

MARCH 1971

MDC G0985

N71-25922

NASA-CR-103171

**ASTRONAUT/COMPUTER
COMMUNICATION STUDY**

Final Report

MCDONNELL DOUGLAS ASTRONAUTICS COMPANY





**ASTRONAUT/COMPUTER
COMMUNICATION STUDY**

**Contract NAS 825701
Final Report**

MARCH 1971

MDC G0985

Prepared for the National
Aeronautics and Space Administration
Huntsville, Alabama under
Contract NAS8-25701

MCDONNELL DOUGLAS ASTRONAUTICS COMPANY-WEST

5301 Bolsa Avenue, Huntington Beach, CA 92647

PREFACE

This report documents the results of the Astronaut/Computer Communication Study prepared for the MSFC in compliance with the requirements of Contract NAS8-25701.

If additional information is required, please contact any of the following McDonnell Douglas or NASA representatives:

Mr. R. R. Joslyn, Project Manager
Huntsville, Alabama
Telephone: (205) 881-8640 or 881-0611

Mr. H. E. Pitcher, Program Development Director
Southeastern Region
Huntsville, Alabama
Telephone: (205) 881-0611

Mr. L. G. Neal, Contract Negotiator/Administrator
Huntington Beach, California
Telephone: (714) 896-4821 or 896-2794

Mr. B. C. Hodges, Project COR, S&E-COMP-C
Marshall Space Flight Center, Alabama
Telephone: (205) 453-1385

CONTENTS

	LIST OF FIGURES	vii
	LIST OF TABLES	ix
Section 1	INTRODUCTION AND SUMMARY	1-1
	1.1 Background	1-1
	1.2 Study Objectives	1-2
	1.3 Scope of Study	1-2
	1.4 Study Approach	1-2
	1.5 Study Results	1-3
Section 2	PHASE A STUDY	2-1
	2.1 Phase A Task 1	2-1
	2.1.1 Spacecraft	2-2
	2.1.2 Aircraft	2-21
	2.1.3 Task Summary	2-30
	2.2 Phase A Task 2	2-31
	2.3 Phase A Task 3	2-35
Section 3	PHASE B STUDY	3-1
	3.1 Phase B, Task 1 - Structured Vocabulary	3-2
	3.2 Phase B, Task 2 - Software Operating System	3-7
	3.2.1 System Characteristics	3-8
	3.2.2 Functional Definition	3-14
	3.3 Phase B, Task 3 - Minimum Hardware Configuration	3-20
Section 4	CONCLUSIONS	4-1
Section 5	BIBLIOGRAPHY	5-1
Appendix A	FUNCTION CATEGORIES	A-1
	A.1 Mission Control	A-1
	A.1.1 Mission Planning	A-4
	A.1.2 Mission Monitoring	A-5
	A.1.3 Mission Commanding	A-6
	A.2 Data Management	A-6
	A.2.1 Data Acquisition	A-9
	A.2.2 Data Storage Functions	A-9
	A.2.3 Data Dissemination	A-10
	A.2.4 Data Retrieval	A-10

A.3	Communications	A-11
	A.3.1 Message Transmission	A-11
	A.3.2 Message Reception	A-14
A.4	Flight Control	A-15
	A.4.1 Trajectory Flightpath Analysis	A-15
	A.4.2 Trajectory Change Implementation	A-17
	A.4.3 Trajectory Change Evaluation	A-17
A.5	Guidance and Navigation	A-18
	A.5.1 G&N Measurement Determination	A-20
	A.5.2 G&N Data Reduction and Computation	A-20
A.6	Experiments	A-21
	A.6.1 Initiate	A-23
	A.6.2 Execute	A-23
	A.6.3 Terminate	A-24
A.7	Maneuver Management	A-24
	A.7.1 Preparation	A-26
	A.7.2 Execution	A-26
A.8	Operational Status	A-27
	A.8.1 Crew	A-29
	A.8.2 Vehicle	A-29
A.9	Mission Independent Crew Functions	A-30
	A.9.1 Training	A-30
	A.9.2 Recreation	A-32
Appendix B	TYPICAL EXPERIMENT OPERATIONS	B-1
Appendix C	SPACE PROGRAM INTERACTIVE FUNCTION COMPARISON	C-1
Appendix D	PHASE B STUDY, ASTRONAUT/COMPUTER COMMUNICATIONS ANALYSIS	D-1
	D.1 Onboard Operating System Considerations for Interactive Communications	D-3
	D.2 Astronaut/Computer Communication Interface	D-5
	D.2.1 Operational Environment	D-10
	D.2.2 Processing Environment	D-18
	D.3 Requirements	D-21
	D.4 System Configuration	D-31
Appendix E	CREW/COMPUTER COMMUNICATION'S VOCABULARY	E-1
	E.1 Vocabulary Structure	E-1
	E.2 Vocabulary Method of Usage	E-4
	E.3 St. Louis Vocabulary Test	
Appendix F	TECHNICAL DISCUSSION OF THE SYNTAX DIRECTED COMPILER	F-1
	F.1 Approach	F-2
	F.2 Parsing Philosophy	F-5
	F.3 Implementation	F-6
	F.4 Summary	F-10
Appendix G	USING THE SYNTAX DIRECTED COMPILER	G-1

FIGURES

<u>Number</u>		<u>Page</u>
2-1	Gemini Computer Communications Equipment	2-3
2-2	Display Keyboard (DSKY)	2-8
2-3	Display Indicators	2-9
2-4	Apollo Telescope Mount (ATM)/Skylab Control Panel	2-16
2-5	Space Shuttle Cockpit Configuration	2-18
2-6	DMS Block Diagram - Baseline Computer Configuration	2-20
2-7	C-5 MADAR System Components	2-24
2-8	MADAR C5A Crew Control Panel	2-26
2-9	AWACS On-Board Operators Console	2-27
3-1	Vocabulary Breakdown & Development Chart	3-5
3-2	Display Software Interface	3-9
3-3	Onboard System Functional Flow	3-15
3-4	Remote Graphic Display Terminal	3-21
3-5	Experiments (Deck 2)	3-22
3-6	General Purpose Laboratory (Deck 4)	3-23
A-1	Vehicle Subsystems By Function Category	A-2
A-2	Mission Control Function Breakdown	A-3
A-3	Secondary Command and Control Center	A-7
A-4	Data Management Function Breakdown	A-8
A-5	Communications Function Breakdown	A-12
A-6	Flight Control Function Breakdown	A-16

<u>Number</u>		<u>Page</u>
A-7	Guidance & Navigation Function Breakdown	A-19
A-8	Experiments Function Breakdown	A-22
A-9	Maneuver Management Function Breakdown	A-25
A-10	Operational Status Function Breakdown	A-28
A-11	Mission Independent Crew Function Breakdown	A-31
B-1	Operating Procedure Flow Diagram	B-1
D-1	Study Approach	D-4
D-2	Astronaut/Computer Functional Interfaces	D-6
D-3	Onboard Spacecraft Operating System - Interactive Communications	D-32
D-4	Systems Status Display Monitor Station Panel	D-35
D-5	Systems Status Display Monitor Station Panel 2	D-36
D-6	Systems Status Display Monitor Station Panel 3	D-37
D-7	MPC Functional Block Diagram	D-38
E-1	Vocabulary Structure Breakdown	E-2
E-2	Function Categories	E-5
E-3	Space Shuttle Cockpit - Category Select	E-6
E-4	Space Shuttle Cockpit - Function Select	E-7
E-5	Space Shuttle Cockpit - Command Sequencing	E-8
E-6	Space Shuttle Cockpit - Data Update	E-9
E-7	Space Shuttle Cockpit - Function Execute	E-10
F-1	Syntax Directed Compiler Flow Diagram	E-8

Tables

<u>Number</u>		<u>Page</u>
3-1	Typical Crew Computer Interaction	3-19
D-1	Space Station Crew Composition Requirements	D-12
D-2	Processing Function Comparison Chart Apollo vs Space Station	D-13
D-3	Interactive Data Requirements	D-25
D-4	Multipurpose Console Functional Elements	D-27

Section 1
INTRODUCTION AND SUMMARY

This report documents the results of the Astronaut/Computer Communications Study conducted by the McDonnell Douglas Astronautics Company for the MSFC under Contract NAS8-25701. The tasks and results of the various analyses reported in interim documents and the study conclusions are discussed in this final report.

1.1 BACKGROUND

The need for a man/machine interaction and capability requirements study is evident after a cursory review of the evaluation and current status of manned space programs. Guidance and control of future spacecraft has reached the threshold of autonomous operations. The development of larger, and more sophisticated spacecraft with the capability of extended earth orbital and/or interplanetary flights demands that more mission and spacecraft control decision capability be on board. The physical size and payload capability of future spacecraft will increase the number of vital functions to be monitored. The cost of providing ground control and continuous mission monitoring of extended space flights can become prohibitive. More autonomous operational capability of future spacecraft appears to be the most cost-effective and practical approach.

Each spacecraft will conduct mission-peculiar experiments that require onboard data reduction and analyses. The mission timeline will depend on conclusions drawn from these data. Therefore, this information must be readily retrievable in a form or combination of forms that can be rapidly interpreted by the crew.

Immediate action in emergencies will require computer diagnoses to assist the crew in determining the proper response. The reaction and decision time on any space mission is critically short and therefore requires the computer to

be an integral part of the spacecraft subsystems. This computer must be programmed to automatically control and monitor the spacecraft functions and must possess the attributes of speed, accuracy and ease of communication. It must be integrated into the information system so that decision data can be effectively displayed in a form which can be analyzed instantaneously. Conversely, crew action in responding to alarm situations must be timely and with minimum translation. Therefore, the information system must accept commands and data in a syntax which conforms to the crew's vocabulary and natural way of operating rather than that of the computer. Without these inherent features, it would be difficult for the spacecraft commander to exercise his authority and responsibility.

1.2 STUDY OBJECTIVES

The primary objective of this research effort was to develop advanced crew/computer communication methods for manned spacecraft using graphic displays. Special emphasis was placed on formulating a structured vocabulary and to devising new methods of establishing crew/computer interactive techniques.

1.3 SCOPE OF STUDY

The study considered anticipated crew functions to be performed onboard manned space vehicles within the time frame of 1973 to 1980. Techniques developed during this study are intended to be used with the state-of-the-art hardware available during this period.

1.4 STUDY APPROACH

A two-phase program of analysis and design was established for this study:

Phase A--Identified and documented crew/computer communication methods on existing and near-term manned space vehicles. Methods of determining onboard crew function requirements were investigated.

Phase B--Based on the crew function requirements established during Phase A, a structured vocabulary to be utilized by the crew for ease of communication with the onboard computer was formulated.

As a starting point, existing and proposed methods of crew/computer interaction were evaluated. This resulted in identification of specific criteria to form a study baseline that established trends from which the degree of computer automation could be extrapolated. The function of each subsystem of a spacecraft was analyzed to determine the degree of manned and automatic control requirements and their interrelationships. Vocabulary requirements were formulated and general software and hardware requirements were established.

1.5 STUDY RESULTS

The Astronaut/Computer Communications Study effort achieved the following:

- A. Identified the onboard crew functions that are performed or assisted by an onboard graphic display computer system. Generically related functions were grouped into categories.
- B. Devised methods by which space vehicle crewmen can communicate function requirements with the onboard computer via a graphic-display device.
- C. Formulated a vocabulary structure conducive to crew/computer communication utilizing a graphic-display computer system.
- D. Demonstrated the feasibility of onboard technology-oriented vocabularies within the scope of the formulated structure.
- E. Recognized that graphic-display communication is an interim solution to the man-machine interface problem.

A detailed description of the efforts performed in support of this study is presented in subsequent sections. Sections 2 and 3 discuss Phase A and Phase B respectively. Section 4 presents the study conclusions. Detailed supporting data are included as Appendices.

Section 2
PHASE A STUDY

Phase A of the Astronaut/Computer Communications contract was restricted to analyzing and evaluating existing and near-term space vehicles for state-of-the-art onboard computer capability and technology development, to identifying current and proposed onboard crew functions, and to devising preliminary methods language requirements to be utilized in performing onboard crew functions. Detail tasks performed in support of this phase follow:

Task 1--Document crew-computer (onboard) communication methods on existing and near-term space vehicles. (Documented in Report No. MDC-G0388, dated May, 1970)

Task 2--Establish crew functions to be performed with an onboard computer for space station and space shuttles. Define the extent to which the crew will participate in these functions. (Documented in Report No. MDC-G0388A, dated September, 1970)

Task 3--Devise methods by which the crew can communicate with the onboard computer to perform the functions determined in Task 2 with special emphasis upon the software and hardware required to support these functions. (Documented in Report No. MDC-G0852, dated November, 1970)

2.1 PHASE A TASK 1

This task was the research and data gathering task of the study. During the early stages of research, the absence of onboard graphic display capability on historic and existing manned space vehicles became immediately apparent. For this reason, and because of the relatively young age of the manned space effort, the scope of this task was expanded to include selected representative aircraft. Rational supporting this decision is based on the larger number of on-going

aircraft projects and high-degree of onboard aircraft computer applications that are applicable to spacecraft usage (i.e., the automation trends established by aircraft manufacturers provide a convenient foundation for developing and proving spacecraft onboard computer requirements).

2.1.1 Spacecraft

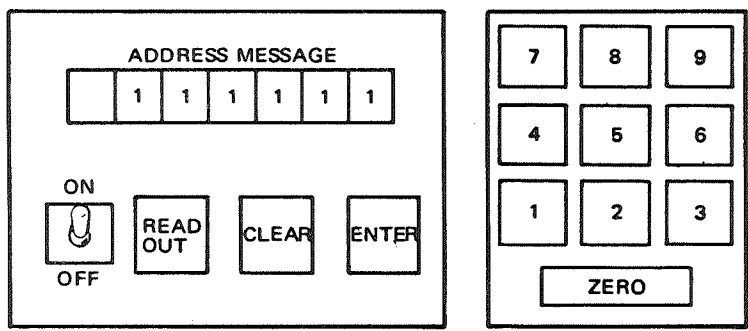
There have been only two manned-spacecraft programs with onboard computer capability requiring crew interaction, the Gemini and Apollo programs. This study was primarily interested in the Apollo system as it is still operational. Future space programs in various stages of development were analyzed during this study are the Skylab, Space Station and Space Shuttle programs.

2.1.1.1 Gemini

The Gemini was the first manned space program to utilize an onboard computer requiring astronaut interface. This was a special purpose computer designed and built to satisfy mission requirements. The primary purpose of this system was guidance and navigation and in case of serious computer malfunction, the astronaut became the sole system backup. The computer was initialized before launch with all prelaunch and ascent software. From then on the astronaut was responsible for loading all other mission phase software (i.e., rendezvous, reentry, etc.) and entering the appropriate data parameters required to initialize processing routines (see Figure 2-1 for computer control equipment).

The Gemini onboard computer was an IBM 4K 39 bit word machine with an online magnetic tape drive, a seven digit display device, a keyboard, and a limited number of switches, lights, and meters. One-third of the computer memory was read only and housed the software executive routines and processing functions critical to mission success. The remaining two-thirds of the core was reusable for data and/or other functions unable to be accommodated by the limited core size. Each machine word was divided into three 13 bit "syllables". The first of which was generally used for addressing, and the remaining twenty-six for data, although, usage of individual machine words was determined by the processing software. The online tape drive was a "read-only" device and

MANUAL DATA INSERTION UNIT



COMPUTER MODE SELECTOR SWITCH POSITIONS

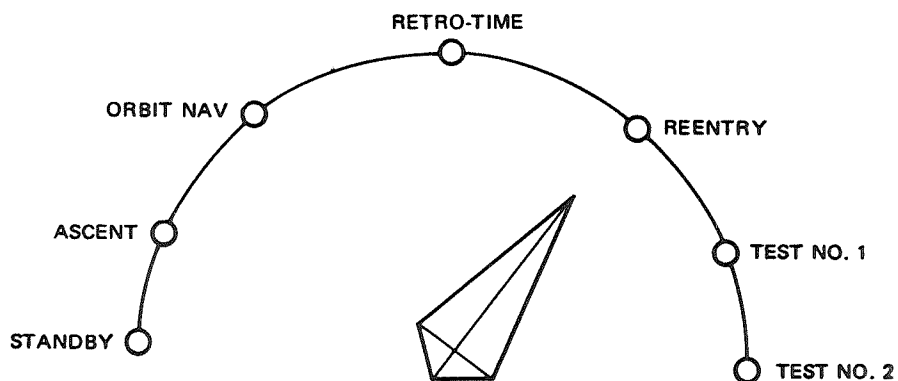


Figure 2-1. Gemini Computer Communications Equipment

accommodated a 550-foot magnetic tape. This tape provided auxiliary storage for software which was not required to be resident at all times during the mission.

All software coding for the Gemini computer used sixteen machine language instructions. Coded programs were assembled and checked out on an IBM 7094. When programs were sufficiently "checked-out", they were ready for loading into the onboard computer system. This was done by punching the programs onto mylar tape, and subsequently loading the data on the magnetic tape or into the onboard computer system via an intermediate hardware process.

The seven window digital display was used in conjunction with a ten digit numeric keyboard. Almost all communications by the astronaut with the computer were via the keyboard, and conversely, almost all communications made by the computer with the astronaut were via the digital display. The first two digits of the display indicated an address or function and the remaining five were meaningful associated data. The following is an example of a typical communication sequence:

- Computer would stop processing and blink the START computer light to designate action required by the astronaut.
- The astronaut would determine the action required by interpreting the codes in the display window.
- If a software routine were required from tape, the astronaut would enter a memory address in first two digits, a tape position number in the third, and the program number in positions 4 - 7 of the digital display. The astronaut would also select a mode on the mode select indicator.
- If data entry was required, the astronaut would enter the required data via the digital keyboard unit, and visually verify the data displayed on the digital display unit.

- In any case, the astronaut depresses the computer start button after performing the required action.
- The computer would then display the results of the action taken to assure the astronaut that he did communicate, and perform action required.

2.1.1.2 Apollo

Apollo vehicles have four onboard computers: one in the launch vehicle, one in the command module, and two in the lunar module. The computer in the launch vehicle requires no direct astronaut/computer communication except in the case of a malfunction. The computer in the command module and one of the computers in the lunar module are there for the same purpose: to support the primary guidance, navigation and control system (PGNCS) of their respective modules. These are the computers which will primarily be discussed. The remaining computer supports the abort guidance system in the lunar module. The launch vehicle digital computer (LVDC) is the major component of the Saturn V launch vehicles (SVLV) guidance system. The LVDC continually calculates vehicle attitude and generates guidance commands for the flight control computer (FCC) (an analog computer). The FCC controls the thruster position of the three SVLV stages. The LVDC system is responsible for guiding the Apollo spacecraft through the ascent, earth orbit insertion, earth parking orbit, and translunar injection phases of the mission. The LV computer also maintains attitude control for the "hook-up" of the lunar module with the command module before commencing the translunar voyage.

Should the LVDC fail, the astronaut with the aid of the command module computer (CMC), the flight director attitude indicator, and a manual attitude rotational control would manually fly the vehicle to translunar injection. When operating in this mode all inputs generated by maneuvering the rotational control are routed by the CMC directly to the FCC for subsequent thruster positioning. At all times during this operation, the CMC displays for the astronaut current inertial velocity, attitude (H) and attitude rate (\dot{H}) calculations which are needed in determining the manual rotational control position.

Should the flight control computer fail prior to translunar injection, the mission must be aborted.

The PGNCS of the command and lunar modules encompass three major subsystems: (1) the inertial subsystem (ISS), (2) the optical subsystem (OSS), and (3) the computer subsystem (CSS). The PGNCS' have been so designed that in the event of a subsystem failure the remaining operational subsystems will continue to function. The functioning subsystem with the aid of the astronaut who, by use of the onboard display and keyboard (DSKY), can perform all tasks necessary to guide and control the vehicle.

The onboard computing hardware for the two modules is nearly the same. The basic difference being: (1) core requirements for software storage as required by difference in missions being performed, and (2) the command module has two DSKY's.

Raytheon manufactures the Apollo computers which have a core size of 38,912 words of which 36,864 are ROPE memory and the remaining 2,048 are scratch pad memory for computer calculations. Memory word size is sixteen bits.

NOTE

ROPE core is the term applied to core which has been generated by actually wiring the binary representation of a software routine, a constant, a table, etc., into memory. The wiring gives the appearance of rope going in and around each bit. The memory produced using this technique is read only and can only be changed by rewiring.

Unlike the Gemini computer system, Apollo has all software routines necessary for mission accomplishment "hard-wired" into core. There is no auxiliary storage associated with the Apollo onboard computers (i.e., magnetic tapes, disk, drums, etc.).

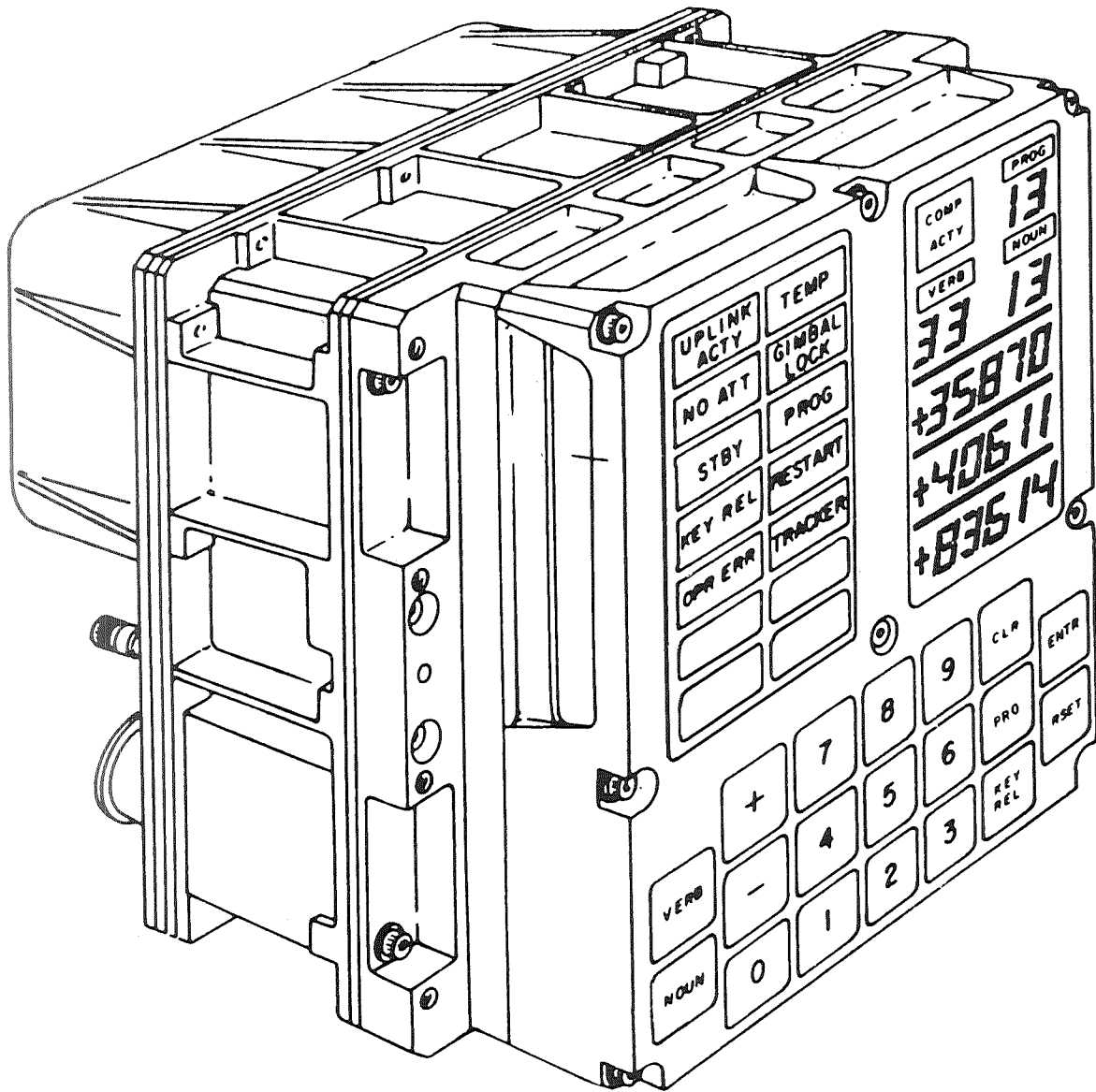
Communication devices for the onboard Apollo computer system consists of: (1) a display panel which is composed of: (a) three digital segmented light register, (b) a two digit program register, (c) a two digit noun register,

(d) a two digit verb register, and (2) a panel of ten condition indicators; and (3) an electroluminescent keyboard consisting of: (a) ten numeric keys (0-9), (b) two sign keys, and (c) seven function keys. The display panel and keyboard are shown on Figures 2-2 and 2-3.

Display panel description and usage:

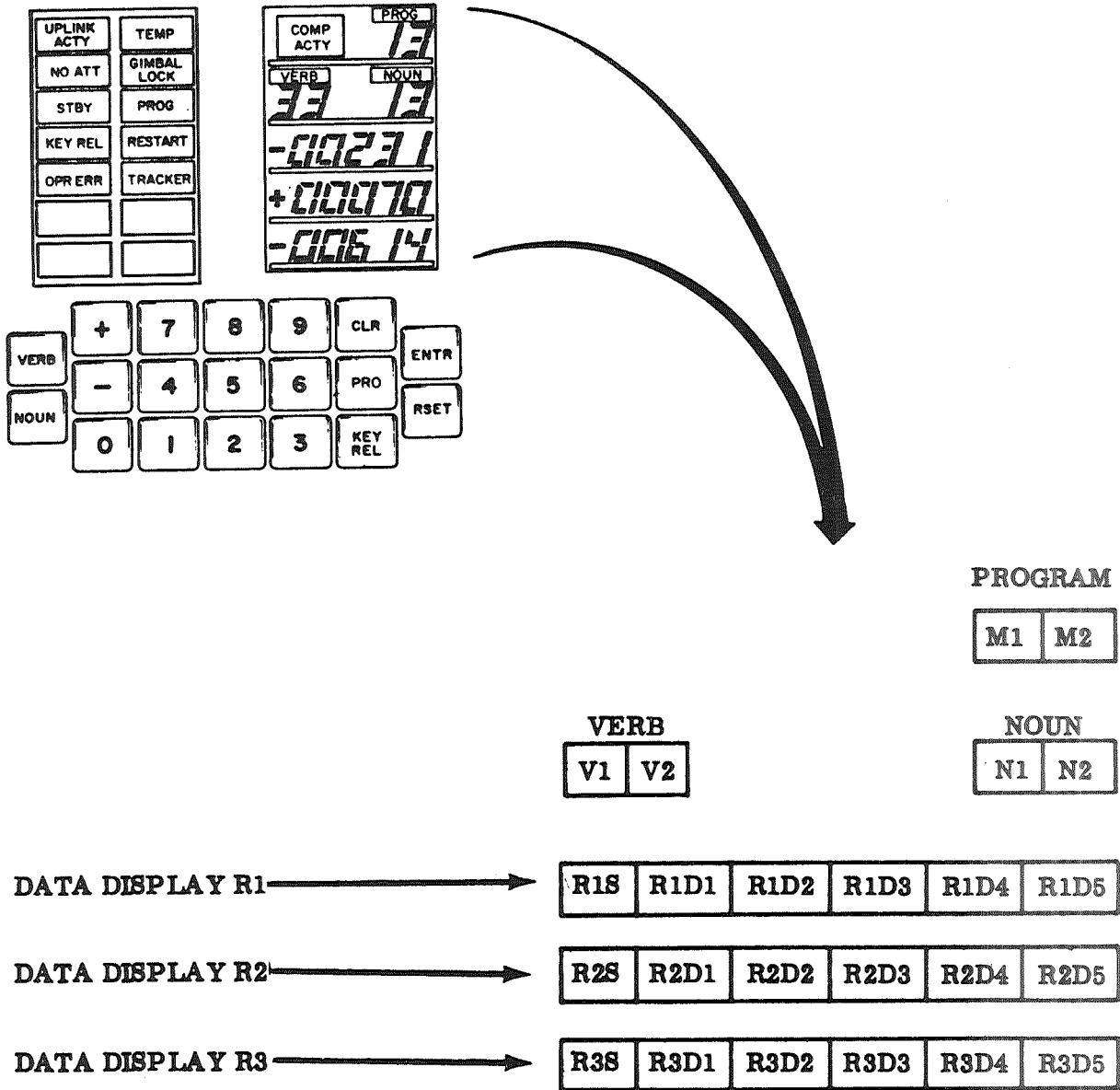
- A. The three segmented light display registers indicate data entered at the keyboard, or a command by the computer and the sign associated with this numerical data if it is in decimal.
- B. Program 2 digit segmented light indicates program being processed.
- C. Noun 2 digit segmented light indicates noun code entered at the keyboard or command by computer.
- D. Verb 2 digit segmented light indicates verb code entered at the keyboard or command by computer.
- E. Panel of 10 condition indicators

Uplink act	Indicates uplink data being received
No att	Indicates that the ISS is not suitable for use as an attitude reference
Stby	Indicates computer is in the standby condition
Key rel	Indicates that the internal program attempted to use the DSKY and found it busy
Opr err	Indicates that the DSKY program has encountered some improper operating conditions



FROM: AC ELECTRONICS DIV. GEN. MOTORS SM-2100
AND SATURN V FLT MANUAL, SA-509

Figure 2-2. Display Keyboard



FROM: AC ELECTRONICS DIVISION
GENERAL MOTORS SM-2100

Figure 2-3. Display Indicators

Temp	Indicates that the stable member temperature has exceeded its design limits
Gimbal lock	Indicates that the middle gimbal has driven through an angle greater than 70 degrees from its zero position
Prog	Indicates that a program check has failed. This indicator is controlled by a computer program
Entr	Informs the computer that the assembled data is complete and/or to execute the desired function
Rset	Extinguishes the status lamps that are controlled by the computer
Pro	Causes the computer to go to standby mode when depressed the first time if channel 13, bit 11 = 1. An additional depression can see the computer to resume operation. Normally this key will be used to signify proceed without DSKY inputs to the computer.

The DSKY is where the majority of the astronaut/computer communication is accomplished. The six uses of the DSKY are listed below:

- A. Loading data into the computer.
- B. Display data from the computer and data loaded into the computer.
- C. Monitoring data in the computer.
- D. Display the computer modes of operation.

E. System control by the initiation of subsystem and system testing and control of the systems major modes of operation.

F. Computer requested operator action.

All communication by computer and astronaut is either accomplished or prompted by the DSKY.

A. Restart Indicates that the computer was reinitialized due to a fail condition.

B. Tracker Indicates an optics CDU failure.

Keyboard description and usage:

A. Ten numeric keys (0-9) used to enter data, address codes and action request code into the computer.

B. Two sign keys (+ and -) informs the computer that the following data is decimal and indicates the sign of the data.

C. Seven instruction keys

Verb Conditions the computer to interpret the following two numerical characters as an action request code and causes the verb display to be blanked.

Noun Conditions the computer to interpret the following two numerical characters as an address code and causes the noun display to be blanked.

Clr	Clears data contained in the data registers, depressing the key clears the display register currently being used. Successive CLR's clear the other display registers.
Key rel	Releases the DSKY displays initiated by keyboard action so that information supplied by a computer program may be displayed.

