware required it is obvious that the voting system ranks above the dual redundant system. The tradeoff must then be made between the voting system and the BITE system. The tradeoff point where the two systems are equivalent from the standpoint of the amount of hardware is where each BITE unit is equivalent to $\frac{1}{4}$ of a computer. Using LSI to mechanized the computer central processor units (CPU) and I/O sections, the memory will represent the major portion of the computer. BITE for memory failure detection consists of the addition of parity generation and interrogation logic and additional memory bits for each memory word to store the parity bits for that word. If 2 parity bits are added to each memory word a total of 4 bits in the word must change state before an error will be undetected. If the computer word length is 16 bits (it is assumed that this is a minimum value to be considered for the space shuttle booster), the addition of two parity bits increases the memory size by 12.5%. This is half of the 25% increase which is the tradeoff point between the BITE and voting mechanizations. Since the major portion of the computer is the memory, a detailed tradeoff analysis will probably indicate that the BITE system requires the least amount of hardware. The tradeoff parameters considered when comparing the amount of hardware required to mechanize the DMS are weight, reliability, and cost. Making a simplifying assumption that the type of hardware required to mechanize the BITE is the same as that used to mechanize the basic computer then the tradeoff point of equal cost and weight between the BITE and voting mechanizations is when the BITE is equivalent to $\frac{1}{4}$ of a computer. Because of the differences in mechanization this is not the same tradeoff point for reliability. The reliability tradeoff point is determined from the formula

$$\left[\left(1 + \frac{K}{100}\right) P_f\right]^4 = 5 P_f^4 (1 - P_f) + P_f^5$$

where P_f is the probability of failure of a single computer and K the amount of hardware required to mechanize the BITE measured as a percent of a full computer. Solving the equation for K yields

$$K = 100 \left[\sqrt[4]{5 - 4P_f} - 1\right]$$

assuming $P_f << 5/4$ then $K = 49.5\%$. Thus the reliability tradeoff point occurs when the BITE approaches $\frac{1}{2}$ of a computer in size. As a result of this simplified tradeoff the BITE mechanization is chosen for the comparative evaluation study.

Using this method of failure detection three configurations must be mechanized. These are:

- centralized linked unit system
- centralized multiprocessor system
- decentralized linked unit systems

In order to mechanize the two centralized configurations the number of computers required to perform the DMS computational tasks excluding redundancy requirements must be known. Computer capability is measured by three values : memory size, memory speed, and processor speed. Technology constraints place an upper bound on memory speed and processor speed. Memory speed must be adequate to supply the demands of the I/O equipment and processor. Processor speed must be adequate to perform the required computational tasks. Limits on memory and processor speed for a particular computer are due to a combination of hardware limitations and computer design. It is assumed for the space shuttle booster that hardware limitations can be overcome by proper computer design so that a single computer will be capable of performing the non redundant DMS requirements. Hardware limitations on memory speed can be overcome by designing memories to operate in a modular fashion so that several words can be read and written simultaneously. Processor speed is increased by reducing instruction execution times and mechanizing special instructions which have multiple operation capability. Figure 6-12 is a block diagram of the assumed Centralized linked unit system. It is composed of four identical computers each monitored by its own BITE. The output of each computer is connected to a switch with each switch connected to a different data bus. With all computers operating each data bus will be supplied from a different computer. When a computer fails, its associated data bus will be supplied from another computer which has not failed. With three computer failures all four data bus systems will be supplied from the same computer. All four computers are operating in perfect synchronism with parallel to serial conversion for data supplied to the data bus system occurring in the computer I/O equipment. The logic equation for mechanizing each switch is

$$O = A \cdot \bar{F}_a + B \cdot F_a \cdot \bar{F}_b + C \cdot F_a \cdot F_b \cdot \bar{F}_c + D \cdot F_a \cdot F_b \cdot F_c$$

where O is the switch output, A, B, C, and D the four outputs of the four computers, and F_a, F_b, F_c, and F_d the four BITE outputs of the four computers.