The language used to communicate with the computer is an encoded "verb" and "noun" language where the verb indicates the action to be taken and the noun indicates on what the action should be taken. The computer prompts the astronaut for action by loading the verb, noun, and data registers with meaningful and related information and blinking the verb/noun display lights.

The astronaut interprets the request, enters a meaningful response in the verb, noun, and data registers, and depresses the enter key. The astronaut can also initiate computer request or enter data into the computer in the same manner as he responds to computer requested action. All communication done by the astronaut is limited to predefined functions programmed in the ROPE core.

The astronaut can also be prompted to communicate with the computer when the condition indicator lights on the display panel are illuminated.

Briefly, the technique used by the astronaut to communicate with the computer is to depress the verb key and enter desired two digit VERB code, and then depress the noun key and enter the two digit NOUN code. (The order of entering verbs and nouns is not critical to this operation.) He then depresses the enter key. If the computer requires more data to complete the astronaut's request, it blinks the "VERB-NOUN" lights. This indicates to the astronaut that the computer is not prepared to accept data entries. Data is then entered via the keyboard.

The "VERB-NOUN" language is composed of 99 verbs and 99 nouns including spares. The astronaut has a guide (Cookbook) on board to which he can refer if necessary. To enter a request requires no more than seven keys be depressed. Example: If the astronaut desires to display in decimal the time to perigee he would depress the following seven keys:

- | | | |
|----------|---|-----------------------------|
| A. Verb | | Next two digits = verb code |
| B. 0 | } | Verb 06 "Display Decimal" |
| C. 6 | | |
| D. Noun | | Next two digits = noun |
| E. 3 | } | Noun 32 "Time to Perigee" |
| F. 2 | | |
| G. Enter | | Execute request |

If data is required, the computer blinks the "VERB-NOUN" lights. The verb and noun remain in their respective displays until the total request is completed. This indicates to the astronaut, at all times, exactly what action is expected by the computer.

Noteworthy is the fact that the ground control center has the capability of communicating with the crew through the onboard computers. Just as the computer makes a request of the astronaut by the "VERB-NOUN" technique, the ground control center can initiate a request via uplink data. This is a technique which is not frequently used, but most remembered as a desirable attribute for future onboard communication systems.

All software programming for the PGNCs computers is done by the Massachusetts Institute of Technology (MIT) Instrumentation Laboratory. Software programs are generated using the fifty-six machine language instruction set of the

Raytheon computer, but checkout is accomplished through simulation on an IBM 360/75. Prior to obtaining an IBM 360/75 MIT simulated software testing on a Honeywell 800. Software to be used on a specific flight is required to be operational three months prior to flight. This time is required to allow Raytheon adequate time to "ROPE" wire and check out the computer to be used. Software checkout continues until the vehicle is flown.

The lunar module abort guidance system centers around a small four-K, TRW "man-rated" computer which monitors the PGNS. In the event it becomes necessary to abort the lunar mission, the lunar module abort guidance system can take control of the vehicle and guide it to the predefined rendezvous with the command module.

This system has a data entry display assembly which functions much in the same manner as the DSKY of the PGNS. The encoded language used in communicating with this system is not as sophisticated as the "VERB/NOUN" language, but the communication technique is the same.

2.1.1.3 Skylab

The Skylab project is an extension of the Apollo program. The immediate purposes of Skylab mission are: (1) to conduct medical experiments associated with extended manned space flights, and (2) to conduct a number of solar astronomy experiments. The solar astronomy experiments are of interest to this study as they are performed utilizing the Apollo telescope mount (ATM) that directly interfaces with the onboard digital computer which requires crew communication.

The Apollo telescope mount digital computer (ATMDC) system centers around two IBM 4-pi TC1 computers. Only one computer is operational at a time, and should the operational computer detect a failure, as a result of self-testing, it will automatically power-up the backup computer and switch over. There is no auxiliary storage associated with this system, and all software required for the mission is loaded into memory prior to launch. The communications link between the astronaut and the ATMDC is provided by the digital address system (DAS).

Astronaut/computer communication is accomplished by entering a four character octal code in the DAS keyboard. (The DAS keyboard consists of numeric keys 0 - 7, a clear key and an enter key - see Figure 2-4). Momentarily the four digit octal code appears in the DAS Orbital/Phase Time Remaining display for visual verification. If the astronaut verified the command, he depresses the enter key and the designated function is performed. Currently there are approximately ninety-five octal codes, including spares, which may be used to communicate with the ATMDC. Some of the available commands cause the ATMDC to interrogate switch settings on the control and display (C&D) panel. These commands may require the astronaut to preset switches before entering the command. Capabilities of the communication system include having data displayed from memory on one of three counters located on the monitor sector of the C&D panel. The astronaut may also have data displayed on these counters by selecting a desired data reading on the counters respective twelve position rotary switch.

All communication with the ATMDC is initiated by the astronaut. The computer system has no method of requesting data from the astronaut or indicating required action to be performed by the astronaut.

Example:

To execute the command "update time from midnight," the astronaut performs the following functions:

- A. Using DAS Keyboard, the astronaut enters the code "J2047" (J represents the address code of the ATMDC and the octal code 2047 represents the request "update time from midnight.")
- B. Visually verified the code when displayed on the DAS orbital/phase time remaining display.
- C. Depresses the enter key.
- D. Enters the two data commands in the same manner as the original command.

COMPOSITE OF SHEETS I THRU 10
OF ICD 40M3787I REV. J
ATM C&D PANEL CONFIGURATION
20 JAN, 71

LEGEND (SHEETS 1 THRU 10)
POSITION CODE
MOMENTARY
LASTING LOCKING SWITCH ONLY
SYMBOL AS L NOT TO BE INCLUDED
ON PANEL

FLAG INDICATIONS
COLOR - GRAY
NORMAL, NOT INHIBITED, ON, OPEN
COLOR - WHITE
ABNORMAL, NOT COMPLETE
COLOR BLACK & WHITE
DEACTIVATED, INHIBITED, OFF, CLOSED
WHITE LETTERS ON
GRAY BACKGROUND
INDICATION (ACS) TO BE BLACK LETTERS
ON WHITE BACKGROUND (2 PLACES)

INDICATOR LIGHT COLORS
PER MIL-C-25050, TYPE 1
METER FAILURE - YELLOW
ALERT - BLUE
PANEL COLOR
BACKGROUND - GRAY PER FED-STD-595 COLOR 3623H
NOMENCLATURE - WHITE COLOR 37875 ELECTROLUMINESCENT EXCEPT MON1, MON2, ON
PANELS 788, AND UPPER DRWG, RING, L/OG ON MPC ARE
ENGRAVED AND FILLED WITH WHITE PER FED-STD-595 COLOR 37875
STYLE OF LETTERS & NUMERALS GORDON NORMAL (ENGRAVED)
STYLE OF LETTERS & NUMERALS FUTURA DEMI BOLD (ELECTROLUMINESCENT)
BORDER & GRID - ATM POWER OFF BORDER
LINES TO BE FLUORESCENT RED, VELVA-GLO RED BY RADANT COLOR CO
EXPERIMENT AND COMPONENT GROUPING LINES, BOTH
VERTICAL & HORIZONTAL WHITE PER FED-STD-595 COLOR 37875
SWITCH CHANNEL GROUND/DAS GRID CMD ONLY WHITE PER FED-STD-595 COLOR 37875.

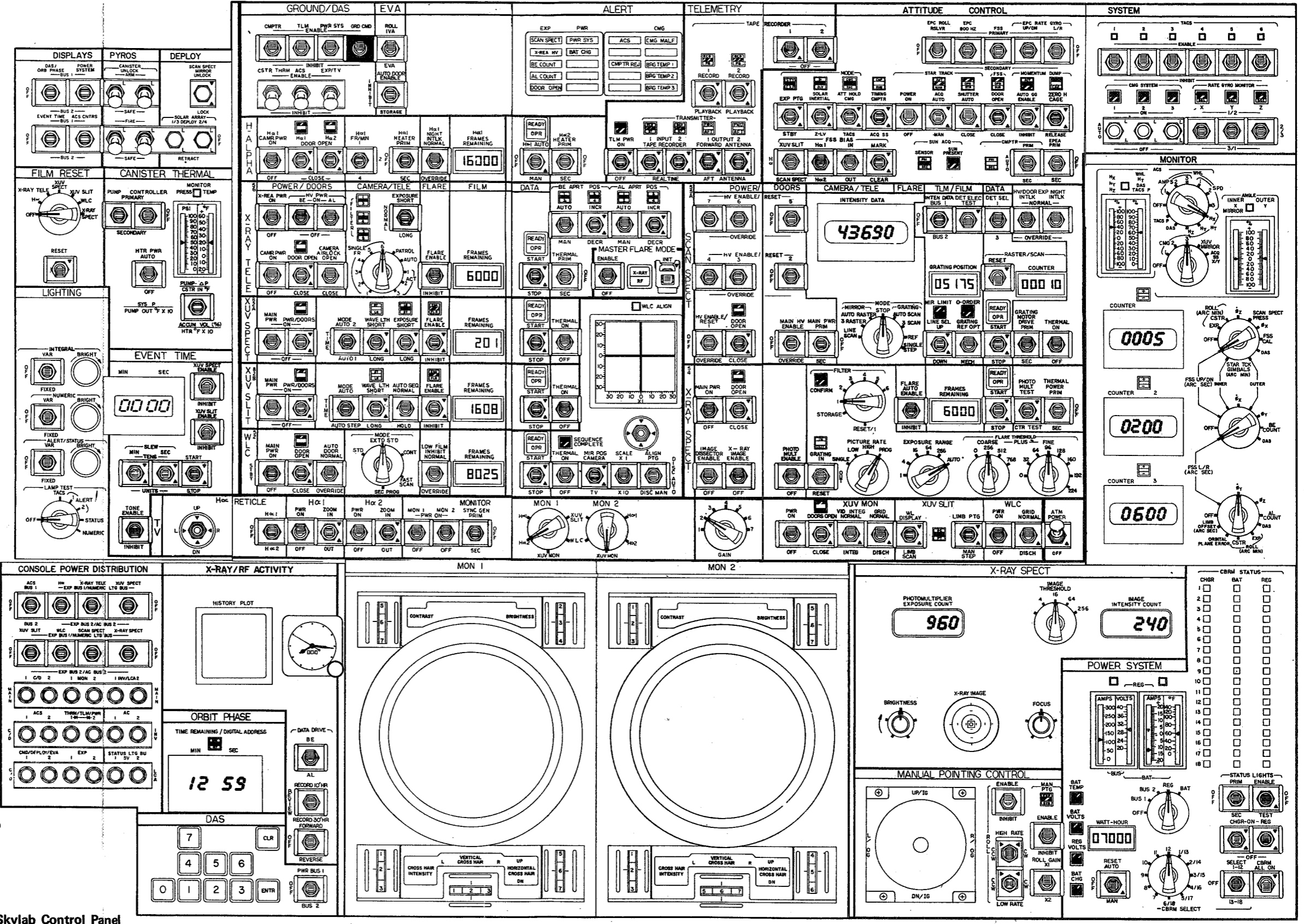


Figure 2-4. Apollo Telescope Mount (ATM) Skylab Control Panel

As alluded to by the character J, other hardware can be addressed from the DAS Keyboard. The extent of this capability could not be determined within the time constraints of this portion of the study.

2.1.1.4 Space Shuttle

The Space Station and Space Shuttle programs are the two most progressive space programs ever to be undertaken. In keeping with this theme, the engineers designing these vehicles are incorporating the latest hardware technology advances. Some of the hardware components being proposed have not been fully developed. An example of this revolutionary design can be found in the cockpit of a shuttle vehicle.

Gone are the familiar rows of gages, meters and switches, and in their place are found computer keyboards, multipurpose cathode ray tube display devices and a head-up display (HUD) (Figure 2-5).

This complex equipment is all part of the most sophisticated integrated avionics systems to be developed for an airborne vehicle. The proposed minimum for this system is a 64K 32 bit general purpose computer, and due to the importance of the computer on the shuttle vehicle it has quadruple redundancy. One computer will be operational at all times and during critical periods all four will be operational with one in command.

The computer system will support all vehicle subsystems with one of its primary functions being onboard checkout and fault isolation. Operating in this mode, crewmen can isolate and correct subsystem malfunctions to the line replaceable unit (LRU) level. This is one of the features that permits the proposed shuttle to be ready for launch in hours instead of months. Other major subsystems which the computer will support either by monitoring or controlling functions are (1) the Guidance and Navigation Subsystem, (2) the Environmental Control and Life Support Subsystem, (3) the Electrical Power Subsystems, and (4) the Flight Control Subsystem.

The mode of crew communication with the onboard shuttle computer system has been defined as requiring a keyboard and a cathode ray tube (CRT) display. The

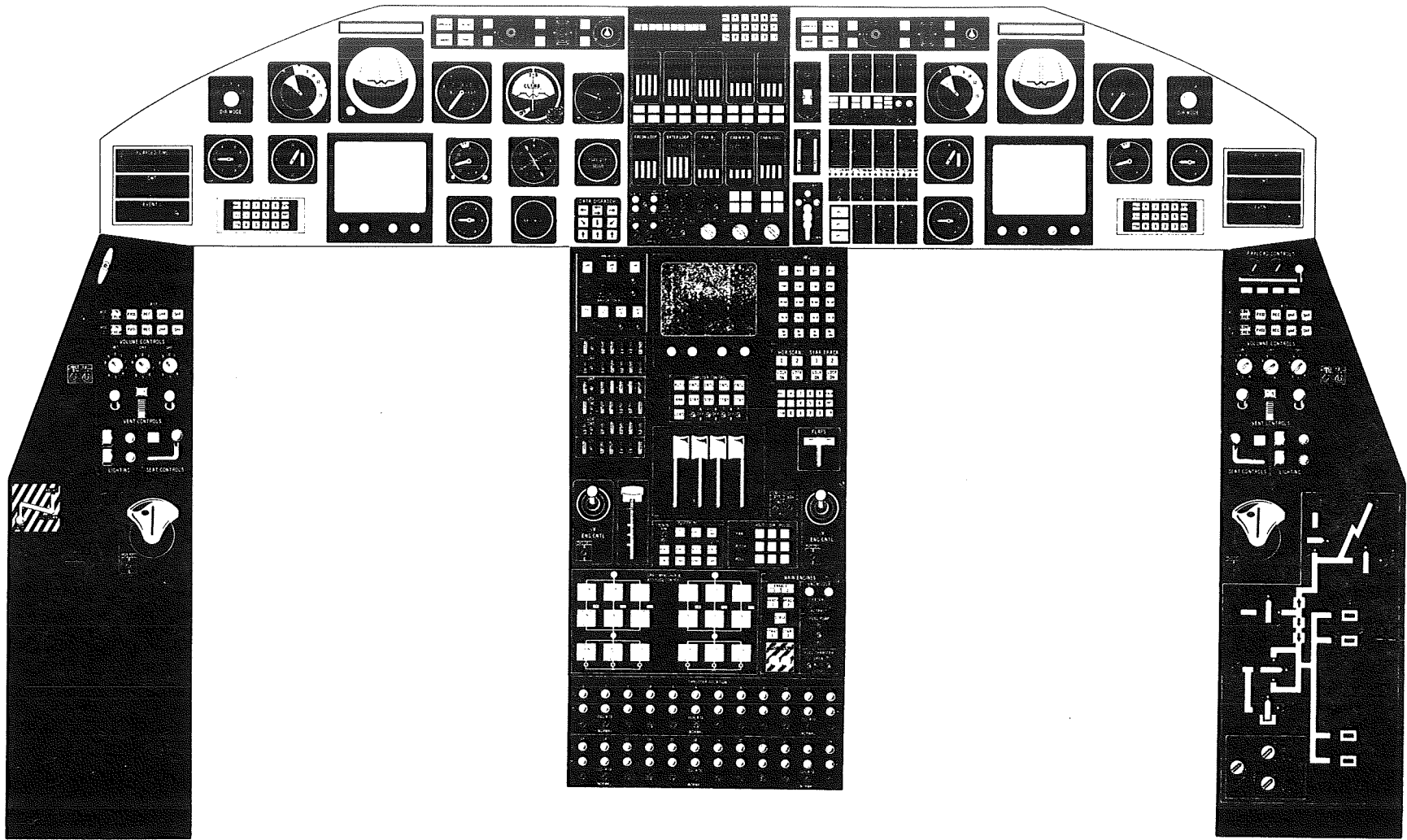


Figure 2-5. Space Shuttle Cockpit Configuration

keyboard will be alphanumeric with function keys, and the CRT display will be multipurpose (i.e., supports hardware other than the keyboard). Several of these devices will be strategically located onboard.

The communications language for this system will be a man/machine vocabulary to be defined in detail and developed at a future date. The Echo plast technique will be used to enter data into the computer (i.e., the character images on the CRT are generated by the computer). The use of this technique assures the crewman that the character he sees is the one the computer will interpret and he hits the enter key which is energized. This method of data entry requires more computer time, but the reliability factor is enhanced many times.

2.1.1.5 Space Station

The computation equipment consists of two multi-processor computer complexes physically located on separate decks, one for flight operations and subsystem control and one for experiment control and scheduling (Figure 2-6). The former functions may actually be performed by either complex if the need should arise due to evacuation or loss of the primary station control center. In addition simplex processors have been incorporated in dedicated areas. During assembly operation, one of the processor modules will contain the executive program and function as a master controller. It will assign tasks and equipment to the other processors as the computation work load varies. When equipment is not actually operating, it will be placed in a standby condition to conserve power. Periodically, each processor will run self-test routines to detect failures. If a failure is detected, the failed unit will be switched out of the system and replaced by another module. The crew will be notified via the Operational Checkout System (OCS) so corrective action can be initiated. An auxiliary memory provides the capability for reading a variety of stored programs into the computer on an as-needed basis for selected and/or intermittent computer operations. The programs will have been prepared in advance and kept in storage in anticipation of the above, or new programs will be generated as required either by transmission from the ground or by means of an onboard magnetic keyboard.

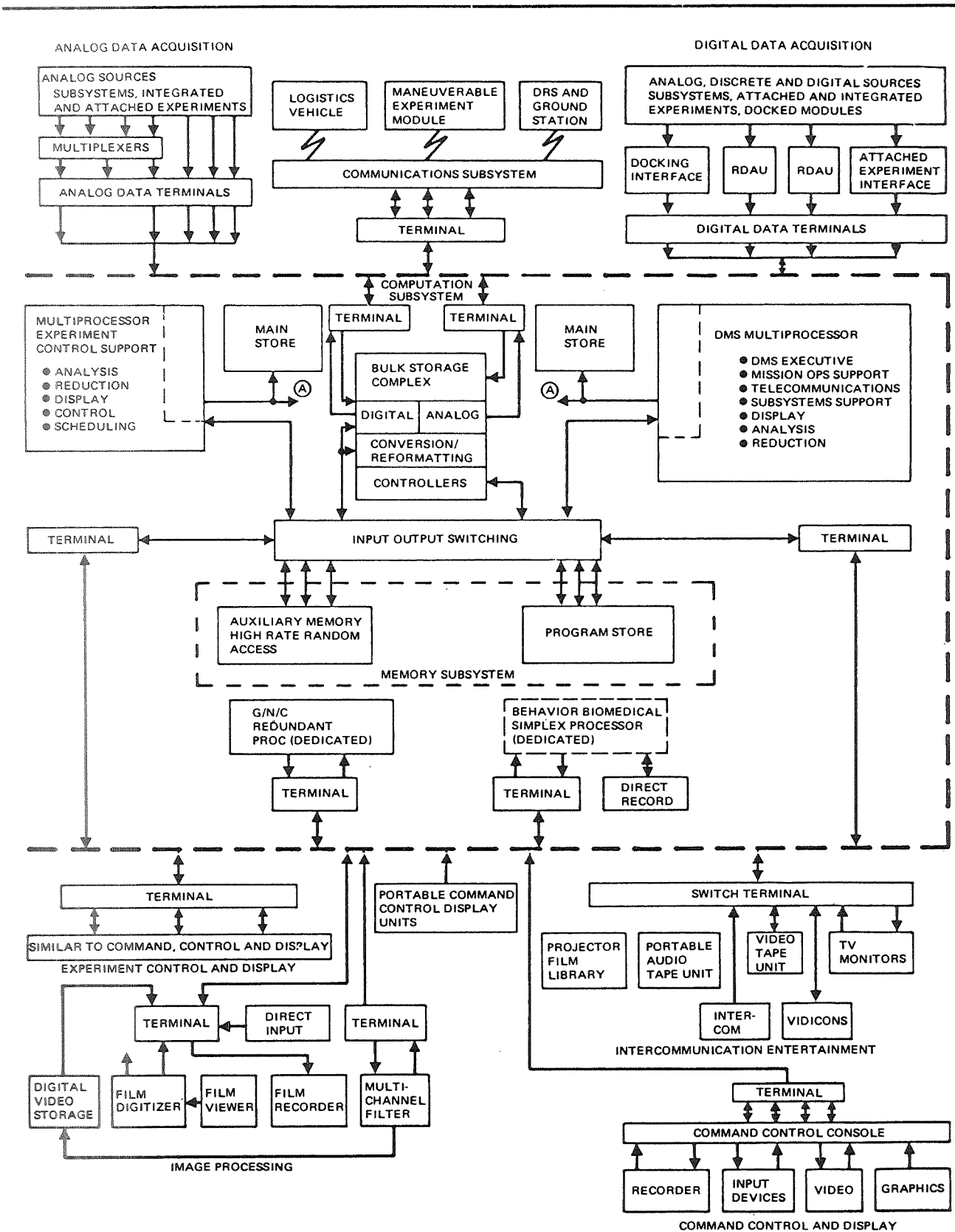


Figure 2-6. DMS Block Diagram – Baseline Computer Configuration

In addition, the Space Station will have a number of portable CRT display device with keyboard and hardcopy capability. This instrument would be capable of being remotely "plugged-into" the computer and used where needed.

As with the Shuttle, the communication language and techniques to be used in communicating with the Computers needs more definition.

2.1.2 Aircraft

The first aircraft onboard computers centered around complex and sophisticated guidance and navigation systems. As the onboard computer became established as a guidance control tool, engineers were already finding more and more applications for the onboard computer. Today, onboard computers are the major components of weapons delivery systems, onboard checkout, data management and recording systems as well as in conventional guidance and navigation systems. The two operational onboard computer systems surveyed during this effort were the C5A and F111D. The more advanced near-term onboard computer systems for the AWACS and F15 projects were also investigated.

2.1.2.1 F111D

The F111D is a tactical air command aircraft manufactured by General Dynamics. The onboard computer systems consist of two general purpose computers which support the aircraft's weapons delivery and navigation systems. Two computers are used to provide backup in the form of a selective functional redundancy for greater probability of mission success.

The two general purpose computers used on the F111D aircraft are IBM 4-pi, Model CP-2 computers. One is the general navigations computer (GNC), and the other is called a weapon delivery computer (WDC). As implied by the computer names, the GNC supports the aircraft's navigation system by performing such functions as route point sequencing, airplane steering, fix taking, fuel monitoring, data entry and display, etc. The WDC system supports the weapons system and perform system self-test. Should a system failure be detected by the self-test routine, the computer automatically switches to the backup mode.

The processing programs for the onboard systems are controlled by an executive routing which functions as an operating system and controls subprogram execution, thus providing a total software system. Crew communication with the system is accomplished by interrupts which allows the computer to display data or receive input.

The method of crew/computer communications is very similar to the one used on Gemini and Apollo (e.g., similar keyboard and display hardware are used). The electroluminescent keyboard consists of ten numeric keys (0 through 9), a clear key, and an enter key. The language is a numerically encoded system which is limited to displaying crew requested data and entering routine parameters required for subprogram processing. In conjunction with the keyboard is rotary mode selection switch for crew/computer communication. This switch indicates to the executive control software routine a subprogram to be executed. The particular subprogram may or may not require additional communication with the crew. If the selected mode requests a display, the subprogram will display data in the appropriate segmented digital display register, and if the mode calls a subprogram which needs input data from the crew, the data must be supplied via the electroluminescent digital keyboard. Data supplied is limited to the data required for the subprograms processing functions.

2.1.2.2 C5A

The C5A, the world's largest aircraft, was designed and manufactured by Lockheed Aircraft Company for the U.S. Air Force. This aircraft is designed to transport large payloads of defense materials where needed. Two onboard computer systems have been incorporated in its design, one to support onboard checkout and the other supports a conventional guidance and navigation system.

The onboard checkout system is the most publicized of the C5A computer system and is known by its acronym "MADAR," Malfunction Detection, Analysis and Recording System. The MADAR system is driven by a Nortronics 1060 eight thousand word computer. MADAR monitors approximately 1900 line replaceable

units throughout the following systems (1) propulsion, (2) environmental, (3) secondary power, (4) mechanical, (5) flight controls, (6) instrumentation, (7) MADAR, (8) radar, (9) communications, and (10) navigation. When MADAR detects an out-of-tolerance reading from a system sensor, MADAR prints a message to the operating technician stating the particulars concerning the doubtful reading (Figure 2-7).

A questionable sensor reading stimulates the operator to begin diagnosing the problem. MADAR assists in this effort by projecting the requested system diagnostic schematic on the Data Retrieval unit's screen. The Data Retrieval Unit contains over 10,000 16mm film frames of information such as troubleshooting schematics and wave-forms of system test points. The operator can "call-out" a troubleshooting schematic of a system. The schematic provides an analytical approach to troubleshooting the questionable system by indicating test points to be checked against filmed wave-forms. Proceeding through the schematic, comparing live images on the oscilloscope with healthy images on film leads the operator to the faulty line replaceable unit (LRU). After finding the bad LRU, MADAR prints on the online printer and records on the magnetic tape information which identifies the faulty LRU. The operator can then take corrective action.

Even though the MADAR system is restricted to checking system status and performing onboard, online system checkout. The method by which onboard checkout is performed is of particular interest because of the man machine interface in an online checkout mode, the utilization of an alphanumeric keyboard, and the hardcopy printer. This system will be analyzed in more detail, but lack of available data prohibits further investigation at this time.

The C5A navigation system is conventional in that all military and commercial aircraft are not being manufactured with a computerized navigation system. The C5A guidance and navigation system incorporates two Nortronic 1051A computers (one for redundancy backup) utilizing digital segmented light registers and a numeric keyboard with function keys as the communication media.

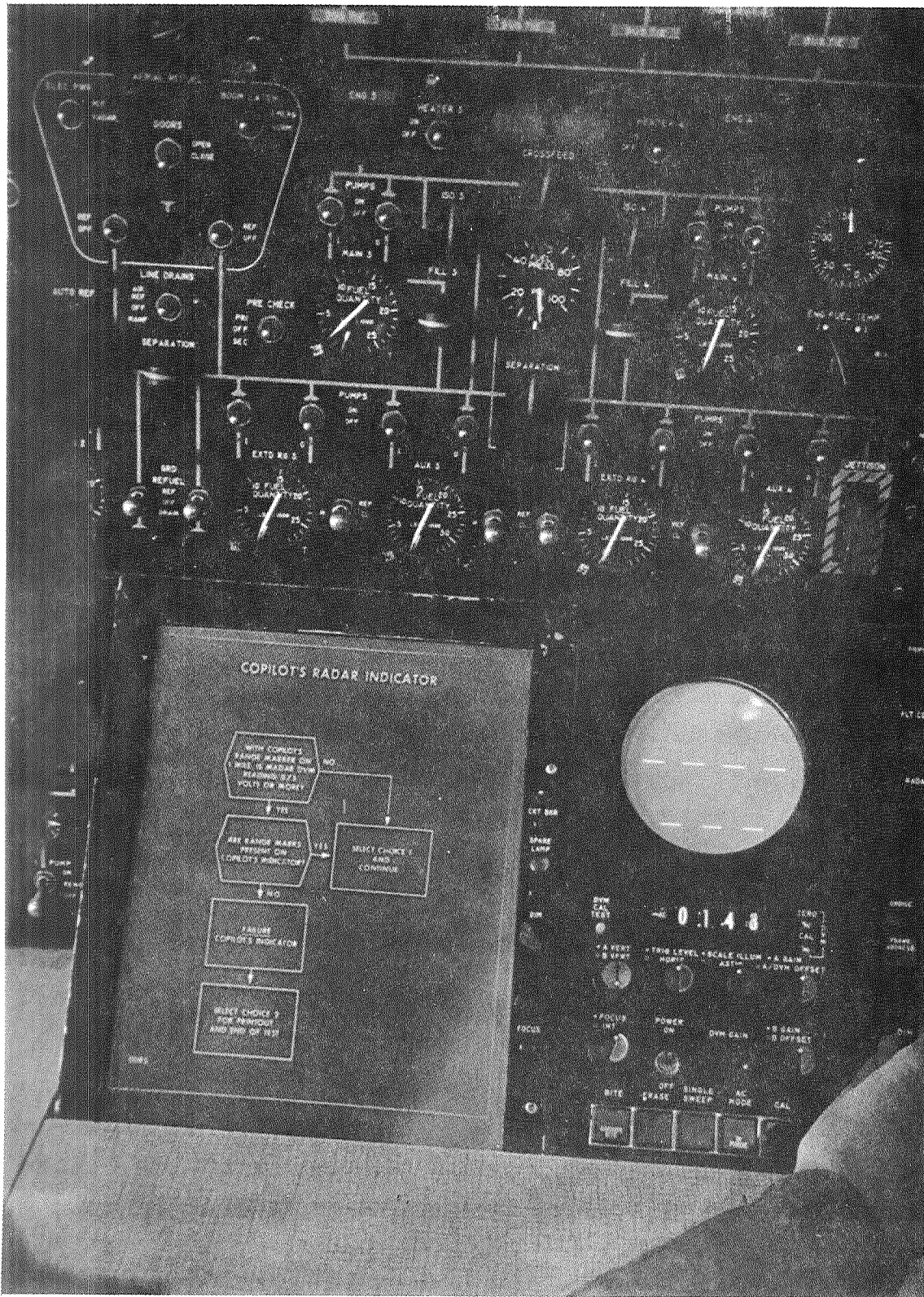


Figure 2-7. C5 NADAR System Components

Computer communication with the crew is limited to data tolerances editing of aircraft flight capabilities such as turn angles, which the computer verifies. Should a command be given which the aircraft is incapable of performing, the computer indicates an error condition to the operator. The operator, after determining the command problem, clears the incorrect command from the processing system, and enters the correct command.

The operator's keyboard is digital, containing numeric keys 0 through 9 and several function keys (North, South, East, West, Plus, Minus, Clear, and Enter). The operator indicates to the computer, via a numeric encoded language, the function to be processed and enters the required data parameters needed by the processing software. This technique is fairly standard throughout the aircraft and spacecraft industry (Figure 2-8).

2.1.2.3 Airborne Early Warning Command System

The Airborne Early Warning Command System (AWACS) command control system is an onboard command center controlled by digital and analog computers utilizing integrated display function consoles. The onboard AWACS computers are multiprocessors responsible for acquisition, reduction, and formatting of all sensor surveillance data, correlating air traffic data with all air movement within given sectors, and performing weapons assignment and direction. Dynamic updating of the onboard data base containing weapon status and characteristics, flight plan data and any enemy activity, is performed continuously by means of special ground station uplink. All data is stored online in mass drum storage and auxiliary memory.