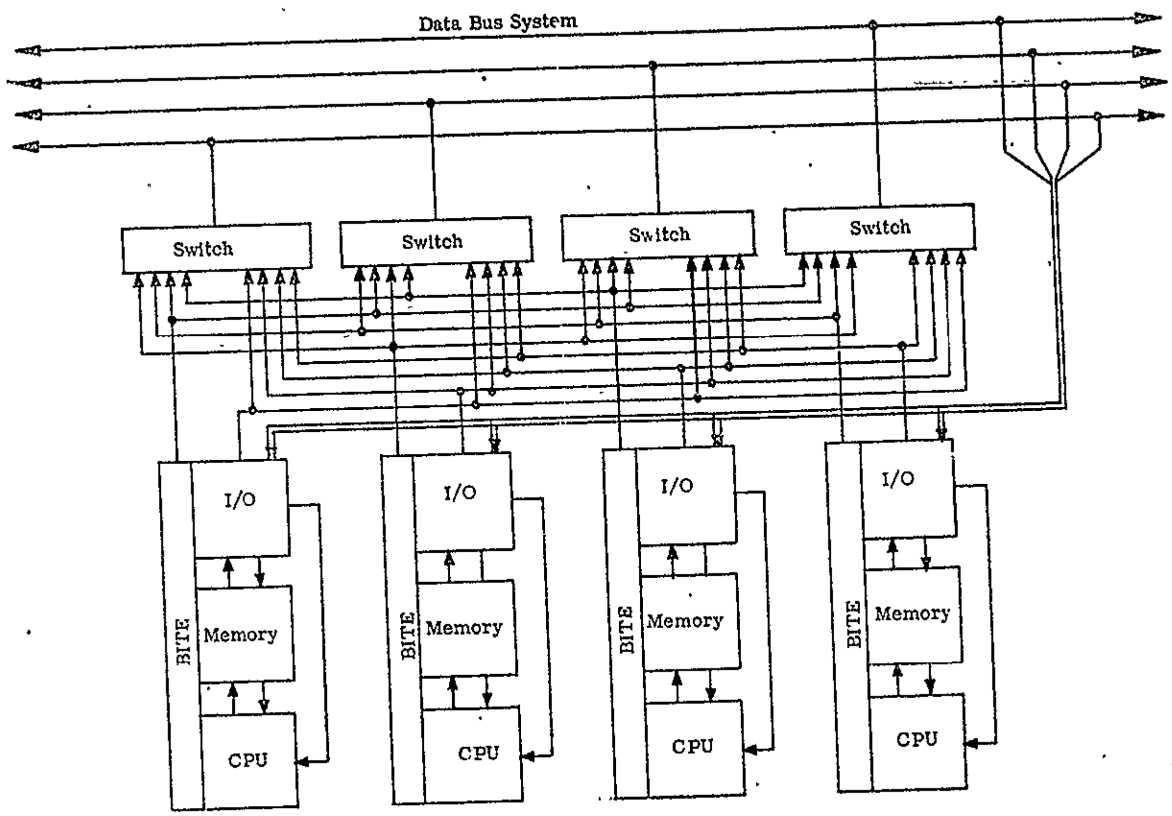


Figure 6-12 Centralized Linked Unit System

In the actual mechanization each data bus will consist of two lines, one carrying address, command and synchronization data and the other line carrying the data transmitted between systems. If the DMS computers are given the task of generating the address line data, a second set of switches as shown in Figure 6-12 will have to be included for the address line interface. All four data bus systems interface as inputs to each computer. The data and address information on each data bus line will include parity bits. Each computer I/O section will serial to parallel convert all four data bus inputs, reject those imputs having parity errors, and majority vote the remaining inputs in obtaining the actual computer input. Data flow between the computer and data bus system will primarily enter and exit the DMS through memory cycle steals initiated by the I/O section. Special high priority data will be transferred directly from the I/O section to the CPU as an interrupt.

Before a description of the centralized multiprocessor can be described the elements of a computer must be described in more detail. Figure 6-13 is a detailed block diagram of a memory system. It is assumed that the memory is a destructive readout (DRO) device. A clock controls the basic memory cycling of reading a word out of memory into a data register and then writing the contents of the data register into memory in a cyclic fashion. The contents of an address register determines which memory word is read from and written into. To read data, the contents of the data register is read after the memory read cycle. To write data into memory, the data register is filled with the data to be written between the data read cycle and data write cycle. The memory system used in the space shuttle booster must communicate with both the processor and I/O computer sections. Each request given to the memory must include the address of the desired memory word and a read or write request. In addition if a write is requested input data to the memory must also be provided. The memory contains logic which looks at the read and write request lines from both the processor and I/O inputs.