This information is available to the console operator at his option. The console is made up of a graphic cathode ray tube (CRT), keyboard, light pen, expansion control and category switches. Figure 2-9 shows a typical console configuration. In the AWACS system there are three major onboard console functional allocations (1) commander, (2) weapons director, and (3) air surveillance. All consoles allocated to these are identical in configuration, primarily to meet system availability requirements. There can be up to

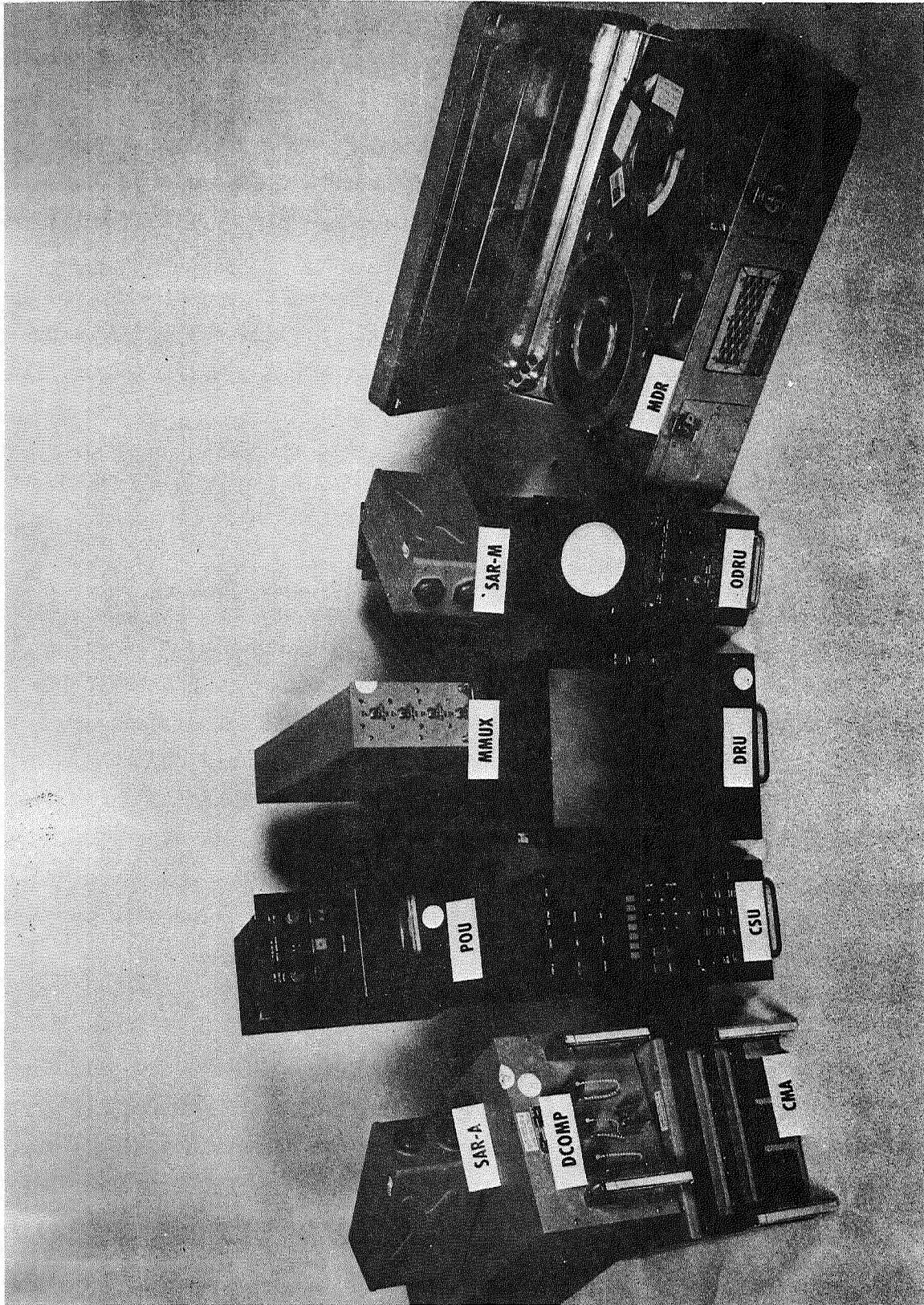


Figure 2-8. NADAR C5A Crew Control Panel

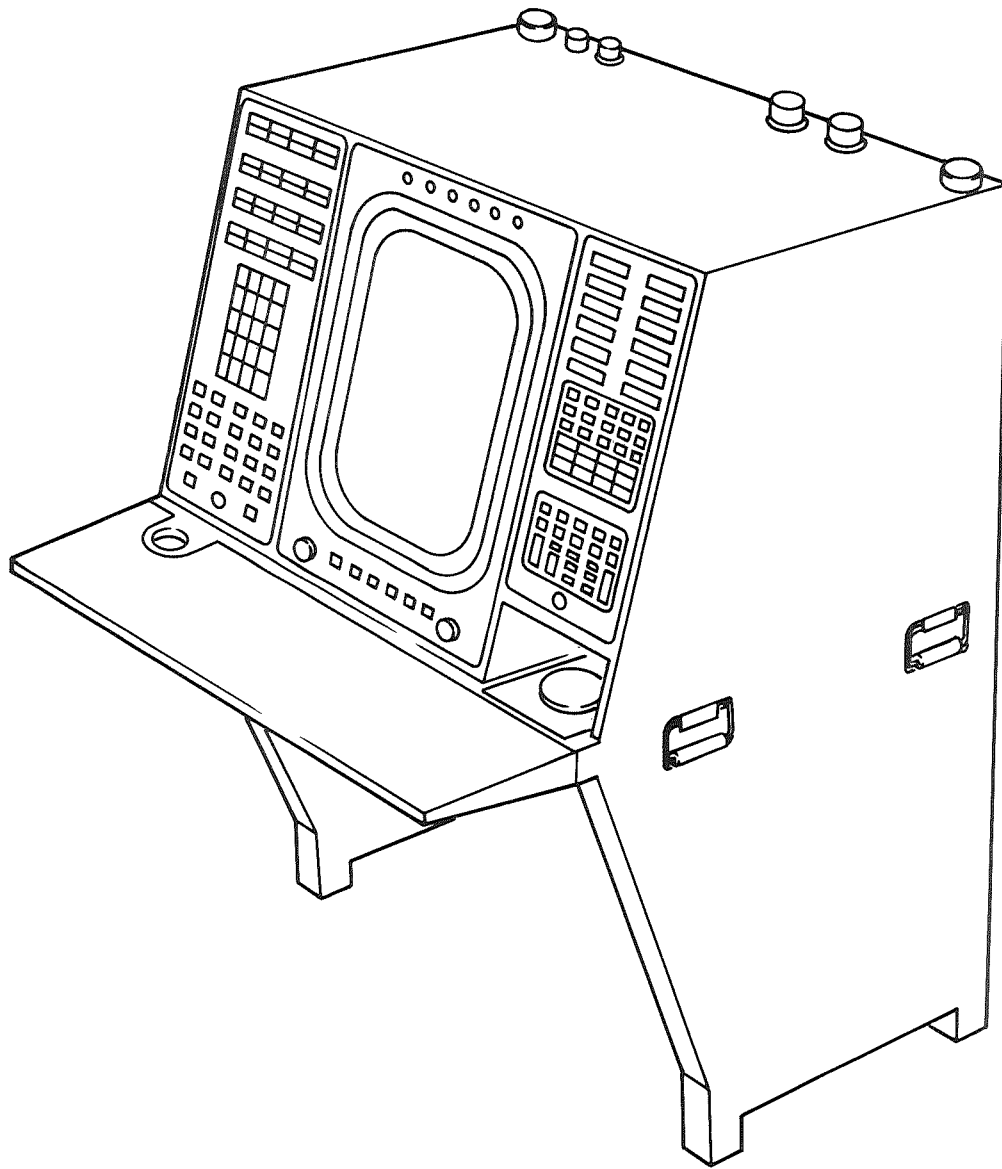


Figure 2-9. AWACS On-Board Operators Console

fourteen or fifteen consoles onboard and should one malfunction, processing tasks associated with that console can be redirected to the remaining consoles, under processor control.

The air surveillance console (ASC) is operated by many different officers responsible for various surveillance functions (track recognition, track identification, track correlation, etc.). The processing of all incoming radar returns is performed by analyzer software. Those which have been filtered through as a result of a quality analysis are displayed on the CRT.

The selected returns are categorized by type, and are then used for association with certain tracks meeting rigid criteria (point of origin, velocity, trajectory or flight path, etc.). This association enables the computer to aid the operator in presenting a true-up-to-minute air picture to all operational personnel. Those data which appear to be representative of a track carried in the AWACS system will be displayed as correlated, and those which fail to associate will be displayed as uncorrelated. Once a track has been established the operator points to it with the light pen and enters it in the tracking system for identification. Identification is accomplished analyzing all air movement for that sector. If the track is identified as one designated to that sector, it is monitored. However, if it is not identifiable, it is designated hostile and turned over to the Senior Director.

Because of the traffic in surveillance volume, there is often more than one air surveillance console operating simultaneously. In this event, each ASC is delegated a portion of the many air surveillance duties. The ASC has a warning light matrix to alert the operator to certain system conditions. Track is accomplished through monitoring of radar patterns and the assignment of meaningful symbols to these trails as a computer representation of a track. Then to maintain and, if necessary, modify this representation in accordance with peculiarities and deviations of the returns. The computer program essentially performs this task with the help of the operator, who manually assists the program by maintaining a close surveillance of all tracks on his screen.

The commitment of a weapon against a target is based on an analysis of the air picture (threat), combined with thorough knowledge of available weapons. The proper pairing is generally based on a minimum "Time To Go," to intercept, with emphasis on weapons tactics, armament capabilities, and fuel considerations. The computer program provides both situation and tabular aids to assist the operator in his decision. This information is analyzed continuously by the Weapons Director (WD). The data is available and displayed to him as required in tabular and graphic form showing a constant moving operations target to point of intercept.

The senior officer in charge of the crew is the Senior Director (SD). All final tactical measures of the air battle are his responsibility, and that of his designees. He can assume the job of WD. It is the WD's task to assign and direct weapons against all target tracks in the system. He is in complete command from the first evaluation of the enemy threat through the final moments of the interception.

With the expansion capability of the console each officer can handle multiple tracks at a time. All data is routed to each console and is selected by the operator by means of category select switches. Therefore, by merely switching from one category to another the operator can handle many types of tracks.

2.1.2.4 F15

The F15 is a tactical air command fighter plane being developed for the U.S. Air Force. Its current stage of development and the national security involved with this aircraft prohibit total disclosure of its proposed computer controlled function. The computer being selected to support this aircraft's needs will be a third generation general purpose computer. It will be word machine and will not have auxiliary storage. As with the F111D, the guidance and navigation and the weapons deployment system will be computer driven, with backup modes of operation. Unlike the F111D, the F15 will utilize CRT's and alphanumeric keyboards for data display and entry. The application communication language to be used by the crew when entering data or requesting data from the computer is undefined at this time.

2.1.3 Task Summary

In summarizing the results of this task the following statements are of particular significance:

- A. No manned spacecraft programs were found to utilize onboard graphic-display computer systems (i.e., Mercury, Gemini, or Apollo).
- B. Project Gemini's interactive system utilized segmented light registers for displaying data and a Manual Data Insertion Unit (MDIU) for entering commands and data into the computer. A computer mode selector switch was also used to signal computational modes to the computer. The astronaut via numerical encoded language was capable of communicating commands and data to the onboard computer through the MDIU and computer mode selector switch (see Figure 2-1).
- C. Apollo vehicles utilize a display and keyboard (DSKY) to provide a media for crew/computer communication. The keyboard is digital with nine functions and the display panel consists of five segmented light registers and a computer operating status panel. The language selected for crew/computer communications is referred to as a VERB/NOUN language. Briefly, the astronaut simply reduces the function he wishes to perform to a descriptive verb (action to be performed) and noun (object to which action is to be directed), converts the verb and noun to their predefined two digit numeric codes, and enters the codes into the computer via the DSKY for subsequent computer processing (see Figure 2-2).
- D. The primary functions of onboard spaceborne computers are guidance, navigation, and control (i.e., maintaining desired trajectory and vehicle attitude required to achieve mission objectives).

- E. Automation trends for future manned space missions as evident by aircraft designs are being directed toward onboard checkout, mission control, mission planning, etc. The underlying ground rule promoting the increase in onboard computer compatibility is vehicle autonomy for future manned space vehicles.
- F. Manned aircraft projects substantiate the feasibility of the automation direction selected for future manned space missions.
- G. Crew/computer languages for past manned space vehicles were limited to guidance and navigation or did not exist at all. Further study has been recognized as a necessary requirement for future space vehicles although little effort has been expended to date. There was no good base to establish a crew/computer communication language requirement, therefore language structures that were found to exist (e.g., Apollo verb/noun) were limited in capability and not deemed sufficient for application to data entry devices and processor configurations available for use in future space vehicle designs.

2.2 PHASE A TASK 2

Phase A Task 2 was the data assimilation and evaluation task of the study with the purpose of establishing a list of crew functions to be performed onboard future spacecraft and to identify the areas where the onboard computer can assist in performing these functions. This task was only concerned with interactive functions (i.e., those functions in which the crewman utilizes the onboard computer in performing a task).

Task 2 was also the critical task of this contract as it established the basis for the remainder of the study, the ultimate purpose of which is to define a structured vocabulary and establish methods by which space scientists and astronauts can communicate (in performing onboard functions) with the onboard computer.

In determining onboard crew functions for future spacecraft, an assumption was made that future spacecraft would be as autonomous as state-of-the-art technology would permit. Ground support functions performed for Gemini and Apollo missions were considered as onboard function requirements for future spacecraft. To compile a composite list of spacecraft functions an in-depth analysis was conducted into the following areas:

- Ground support functions required by existing and past spacecraft and space missions.
- Onboard functions required by existing and past spacecraft.
- Proposed functions for future spacecraft (Skylab, Space Shuttle, and Space Station).

Although much of the advanced system concepts were found in future manned space requirements, a high percentage of supporting data was obtained from the aircraft industry. The aircraft industry has more on-going projects than does the spacecraft industry, and has a more definitive automation direction. A review of the automation direction being followed by aircraft manufacturers provided a point of departure or a building block for developing mechanized spacecraft functions.

A careful investigation of current methods of documenting or listing onboard functions of both aircraft and spacecraft disclosed that several techniques were used. The more conventional methods were oriented toward mission timeline, mission phase, or vehicle hardware subsystems. These techniques were generally voluminous in nature and redundant in content and therefore unsatisfactory for the purpose of this study.

As previously mentioned, one of the objectives of this study was to define a structured vocabulary and establish methods by which space scientists or astronauts can communicate with the onboard computer. In identifying the onboard function by technology or disciplines, a structure is thus established

and a direct function, method and vocabulary relationship defined. It is for this purpose that the technique of grouping related functions into categories was adopted.

The nine function categories listed below have been established as a result of the analyses conducted as Task 2 of this study.

- A. Mission Control--Functions required to monitor and evaluate actual versus planned mission data, and to adjust mission plan or correct vehicle performance as required to achieve desired mission goals.
- B. Data Management--Functions required to acquire, process, store, maintain, and retrieve data as required to support the vehicle and related mission functions.
- C. Communications--Functions required to select the desired communications media (i.e., radio, TV, radar, etc.) and related frequency/channel, antenna and route of communications.
- D. Flight Control--Functions required to maintain vehicle attitude and trajectory as required by the flight plan.
- E. Guidance and Navigation--Functions required to acquire and reduce G&N data so that the vehicle's current versus planned position and trajectory can be determined and the degree of deviation and compensating control functions can be computed.
- F. Experiments--Functions required to perform, monitor, and control experiments being performed from space vehicle.
- G. Maneuver Management--Functions required to initiate, execute, and evaluate success of a vehicle maneuver.

- H. Operational Status--Functions required to monitor and maintain vehicle and crew operational status.
- I. Mission Independent Crew Functions--Functions required to maintain crew proficiency and efficiency during long space missions.

These categories have been defined in terms of functions and detail subfunctions (Appendix A) to a level that Task 3 can logically devise methods and techniques for crew/computer interaction. As an example, a proposed Space Station Astronomy Experiment shows the versatility of using function categories. The breakdown of this experiment shows the interrelationship of functions and how methods, languages, and commands are derived. Typical experiment operation is discussed in Appendix B. Appendix B contains an excerpt from a System Development Corporation report which describes the "X-ray polarimeter experiment" proposed for Space Station. Figure B-2 depicts where and how this particular experiment would show in the Experiment function category structure. Figure B-3 provides an example of the procedural and vocabulary language formats which could be developed to perform an experiment function.

Appendix C contains a breakdown of the function categories in Appendix A with respect to computer assistance in performing the identified functions. The actual breakdown covers past, present, and proposed spacecraft programs. The results show that industry is leaning toward a completely mechanized vehicle.

In summary current aircraft and spacecraft crewmen are capable of interfacing with the onboard computer as follows:

- A. Loading data into the computer.
- B. Modifying data already in the computer.
- C. Querying for data status, in the form of; lines, points, tabular summaries, graphics, etc.
- D. Initiating commands to vehicle subsystems.

The crew/computer interactive capability for the Gemini, Apollo, and space vehicles currently under development is identified in the matrices in Appendix C for each of the function categories established and defined during this study. Although this list includes all function categories established during the study, it is not intended for the reader to assume that the named function categories have been fully developed. There is need for much additional work to identify and develop total interactive computer capability for an autonomous space vehicle.

The results of this task support the philosophy that each function category will have its own unique interactive hardware and language requirements. This is not unreasonable to expect when considering that individual ground disciplines have dedicated computer/communication requirements and that future space language will, by necessity, have to be easy to learn and easy to use.

After a thorough review of the Syntax compiler development, i.e., the McDonnell Douglas Syntax Directed Compiler, the above stated approach does not appear to be impractical or unreasonable. Future space missions will probably find the onboard computer user performing his job functions using a language and tools that are normally associated with his job function and/or discipline in brief the Syntax directed compiler is defined as a computer software system which provides a means for defining Syntax in terms of a computer's "machine language." The defined Syntax can then be used to communicate problem requirements with the computer. The output of the McDonnell Douglas Syntax Directed Compiler can be used to recognize and analyze statements written in the defined language. Appendix F gives a detailed description of operation.

2.3 PHASE A TASK 3

Phase A, Task 3 was the initial development task of the contract. Specifically, this task devised methods for the onboard crew to communicate function requirements to the computer. The derivation and recommendation of methods by which the crew can communicate with the onboard computers to perform the functions defined in Appendix A has been shown from analysis (Ref Appendix D) to require considerations of a broad range of technologies associated with an

information system. As such, we have found it necessary to establish a broad definition of the term "method" as used in the context of this study. Method relates to the integrated operation of the processing equipments, the graphic-display terminals and the computer software, driven by operator commands to provide for, and allow, crew members a systematic means for performing their assigned tasks. The combination of the equipment and software is considered as the operating system needed to interpret and act upon the operator commands. The operator commands relate to the vocabulary. By this approach, we have found it feasible to define a structured vocabulary around which can be formulated equipment operational characteristics and software functions. In this manner, assessments of the interaction of each can be made in relation to overall operating requirements. However, it becomes necessary at this time to emphasize that the scope of this contract is not to establish a hardware design. Any hardware portrayed in this document is done so to illustrate that the communication methods and techniques being devised by this study are feasible. Also, the software discussed in this report is limited to that software which initiates, drives, and/or queries the status of application programs (i.e., those computer programs which perform actual functions such as trajectory determination, onboard checkout, etc.). These two points will be reiterated as required to ensure proper understanding of material being presented.

After beginning this phase of the study, it became evident that no one man/machine communication method could be devised to satisfy the requirements of all functions in all function categories. For this reason a decision was made that any method devised for performing onboard spacecraft crew functions should be:

- A. Flexible in Design--Methods must be "open-ended" for future enhancements and readily susceptible to change for ease in modification.
- B. "Straight-Forward" in Use--Methods must be easy to learn and easy to apply so as to minimize the training process requirements.

- C. Consistent in Approach--Methods must be similar in performance for different functions in all function categories, and methods for performing identical functions for different space vehicles.

Methods devised applying the preceding rationale were found to have basic communication hardware and language similarities. In general, interfacing communications hardware will require a minimum configuration of a general purpose Cathode Ray Tube (CRT) with light pen and/or automatic cursor capability, a multipurpose keyboard consisting of selected predefined function (i.e., on/off, clear, enter, etc.) keys, general purpose command keys, CRT line select keys and a numeric digital keyboard.

The man/computer communications media identified for development to support on-board interactive requirements is a technology oriented structured vocabulary (i.e., each onboard discipline will have an indigenous vocabulary for communicating respective function requirements to the computer). Individual discipline languages will be accessible through the use of function and CRT line select keys that provide a conversational means for selecting the desired language level. The conversational approach to obtaining a language allows time for preprocessing software and hardware initialization. Many objections appear when manned space flights are discussed along with conversational crew/computer communications. One must remember that each discipline is being handled separately, and preplanning should solve any critical timing problems.

The vocabulary structure being developed will permit onboard computer users to select the language working level conducive to solving the particular problem of interest. A language will have a function, command and procedural levels of problem solving capability. The "function" level will perform gross task (e.g., display current G&N measurement data), the "command" level performs a more controlled task (e.g., display the vehicle's current attitude using the inertial reference system), and the procedural level will permit finer control over the task being performed (e.g., determine the vehicle's current attitude by using telescopes "A" and "B" to sight celestial bodies "C" and "D").

respectively). The flexibility generated using this technique centers around the users ability to obtain the exact data he desires and to chain vocabulary segments to perform even more sophisticated tasks. Appendix E contains a central discussion of the vocabulary and hypothetical of usage.

In addition to the user vocabulary described, this study did reveal future requirement to modify onboard software. This too is a controversial subject which is challenged by the onboard programming and software (i.e., compilers, assemblers, etc.) support requirements. The need for this requirement is outside the scope of this study and will not be discussed further. It will suffice to say that the requirement has been identified and that its solution must be treated as the subject of a future study.

Phase B, Tasks 1, 2, and 3 studies develop and enhance the preliminary efforts begun in Phase A in formulating the structured vocabulary, identifying software requirements and identifying hardware requirements respectively. Paragraph 3.3 is a discussion and description of a minimum hardware configuration of a graphic-display device which is feasible and capable of assisting crewmen in performing onboard functions. The lack of multiple configuration should not be interpreted to mean that no requirement exists for multiple onboard displays or different and enhanced display configurations. There is an absolute need for multiple displays (terminals) and additional peripheral support equipment most of which is contingent on the vehicle and mission to be performed. The displays depicted in Phase B, Task 3 are consistent with the study objectives for performing a requirements analysis.

Section 3
PHASE B STUDY

It is readily apparent from Phase A studies that onboard communication methods between man and computer applicable to last half of the 1970's and early 1980 will require advances in both hardware and software technologies; first, to overcome the users' limitations prevalent in today's airborne/spaceborne vehicles and secondly, to accommodate a broad spectrum of new requirements that will be posed by future space missions. Space Systems envisioned for the next ten year period have emphasized concepts of sophistication, quick reaction time, extended deployment, and multiple missions not yet realized on any manned airborne or spaceborne system operational at this time.

These concepts differ drastically from previous manned space programs in that they are intended for application in an operational environment rather than be R&D in nature, thereby creating a need for autonomous onboard operations and minimal ground support operations for real-time activities.

Phase B was the development phase of the study and included performance of the following tasks:

Task 1--Formulate a structured vocabulary which will allow the crew to communicate easily with the computer by means of graphic-displays.

Task 2--Perform a requirement analysis, based upon the crew functions and the vocabulary, to establish the software operating system necessary to accomplish crew-computer interaction.

Task 3--Perform a requirements analysis, based upon the results from Task 1 and 2 above, to establish the hardware operating system necessary to accomplish crew-computer interaction.

During this phase of the study, primary emphasis was placed on development of a vocabulary structure. It was determined that this control software was the link between the crew and the software applications program.

3.1 PHASE B, TASK 1 - STRUCTURED VOCABULARY

Phase B, Task 1 was devoted to formulating a structured vocabulary which will facilitate crew communication with an onboard computer via a graphic display device. One of the major problems in developing a tool such as a structured vocabulary, is to perfect a feasible problem solution which will receive minimum resistance from the anticipated user. Therefore, before the task was formally initiated a list of basic requirements were established as guides for this design effort:

- A. The structure must be conducive to graphic-display usage.
- B. The structure must satisfy the requirements of all function categories determined during Phase A of this study.
- C. The structure must present a logical and normal communications technique to the user (i.e., the technique must be easy to learn and easy to use).
- D. The structure must be flexible and susceptible to change without incurring major rework.
- E. The structure must be consistent for all function categories.
- F. The lowest level of the structure should support a command capability for performing functions (i.e., the user commands an onboard function to be performed via the computer, but the computer may be required to perform several calculations and initiate several hardware components, the complexity of which, remains transparent to the user).
- G. The structure must be capable of transferring from any level of one category to any level of another and return, closing all open category segments.

Using the structured vocabulary approach and applying the preceding ground rules, a vocabulary structure that parallels the function category breakdown established during Phase A was formulated. This structure not only satisfies the basic requirements, but gives the added capability which permits development of individual technology oriented vocabularies.

To establish the adequacy of this vocabulary structure, selected functions were implemented on the Space Shuttle simulator cockpit configuration in St. Louis. The simulator is driven by a time sharing Sigma 7 computer. Under program control of the computer was a microfilm carousel projector (which acted as a graphic-display), and the function keys, line select keys and numeric keyboard described in Task 3. The purpose of this experiment was to determine how the vocabulary would "stand-up" under test in a space vehicle environment. Tests confirmed that the vocabulary structure does not break down when changing display configurations. Photographs of test performance are included in Appendix E to illustrate the communication technique. The vocabulary structure was also found to have the following attributes.

- A. It is conducive to conversational interaction with graphic-displays.
- B. It is flexible to change (i.e., as additional onboard functions are mechanized, no real unexpected problem arises).
- C. Implementation of vocabulary components can be phased by requirements.
- D. It is straight-forward, thus, giving to simplicity.
- E. Individual elements can be massaged to meet specific requirements without impacting other defined elements.

The structures lowest level is the vocabulary, which is the basic means of communicating onboard function requirements to the computer.

Figure 3-1 illustrates the vocabulary structure highlighting the parallelism with function category structure and the usage technique. The primary reason for the vocabulary structure is to allow keyword identification of the desired task to be performed. Using this technique a crewman can be assisted by the onboard computer in locating the function to be performed. The language structure is four levels with the lowest level, the vocabulary, subdivided into two modes. The four levels of structure provide a logical approach to attaining the desired working level. The vocabulary level of the structure was divided into two modes the command or procedural mode. Either mode is capable of initiating a desired crew function. The normal mode of operation for the vocabulary software is the command mode. However provision has been made for a procedural mode of operation. This mode of operation will allow more control of the structure by the crewman.

Modes of Operation

Command Mode--A command is one or more words used to identify an action to be performed by the computer. The extent of the action to be performed depends on the subfunction being executed. Characteristics of a command must permit the crewmen to communicate easily and naturally with the computer, and yet, provide a formatted statement to the computer which can be efficiently recognized and processed. The commanded format includes Operation, Class, Arguments, 1, 2, ... N-1, N:

- Operation: The operation is the action to be performed by the computer.
- Class: The "Class" is a coding which identifies a type of command for efficient software editing and processing.
- Argument: Arguments identify the action operator and operand and defines data control parameters involving execution tolerances, display requirements, etc.

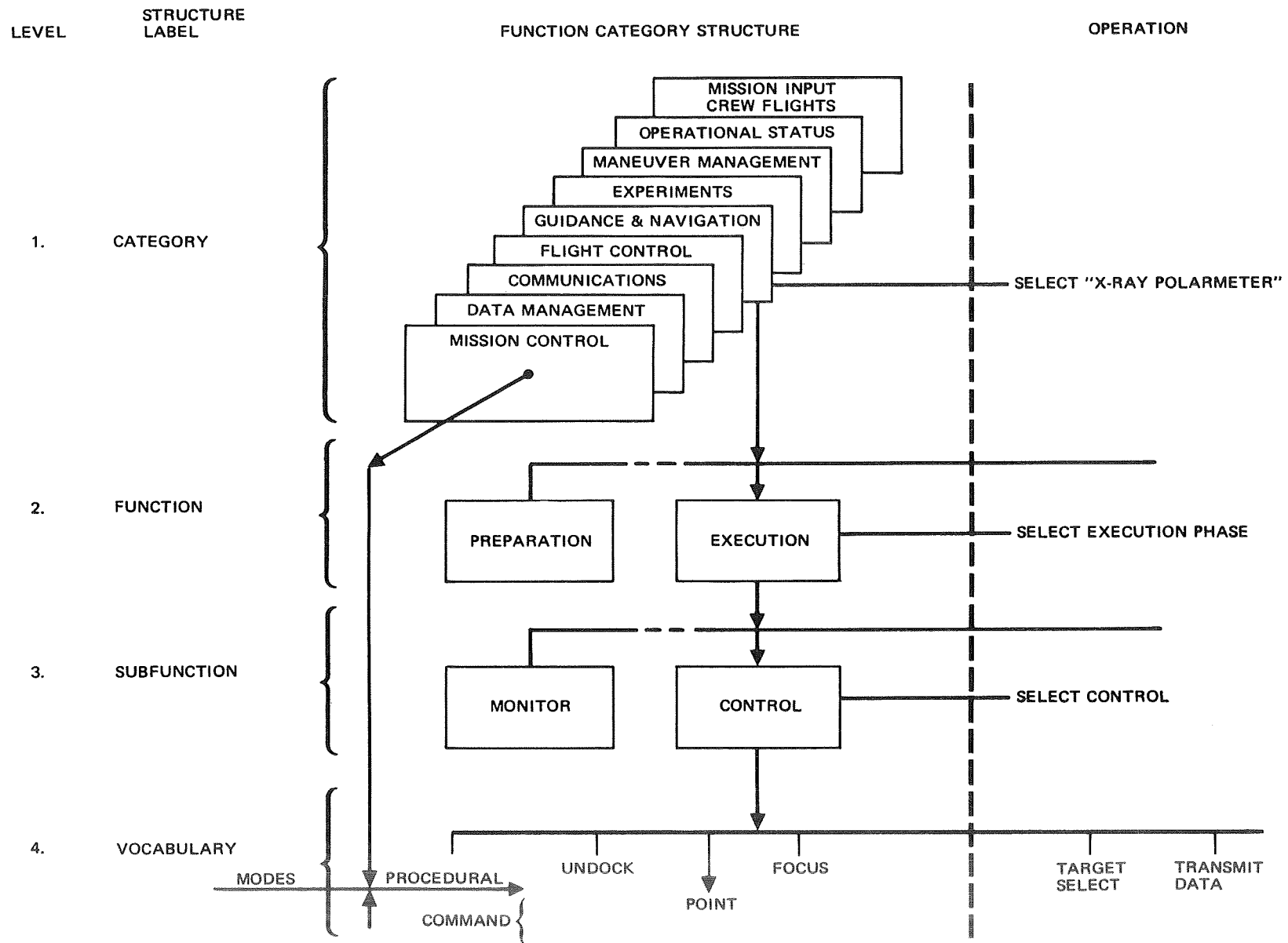


Figure 3-1. Vocabulary Breakdown & Development Chart

Procedural Mode--The procedural mode of operation permits the knowledgeable user the capability of chaining/linking commands, both within and without of the category being performed for the purpose of performing complex functions. Once a procedure has been established, it can be saved and reused. This technique of creating procedures is accomplished through the manual mode of the vocabulary software.