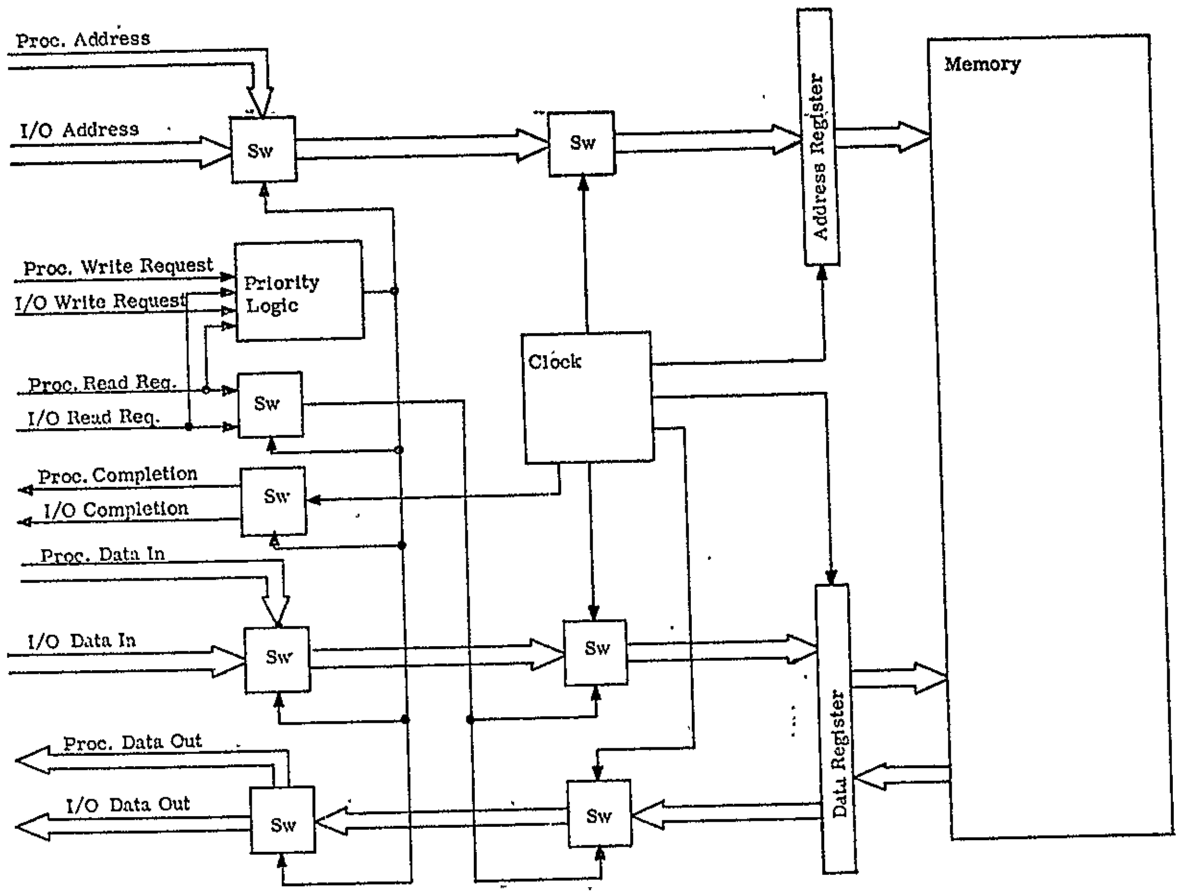


Figure 6-13 Typical Memory Mechanization

This logic resolves a conflict if two requests are received simultaneously and sends out switching commands to route either the processor or I/O data into the memory. Upon completing the requested operation the memory issues a completion signal to the proper unit.

Figure6-14 is a block diagram of a typical CPU. The CPU performs the process of reading an instruction and executing it. Starting this process with an instruction in the instruction register the CPU performs the following sequence:

1. The control logic interprets the instruction. Instructions can be classified into one of three major groups: the instruction requires data from memory, the instruction requires data be sent to memory, or the instruction requires no memory function. If data is required from memory a command is sent from the control logic to toggle the switch on the memory data lines to the number register and an address is generated within the control logic which specifies the memory location to be read. A read request is then issued to the memory. For any instruction the control logic determines if an arithmetic logic function is to be performed and transmits the information to the arithmetic logic. If the instruction requires data to be written in memory, the address is formed from the instruction, index and/or base registers and a write command issued to the memory. A transfer instruction takes the address from the instruction as modified by index and/or base registers and copies it into the instruction register.