The procedural mode will satisfy the projected requirement of performing periodic repetitive tasks. This requirement is particularly evident when examining onboard experiments. An experimenter is periodically required to check the status and display results of his experiment, and contingent on the data displayed perform an action. Through the use of the procedural mode, an experimenter will be capable of fabricating a reusable procedure which can be initiated as required, and thus, free himself for more important tasks.

Under normal operating conditions both the command and procedures modes are controlled by the computer (automatic). However in certain cases it may be desirable for the operation to intervene, in which case the automatic sequence can be interrupted and manual action taken. Future study in this area will be required to determine the exact methods.

Appendix E presents a general but brief discussion of the vocabulary breakdown, reiterating points already made. Also contained in this appendix is an example illustrating the usage of the vocabulary and graphic-display.

Development of a "computerized" structured vocabulary consisting of onboard technology oriented vocabularies has become feasible with the advent of language compilers. McDonnell Douglas is currently developing such a compiler which provides a means for defining syntax in terms of a computer's machine language. The compiler generates a software system which is capable of recognizing and analyzing statements written in the defined syntax. The syntax can then be used to communicate function requirements to the computer. The McDonnell Douglas language compiler is being developed in Huntington Beach, California, and is referred to as a Syntax Directed Compiler (SDC). The significance of this technique is that individual vocabularies can be expanded and/or changed without affecting other vocabularies or the operating system.

A technical discussion of the concept and development of the SDC is contained in Appendix F. Appendix G is an example of how the SDC would be used to generate an onboard command.

3.2 PHASE B, TASK 2 - SOFTWARE OPERATING SYSTEM

The software requirements analysis is based on Appendix D for crew/computer interaction. The structured vocabulary is to be used in the performance of those crew function categories requiring interactive communication.

For purposes of this discussion, the vocabulary supporting software is separated by categorizing it as follows:

- A. Operating System Supervisor (Executive)--That software responsible for performing the multiprocessing functions of the onboard system.
- B. Application Program Software--That software (or program module) responsible for performing a specified task/application as supervised by the operating system.
- C. Display Controller Software--That software responsible for direct communication with onboard display system in support of the previous two software users.
- D. Subroutine Library--Programs universally used by the Supervisor or Application Programs.

The vocabulary controller software is, in reality, a part of the subroutine library, but is considered separately because of the importance of the interface with the display console. Thus, the vocabulary software is considered to be that software which interfaces the display terminal with the onboard operating system software, as well as application programs (or program modules) and subroutines used in the performance of the crew functions requiring interactive displays. The configuration of the software is depicted in Figure 3-2. The onboard system is considered to be under the control of an operating system executive or supervisor, responsible for executing the application programs,

as well as controlling the system data bus. Within this system configuration, the vocabulary software is responsible for the display terminal data transmissions (as received and sent on the system data bus) in support of the applications programs as controlled and monitored by the operating system executive. To meet these responsibilities, two functional areas of vocabulary software can be defined. They are:

A. Display Controller

B. Subroutine Library

Figure 3-2 shows that the controller interfaces with the display terminal(s) and the operating system supervisor while the subroutine library provides the interface with the application programs. The total environment of the vocabulary software is defined in Subsection 2.2.

3.2.1 System Characteristics

The vocabulary software must support a high degree of operator interaction with the operating system. With the communication envisioned, it becomes apparent that the operator has an almost unlimited amount of control. The software must support action on a step by step basis or execute procedures from beginning to end with no need for intervention.

Primarily, this study recognizes that the man is continuously in the loop and should have the capability to exercise maximum control of the system. It also recognizes the creativity of man and provides him with the means of creating online any actions or sequence of events he wishes to perform without having to rely entirely on preprogrammed procedures.

The goal here is to provide the operator with the ability to initiate programs, provide data input, receive outputs and generally control the operations of the system.

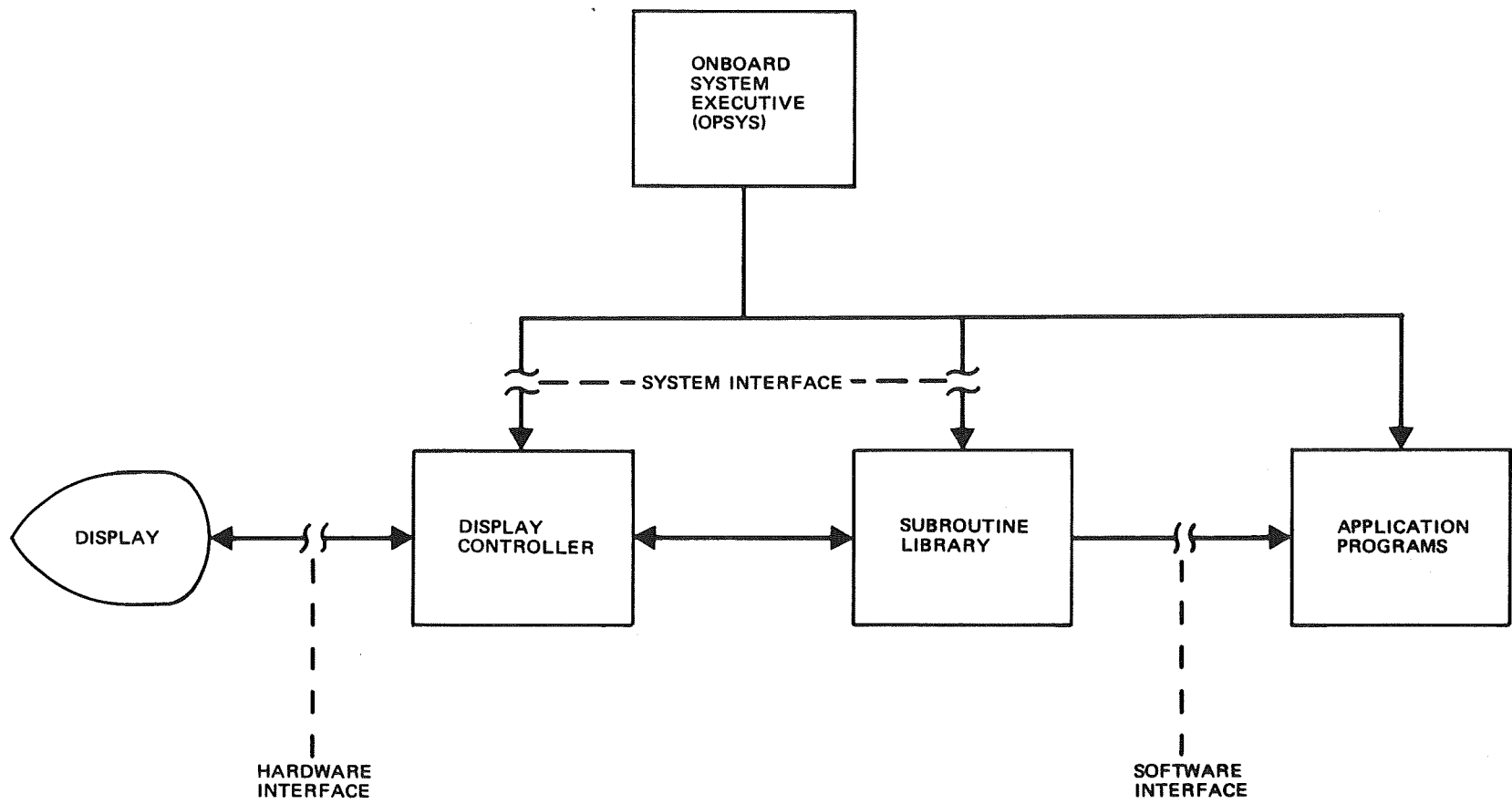


Figure 3-2. Vocabulary Software Interfaces

Additionally, from the considerations of flexibility and autonomy, it is apparent that the software must be capable of supporting automatic console reassignment since the primary consoles provide backup for one another in the event of equipment malfunction, as well as providing software support necessary to allow each console to have the capabilities required for a given application, that is, the software augments the general purpose display to allow performance of specific, application-oriented functions.

To provide the operator with a positive indication of being "in the loop" and to inspire confidence in the interactive system, the system must indicate, in a timely manner, that it is responding to operator actions.

3.2.1.1 Display Terminal Characteristics

The display hardware provides the following data items of vocabulary software:

- A. Numeric Keyboard Data.
- B. Function Keys.
- C. Vocabulary Command Keys.
- D. Screen Pick Device (i.e., light pen, cursor).

The display has eight function keys to be recognized. The keys are used in conjunction with the structured vocabulary. The interpretation of each key is as follows:

- A. Category Select--Request for display of the nine crew function categories.
- B. Function Select--Request for selection of execution phases 'initiate,' 'execute,' and 'terminate.'
- C. Procedure Select--Selects procedural mode of execution.

- D. Data Select--Request for data from system data base.
- E. Enter--Signals end of data entry.
- F. Execute--Signals request for command/procedure execution to begin.
- G. Clear--Cancel current operation.
- H. Manual--Signals bypass of normal operations to go directly to vocabulary level.

The twelve vocabulary command keys are controlled by the application program to display the instruction repertoire required for performing a selected category task. The keys require updating as the operator proceeds from task to task. Arguments for the commands are entered via the keyboard.

The screen pick device is used to select specific lines or entries on the display, such as selecting an entry in a category or subcategory list. Each item provides a unique data type for software processing.

The display receives the following data items from the vocabulary software:

- A. Test data displays.
- B. Graphic displays.
- C. Vocabulary Command Key control signals.

3.2.1.2 Application Program Requirements

The application programs provide the primary link between the crew and the onboard system. Each application is programmed to meet the requirements of the crew function categories previously defined. Within each application many specific tasks are performed. When these tasks involve the use of the displays, the interface is via the display subroutine library.

To facilitate both the programming and the execution of the various tasks, the subroutine library must provide for the two-way exchange of all display data types required by the tasks while minimizing the amount and complexity of programming the tasks.

The application programs are responsible for interpolation of the man/machine information exchange via the display system. The variety and classes of data is almost limitless (since any application within the automated system capability can be a candidate for implementation). However, the following list of data items, extracted from the communication analysis given in Appendix D, is presented as a baseline and guide to establishing at least a minimum set of requirements upon the software data handling capabilities.

- A. Text Messages--Direct English language statements pertinent to the presently executing program used to indicate status or request operator action. These messages may be relegated to a specific area of the screen.
- B. Tabular Data--Program output data optionally containing variable name, value, and units. Standardized format is recognized as a trade-off of flexibility versus program simplicity. The program has the responsibility of providing display/column identifiers to enhance readability.
- C. Menus--Essentially lists of items to be selected from as requested by the application program. Data types will include program/procedure names, command lists, or program peculiar options. Menus should also be considered for display on specific portions of the screen.
- D. Status Background Data--Depending on the hardware capability, this will encompass slide/film projection, TV camera, or other similar mechanisms for providing display background information which stands alone or is augmented by dynamic, program supplied data. Static data will also include display or canned software generated graph formats, schematics, table overlays, or any type of fixed data not economically or practically feasible to implement with a 'picture.'

The individual application programs are the key to the implementation of the structured vocabulary. It has been previously identified that each crew function category has unique vocabulary requirements. Within the application program, then, the operator inputs are related to the vocabulary defined for a specific task. The application program includes the related semantic procedures and data input processing to permit usage of the syntax-directed compiler. When the data has been received from the console, the application program activates the SDC to interpret the data, the SDC being available in either the subroutine library or accessible through the operating system. The compiled data is then executed and control returned to the application program. Note that the interpretation of the input data may be more efficiently performed by the application program when considering fast reaction time and limited core.

3.2.1.3 Operating System (OPSYS) Interface

The onboard operating system must control the allocation of resources and the execution of crew tasks. As such, it must provide the necessary status and mode information to allow assignment of shared displays to specific tasks. Additionally, it must be capable of activating any display as well as subverting any current task in favor of a higher priority task. To meet these requirements the operating system must interact with the vocabulary software to perform the following system-oriented tasks:

- A. Recognizing/allowing console starting.
- B. Restriction of console use to allowable applications.
- C. Reassignment of console to higher priority application.
- D. Current display override/add-on for critical message/parameter display.

This study recognizes that display usage conflicts are inevitable whenever a higher priority communication task occurs involving a display currently in use by another application. The problem existed on Apollo and will exist in future missions. The operating system executive will be required to inform the crew

of critical conditions or failures. To meet this requirement, the operating system must be capable of forcing display of data relative to the critical condition.

The requirement then exists to retire current user by either (1) task immediately cancelled, (2) task retired on an orderly manner allowing for retention of data collected to this point, data base updates/resets, equipment shutdown/safing, and (3) task placed in hold for future restart.

At this point, it must be assumed that the following ground rules apply.

- A. OPSYS recognizes the need for a forced display.
- B. OPSYS has the capability and need to perform resource allocation.
- C. OPSYS has access to the Display I/O routines and status tables.
- D. The application program, alone, is aware of its status and the status of equipment dedicated to it.
- E. The application program will have within itself the ability to hold, stop, or shutdown in an orderly fashion.
- F. If the application cannot be terminated/held, and must be abruptly terminated it should, upon restart, allow for manual/automatic corrective measures to be undertaken before resuming.

3.2.2 Functional Definition

Figure 3-3 illustrates a functional flow typical of an onboard operating system with the characteristics previously defined in Subsection 2.2. The vocabulary software is available to both the operating system and any currently active application program. The subsystem executives may be considered to represent either continuously active processors (i.e., Mission Control) or periodically

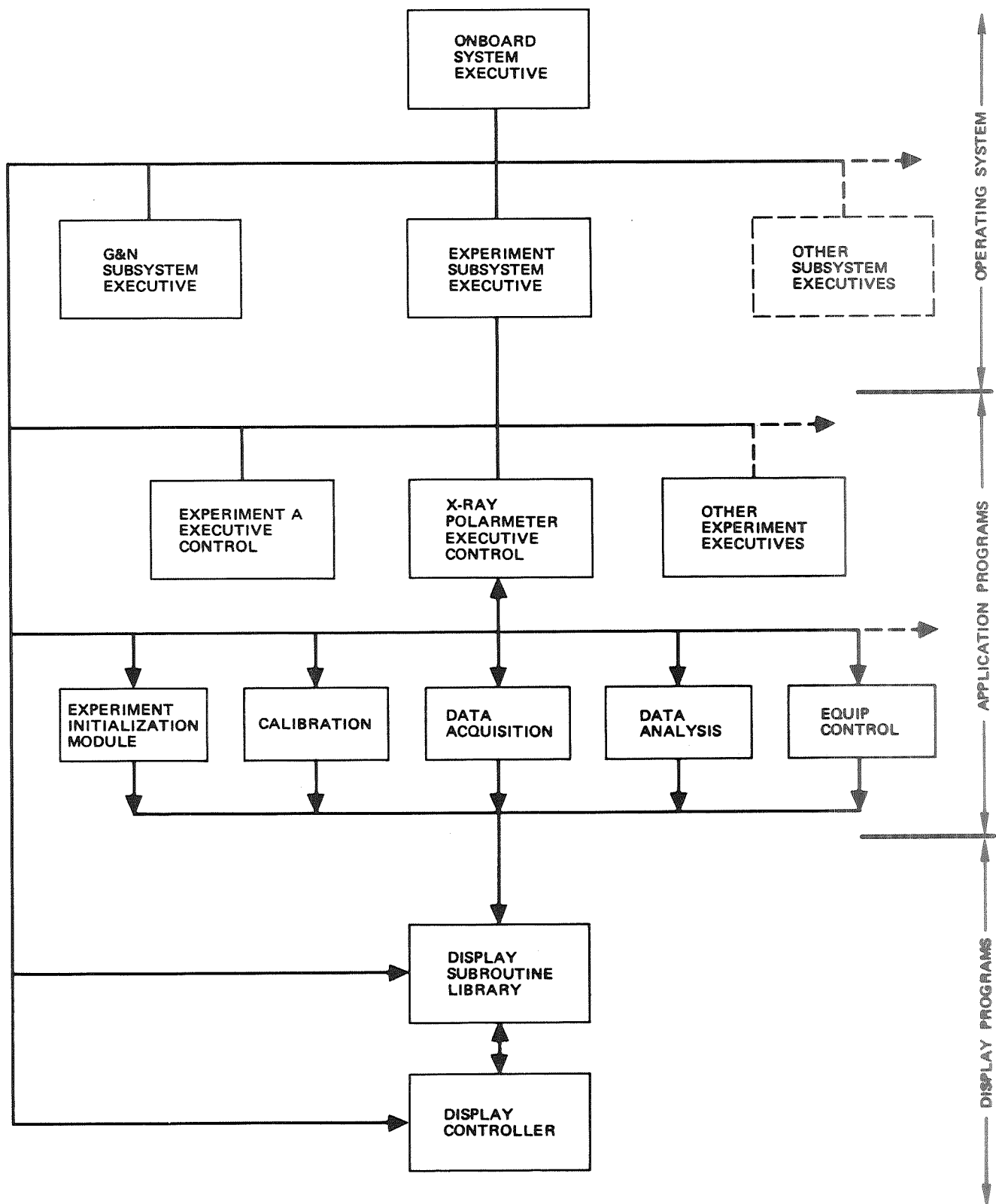


Figure 3-3. Onboard System Functional Flow

active processors (i.e., Experiments). The concept is adaptable to the general class of onboard systems currently envisioned for the next decade of manned spacecraft.

3.2.2.1 Display Controller

The display controller software package directly interfaces the display with the operating system. Insofar as possible, the software herein should isolate the hardware from the remaining software. The functions of the controller are:

- A. Data Input--Processes all data input from the display and passes data to applicable subroutines.
- B. Data Output--Transfers all data to the display per request of the display subroutines.
- C. Status Monitor--Responsible for establishing, maintaining, and monitoring the operational status of the display. Determines legality/relevance of both input and output data based on current status.

When the onboard system is activated, the display controller is notified to initialize the display. The controller is responsible for establishing the operating condition for the display. The status and/or mode of a display is established to the level of honoring inputs from the display operator or output data from an application program.

Primarily, the controller is to perform display communication as requested by routing in the subroutine library. The subroutine library, in turn, provides the link between the application programs and the display. In so doing, the subroutine library insulates the display system from the application world and provides an optimum programming interface for the applications programs.

The controller performs final formatting and all communication with the onboard data transmission link within constraint of the operating system.

3.2.2.3 Subroutine Library

The subroutine library provides the software interface between the application world and the display. These subroutines are universally used in support of those nine crew function categories requiring the use of onboard display system. The functions of the subroutine library are:

A. Output Data

1. Text Data Displays--Category and subcategory lists, procedure statements, status messages, error messages, menus.
2. Graphic Displays--Plot trajectories, attitude displays.
3. Tabular Displays--Parameter lists, status tables.
4. Vocabulary command key control signals.

B. Input

1. Function keys.
2. Vocabulary command keys.
3. Numeric data input and verification (includes requirement for automatic playback and update of entered data).
4. Screen Picks--Manual selection, category or subcategory selection, line selection.

To facilitate the use of the display, certain utility routines require inclusion in the subroutine library to provide the following services:

- #### A. Output Data Formatting--Conversion of tabular data to plots, text data manipulation, data conversion (e.g., floating point to display code), vocabulary command key control command formatting.

- B. Input Data Formatting--Screen pick interpretation, data conversion, vocabulary command key interpretation.
- C. Status Checks--Interpretation of status tables.

This study recognizes that additional library routines may be required depending on the actual hardware configuration; thus, the above requirements represent a minimum level of application program support.

As can be seen, the subroutine library performs a wide variety of services. However, in each case, the goal is performing a specific function for an application program. The primary services performed in astronaut communication are:

- Identification of items picked from the screen by the operator.
- Identification and indication of function key activation.
- Providing formatted data entries as input from the operator as well as displaying those entries for operator verification and/or modification.
- Vocabulary command key setup.
- Vocabulary command key interpretation.

To further illustrate the role of the vocabulary software, Table 3-1 gives a brief description of the activity involved in the performance of an experiment by a crew member. The interplay between the operator, the application program, the operating system and the display through experiment oriented, is generally applicable to the entire range of crew functions. Therefore, the prime responsibility of the vocabulary software is to provide the line between the crewman and application software programs.

Table 3-1
TYPICAL CREW COMPUTER INTERACTION

Operator Action	Display Input Processing	System Activity	Display Status
Category select key activated	Receives and interprets function key	Operating system requests display of category menu and requests operator response	Category menu displayed
Category selected (experiments)	Receives and interprets menu selection	Operating system activates experiment exec which, in turn, requests display of available experiments	Experiment menu displayed
Experiment selected (X-ray polarimeter)	Receives and interprets menu selection	Experiment exec activates exp controller which, in turn, displays data as required by the chosen experiment. Voc cmd keys are set-up for the selected experiment	Voc cmd key set-up screen display as required
*Operator selects command	Voc Cmd - key identified	Exp controller assumes command mode posture. Command data anticipated from operator	As required
Operator enters cmd related data	Data collected and merged with command	Command readied for execution	Data displayed
Operator enters execute cmd	Execute request received	Command executed. Controller awaits next step	Cmd results displayed as required

*Procedure mode selection at this point would allow experiment to proceed as programmed.

3.3 PHASE B, TASK 3 - MINIMUM HARDWARE CONFIGURATION

Although the design of computer communications hardware (Data Entry/Display Terminal) is beyond the scope of this study, it is recognized that the requirement features are an integral part of the method for communicating with a computer. Therefore to facilitate discussion of crew/computer communication methods, a minimum hardware configuration of a remote graphic-display terminal is shown on Figure 3-4. Two factors which highlight the differences in function communication requirements are (1) manned space vehicles have common function requirements, but have different missions, objectives, and configurations. All of which are influencing factors in devising interactive methods. Examples would be the difference in guidance and navigation, onboard checkout, and flight control requirements for Space Shuttle and Skylab. Both vehicles have a need to perform all the functions, but their requirements are very different and (2) interactive functions performed onboard manned space vehicles have a common requirement to communicate with the onboard computer, but each has unique data acquisition problems which prohibit a common hardware solution. Example: Manual control of the in-flight vehicle, a flight control function, enters data to the computer via remote sensors which transmit the position of the manual controls (i.e., yoke, translation and/or rotation devices, etc.). On the other hand, updating ephemeris data in the computer's data base, a G/N function, is accomplished by the graphic-display terminal keyboard. (Figures 3-5 and 3-6 show the various display and control requirements for decks 2 and 4 respectively of the proposed Space Station.)

This graphic-display terminal configuration not only provides an instrument for discussion and developing a communications vocabulary but a prototype for future designs. A discussion of the components which comprise the graphic-display terminal depicted in Figure 3-4 follows:

- A. Cathode Ray Tube (CRT) Display--The CRT display is a multipurpose device capable of receiving and displaying data from the computer, remote television camera and/or mechanical projectors (i.e., micro-film projectors which display an image on the back of the CRT by

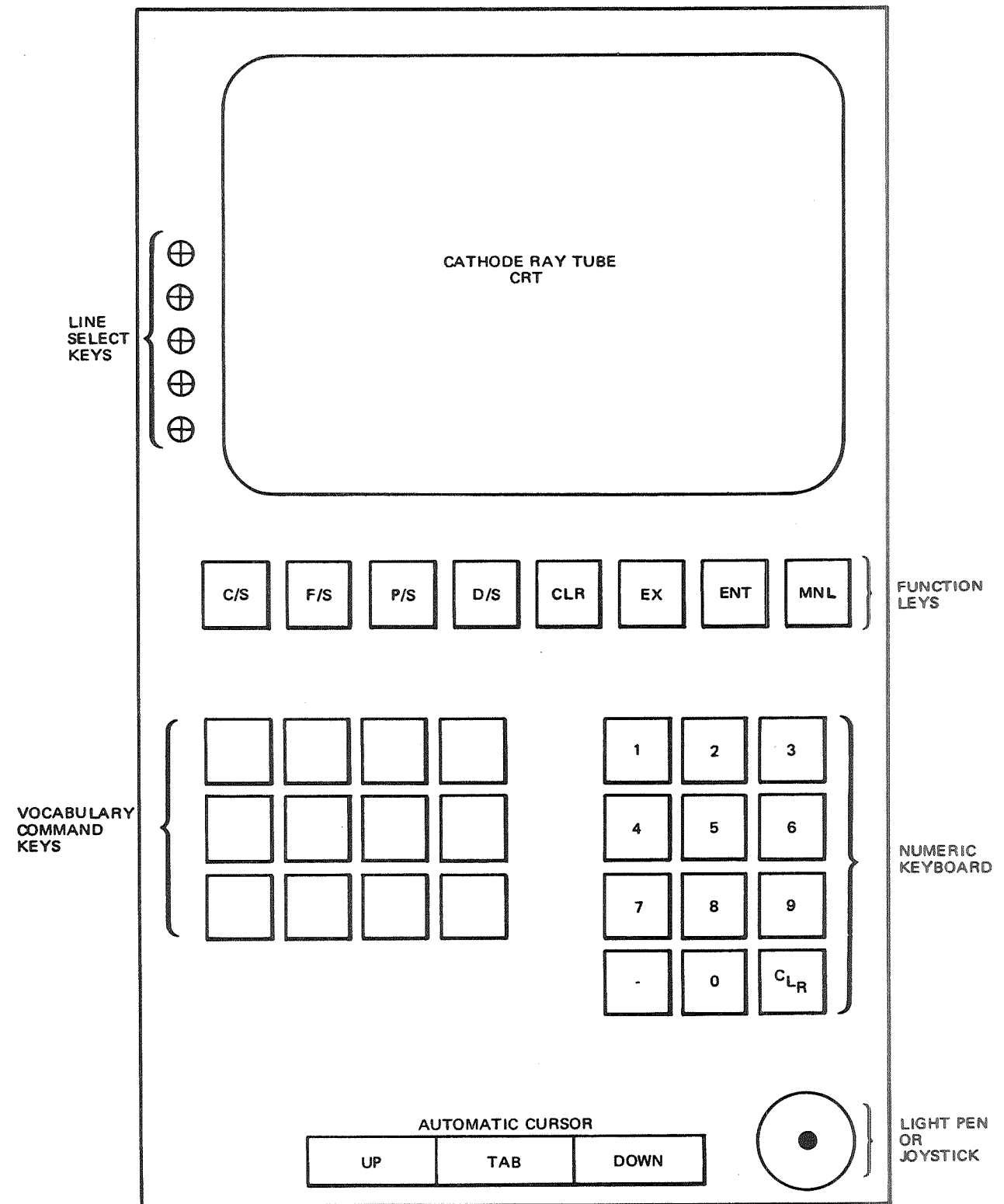


Figure 3-4. Remote Graphic Display Terminal

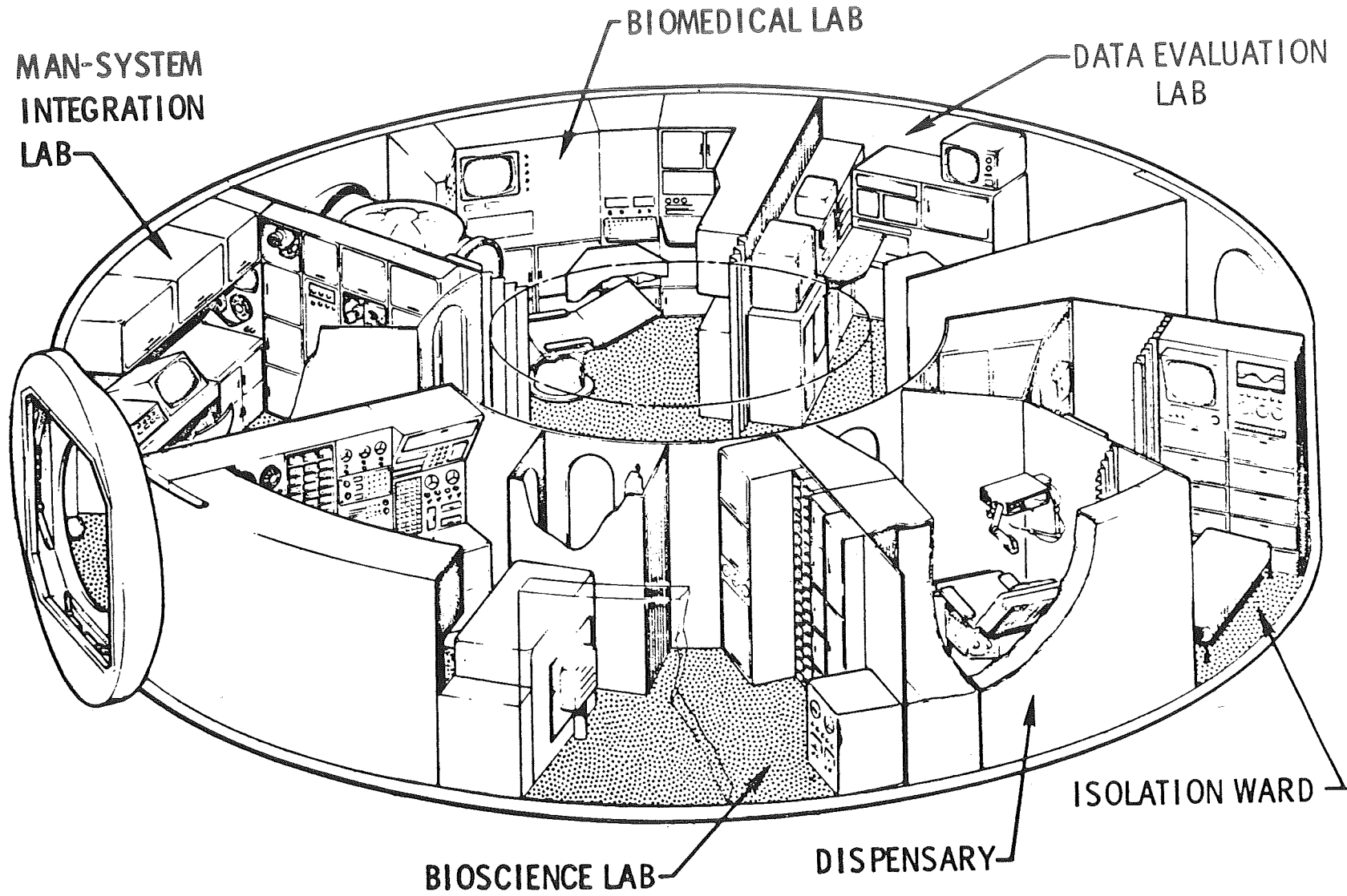


Figure 3-5. Experiments (Deck 2)

3-23

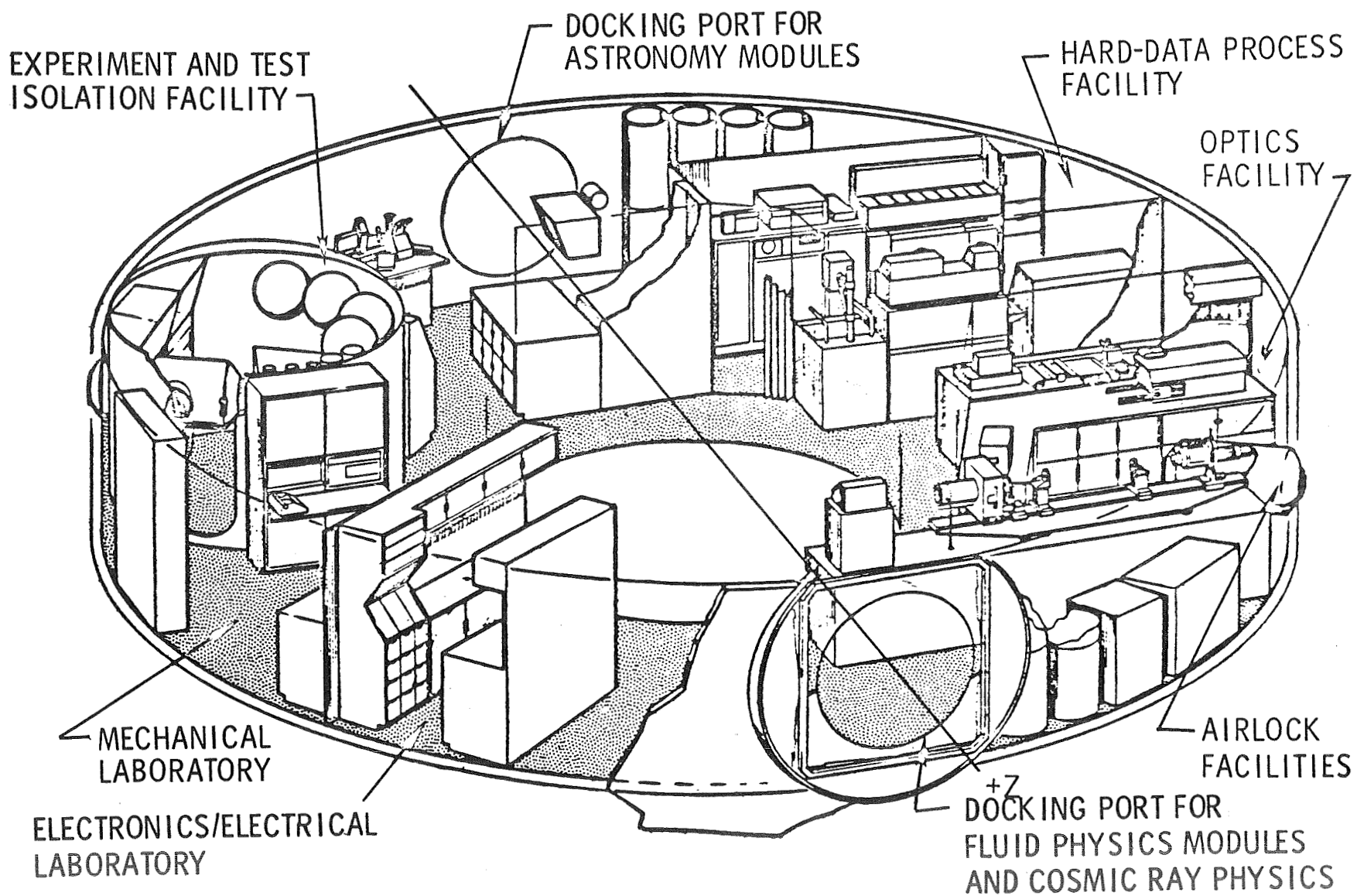


Figure 3-6. General Purpose Laboratory (Deck 4)

projecting the image through a window in the rear of the tube). The CRT via a light pen or cursor position also provides a media for entering data into the computer.

- B. Line Select Keys--Line select keys correspond to data lines on the face of the CRT. When a line select key is engaged, the computer receives a discrete impulse. This impulse is correlated to its respective data line, and the software, being cognizant of the data displayed, performs those preprogrammed functions required by the data on the line.

- C. Function Keys--A graphic-display terminal is a general purpose device capable of performing or assisting in the performance of all onboard interactive functions. The function keys associated with the terminal are used to inform the operating system that the terminal is active and preparing to perform a specific task. As a crewman manipulates the function keys to assign meaning to the vocabulary/command keys, the operating system initializes the application program software. Function keys retain their predefined and preprogrammed identity and can be used to select and perform various tasks as required. Function keys are also used to initiate data transmission to the computer and indicate to the software end of data, begin execution.

Eight function keys are proposed to satisfy the structured vocabulary and execution requirements of crew/computer communication methods. The identified keys are listed for information:

- 1 C/S - Category Select
- 2 F/S - Function Select
- 3 P/S - Procedure Select
- 4 D/S - Data Select
- 5 ENT - Enter
- 6 EX - Execute
- 7 CLR - Clear
- 8 MNL - Manual

(For detail description of function key usage refer to Appendix E.)

- D. Vocabulary/Command Keys--Once the graphic-display terminal has been initialized, via the function keys, the vocabulary/command keys will display the instruction repertoire for performing the desired task(s). Each activated key represents a command for the computer to perform an associated task(s). The vocabulary/command keys are general purpose (i.e., will accommodate all onboard functions). Normally commands require arguments to control the function being performed. Arguments will be entered through the numeric keyboard.
- E. Numeric Keyboard--The numeric keyboard provides the terminal user the capability of entering data to the computer. The data will be in the form of arguments to commands. Example: POINT telescope 1, 20 degrees elevation and 5 degrees azimuth, FOCUS 500 nmi. "POINT" and "FOCUS" are commands. "Telescope" 1, 20, 5, and 500 are arguments entered by the terminal user.
- F. Light Pen/Joystick--The light pen or joystick will be used for course cursor alignment and rapid answers to "binary" type conversational statements. The light pen is the preferred device and yields the greater capability.
- G. Automatic Cursor--The automatic cursor keys are used for fine alignment of the cursor to the beginning of data fields.

In summary, the minimum hardware configuration identified in this section is concerned with the general purpose components that can be utilized by all spacecraft and all functions. This study did reveal and does recognize the requirement for other general and special purpose communication's hardware in performing onboard interactive functions. These components and requirements are discussed in Appendix D.

Section 4
CONCLUSIONS

The primary objective of this research effort was to develop advanced crew/computer communication methods. This objective has been achieved and is represented by the structured vocabulary discussed in Appendix E of this report. Preliminary tests of this vocabulary were made on the Space Shuttle Cockpit Simulator at the McDonnell Douglas Astronautics Company, St. Louis, Mo. Although these tests were limited to a single task in one of the nine function categories identified during this study, they verified that the basic structure of this vocabulary is a sound foundation on which to build.

During the first phase of this study, analysis of spacecraft functions and methods of astronaut/computer communication in the performance of these functions was performed. These analyses demonstrated the feasibility of the onboard technology-oriented vocabulary and substantiated the premise that a graphic-display communication system should be an integral part of the man/machine interface.

Development of a computerized structured vocabulary consisting of onboard technology-oriented languages has become feasible with the advent of language compilers. McDonnell Douglas is currently developing such a compiler which provides a means for defining syntax in terms of a computer's machine language. The compiler generates a software system which is capable of recognizing and analyzing statements written in the defined syntax. This syntax can now be used to communicate function requirements to the computer. The significance of this technique is that individual languages can be expanded or changed without affecting other languages or operating systems. These techniques, as presented in this document, have been implemented at MSFC on an IBM 7094, and at MDAC in a Space Shuttle cockpit simulator. Through these efforts, this technology has proved effective and usable.

This research effort resulted in new approaches to further develop the techniques and technologies discussed in this report. The basic structure of the vocabulary developed in this study should be further expanded to encompass a crew/computer conversational procedural language for application support. A general programming language for the development of onboard software generation, update and modification should be considered. Implementation of a total structured vocabulary representative of the nine function categories should also be considered for future refinement and testing.

The ultimate crew/computer communication is voice conversation, both from the crewmen and the computer. Currently efforts are underway to develop techniques for accomplishing this and the results should be incorporated in future studies of crew/computer communication.

In summary, this study established the communications link between the crewmen and the applications program stored within the onboard computer. This link ties the functions to be performed with the onboard computer programs necessary to perform them. This study has laid the ground work and basic foundation upon which future spacecraft interactive software can be developed.

Section 5
BIBLIOGRAPHY

GEMINI

All information on Gemini was obtained from Don Frankie and Ray Wamser,
McDonnell Douglas - St. Louis, Missouri.

APOLLO

1. "Apollo Guidance and Navigation System Lunar Module Student Study Guide"
Prepared by: AC Electronics, Division of General Motors, Milwaukee,
Wisconsin

(Computer Utility Prog)	15 January 1967
Rev A	5 April 1967
Rev B	1 November 1967
Rev C	1 October 1968

2. "Apollo Lunar Module Primary Guidance, Navigation and Control System
Student Study Guide"

Same Author	
(Familiarization Course)	25 August 1966
Rev A	20 January 1967

3. "Apollo Command Module Primary Guidance, Navigation and Control System
Student Study Guide"

Same Author	
(Familiarization Course)	7 November 1966
Rev A	15 February 1967
Rev B	27 May 1968

4. "Apollo Command Module Primary Guidance, Navigation and Control System
Student Study Guide"

Same Author	
(System Mechanization Course)	25 August 1966
Rev A	31 July 1967
Rev B	25 April 1968
Rev C	3 September 1968

5. "Apollo 12 Flight Plan AS-507/CSM-108/LM-6" - 21 April 1969
Prepared by: Manned Spacecraft Center, Houston, Texas

6. "Flight Operations Plan G-1" - 21 March 1969
Prepared by: Manned Spacecraft Center, Houston, Texas
7. "Apollo Navigation Ground and Onboard Capabilities" - 1 September 1965
Technical Report No. 65-AN-2.0
Prepared by: Goddard Space Flight Center, Greenbelt, Maryland
Manned Spacecraft Center, Houston, Texas
8. NASA Facts 0-6/12-67 - John F. Kennedy Space Center, Florida
(NASA) 1968
9. "Saturn V Flight Manual SA-508" - 15 August 1969 - changed 1 March 1970
MSFC-MAN-508
Prepared by: Marshall Space Flight Center
10. Phone conversations with:
Raytheon - Sudsbury, Mass.
MIT Instrumentation Laboratory (Margaret Hamilton)
IBM-Huntsville (Olen Britnell) - LVDC, FCC
TRW - California (Bruce Sanford)
11. LVDC Equation Defining Document for the Saturn V Flight Programs
Revision C - 27 August 1969 - IBM No. 69-207-001 - MSFC No. 111-4-423-15

SKYLAB

1. "Apollo Applications Program Base Line Reference Mission" - 9 March 1970
Prepared by: Manned Spacecraft Center, Houston, Texas
2. "ATMDC Interface Program Requirements Document (IPRD) SVWS Version" -
20 January 1970 - 50M-37 and 39
Prepared by: NASA S&E-ASTR-SG
3. "Saturn I Mission Implementation Plan Skylab I Mission" - 20 April 1970 -
PM-SAT-8010.11 (SL-2, 3, and 4)
Prepared by: NASA/MSFC PM-SAT-E
4. "Skylab A ATM Digital Computer Program Requirements Document (PRD)" -
1 July 1970 - 50M-37941
Prepared by: NASA/MSFC S&E-ASTR-SG
5. Phone conversations:
IBM - Huntsville (Olen Britnell)

C5A

1. Phone conversations:
Northronics (Northrop), California
Local Lockheed Office
Astrionics Department at Lockheed, Marietta, Ga.
Guidance and Navigation Department at Lockheed, Marietta, Ga.

2. "Aids Concept Proven"
Paper presented by: W. F. Wall, Diagnostic System Group Engineer/
J. T. Mash, Aircraft Dev Engineer Spec
Lockheed, Georgia Company
3. "Air Lifter" - Periodical Spring 1969
Published by Lockheed - Georgia Company
Article: "MDAR" by C. F. Elgin and J. T. Mash

SPACE STATION

1. Walt Rabinoff, McDonnell Douglas Corporation, Huntington Beach, Calif.
2. "Space Station Electronics Subsystem Study" - 28 April 1970
DRL 8, Volume V, Book 4
Prepared by: McDonnell Douglas Astronautics Company
3. "Information Management Study" - 28 April 1970
DRL 8, Volume V, Book 3
Prepared by: McDonnell Douglas Astronautics Company
4. "Space Station Definition MSFC-DRL-160 Line Item 8 - Volume V
Subsystems, Book 4 Electronics" - July 1970
Prepared by: McDonnell Douglas Astronautics Company
5. "Space Station Crew Operations Definitions MSFC-DRL-160 Line Item 9"
Prepared by: McDonnell Douglas Astronautics Company
6. "Crew Operations Definitions MSFC-DRL-160 Line Item 9" - August 1970
Prepared by: McDonnell Douglas Astronautics Company
7. "Space Station Development Definition Software Requirements Documents"
DRL 18 Volume III - 1 May 1970
Prepared by: McDonnell Douglas Astronautics Company
8. "Information Management System Study - Space Station Program Volumes I
and II" - April 1970 - IBM No. 70-K34-0001
Prepared by: IBM, Federal Systems Division, Space Systems Center,
Huntsville, Alabama

F-111D

1. Local General Dynamics Office - Huntsville
2. Periodical: "Journal of Aircraft," Vol 6 No. 5
Title: "F-111D Computer Complex"
Authors: D. H. Daggett and R. Q. Lea
General Dynamics, Fort Worth
3. Phone call: D. H. Daggett, General Dynamics, Fort Worth, Texas
IBM Local on 4 pi-TC1

AWACS

McDonnell Douglas Corporation
E. Witten, Long Beach
P. D. Janes, Huntington Beach

F-15

McDonnell Douglas Corporation, F-15 Project Office, St. Louis

SPACE SHUTTLE

1. "Integrated Avionics System Space Shuttle Program" - 21 December 1969
Prepared by: McDonnell Douglas Corporation
2. "Integrated Avionics System Tradeoffs Space Shuttle Program" -
11 February 1970
Prepared by: McDonnell Douglas Corporation
3. Plant visits: J. K. Gates and M. Cowan - McDonnell Douglas Corporation,
St. Louis, Missouri
Subject: Space Shuttle Control and Display Application Usages, and
Technology
4. "Proposal to Accomplish Phase B - Space Shuttle Program" -
30 March 1970 - Report MDC E0120
Prepared by: McDonnell Douglas Corporation
5. "Integral Launch and Reentry Vehicle System - Volume 1/Book 1" -
November 1969 - Report No. MDC E0649
Prepared by: McDonnell Douglas Corporation
6. Paper: "Shuttle News" - Vol 1/No. 1 - March 1970
Publisher: McDonnell Douglas Corporation
7. Periodical: "Information Display" - Vol 7/No. 4 - April 1970
8. Article: "Holography in Airborne Display System"
Author:
 - T. J. Harris
 - R. S. Schools
 - G. T. Sincerbox
 - D. Hanna
 - D. Delay

} IBM System Dev Div, Poughkeepsie, NY

} Fed Sys Div, Oswego, NY

GENERAL

1. George E. Townsend, North American Rockwell Corporation,
Huntsville, Alabama
Subject: State-of-the-art-Guidance, Navigation and Control Functions
2. "Supplement AIAA Technology for Manned Planetary Missions Meeting" -
March 4 - 6, 1968 - New Orleans, Louisiana
Prepared by: American Institute of Aeronautics and Astronautics
3. "Analysis for the Requirements for Computer Control and Data Processing
Experiment Subsystems - Experiment Control and Data Processing Require-
ments Specification Report" - 15 May 1970
TM-(L)-HU-033/000/00
Prepared by: System Development Corporation, Santa Monica, California
4. "Astronaut/Computer Communication Study - Phase A Task 1 Report" -
May 1970 - MDC-G-388
Prepared by: McDonnell Douglas Astronautics Company
5. "Astronaut/Computer Communication Study - Phase A Task 2 Report" -
September 1970 - MDC-G0492A
Prepared by: McDonnell Douglas Astronautics Company
6. Paper: "Trends in Computer Technology" by G. M. Harold -
Aerospace Research Data Control, Data Corporation
7. Brochure: "Control Data Plasma Display"
No. 100, 729B Litho in USA 9/69 - Control Data Corporation
8. Messrs T. W. Miller and R. G. Koppang, MDAC-WD, Data Applicable to the
"Syntax Directed Compiler" Consultants
9. Mr. J. T. Hine, MDAC-WD, Software Consultant
10. Messrs P. D. Janes and C. L. Hoyt, MDAC-WD, Hardware Consultants
11. Mr. W. Schramm, MDAC-ED, Space Shuttle Simulator Consultant
12. J. S. Miller, et al, Multiprocessor Computer System Study,
Intermetrics, Inc., March 1970
13. Space Station Program Development Definition, MSFC-DRL-160
Line Item 18, Volume III, Software Requirements Document
14. Computer Display Review, No. 499, Keydata Corporation
15. Saturn V. Flight Manual SA509, dated 1 January 1971
16. Language Considerations for Manned Space Flight, dated 7 October 1969,
IBM Boston Programming Center

17. Modern Data, July 1970 Issue, Interactive CRT Display Terminals by Brick and Chase
18. IBM Systems Journal, "Interactive Graphics in Data Processing"
19. On Man-Computer Interaction: A Model and Time Related Issues, J. R. Carbonell; Bolt, Beranek, and Newman, Inc., September 1967
20. "Recent Advances in Display Media," NASA SP-159, A Symposium held in Cambridge, Massachusetts, September 19-20 1967
21. "Display Problems in Instrument Approach and Landing," Lewis F. Hanes and Malcolm L. Ritchie, Ritchie, Inc. Technical Report RTD-TDR-63-4000, June 1965, Air Force Flight Dynamics Laboratory Research and Technology Division, Air Force Systems Command, Wright-Patterson Air Force Base, Ohio
22. "Preliminary Development of a Solid-State Matrix Display," B. J. Lechner, et al, Radio Corporation of America RCA Laboratories, Technical Report AFFDL-TR-66-55, January 1967, Air Force Dynamics Laboratory Research and Technology Division, Air Force Systems Command, Wright-Patterson Air Force Base, Ohio
23. "Solid State Display Techniques," Carlton J. Peterson, Captain, USAF, AFFDL-TR-66-123, October 1966, Air Force Flight Dynamics Laboratory Research and Technology Division, Air Force Systems Command, Wright-Patterson Air Force Base, Ohio
24. "Development of a Solid State Matrix Display," B. J. Lechner, G. W. Taylor and J. Tults, RCA, Technical Report AFFDL-TR-67-71, July 1967, Air Force Flight Dynamics Laboratory Directorate of Laboratories, Air Force Systems Command, Wright-Patterson Air Force Base, Ohio
25. "Control Display Subsystem for Manned Advance Aeronautical and Aerospace Vehicles," E. H. King, M. W. Story, et al, Lear Siegler, Incorporated Technical Report No. AFFDL-TR-66-182, December 1966
26. "Development of High Contrast Cathode Ray Tubes," W. C. Hoffman, P. Damon, G. Slocum, K. Fujimoto, Hughes Aircraft Company Technical Report AFFDL-TR-67-77, July 1967, Air Force Flight Dynamics Laboratory, Directorate of Laboratories, Air Force Systems Command, Wright-Patterson Air Force Base, Ohio
27. "Human Factors Problems in Computer-Generated Graphic Displays," Joseph E. Barmack, et al, Institute for Defense Analyses, April 1966
28. "Survey of DAC Commercial Aircraft Techniques in Man/Computer Communications," MDAC memorandum A3-830-BEFO-016, dated 26 February 1971

Appendix A
FUNCTION CATEGORIES

This appendix discusses the nine function categories determined in Phase A, Task 2. The intent is to define the functions comprising each function category and to emphasize the requirement(s) for said functions. The format for this discussion will be to present a pictorial overview of the function category being described with accompanying prose citing examples for clarity. Many of the functions described herein are proposed or anticipated to be operational on manned spacecraft during the time frame considered for this study. (1973 to 1980.) For this reason all functions are discussed collectively as if they were operational today.

NOTE

Figure A-1 shows the functions which must be performed by the pilot and/or copilot of the proposed Space Shuttle orbiter. It also provides emphasis on the control and display capability needed to perform the onboard functions.

A.1 MISSION CONTROL

Mission control (Figure A-2) encompasses all functions and subfunctions that must be performed to assure crew safety and mission success. Basic functions associated with this category are (1) mission planning, (2) mission performance monitoring, and (3) commanding. Currently, the performance of these functions is a ground support responsibility, but to achieve the degree of vehicle autonomy being proposed for Space Shuttle and Space Station and to accomplish future manned space missions (e.g., Mars Fly-by, Grand Tour, etc.), mission control functions will be required to be performed onboard the in-flight vehicle. This is substantiated by the economics and logistics involved in maintaining a dedicated ground support mission control center.

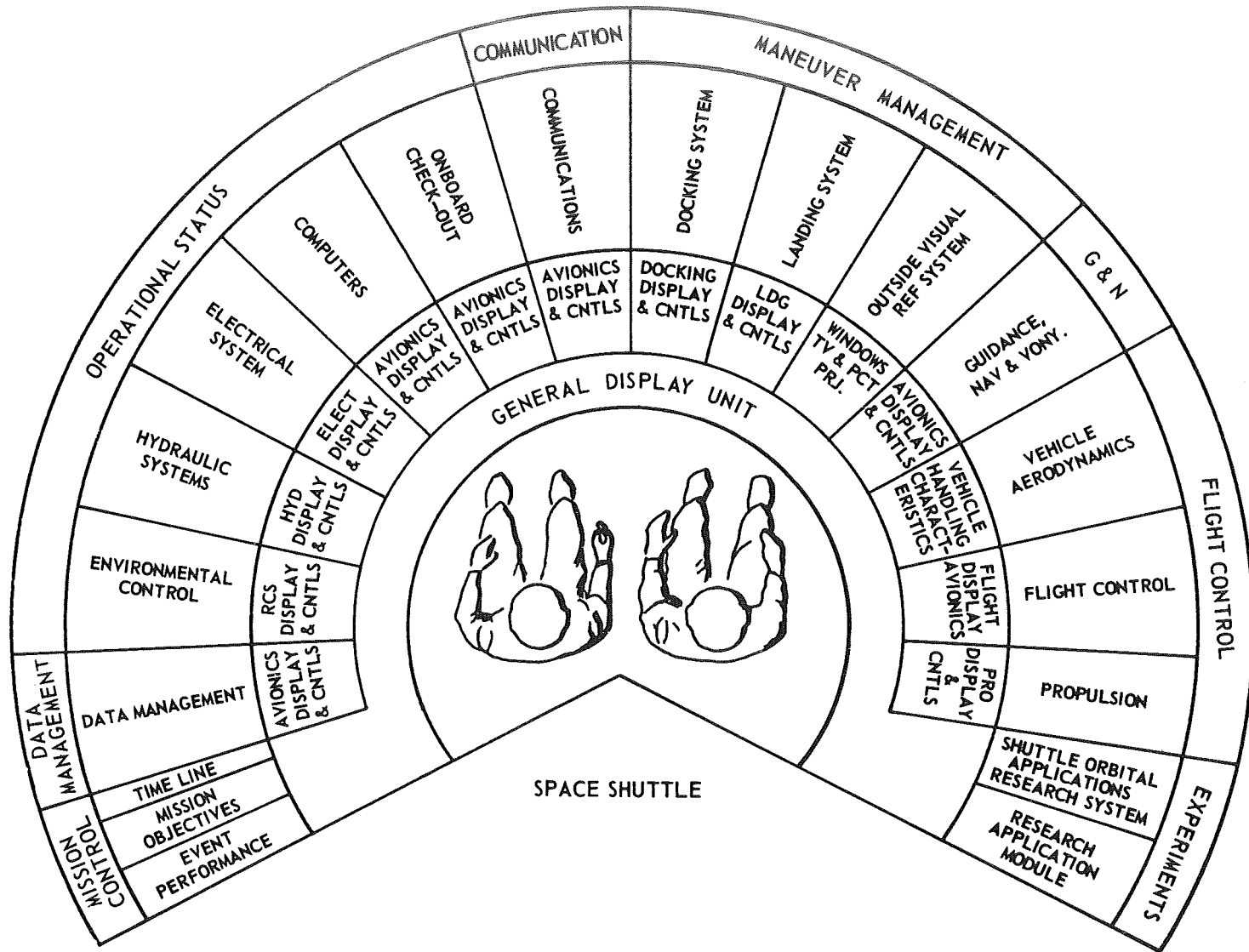


Figure A-1. Vehicle Subsystem by Function Category

A-3

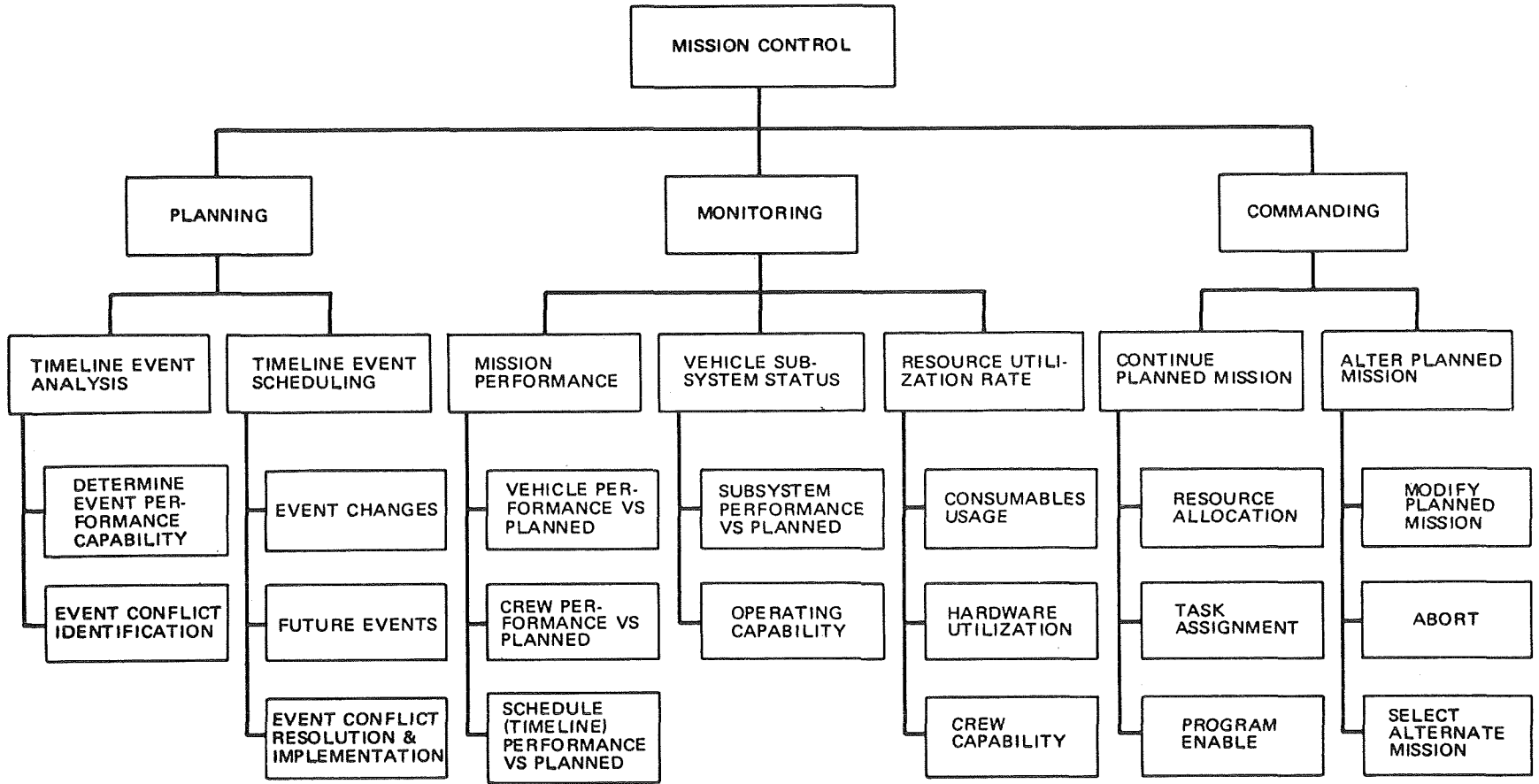


Figure A-2. Mission Control Function Breakdown

A.1.1 Mission Planning

Mission planning functions are primarily concerned with the generation and maintenance of the mission timeline, a chronological list of mission events to be performed. The mission timeline also contains the following data related to event performance requirements:

- A. Event support requirements (i.e., subsystems, resources, etc.).
- B. The estimated status of the vehicle and subsystem before and after an event is performed.
- C. Resources and expendables required by the event.
- D. Crew participation requirements.
- E. Ephemeris data associated with the vehicle prior to event execution.
- F. Other necessary data required for judgment decision involving event execution.

The mission timeline is also used as a "yardstick" for measuring mission performance (i.e., actual requirements versus planned).

Timeline Event Analysis--Before scheduling a timeline event the event requirements must be identified to establish the capability of the vehicle to perform the event. Once this is established, an analysis is conducted of other events already scheduled to identify conflicts which could effect event scheduling and performance (i.e., computing for resources, counteracting events, etc.).

Timeline Event Scheduling--Once the vehicle is found capable of performing the event, and the timeline can accommodate the event, the event is scheduled. Event scheduling is a tedious task for any event change. A critical analysis must be made of the timeline to determine exact times available for event

execution and should the event being scheduled require a specific time, scheduling conflicts (i.e., resource and subsystem availability) must be resolved via predefined shifting or rescheduling algorithms.

A.1.2 Mission Monitoring

The vehicle "commander" must continuously monitor mission performance, vehicle subsystem status and resource utilization rates. In this manner he is able to make decisions concerning the mission. For past manned space programs (Gemini and Apollo) this activity was performed on the ground with the assistance of a large mission control staff. For future manned space programs this function will be performed onboard with the assistance of the onboard computer and computer software.

Mission Performance--The performance of the mission is determined by comparing actual data generated by the mission in progress with planned data generated prior to the mission and predicting mission success based on timeline event requirements and new data rates. These functions are performed automatically by the onboard computer by comparing the mission timeline with data obtained directly from vehicle subsystems. In this manner vehicle, crew and schedule performance are determined statistically and displayed for information and action by the flight commander.

Vehicle Subsystem Status--Monitoring vehicle subsystem status is another function automatically performed by the computer for the "flight commander."

Resource Utilization Rates--The flight commander will monitor the utilization rates of vehicle hardware and consumables for statistics generated by the computer. The raw data used in computing these statistics is obtained directly from hardware sensors. Interpretation of the computed statistics provides visibility of resource utilization performance. Crew capability and availability will be maintained in the computer's data base of skills inventory and duty rosters.

A.1.3 Mission Commanding

Commanding functions are those decisions made by the flight commander in determining the future of the mission being performed. The commander makes the final decision whether or not to continue or alter the planned mission and implements his decision by issuing commands to effected areas. Commands can be verbal orders to crew members, but in many instances commands will be computer oriented functions which are initiated by the commander from the control and display panel on the mission control console. (Figure A-3 depicts a version of the proposed Space Station mission control console.)

Continue Planned Mission--Decisions made to continue the planned mission will be of the more routine type commanding functions that allocate resources, assign tasks and enable computer programs required to maintain the planned mission.

Alter Planned Mission--If the flight commander decides a mission change is required, he must determine (1) if a modification to the planned mission is adequate, (2) if an alternate mission is possible, or (3) if an abort situation exists. The ultimate decision will be made by the commander assisting the mission change requirement and the capability of the vehicle and crew to perform.