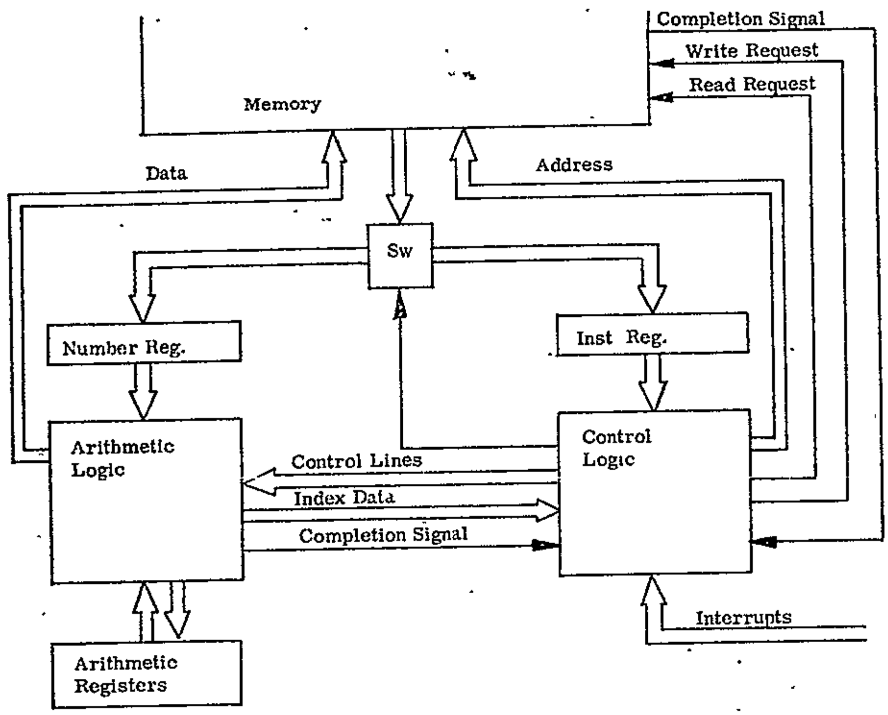


Figure 6-14 Typical Processor Organization

2. With the completion of the instruction execution the control logic checks the interrupt lines. If the interrupts are enabled and an interrupt exists the control logic stores the instruction counter in a special register or in some computer designs goes through a memory store sequence storing the instruction register into a special memory location. The control logic then forces the instruction register to a value dependent upon which interrupt has occurred. The control logic also disables the interrupts either temporarily or until an enable interrupt instruction is executed dependent upon the computer mechanization.

3. The control logic causes the output lines from the memory to be switched to the instruction registers, causes the instruction counter to be fed to the memory address lines and issues a read request to the memory. Upon receiving a completion signal from the memory the new instruction has been placed in the instruction register and the process is repeated starting with step 1.

Figure6-15 is a block diagram of a typical Input/Output mechanization. Data enters the I/O section from four redundant data bus address lines and four redundant data bus data lines. The sequence of an input or output operation starts with serial digital data appearing on the address lines. This data is accumulated in four registers, one for each data bus address line and the parity tested with those inputs showing bad parity rejected. The remaining inputs are majority voted and the majority results transferred to a register from the parity verification and synchronization circuitry.