A.2 DATA MANAGEMENT

Onboard manned spacecraft data management (Figure A-4) is concerned with the computer functions executed in acquiring, dissemination, storing, and retrieving data. These functions are stimulated at the request of executing software (i.e., application's programs or operating system) or interrupting hardware. The significance of this statement and the point intended to be emphasized is that no direct crew interface is identified with data management functions. Onboard data management functions are analagous to a sophisticated "input/output" control system with all crew requests being made through interfacing software.

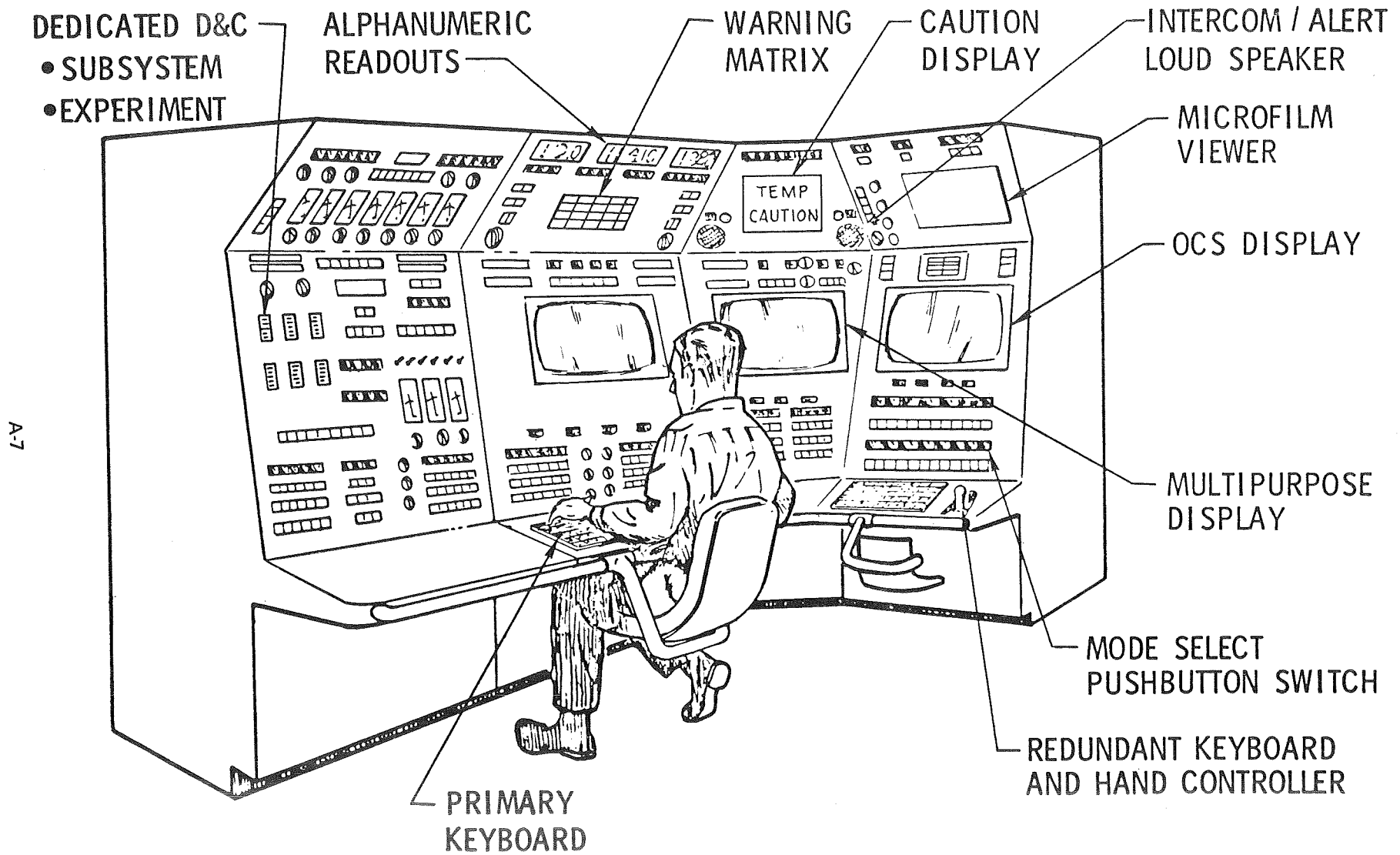


Figure A 3. Secondary Command and Control Center

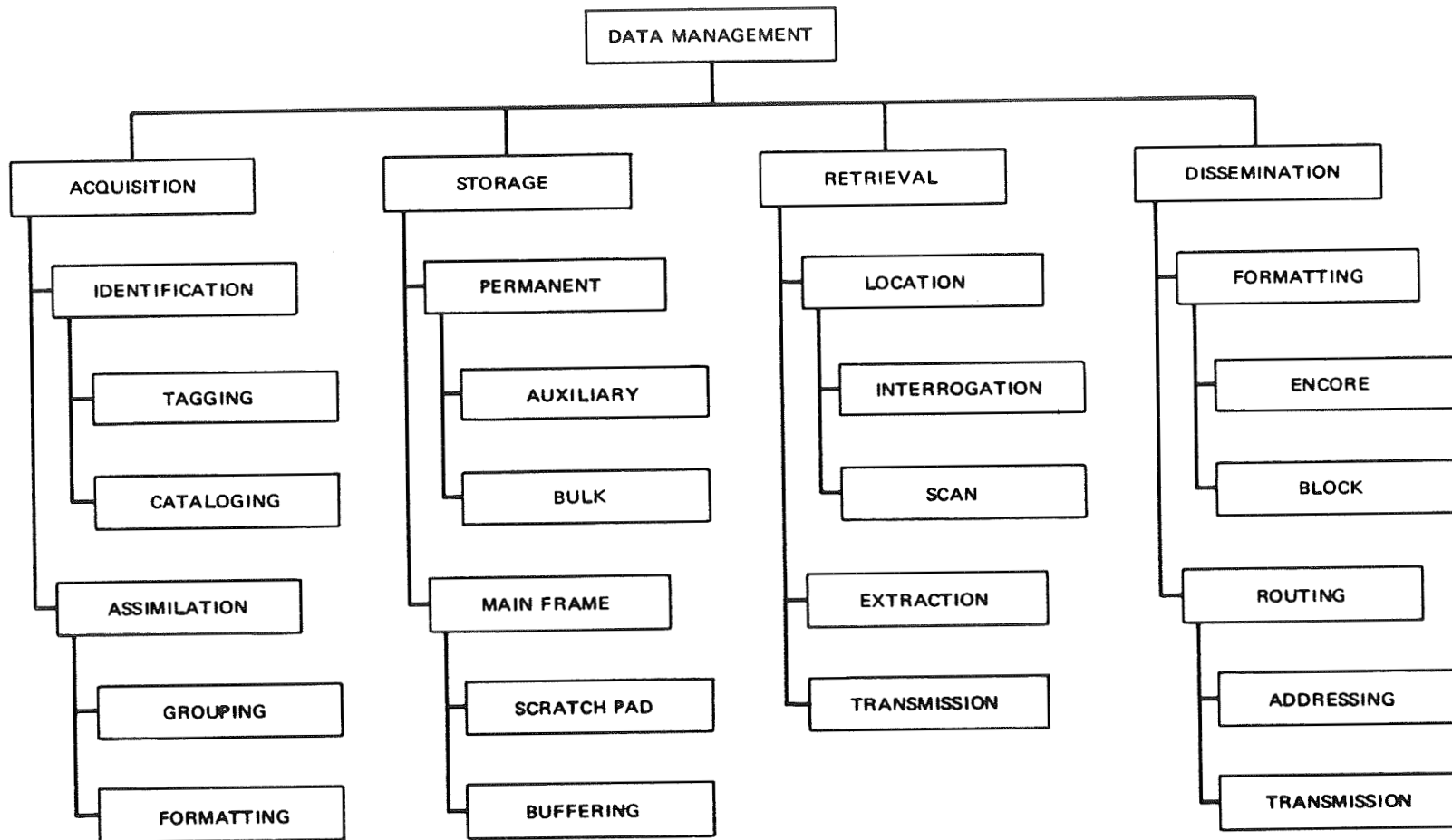


Figure A-4. Data Management Function Breakdown

A.2.1 Data Acquisition

Onboard data acquisition deals with data as it is received from the data buses. As there is no absolute way of predicting the exact type, amount, or sequence of data that will be obtained from the data bus(es), the responsibility of data identification and assimilation is assigned to the data acquisition procedures.

Data Identification--As incoming data is identified, a data management identification code is generated and "tagged" to the data for internal processing use by data management procedures. To minimize data location problems and to optimize data retrieval processing problems, an entry is made in the data availability index. This is referred to as cataloging.

Data Assimilation--Due to the random and sporadic manner in which data is received from the data bus(es), assimilating data in a working area of storage becomes a necessary function. This provides a convenient and simple method for grouping and formatting data without interfering with other processing functions. Data grouping functions assure data integrity by verifying reception of the total message, while data formatting optimizes data core requirements for efficient internal computer processing.

A.2.2 Data Storage Functions

Data storage functions are required to manage the variety of storage media supporting the onboard computer. Future onboard computers are expected to have auxiliary and bulk storage capabilities as well as increased main frame storage. Data storage tasks must maintain a continuance inventory of storage utilization and allocate remaining storage as required.

Permanent--Permanent storage will be provided in various forms (i.e., discs, tapes, etc.). Auxiliary storage is the faster access storage which is expected to accommodate the operating system, subroutine library, and frequently used data parameters and/or files. Bulk storage will store large volumes of data with limited access requirements and will be slow access storage.

Main Frame--Main frame storage will provide "scratch-pad" memory for intermediate processing by application programs and buffering areas for file blocking and processing functions.

A.2.3 Data Dissemination

Data dissemination is the term associated with the computer functions that process all requests for the computer to transmit data to remote stations on a data bus. Such a request can originate from the remote station itself, from a different remote station, or from functions performed by the operating system. Data dissemination functions analyze the transmission request to determine format and routine tasks to be performed. Example: A message is originated as a G&N function to display current G&N data on mission control and flight control displays. Data dissemination functions interpret the request, format the message for line transmission, and route the message (i.e., add the hardware address of the remote station(s), place the message in a data transmission buffer, and initiate hardware transmission).

Data Formatting--Data encoding and message blocking are performed by the software to ensure hardware and line compatibility, respectively.

Data Routing--Data dissemination functions provide the message hardware and line addressing data and initiate transmission sequence.

A.2.4 Data Retrieval

Data retrieval functions perform the more obvious data management tasks of locating, extracting, and transmitting data for subsequent computer processing.

Location--Data location involves interrogating the data availability catalog to rapidly determine the existence of data and scanning the data location index for absolute data address should the data exist.

Extraction--Once the desired data has been located, it must be extracted from the data file as requested.

Transmission--The data is then transmitted to a working area of main core for application program processing.

A.3 COMMUNICATIONS

Data management discussed one type of onboard communication, online computer communication with remote stations located on a data bus. This section discusses the remaining types of communication which deal with the more conventional communication techniques (i.e., radio, television, telemetry, etc.) (Figure A-5). Examples of this type of communications' requirement would be:

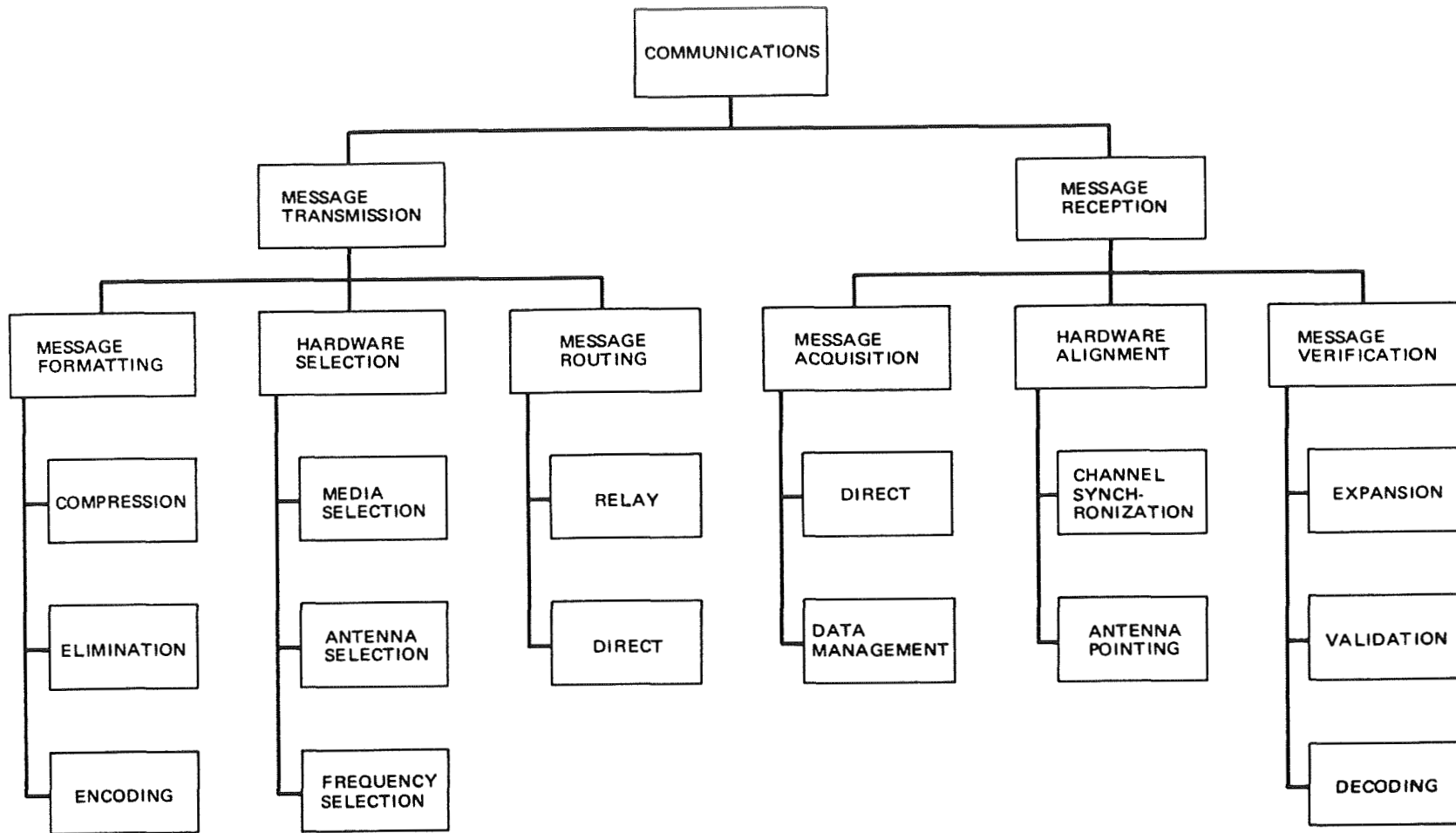
- A. Performing coordinated crew task onboard.
- B. Performing landing/docking maneuvers.
- C. Ground support communication.
- D. Performing external vehicle activities.

For discussion and study purposes onboard communications are divided into two functions, transmission and reception.

A.3.1 Message Transmission

The onboard computer is expected to perform most of the intricate and time consuming functions associated with message transmissions. The communications operator determines that transmission should occur and the messages to be transmitted. The computer will assist with message formatting, hardware selection, and message routing. The communications operator selects from a menu of options presented by the computer the communications method for performing his task.

Message Formatting--One of the major problems confronting future manned space missions is infinite data generation capability existing onboard future space vehicles versus the limited storage and transmission capability for saving



FigureA-5. Communications Function Breakdown

data. Communications software will, of necessity, require data compression, elimination and encoding algorithms for limiting the amount of data being transmitted. These terms may appear to connote similar meanings but each algorithm performs a unique task.

- A. Data Compression--Those algorithms reduce the bit requirements for digital character representation. Example: If a transmission were known to contain all numeric data, a character could be represented in four bits as opposed to eight or the binary representation of the numeric field could be transmitted.

- B. Data Elimination--Algorithms eliminate similar or redundant data. Example: If a thermometer were being sampled and a constant reading was obtained, the first and last samples with respective times would suffice for data reconstruction purposes.

- C. Data Encoding--Algorithms abbreviate data. Example: If a limited number (10) of data parameters are being transmitted in random sequence, a single unique character would adequately identify the data.

Hardware Selection--Selection of data transmission hardware is contingent on the type of data to be transmitted and the receiving capability of the station being contracted. For this reason and the uncertain attitude of the vehicle when message transmission occurs, message transmission systems must be coordinated. This means the communication media, compatible antenna and transmission frequency for data transmission must be selected and aligned before successful transmission can occur. The computer performs many of these tasks at the direction of the communications operator.

Message Routing--Message routing functions determine the route a message must take to reach the addressee. Message will be sent directly to the station for which the message intended, but there will be times when contact is required with stations not in transmission range of the vehicle. For this type of transmission, message routing will determine the message relay route for

transmission. This route may utilize one or a combination of data relay satellites, ground support stations, or other manned space vehicles.

A.3.2 Message Reception

Message reception functions are required for message reception, communications hardware alignment and message verification. These functions are initiated prior to station contact and continue until all data received has been evaluated/interpreted either by crew members or the computer system.

Message Acquisition--Message acquisition can be performed directly by the crew or by the computer data management system (i.e., teleprocessing techniques) providing a requirement for a wide range of interactive tasks. Direct message acquisition could either be visual or data obtained for immediate action or information. Even though the actual data communication is coordinated with the assistance of the computer, the text of the message travels directly to the addressee. Data management, on the other hand, intercepts messages for computer interpretation prior to disseminating the data.

Hardware Alignment--Onboard message reception requires hardware alignment interactive functions. The communications operator must synchronize the receiving channel with that of the transmitting station and optimize the position of the receiving antenna for "static free" reception. Hardware alignment functions are accomplished via the onboard computer at the direction of the communications operator.

Message Verification--Message verification functions are performed by the data receiving station to verify the integrity of the message being received. Messages received directly (i.e., bypassing the computer system) are verified by the receiver at the time of reception or inspection. Message received the data management system are routed to communications software applications programs for processing. Software procedures analyze the message and perform the following functions as required:

- A. Expand the data by reversing the data compression and elimination algorithms applied prior to transmission.

- B. Validate the data by computing and comparing data check words transmitted as a part of the message text.
- C. Decode data by replacing predefined data abbreviations with respective meaningful data parameters.

A.4 FLIGHT CONTROL

Flight control (Figure A-6) encompasses those functions performed in flying a space vehicle from launch to landing. Flight control interfaces closely with functions described under mission control, guidance and navigation, and maneuver management. The distinction between these functions and those described herein is that flight control functions are concerned with the immediate and near-future problems associated with maintaining a given flight trajectory to achieve the mission objective. The major functions identified with flight control are:

- A. Performing trajectory/flight path analysis.
- B. Implementing trajectory changes.
- C. Evaluating trajectory changes.

A.4.1 Trajectory Flightpath Analysis

In the course of maintaining a given flightpath/trajectory continuous analysis must be made of the vehicle's actual position versus required position. Should a flightpath deviation of a magnitude requiring corrective action be detected, an in-depth study of the course change requirements must be made with particular consideration given to resources requirements and subsystem availability.

Course Change Analysis--Course change analysis is instigated to determine the corrective action required. Inquiries are made of the guidance and navigation, mission control, and maneuver management functions in determining course correction requirements. After determining course correction requirements, a further study is performed to determine the optimum time to execute the corrective functions. Such optimization must take full advantage of the vehicle's position, external vehicular forces, and percent probability of success.

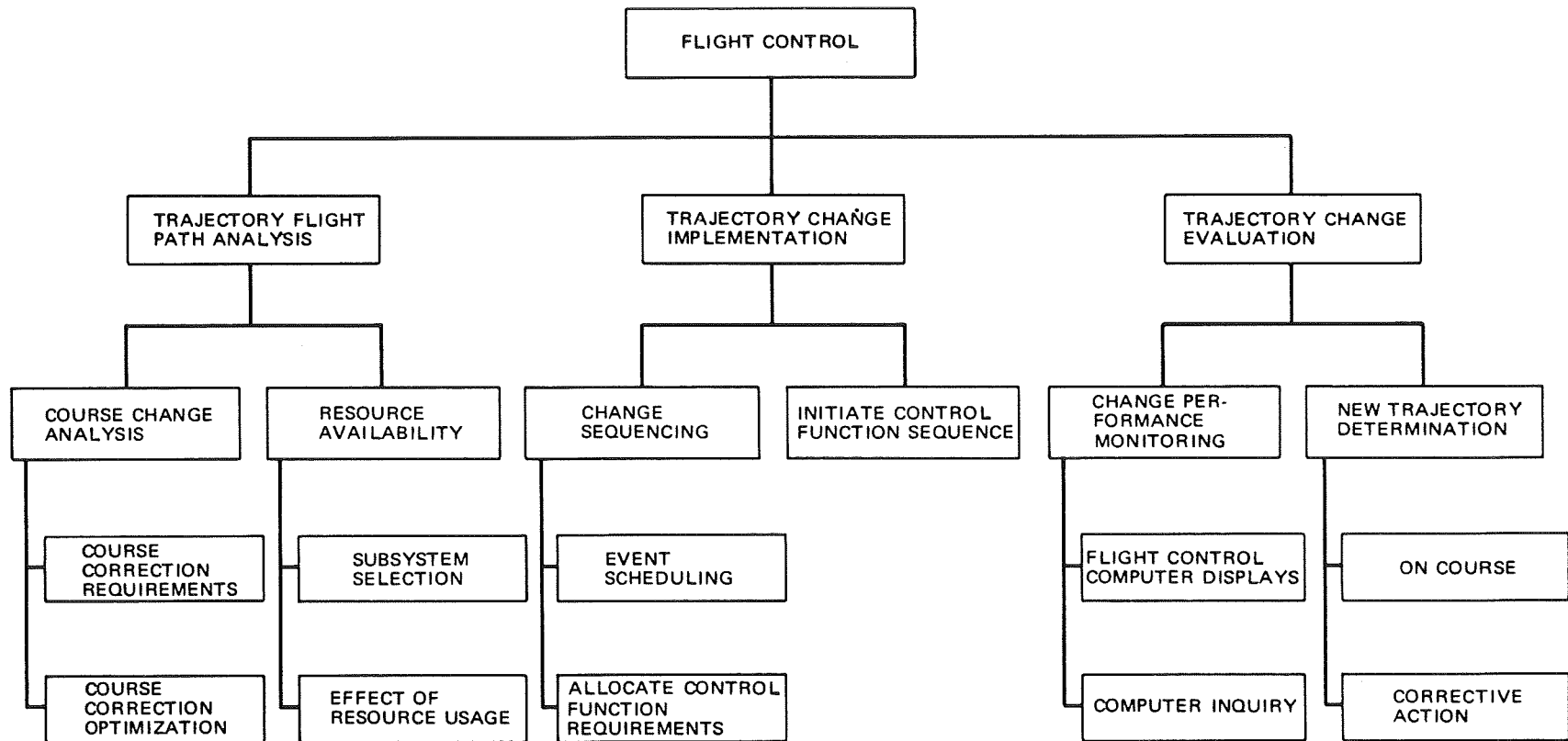


Figure A-6. Flight Control Function Breakdown

Resource Availability--In conjunction with the analysis performed to determine course change requirements, functions must be executed to ascertain subsystem and resource utilization. It is important that the subsystem selected be capable of performing the required maneuver and not consume excessive quantities of valuable resources. The effects of resource usages must be considered in determining ability of the vehicle to complete the mission without endangering the crew.

A.4.2 Trajectory Change Implementation

After determining a trajectory change is to be implemented, sequencing functions must be determined and initiated for performing the change. Sequencing determination is essential and requires careful planning to assure no step will be overlooked. In some instances a "dry-run" of the planned sequence will be required.

Change Sequencing--A major function in establishing a change sequence is event scheduling. Event scheduling coordinates all subsystem and resource requirements to assure availability and timely execution. Event execution timing is critical. As event scheduling is accomplished, control function requirements are allocated. Control function allocation may be performed by the flight controller in indicating to the onboard computer the sources of control data (i.e., autopilot, hand controls, rudders, etc.).

Initiate Control Function Sequence--The flight controller performs the minimum function of initiating the control function sequence that starts the trajectory change maneuver. From that point on the flightpath change is performed by the computer under the auspices of the flight controller. The degree of interaction required by the flight controller during the execution of a trajectory change is a discretionary decision made by the flight controller based on trajectory change requirements.

A.4.3 Trajectory Change Evaluation

The initiation of a trajectory change sequence signals the time to begin trajectory change evaluation. The flight controller, utilizing the onboard

computer, monitors the performance of the change being executed by viewing critical flight control data displayed on the control and display panel. Features of the control and display panel permit manual control of the changes in progress, if required. At completion of the trajectory change the flight controller makes a detailed evaluation of the new trajectory. Many of the tasks performed evaluating the vehicle's new trajectory are identical to those discussed in Subsection A.4.1.

Change Performance Monitoring--One of the major tools available to the flight controller is the onboard computer and its display capability. Utilizing this tool the flight controller can monitor the performance of the vehicle. He does this by directing the computer to display and keep current that information of interest. He also queries the computer to obtain data from the computer's data base as required in performing his job duties.

New Trajectory Determination--The flight controller determines the vehicle's trajectory, as altered by the change, by interrogating the guidance and navigation system. If he has achieved the desired course, he enters a flight trajectory monitoring mode, but if he has not, he begins again the sequence of functions to implement a trajectory change.

A.5 GUIDANCE AND NAVIGATION

Guidance and navigation (G&N) functions (Figure A-7) determine the trajectory and guidance requirements for flying a manned space vehicle to a predetermined destination. To perform these functions measurement data indicating the vehicle's attitude and position must be determined. This data, in turn, is reduced and processed to compute the vehicle's velocity and estimated actual trajectory. Continued processing yields the trajectory deviation, if any, and control requirements for deviation compensation. The results of G&N calculations are used by mission control, flight control, and maneuver management in performing their respective functions.

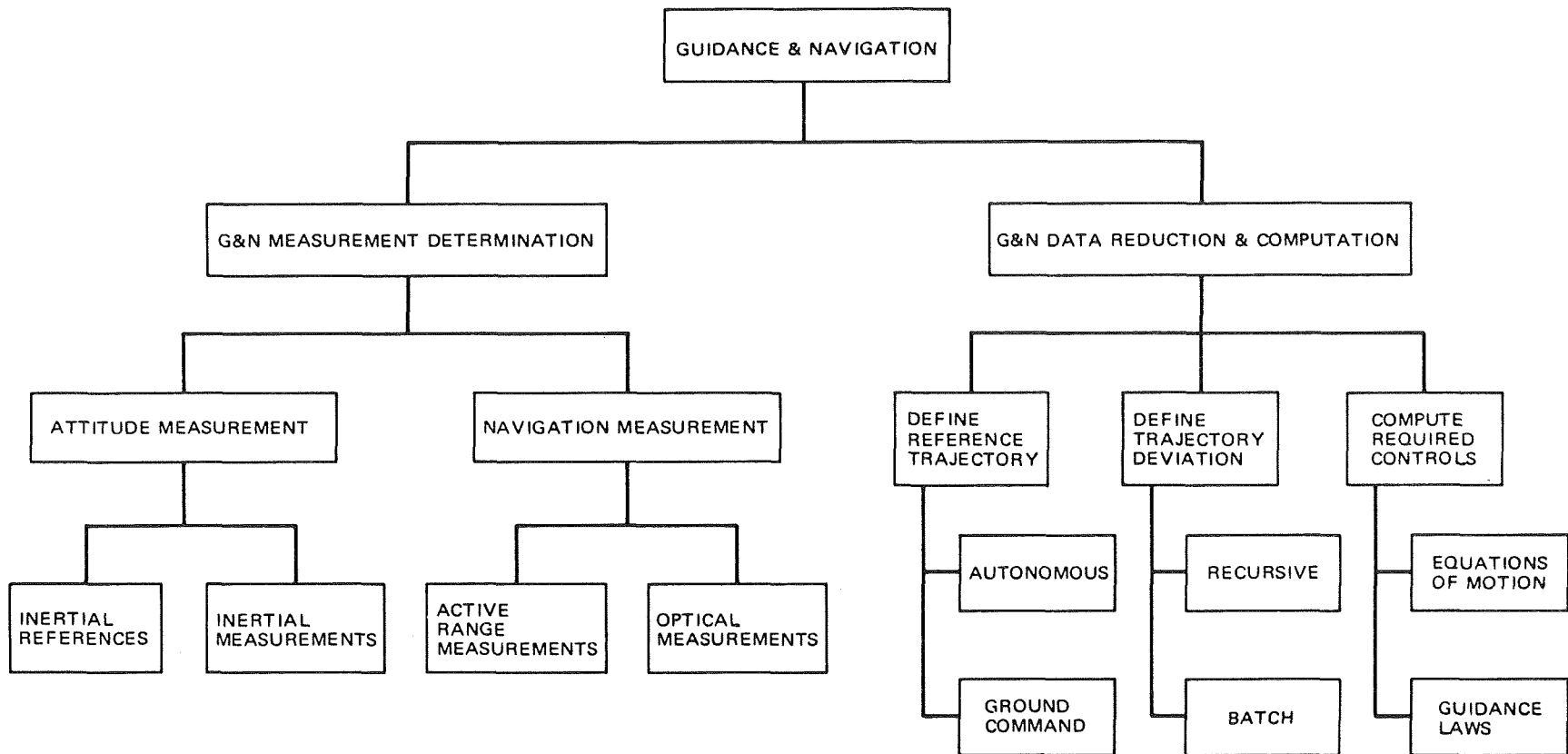


Figure A-7. Guidance & Navigation Function Breakdown

A.5.1 G&N Measurement Determination

One of the first functions to be performed is determining the vehicle's attitude. This data is used in stabilizing the vehicle's attitude so that navigation measurement functions can be initiated.

Attitude Measurements--Attitude measurements are obtained utilizing both an inertial reference and an inertial measurement unit (IMU). Angular measurements are made from optical sightings of predetermined stars. An IMU is a gyro system, either gimbaleed or strap-down, which generates measurement data of the attitude of the vehicle with respect to the reference derived from the stars.

Navigation Measurements--Functions performed to obtain navigation measurements utilize active range (i.e., radar) or optical hardware. Radar is used in performing station keeping and docking type functions (i.e., radar is used to determine the relative position of the vehicle with respect to other free flying bodies). Radar can also be used in determining the vehicle's position relative to the earth when the vehicle achieves adequate distance at which a meaningful measurement can be obtained. In most instances an optical system is used to determine the vehicle's position relative to the earth or any other celestial body sufficiently close to the vehicle. Optical techniques determine the vehicle's relative position from optical sightings of earth land marks or celestial bodies.

A.5.2 G&N Data Reduction and Computation

Data obtained from the performance of G&N measurement determination functions is reduced and processed to determine (1) the actual trajectory of the vehicle, (2) the trajectory deviation, and (3) the controls required to correct the deviation. G&N data reduction and computation functions are executed automatically by the onboard computer system, and maintain current data parameters for use in performing mission control, flight control, and maneuver management functions. These functions can also be interactively controlled for mission planning. An example would be to establish a new trajectory for an alternate mission not contained in the mission timeline.

Define Reference Trajectory---"Recursive" and "batch" are terms applied to techniques used in computing a space vehicle's trajectory deviation (i.e., the difference between the reference and planned trajectory). The recursive technique is interactive in approach and lends itself nicely to a real-time environment with limited computing resources. The batch technique accumulates data points and increases in computation time requirements with the addition of each data point. The recursive technique is adequate for the job that must be accomplished and is recommended for limited onboard computer facilities. This technique can also be used in a nonreal time mode to assist in defining trajectories in support of mission planning.

Compute Required Controls---Once the vehicle's reference trajectory has been computed and the trajectory deviation established, the G&N functions apply the equations of motion and guidance laws in computing the control requirements for achieving the planned trajectory. Since all G&N functions are educated guesses based on scientific computation, the calculation of control requirements becomes an interactive technique with the flight controller in the "loop."

A.6 EXPERIMENTS

Onboard space experimentation is one of the more interactive functions (Figure A-8) to be performed on future manned space vehicles. The nature of experimentation prohibits discussion of unique functions required in performing individual experiments.

This is due to the current number of planned space experiments, the continuous identification of new experiments, and to the unique solution requirements of each experiment. For this reason experiments are discussed collectively so as not to prohibit growth or limit solution requirements. Basic experiment functions are to initiate, execute, and terminate the experiment within the constraints of the mission environment. These functions are clearly separated from other mission functions and are considered by some organizations the responsibility of a mission director. This report does not intend to argue this point, but to discuss the functions named above.

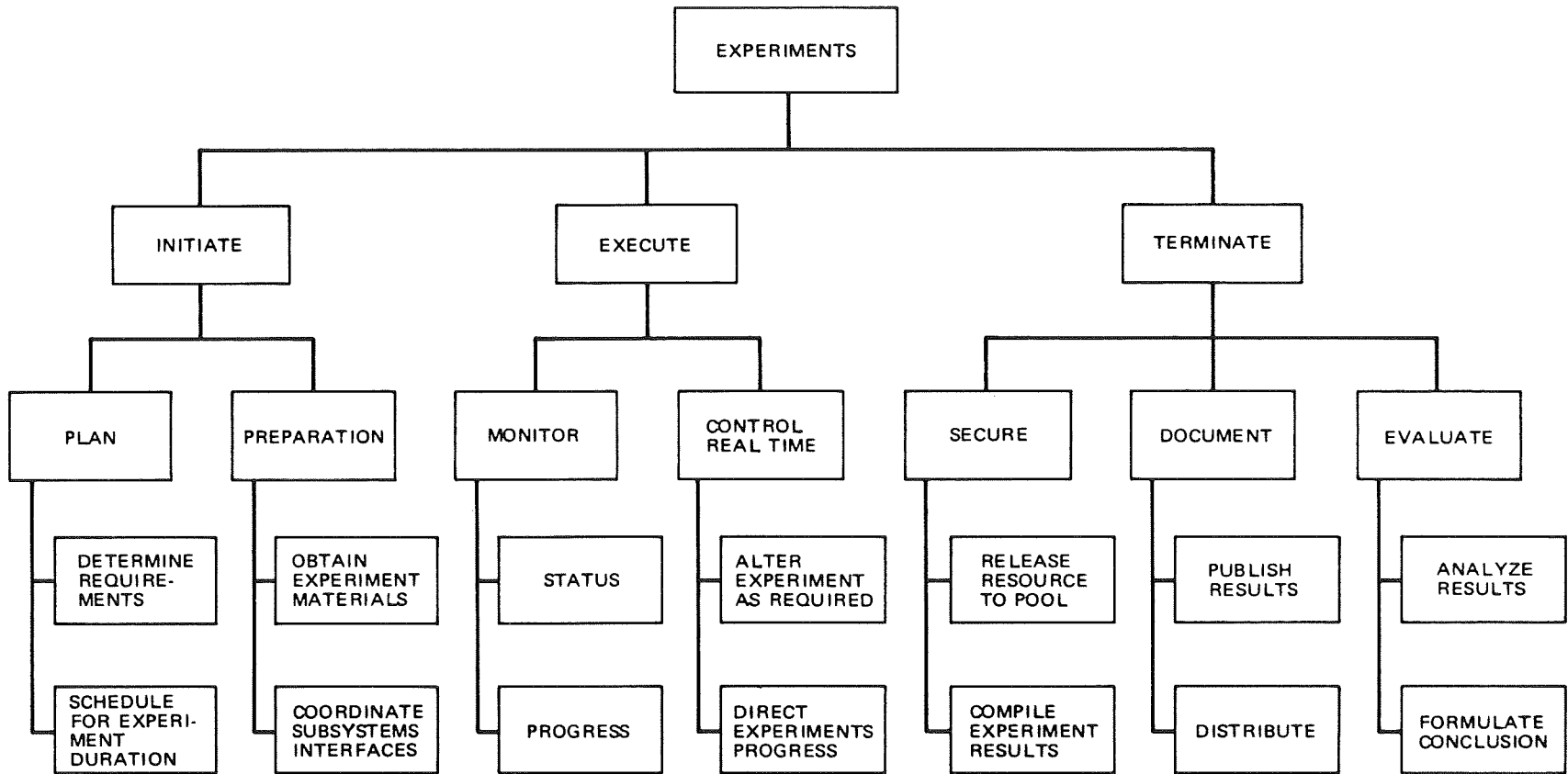


Figure A-8. Experiments Function Breakdown

A.6.1 Initiate

This interactive function assists the space scientist in planning and preparing his experiment. This function assures a coordinated effort and that the experiment can be completed once started.

Plan--Experiment planning calls for a detailed determination of the experiment requirements (i.e., subsystem support, expendables, hardware, etc.) and scheduling needs. An experiment must be scheduled considering the total onboard requirements with respect to the limited resources. Gross scheduling requirements are performed by the space scientist with the computer determining detail schedule requirements and conflicts.

Preparation--Once the experiment is planned the space scientist obtains experiment materials and prepares his laboratory for execution. He coordinates subsystem interface requirements by verifying connection and the operating status of his experiment related equipment.

A.6.2 Execute

The space scientist must monitor and maintain real-time control over his experiment during execution. These functions provide him with the visibility he needs in deciding the direction of his experiment.

Monitor--Monitoring the experiment's status and progress is a computer assisted function which frees the experimenter for other duties. While monitoring the experiment the computer records pertinent data for reference at future times.

Control Real-Time--Real-time control of onboard experiments will provide the space scientist with a "computerized" approach executing his experiment. He will be able to direct or alter his experiment's progress by initiating stimuli from his experiment computer control panel.

A.6.3 Terminate

Experiment termination is an important phase of experiment processing. The basic functions involve securing the experiment, documenting the experiment, and evaluating the experiment results. The space scientist will find the onboard computer a valuable tool in performing these functions.

Secure--As the experiment is completed the space scientist will release his hardware interfacing subsystem and unused resources to the availability pool for use by other experiments. He also organizes his experiment data for documentation and evaluation purpose by compiling experiment results and indexing for easy access.

Evaluate--Experiment evaluation becomes a simplified task since the computer has recorded the data during execution monitoring. The space scientist can now analyze experiment results by automatically messaging the computerized data to formulate experiment conclusion for publication.

Document--Documenting will be a manual task with exception of statistics and charts which can be automatically generated from the computerized data. Once documented the experiment results are distributed for consideration and interpretation by other space scientists having similar interest.

A.7 MANEUVER MANAGEMENT

Maneuver management (Figure A-9) is a unique category of functions established to coordinate the various interfacing functions required to identify, define, and execute a vehicle maneuver. In many instances a maneuver requirement will be identified by mission control, in others by flight control. G&N determines the control requirements of the maneuver and flight control performs the maneuver. The responsibility of maneuver management functions is to coordinate function interfaces preparing for the maneuver, executing the maneuver, and evaluating success of the maneuver. The functions described here are distinguished from those discussed under flight control in that flight control

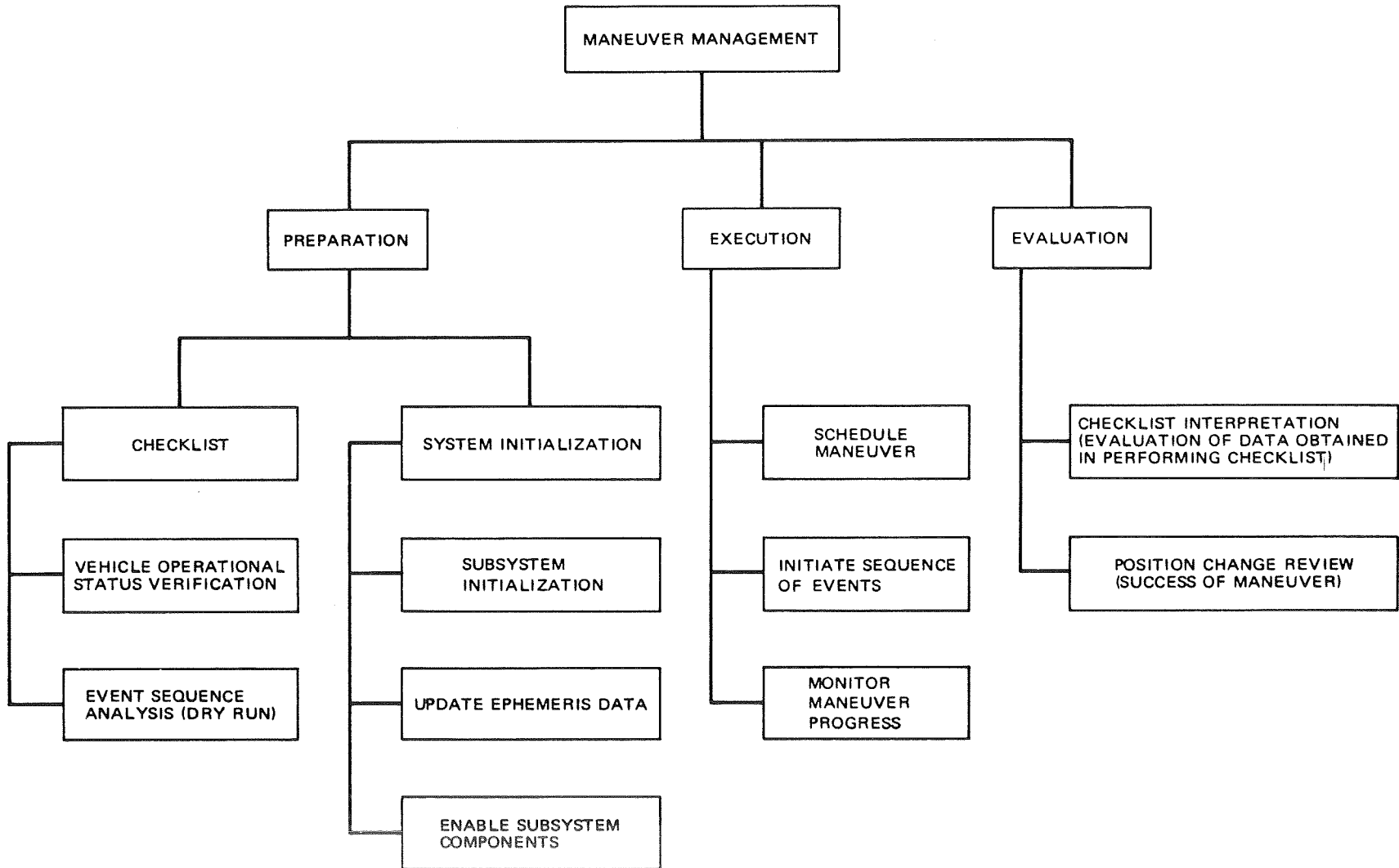


Figure A-9. Maneuver Management Function Breakdown

in concerned with maintaining a predefined flight path. Maneuver management on the other hand is concerned with rendezvous, docking, station keeping, and other maneuvers which cease to be trajectory oriented.

A.7.1 Preparation

Vehicle maneuvers are scheduled on the mission timeline at sufficient time intervals to allow for the extensive preparation requirements. A checklist of functions must be performed and systems must be initialized in anticipation of the maneuver requirements.

Checklist--Checklist functions are performed to verify condition requirements of the maneuver being planned. The checklist is an accepted technique for reminding crewmen of the tasks which must be performed in verifying the vehicle's operational status and readiness to perform the maneuver. Checklist functions for manned space vehicles are expanded to include an analysis of the events required to perform the maneuver. This analysis establishes the timing and interactive function requirements for performing the maneuver. The purpose of checklist functions is to minimize, if not eliminate, operating problems once maneuver execution is started.

System Initialization--Prior to performing a maneuver, vehicle onboard systems must be initialized. This requires that the intital conditions for respective subsystems be set to accommodate maneuver requirements. Some of the subsystems will have components which must be enabled before the components respond. (An example of a component-enable requirement would be hardware settings required for abort situations. This enabling technique requires double consideration be given the task being performed.) Software, as well as hardware conditions must be initialized. Guidance & navigation ephemeris parameters must be updated to reflect current status of the vehicle so that any accumulated G&N data errors are corrected prior to the maneuver.

A.7.2 Execution

Maneuver execution functions actually schedule the maneuver, initiate the sequence of events which perform the maneuver, and monitor the maneuver's progress to completion.

Schedule Maneuver--Maneuver scheduling functions allocate the actual time and time duration for the maneuver, ensuring subsystem and resource availability requirements.

Initiate Sequence of Events--Initiating the sequence of events as described as that function(s) which are required to issue the control command(s) that will execute the events which perform the maneuver.

Monitor Maneuver Progress--Future manned maneuver progress functions will be restricted, almost entirely, to computer interpreted and displayed data. There will be television personnel viewing of maneuvers but decisions will be based on performance evaluation data being displayed by the onboard computer system.

A.7.3 Evaluation

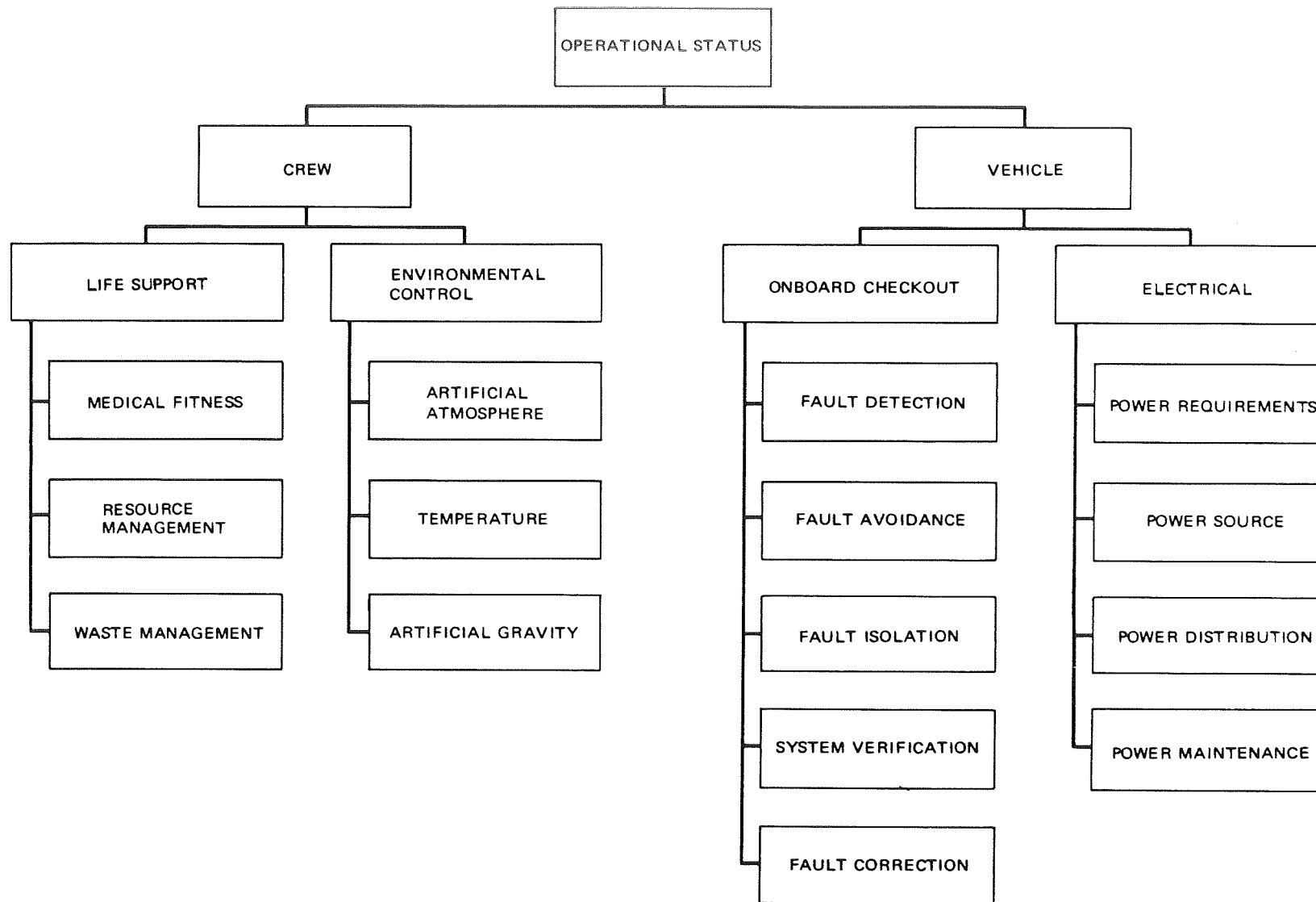
Maneuver management requires both pre- and past-evaluation of maneuvers.

Checklist Interpretation--Pre-evaluation of the maneuver is limited to the interpretation of data obtained as a result of performing checklist functions.

Position Change Review--Past-evaluation of a maneuver entails a review of the maneuver objectives (i.e., position change) and an interpretation of the success of the maneuver (i.e., were objectives achieved).

A.8 OPERATIONAL STATUS

Operational status (Figure A-10) is the term applied to the interactive functions which are performed onboard the in-flight vehicle to maintain the vehicle's and crew's operating capability. The requirement for this type of function can be attributed to the proposed long duration mission and vehicle autonomy. To achieve these objectives the capability must reside onboard the in-flight vehicle to detect, diagnose and correct subperforming systems. This requirement is satisfied by the capability provided by the operational status functions.



A-28

Figure A-10. Operational Status Function Breakdown

A.8.1 Crew

Maintaining the operational capability of the flight crew is the concern of the life support and environmental control functions performed to sustain life in a space environment. Crew safety is the major factor in manned space flights, and crew operational status functions are designed to detect and identify for corrective action any symptoms which could prove hazardous to human life. These functions also provide a means for making life in space as comfortable as possible considering the imposed limitations.

Life Support--Life support encompasses those functions which are concerned with the physical needs of the body. Specifically, (1) is the body medically fit to perform (i.e., mental or physical), (2) are adequate resources available to sustain life (i.e., food, water, oxygen, etc.), and (3) is the waste management system sufficient and operational?

Environmental Control--Environmental control functions maintain cabin environment as well as space suit environment. These functions maintain a "shirt-sleeve" environment in the space vehicle by controlling the temperature, pressure, and atmosphere within the cabin. Future spacecraft are being proposed to support an artificial gravity system. This function when defined will be considered an environmental control function.

A.8.2 Vehicle

Vehicle operational maintenance functions deal with the more popular subjects of onboard checkout and electrical requirements. These interactive functions permit rapid preflight preparation of the space vehicle and in-flight error detection and correction. Primary requirements for vehicle operational status functions are in response to vehicle autonomy objectives and proposed long duration mission.

Onboard Checkout--Onboard checkout is an obvious function which permits onboard hardware subsystem analysis and correction of hardware to the line replaceable unit (LRU). A crewman with the assistance of the onboard computer can detect

hardware faults, avoid faults by detecting marginal functioning parts, isolate failures which are not immediately apparent, correct the fault, and verify the success of the corrective action taken.

Electrical--Electrical functions are separated from those of onboard checkout because of the dependence of onboard checkout functions on the computer which requires electrical power. Rationale is based on the need for troubleshooting power failures in the absence of electrical power. For this reason electrical power is separately considered and assigned functions for determining and supporting total power requirements, allocating the source of power, overall distribution and load leveling of power, and of course, its own optional maintenance system.

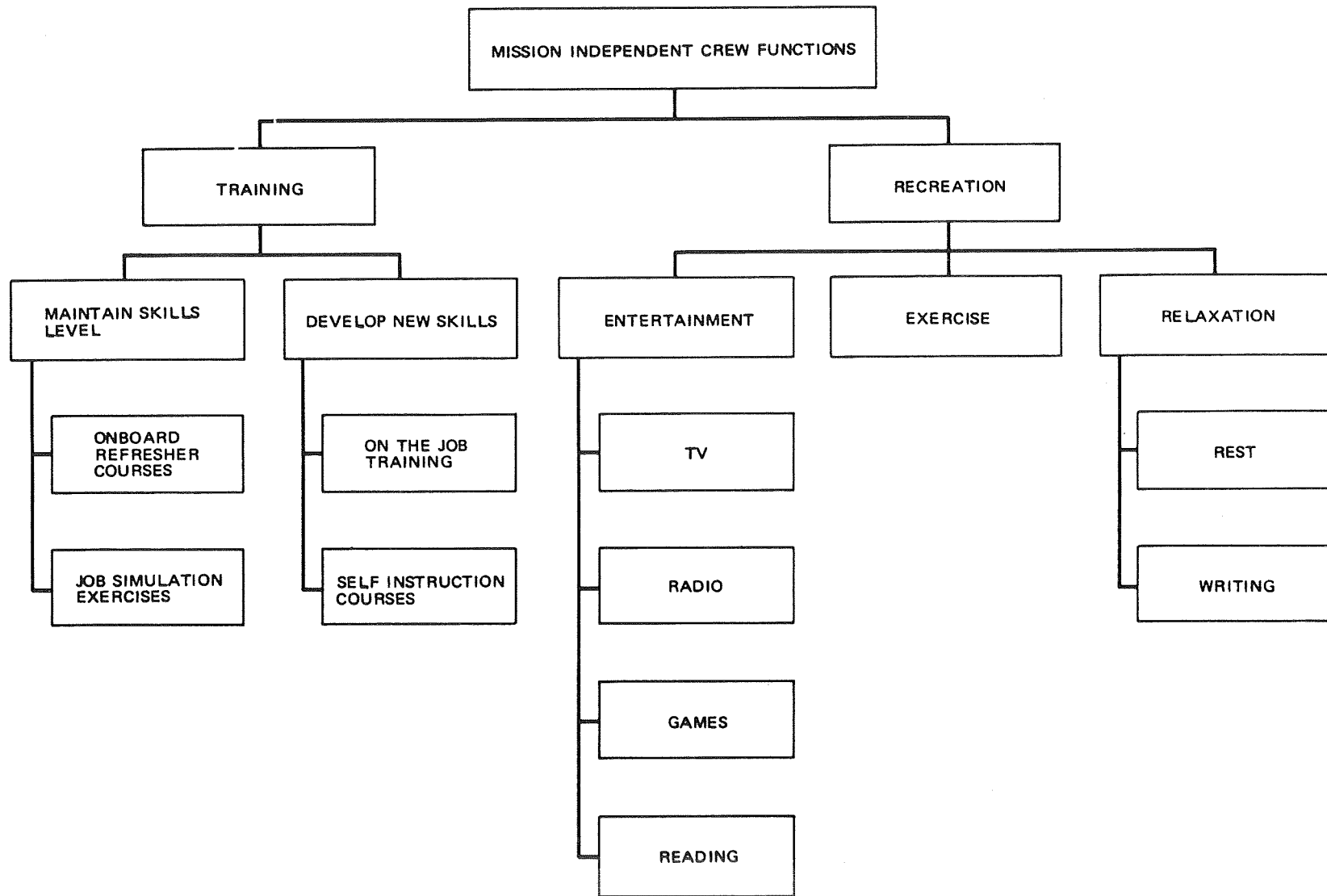
A.9 MISSION INDEPENDENT CREW FUNCTIONS

During the course of manned spaceflight, particularly of the type proposed for space stations, crew morale and technical competence become a problem. It doesn't require much imagination to anticipate problems in maintaining a skills level onboard a vehicle with high reliable parts and infrequent function performance problems. This problem has not been solved, but it has been identified. It is the purpose of this section to emphasize the problem and discuss training and recreation functions which will alleviate the problem (Figure A-11).

A.9.1 Training

Training is composed of two basic problems (1) maintain a skills level (i.e., keeping crewmen proficient in their job duties) and (2) developing new skills (i.e., training a crewman to become proficient in a job skill contrary to his basic job function). Training requirements are based on the limited vehicle support crew which will be onboard, and contingency problems which may occur during a crew illness or other unanticipated problem.

Maintain Skills Level--Onboard refresher courses will be a technique for a crewman to use in reviewing his job duties and becoming mentally aware of the



A-31

Figure A-11. Mission Independent Crew Function Breakdown

requirements of performing his job. He will also perform job simulation exercises to maintain dexterity associated with respective job functions.

Develop New Skills--A formal program of on-the-job training will exist. This will permit vehicle technology support crewmen to assist one another, and thus become acquainted with functions performed in a different skill. For a more concentrated approach self-instructional courses will reside onboard for use as required. This type of course is encompassing but assumes the trainee has a reasonable background on technology vehicle support.

A.9.2 Recreation

Recreation functions are those which are associated with maintaining the morale of the crew. The degree of interaction required during the scope of this study has not been determined, but with evolution of the manned space program and continued mechanization of onboard functions, there is no question that the onboard computer will assist in the performance of these functions. Basic function requirements considered as recreation are entertainment, exercise, and relaxation.

Entertainment--Entertainment functions, as expected, provide a means for crewmen to divert their attention to lighter subjects and amusement. Typical functions would include watching television, listening to the radio, reading and/or playing games.

Exercise--Exercise is a required function for maintaining physical fitness in a space environment. A need will exist for tracking crew performance and adherence to predefined physical fitness programs.

Relaxation--A hard requirement exists for each crewman to spend time relaxing (i.e., sleeping, resting, writing, etc.). This function will be a crew timeline function and strictly enforced. The rationale is to have each crewman capable of performing when required, not as he desires.

Appendix B
TYPICAL EXPERIMENT OPERATIONS*

The Operating Procedure Flow Diagram (Figure B-1) depicts the operations performed during a normal experiment observation sequence. This flow is presented as a typical series of operations to illustrate the sequence of events in a prescheduled operations plan, and is not intended to reflect the many possible operations which might be performed at the discretion of the experimenter. The procedure is independent of any automation system that may be employed. However, it is assumed that for these operations the sequence begins with the astronomy module in position relative to the Space Station, and will all attitude control, power, and data handling systems active. Real-time control and monitoring is maintained on board the Space Station by the onboard experimenter, and overall experiment control is exercised from the earth by the principal investigator.

The following discussion provides details on each activity of the Operating Procedures Flow Diagram:

- A. Calibrate Telemetry--Astronomy module telemetry should be calibrated periodically during the operations. This is necessary to provide a high level of confidence in the resultant data and to provide calibration benchmarks for use in telemetry data reduction.

- B. Power-Up and Initialize Experiment--Power application to the experiment apparatus must be accomplished in a step-by-step fashion, verifying the proper completion of each step before the next is taken.

**This Experiment and Solution Requirements were extracted from a report produced for NASA by System Development Corporation. The report title and number are "Analyses for the Requirements for Computer Control and Data Processing Experiment Subsystems--Experiment Control and Data Processing Requirements Specification Report" and TM-(L)-HU-033/000/00 respectively, dated 15 May 1970. These pages are direct excerpts from the above report.*

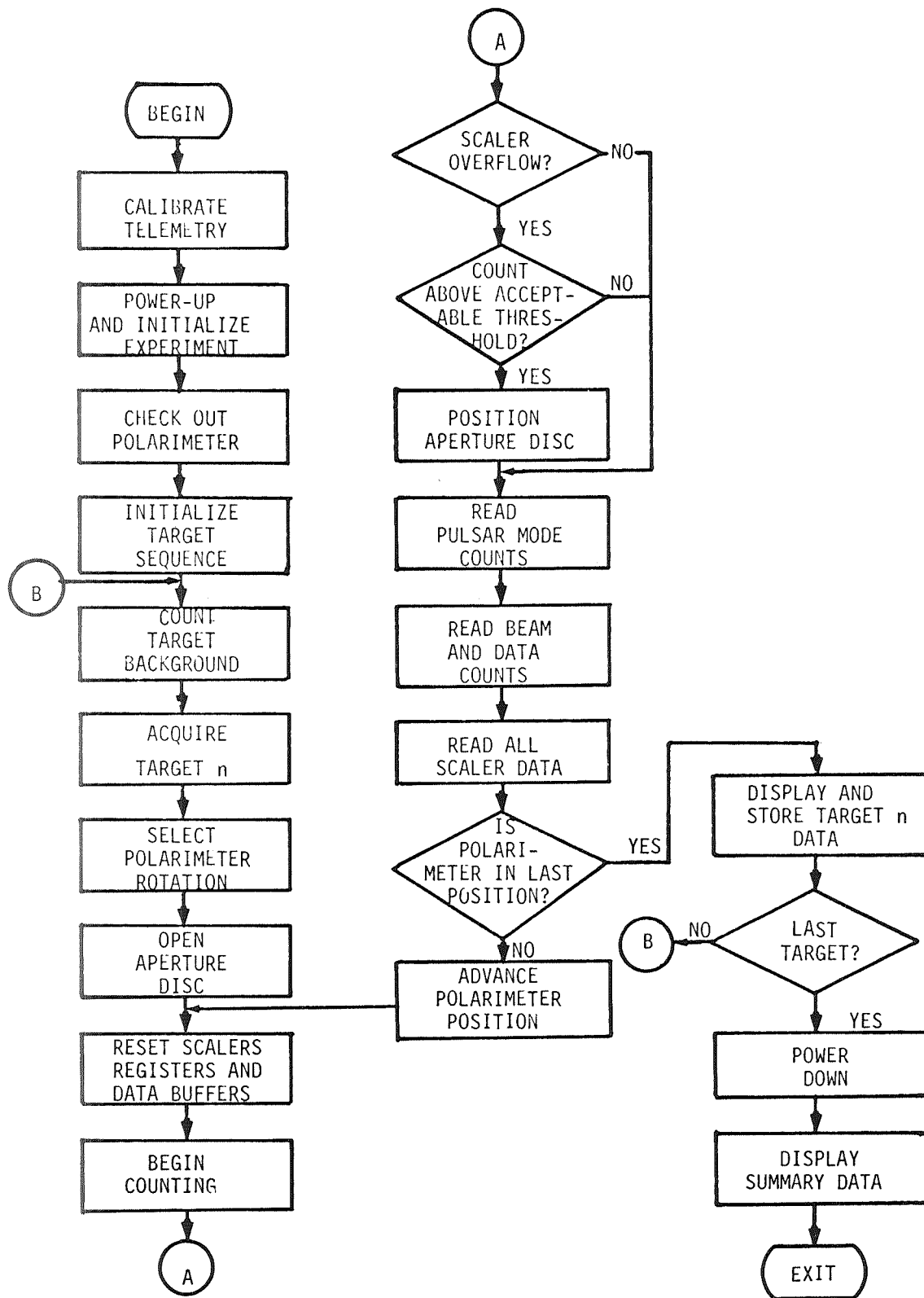


Figure B-1. Operating Procedure Flow Diagram

- C. Checkout Polarimeter--An active checkout of the polarimeter apparatus must be performed either as an integral part of the power-up sequence or after all power is applied. A function generator is used to check all experiment electronics except the proportional counters. A short scan of certain well known celestial X-ray sources will provide an adequate counter verification.
- D. Initialize Target Sequence--A list of targets and the order in which they are to be studied is prepared in advance of the experiment operation. In carrying out a normal observation sequence, the targets are scanned in the order indicated.
- E. Count Target Background--The X-ray background will affect the data study of a discrete X-ray source by adding to the total count. By noting the level of this background, compensation can be made in the data. Also, since the background count can provide valuable information about the level and wavelength of absorption by the interstellar media, it is an important clue to the density and composition of that media.
- F. Acquire Target--Target acquisition is accomplished automatically by the astronomy module pointing and control system on command from the Space Station. The Space Station observer accomplishes target verification by comparing aspect camera images with reference star fields.
- G. Select Polarimeter Rotation--The degree and rate of polarimeter rotation is specified by the principal investigator for each target. This information is stored along with other data in the target sequence list.
- H. Open Aperture Disc--Since the aperture disc partially occults X-ray emission, this operation assures that the disc is removed before observations of a target begin.