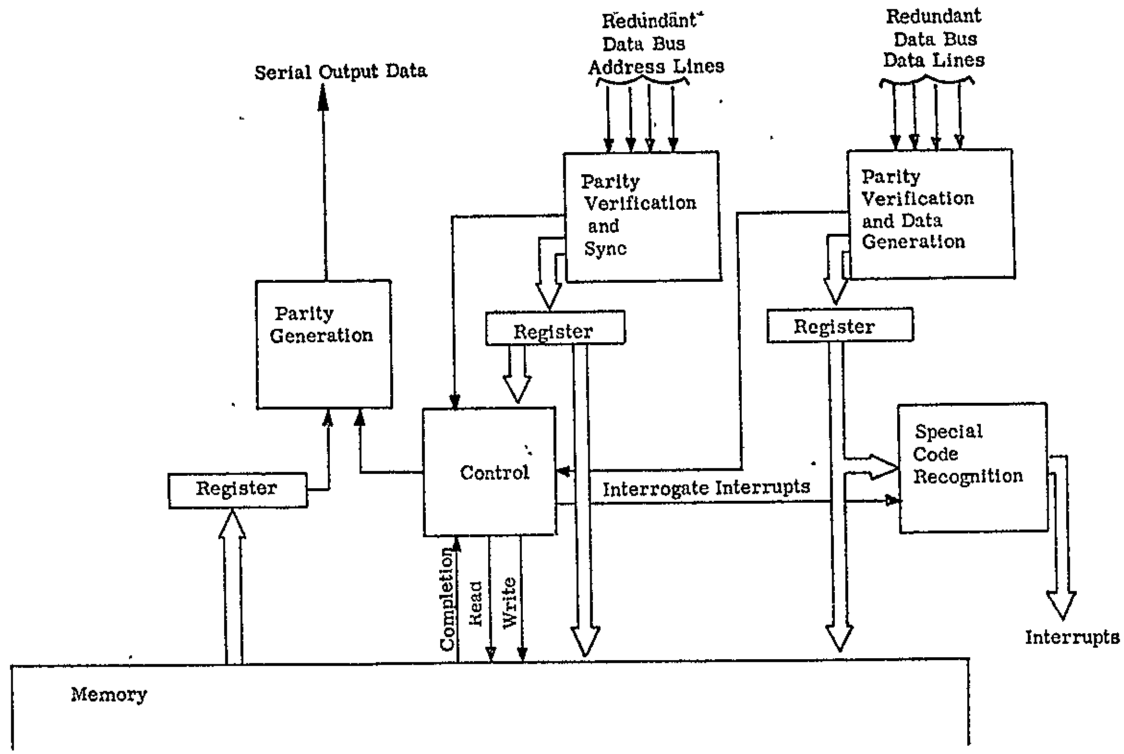


Figure 6-15 Typical Input/Output Mechanization

A portion of the contents of this register is made available to the control logic and is interpreted as a command by the control logic. One of four possible interpretations is made by the control logic: 1) the next data to be received over the data bus data lines is to be stored in the computer memory, 2) data is to be read from memory and transferred to the data bus data lines, 3) the next data to be received over the data bus data lines is to be transferred to the computer interrupt lines, 4) the next data on the data bus lines requires no action from the computer. If the next data from the data bus lines is to be written into memory or interpreted as computer interrupts, the control logic waits until the parity verification and data generation logic signals the completion of its reception of data. The parity verification and data generation logic receives its inputs from four redundant data bus data lines rejecting that data with bad parity and majority voting the rest generating a parallel output to a data register. Upon filling the data register it issues a completion signal to the control logic. If this is input data to the computer it will be either stored in memory or directed to the processor as interrupts. To read the data into memory a portion of the register receiving inputs from the data address lines is used as the memory address and a write command issued to the memory. An interrogate interrupt signal is issued to the special code recognition logic if the data are interrupts. The data register will be written into memory. If the input data are interrupts the special code recognition logic will extract the interrupt bits from the data register and transmit them to the CPU. If the control logic determines that the computer is to supply data to the data bus it releases the address from the parity verification and sync logic output register to the memory address lines and issues a read command to memory. The output data from memory is read into a register.

The control logic upon receiving a completion signal from the memory issues a command to the parity generation logic which shifts the output register a bit at a time into the parity generation circuitry causing parity to be generated and a serial output directed to the data bus data lines.

Figure 6-16 is a block diagram of the centralized multiprocessor system. In this system the I/O units and central processor units are capable of operating into and out of any memory. In normal operation with no failures each memory will be connected to a different CPU and a different I/O section making the operation appear as four independent computers each doing the same operation in synchronism. In the event of a failure the outputs of a failed section will be removed from the system by the switching networks shown and its outputs replaced by an identical operating section. For example if a CPU fails the memory normally fed by the failed CPU will be fed by one of the other operating CPU's. This means that the other selected CPU will feed two memory's simultaneously. Since the computers are synchronized the memory will be fed the identical data that it would have normally received from its own CPU had it not failed. This keeps the memory continously updated and available as a replacement if some other memory fails. The switches shown are the same used in the centralized linked unit system. A switch must be mechanized on each line carrying data from a computer section including each line of a parallel data transfer. The advantage of this mechanization is a possible large increase in reliability over the centralized linked unit system. If P_I, O_m, and P_c are the failure probabilities of the I/O section, memory, and CPU respectively (and it is assumed that the switches have very low failure probabiliities in comparison to the computer section failure probabilities) then the probability of total

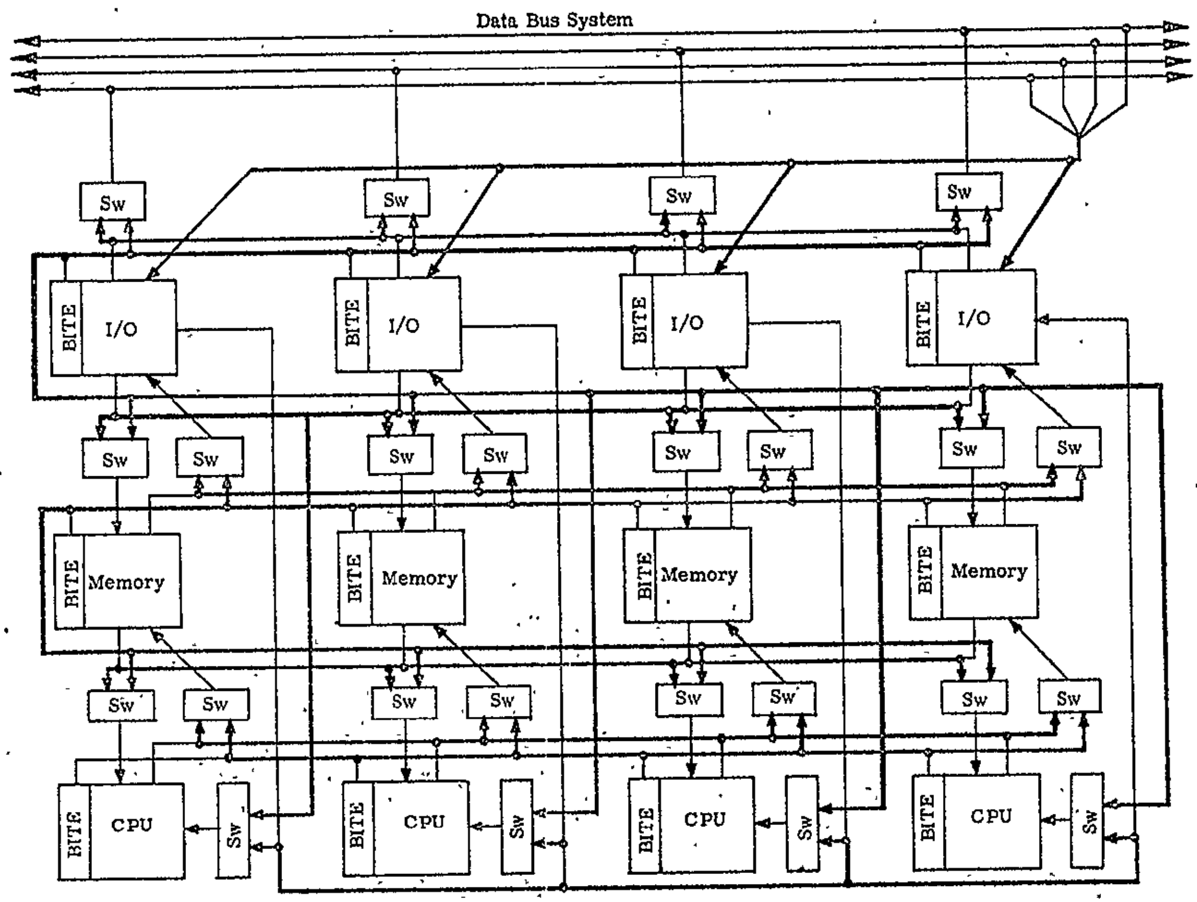


Figure 6-16 Centralized Multiprocessor System

system failure for the centralized linked unit system to

$$P_f = (P_I + P_m + P_c)^4$$

and for the centralized multiprocessor system

$$P_f = P_I^4 + P_m^4 + P_c^4$$

If the failure probability for each section is the same, i.e., $P_I = P_m = P_c = P$ then the centralized linked unit system failure probability is

$$P_f = 81P^4$$

and the centralized multiprocessor system failure probability is

$$P_f = 3P^4$$

This shows that under the condition of identical failure rates on the memory, I/O section and CPU, the centralized multiprocessor system will be 27 times as reliable as the centralized linked unit system. If the failure probability for one of the sections (e.g., the memory) is much greater than the other two failure probabilities the reliability of the two systems will be nearly identical.

Figure 6-17 is a block diagram of one section of a decentralized linked unit processor. Each unit is similar to a centralized linked unit systems, the difference being in the size of the computer, those used in the decentralized linked unit system are much smaller, and in an addition a direct interface between the computer and subsystem equipment. These computer units are located in the vicinity of subsystems equipments allowing direct electrical connection with those subsystems. The number of units is dependent upon the size of the computer selected and total computational task. The various units will communicate with each other through the data bus system. One of the results of this mechanization is an overall reduction in data bus data rate requirements.

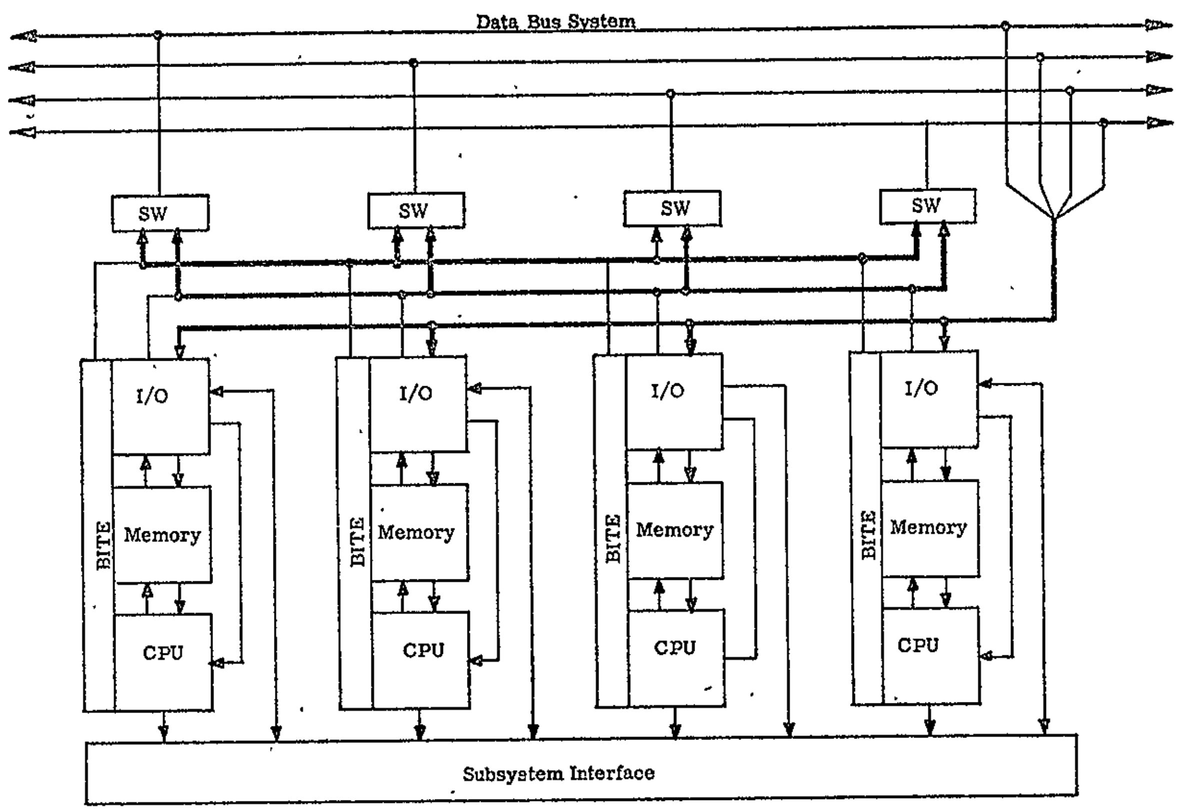


Figure 6-17 Decentralized Linked Unit System