- I. Reset Scalers Registers and Data Buffers--Binary registers provide a running total of proportional counter outputs. This operation resets those registers in preparation for a new counting sequence.
- J. Begin Counting--This operation simultaneously opens the input gates to all of the binary registers to begin taking data.
- K. Scaler Overflow--Scaler overflow occurs when all bits in a binary register read "1" and another pulse is applied to the input.
- L. Count Above Acceptable Threshold--Very intense X-ray sources may tend to saturate the data handling capability of the experiment. This may be signified by a number of scaler overflows occurring in a specified period of time.
- M. Position Aperture Disc--This operation will result in a calculated reduction in counting rate.
- N. Read Pulsar Mode Data--The binary string which is the output of the pulsar mode counter is continuously scanned to reveal any source pulsations.
- O. Read Beam and Data Counts--Proportional counter outputs are sorted by pulse height analyzers into one of eight height levels and stored in appropriate binary buffers. These buffers are read to determine the number of counts and their approximate amplitude.
- P. Read All Scaler Data--High energy radiation such as gamma rays or heavy particle counts are signified by counts from the anti-coincidence circuitry or from the pulse shape discriminators. Binary scalars store these counts and are read to determine count rejects.
- Q. Is Polarimeter in Last Position--The polarimeter table is rotated by applying pulses to a stepping motor. The number of steps and hence, the degree of rotation is included in the target sequence list.

- R. Advance Polarimeter Position--This rotates the polarimeter table one step.
- S. Display and Store Target Data--Target data is automatically provided in preprocessed form to the Space Station observer and to ground personnel, and is stored for later analysis. "Raw" data is available upon request.
- T. Last Target--Operations are terminated when observations of the last target on the target list are complete.
- U. Power-Down--Power is automatically removed from the experiment apparatus in accordance with a pre-established procedure.
- V. Display Summary Data--Overall summary data will be provided in tabular and graphic form to supply "quick look" information on the results of the data sequence and to summarize the performance of on-board systems during the sequence.

A number of operations must be performed aboard the astronomy module which are of a routine nature. Periodic monitoring of secondary instrumentation such as voltage, temperature, pressure, etc., will be controlled by the experiment subsystem.

Appendix C

SPACE PROGRAM INTERACTIVE FUNCTION COMPARISON

The tabular data in this appendix is a comparison of interactive functions of past space programs and those anticipated for near-term space programs (i.e., Skylab, Space Shuttle, Space Station).

MISSION CONTROL

Functions	Gemini	Apollo	Skylab	Space Shuttle	Space Station
Planning	No onboard planning	No onboard planning	Limited to experiments		
Timeline event analysis	N/A	N/A	X	X	X
Timeline event scheduling	N/A	N/A	X	X	X
Monitoring					
Mission performance	N/A	N/A	N/A	X	X
Vehicle sub-system status	Limited	Limited	Limited	X	X
Resource utilization rate	Limited	Limited	Limited	X	X
Commanding					
Continue planned mission	Fixed	Fixed	X	X	X
Alter planned mission	Fixed abort or alternate	Fixed abort or alternate	Abort via return vehicle	X	X

X Fully interactive

DATA MANAGEMENT

Functions	Gemini	Apollo	Skylab	Space Shuttle	Space Station
Acquisition					
Identification	N/A	N/A	Limited	X	X
Assimilation	X	X	X	X	X
Storage					
Permanent	N/A	N/A	Limited	X	X
Mainframe	Limited	Limited	Limited	X	X
Retrieval					
Location	--	--	Limited	X	X
Extraction	--	--	Limited	X	X
Transmission	--	--	Limited	X	X
Dissemination					
Formatting	--	--	Limited	X	X
Routing	--	--	Limited	X	X

COMMUNICATIONS

Functions	Gemini	Apollo	Skylab	Space Shuttle	Space Station
Transmission					
Message formatting	Limited	Limited	X	X	X
Hardware selection	Fixed	Limited	Limited	X	X
Message routing	Fixed	Fixed	X	X	X
Reception					
Message acquisition	Fixed	Limited	X	X	X
Hardware alignment	Fixed	Limited	Limited	X	X
Message verification	Limited	Limited	Limited	X	X

FLIGHT CONTROL

Functions	Gemini	Apollo	Skylab	Space Shuttle	Space Station
Trajectory/Flight Path Analysis	N/A	N/A	Fixed orbit		
Course change analysis	--	--	Attitude only	X	X
Resource availability	--	--	Limited	X	X
Trajectory Change Implementation					
Change sequencing	Fixed	Fixed	X	X	X
Initiate control function sequence	Limited	Limited	--	X	X
Trajectory Change Evaluation			N/A		
Change performance monitoring	Limited	Limited	--	X	X
New trajectory determination	Limited	Limited	--	X	X

GUIDANCE AND NAVIGATION (G&N)

Functions	Gemini	Apollo	Skylab	Space Shuttle	Space Station
G&N Measurement Determination					
Attitude measurement	X	X	X	X	X
Navigation & range rate measurements	X	X	X	X	X
G&N Data Reduction Computation					
Define reference trajectory	Limited	Limited	Limited	X	X
Define trajectory deviation	X	X	X (Orbit only)	X	X
Compute required controls	X	X	Attitude only	X	X

MANEUVER MANAGEMENT

Functions	Gemini*	Apollo	Skylab**	Space Shuttle	Space Station***
Preparation			N/A		Limited requirement
Checklist	Highly manual	Highly manual	--	X	X
System initialization	Limited	X	--	X	X
Execution			N/A		Limited requirement
Identify & schedule maneuver events	--	--	--	X	X
Initiate sequence of events	Manual	Limited	--	X	X
Monitor maneuver progress	X	X	--	X	X
Evaluation			N/A		
Checklist interpretation	Manual	Manual	--	X	X
Position change review	Limited	Limited	--	X	X

*Gemini was a manual system except for ascent and descent - a manual descent capability did exist.

**Skylab will have a fixed orbit mission only requiring attitude changes.

***Space Station will have a limited maneuvering capability which future tugging requirements will also be required to control free fly modules.

OPERATIONAL STATUS

Functions	Gemini	Apollo	Skylab	Space Shuttle	Space Station
Crew					
Life support	Ground based	Ground based	Limited	X	X
Environmental control	Limited	Limited	Limited	X	X
Vehicle					
Onboard checkout	Fault detection	Fault detection	Limited	Ground based	X
Electrical	Fixed	Fixed	Limited	X	X

EXPERIMENTS

Functions	Gemini	Apollo	Skylab	Space Shuttle	Space Station
Initiate					
Plan	N/A	Fixed	Fixed	N/A	X
Preparation	N/A	Manual	Limited	N/A	X
Execute					
Monitor	N/A	N/A	X	N/A	X
Control real-time	N/A	N/A	Limited	N/A	X
Terminate					
Secure	N/A	N/A	Limited	N/A	X
Document	N/A	N/A	Limited	N/A	X
Evaluate	N/A	N/A	Ground based	N/A	X

MISSION INDEPENDENT CREW FUNCTIONS

Functions	Gemini	Apollo	Skylab	Space Station	Space Shuttle
Training					
Maintain skills level	--	--	Limited	No requirement initially	X
Develop new skills	--	--	--	No requirement initially	X
Recreation					
Entertainment	--	Limited	X	Limited initially	X
Exercise	--	Limited	X	Same	X
Relaxation	Limited	Limited	X	Same	X

Appendix D

PHASE B STUDY ASTRONAUT/COMPUTER COMMUNICATIONS ANALYSIS

These studies indicate that it is possible to derive requirements for an operator communication method that can meet expected astronaut/crew experience and psychological factors, satisfy the anticipated onboard tasks to be accomplished during prelaunch, ascent, inorbit and descent operations and be capable of being implemented in an onboard processor configuration of the type anticipated for the 1973-1980 time period.

These requirements differ drastically from those for previous manned space programs in that they are intended for application in an operational environment, rather than one that is R&D in nature. Therefore, the requirements satisfy the need for autonomous onboard operations and minimal ground support operations for real-time activities.

The increased operational complexity, necessitated by autonomous operations, will increase the reliance on automated methods to assist in the utilization and maintenance of the systems, similar to those found in ground command and control systems today. These methods which encompass the use of processors, computer programs, and operating procedures in an effective, integrated manner to provide a responsive man/machine combination have evolved into a function commonly defined as "interactive communications."

The degree to which interactive communication methods are implemented to increase man/machine performance onboard spacecraft is a trade-off against the job(s) to be accomplished, the operator (background, education, skill level, etc.), and the environment in which the job must be run and the operator must work. At one extreme as found in the early days of computer applications, is a highly trained operator, specialized in digital systems, entering octal values into the processing system memory via an input control consisting of binary neon displays and momentary action push buttons and observing results via binary

readouts and meter/annunciator readings. At the other extreme one may visualize a user trained in his particular profession, not necessarily familiar with computer technology, "conversing" with the computer via voice and written statements and observing on a display screen responses presented in a combination of graphical and textual information. Obviously, space systems of the future will tend towards the latter extreme, but an ingredient of the former will also be needed for maintaining the system.

Study Considerations--Thus, in reviewing and analyzing factors contributing to man/machine communication "methods," for space systems, this phase of the study concentrated upon the influence and interaction of both the equipment features and software structures and their net contributions toward increasing performance of the overall space system via conversational type vocabulary. Performance was addressed from two aspects; that which is associated with the onboard processing environment, similar in nature to those conditions inherent in today's bulk data processing facilities, and those factors that pertain to overall operation of the spacecraft, (performance assessment, interface control, propulsion, etc.).

However, it must be recognized that design details on data entry equipment, display devices and types of recording hardware play an integral part in influencing the exact method(s) used in man/machine communications and the resultant responsiveness of the system. To circumvent presenting performance requirements based on one or two applications, the scope of the study has been oriented towards investigations into the functional aspects of "communication methods" applicable to aerospace missions. For the purposes of this study task, then, we identify (1) those data handling characteristics of display/control that are of greatest impact upon responsive interactive communication methods, (2) the operating environment associated with manned control, and (3) the onboard operating system needed to facilitate crew/computer communications, then single out functional performance requirements that best satisfy a method suitable for aerospace applications (as defined by the function categories). The feasibility of the method was then tested by a structured vocabulary approach on the shuttle cockpit simulator in the McDonnell Douglas plant in St. Louis, Missouri. Results of the tests are discussed in Appendix E.

The identified performance characteristics have been further influenced by considering equipment techniques that are within the state-of-the-art and which can be most easily implemented within the 1973-1980 time period. This approach allows us to identify a growth oriented hardware/software operating system necessary to achieve crew/computer interaction without presenting details that should be left to the designer for a particular application.

Figure D-1 depicts the areas and approach taken to accomplish this study. The following sections discuss the findings and results of this study on astronaut/computer communication.

D.1 ONBOARD OPERATING SYSTEM CONSIDERATIONS FOR INTERACTIVE COMMUNICATIONS

From the requirements identified herein, based on crew factors, onboard workloads, the structured vocabulary, processing functions anticipated, and the estimated proximity of equipments in a space vehicle, an operating system capable of providing a responsive and reliable method for crew/computer interaction has been identified. The method is predicated on satisfying three major functions that must take place throughout the entire mission in order to assure successful accomplishment. These are:

- A. Life support (crew operations).
- B. Command and control.
- C. Scientific experimentation and exploration.

The operating system is formulated around a central computerized data source, and a program library, augmented by special purpose display/control software and general purpose data entry and display devices to allow crew control over both the onboard processing environment and the operational environment. The means of communication (commanding) is via a structured vocabulary.

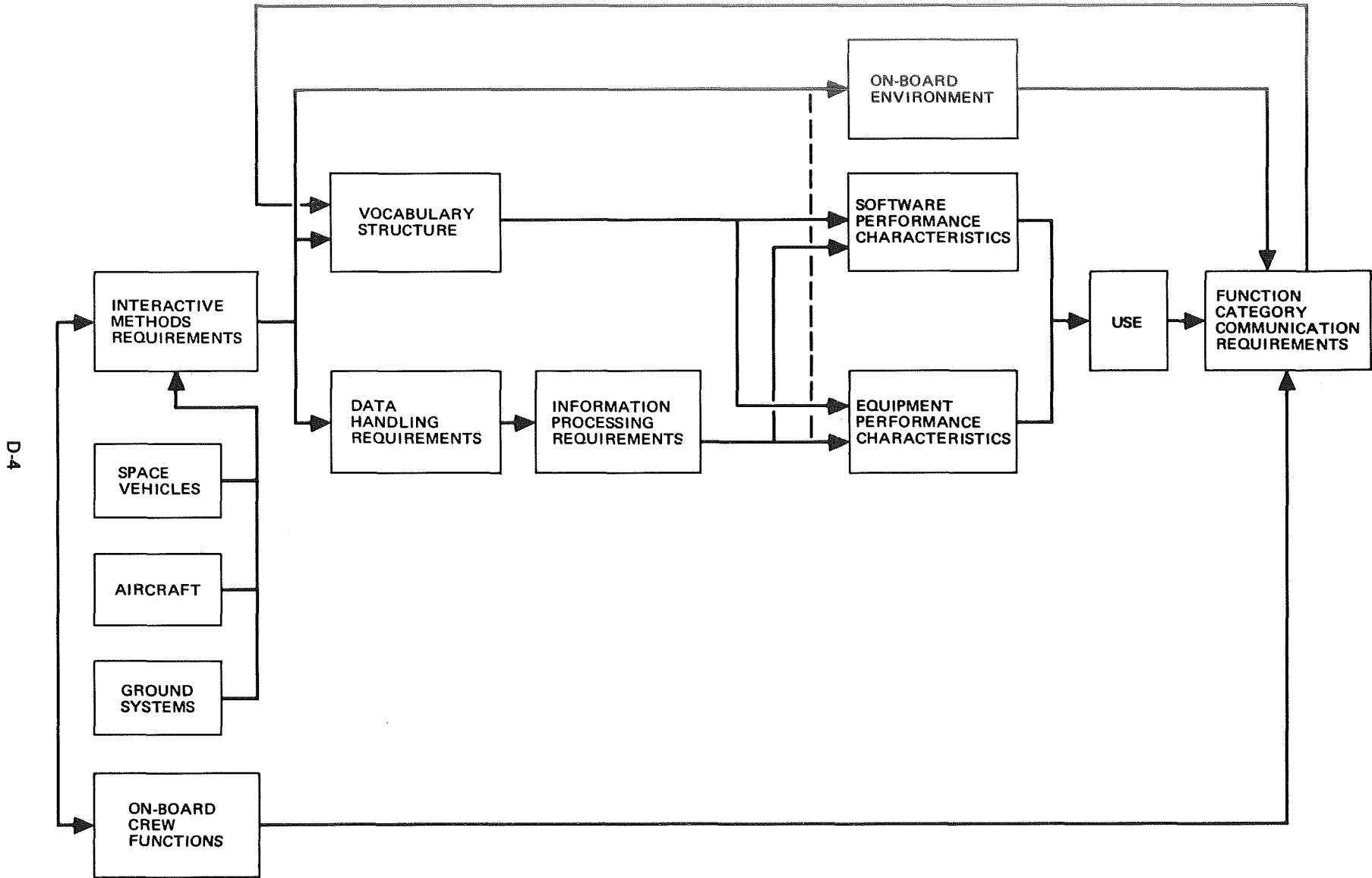


Figure D-1. Study Approach

D.2 ASTRONAUT/COMPUTER COMMUNICATION INTERFACE

The extent to which the individual crew members interface with onboard processors has been determined by examining the onboard function categories. The extent of the interface has been further determined by allocating interactive function capabilities to those manned actions involved with the processor (e.g., resource allocation) and those involved with onboard operations (performance assessment). Appendix C shows the relative magnitude of these operations with respect to each function category. It is postulated and further elaborated on under the discussion of spacecraft operational environment that a number of consoles and processors will be required to perform operations of this magnitude.

The system communication problems involved in such a multiple processor/console configuration have been found to be considerable, but once solved offer a degree of versatility, reliability, and operator convenience not available on the smaller systems used in mission operation today. For example, from a data handling standpoint, massive data transfers between processors may be needed for radical changes of the picture on selected displays. To satisfy this, transmission rates must be high, buffering must be utilized, program partitioning between processors must be implemented, and console/data assignment algorithms developed. Neither hardware or software techniques by themselves will satisfy the above conditions. The combination of equipment and software will have to thoroughly be integrated to form an effective, interactive communication system. Figure D-2 depicts a functional diagram of such a system. As can be seen from the diagram, a distinction is apparent between the hardware and software, however, the common interface with the operator is via the data entry (control) and display devices contained within an operators console.

Therefore, in determining the data handling characteristics for an operating system peculiar to the interactive operations given in Appendix C, and the structured vocabulary, performance features as related to human factors have been considered. In the broad sense, two major areas are identified (1) the means by which man enters and manipulates data based on his background, training, and skill and (2) the factors concerned in his observation and subsequent reaction to information.

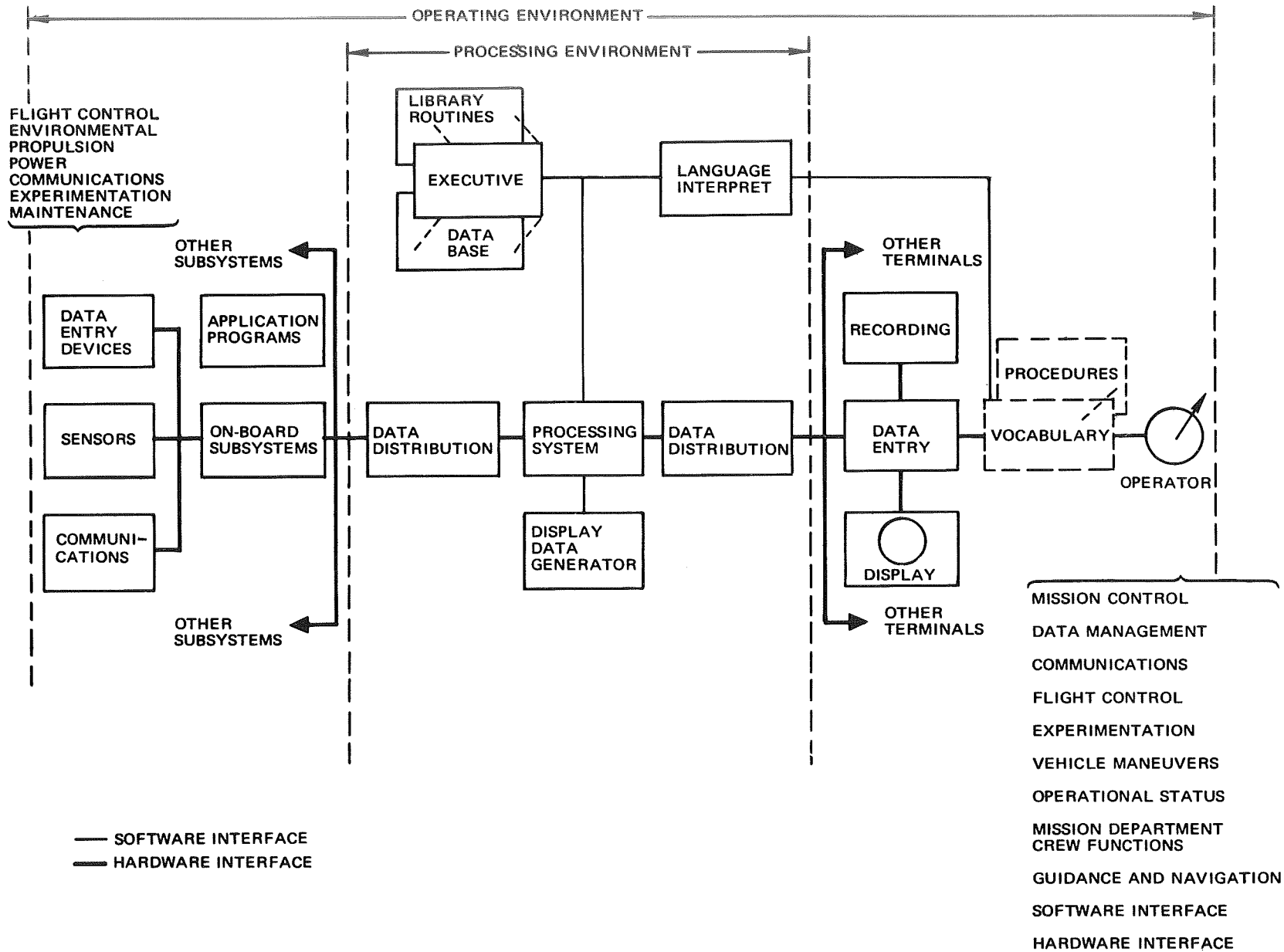


Figure D-2 Astronaut/Computer Functional Interfaces

With respect to his observation of information, two separate sets of performance requirements have been identified (1) those dealing with the display information itself and (2) those that relate to the equipment and software which produce the information. The performance requirements were based upon studies performed on human motion perception* using oscilloscope type display techniques. Those performance requirements are listed below:

- Luminance
- Image Color
- Viewing Conditions
- Frame Rate
- Image Duration*
- Image Display Rate*
- Object Motion*
- Background Information*
- Response Time*

*Can be influenced by software.

In considering data entry means, the term "means" relates to the data entry equipment and the manner in which that equipment is used, which in this study has been determined to be a structured vocabulary.

The above was taken to be baseline considerations and the most important factors in the interactive requirements analysis performed under this study. Important in the sense that the observed data as presented by the onboard processors requires control (formatting, selection, updating, routing, etc.) which can almost entirely be done by a processor (fully automated system), a processor augmented by manual input requests (semiautomatic) or a processor under complete control of manual request (manual). The extent to which the astronaut participates in this control, through the data entry devices available to him and the companion "data code" he employs to use those devices and manipulate the data, represents a further level of definition on the interactive operations given in Appendix A. The data code in this case refers to the structure of data sets into a vocabulary type "high order language" that is easily adaptable to meet a variety of user needs and processing applications. However, a determination must also be made as to

*Motion Perception Using Oscilloscope Display, Bell Telephone Laboratories.

what extent general purpose data entry devices will be applied to satisfy (or constrain) language requirements and the user and what special function switches and panel displays will be needed to meet the application and equipments.

The "language" used by the astronauts onboard the Apollo spacecraft is an example of a rudimentary type language used within the framework of limited computer data entry devices. In the case of Apollo we found that the language was restricted, not because of functional applications, but because of processor related equipments features and operations procedures. A 10-digit numerical keyboard, augmented by seven special purpose function switches (keys) were the data entry means the display was limited to three, 5-digit segmented display registers and three, 2-digit registers, augmented by fifteen computer driven annunciators. The language derivative from this equipment configuration was a numerically encoded verb/noun structure, procedurally oriented, using the maximum 2-digit count available from the keyboard, 0-99. Therefore, 99 (00 was not used) verbs (actions), and 99 nouns (subjects) in combinations thereof were available to the astronaut for computer communications. It should also be noted that approximately 350 other data entry and display devices (e.g., switches, knobs, meters, etc.) were available to the astronaut during flight.

Somewhat surprisingly, this study found a parallel situation also exists in commercial and military aircraft cockpit design. Functionally speaking, the aircraft cockpit display/control system supports two major onboard function categories, performance monitoring by the flight engineer (on commercial aircraft) or systems engineer (on military aircraft) and navigation and control accomplished by the pilots. For comparison these two functions are considered equivalent in characteristics to the communications, flight control, guidance and navigation, and operational status function categories defined during Phase A of this study. For the DC-10 aircraft, the cockpit panel displays/controls consist of approximately 40 dials, 100 annunciators, 90 data entry devices (switches, knobs, etc.) 4 of which are keyboards and 53 digital readout devices. It should be noted that the above count includes the co-pilot panel which contains redundant switches/indicators. On the flight engineers panel, 87 dials, 185 annunciators, 136 data entry devices

(one keyboard) and 14 digital readout devices (mechanical, magnetic, etc.) are employed. The above is based on a three-man crew. By contrast, the DC-9 with a two-man crew, had a total of 453 display and control devices. The pilot and co-pilot of the DC-3 were confronted with 109 such devices. By comparison, Mercury Spacecraft (No. 20) contained 102 devices, Gemini, 312, and design plans for MOL called for 522. The majority of these were electromechanical sensor driven devices rather than computer driven.

It is evident from the above that to equip future space vehicles with the necessary hardware and software to accomplish an effective man/machine communication interface for all functions given in Appendix A, the use of special purpose equipment must be minimized, and a degree of flexibility maintained. The advantages of "communitive" system concept such as now used on aircraft/spacecraft are obvious but also as equally obvious are the limitations, particularly in the area of panel layout and clutter consideration. Command sequences are fairly limited and fixed in structure. Extensive training to learn the language code is required and on-the-spot adaptiveness is limited due to minimum tutorial display methods. The quantity of data to make decisions is held to the most critical and displays are oriented towards text data (command) as opposed to "picture type" (interpretive). Where this approach is suitable for a class of specialized problems it has serious limitations in applications where more than one user needs to rely heavily on the computer and a number of tasks must be performed simultaneously. On the other hand, a data entry and display console, which has the capability both to display large quantities of information rapidly from a central data source and enter multiple operator requests, has an inherent danger of causing operational problems by overloading the processing system.

Two particular areas have been noted during the course of this study.

First, the operator may become impatient with a response to this request, particularly when he is operating in a timeshared processing environment. Second, the chance of his momentarily "losing" critical data displayed on his screen is increased during extended mission operations.

To circumvent this situation, it has been found to be imperative that future software and equipment designs provide the following capabilities:

- Equitable and relatively concurrent computing service to all operators.
- The interactive data handling "system" be attention oriented in application.
- Hard copy documentation of selected information be provided.

Two environments are readily apparent from the above, first, that which deals with the processing approach and that which is concerned with operational conditions. A discussion of these two environments as related to interactive communication for future space missions follows.

D.2.1 Operational Environment

Beginning with the space shuttle, space systems of the future will be directed towards operational deployment rather than R&D. This requirement in itself forces a distinct change in the operating environment for spacecraft as we know it today. The number and complexity of the function categories identified in Phase A of this study (Appendix A) is indicative of the operational environment tasks expected onboard. An equal mix between housekeeping activities and mission operations is readily apparent. Therefore, the designs of the avionics to be flown must be capable of accommodating operations similar to that found in ground systems in operation today. Such activities as simulation and training, maintenance, logistic management, recreation and interstation communications will become necessary.

Tasks of this nature, performed onboard will expand the spacecraft commander's role from one of primarily a pilot to base commander with all its attendant personnel and communication problems.

Further complicating operations, onboard avionics designs will need to accommodate numerous skill levels of scientific and engineering crew personnel performing a broad spectrum of research activities in such areas as astronomy, astrophysics, biomedicine, biology, and space physics.

There is no lack of information on potential experiments to be performed in space (and supported on the ground). For example, the space station definition documents (DRL 9) called "Crew Operations Definitions" lists no less than thirty separate experiments. They include such experiment areas as grazing incidence x-ray, telescope, advanced solar and stellar astronomy, plasma physics, small vertebrates, plant specimens, and many more. Obviously, the categories of information required to be processed and displayed are immense and varied, ranging from "quick look" information to detailed analysis "reports." In most cases the processing tasks will become a mix of procedural and problem oriented jobs.

Table D-1 shows the crew compliment anticipated for a space station of the 1970's. Present planning information reveals the scientific crew members will not be specifically trained for space flights in the sense that astronauts of today are trained. In lieu of specialized training they will become knowledgeable through work in laboratories associated with the ground systems which support space operations or are users of information from it. From time to time they may be required to work in a space vehicle operating environment and as space ferrying vehicles become operational, this may become a frequent occurrence. The future space scientist would thus work in a space vehicle for arbitrary short or long periods without the need for special training as operating personnel of the spacecraft. Table D-2 bears this out with the crew composition being divided into operations personnel, similar in nature to today's aircraft manning, and the users (e.g., scientists). It is natural to assume then, these users will be accustomed to extensive processing facilities to accomplish their day to day tasks, operate their programs and provide them research information. Even the operations crew will be accustomed to some computer assisted flight control and monitoring equipment, although not of the sophistication that the users are.

Table D-1
SPACE STATION CREW COMPOSITION REQUIREMENTS

Crewman	Basic Skills	Secondary Skills
Operations Personnel		
Commander	Pilot/navigator	Mechanical engineer
Deputy	Pilot/navigator	Electrical engineer
Commander operations	Electronics	Mechanical engineer
Operations engineer	Mechanical	Electrical engineer
Scientific Personnel		
Astrophysicist	Astronomy	Physics, metallurgy
Biologist	Biology	Physiology, maintenance
Physiologist	Physiology	General biology, maintenance
Physician	Medicine	Behavioral science, IMBLMS
Physician	Medicine	Behavioral science, maintenance
Engineer	Medical electronics	Centrifuge operation/maintenance
Engineer	Electrical/mechanical engineering	Maintenance, free flyers
Engineer	Electrical/mechanical engineering	Maintenance, free flyers

Table D-2

PROCESSING FUNCTION COMPARISON CHART
APOLLO VS SPACE STATION

Allocation	Processing Functions	Location	
		Onboard	Ground
Control	Executive functions	*X	X
	Failsafe services	X	X
	Self-test capability	*X	X
	Display formatting lang.	X	X
	Data bus control		X
Services	Modeling and monitoring sys		X
	Assemblers	X	X
	Compilers	X	X
	Specialized languages	*X	X
	Utilities	X	X
	Linkage editors		X
	Debug routines	X	X
	Debugging aids		X
	Testing aids		X
	Emulators and translators		X
	System loading & initial.	*X	X
Management information sys		X	
Test (Nonoperational)	Manufacturing facility		X
	Launch facility		X
	MCC facility		X
	Mission exper center fac		X
	Experiments		X
	OCS		X
	GNC		X
	Power System		X
	Propulsion		X
	Environmental control/life support		X
	Space Station		X
	ILS		X
	ALS		X
	FFM		X

*Indicates Apollo onboard computerized tasks.

Table D-2
 PROCESSING FUNCTION COMPARISON CHART
 APOLLO VS SPACE STATION (Continued)

Allocation	Processing Functions	Location	
		Onboard	Ground
Communications Control	Scheduling		X
	Verification		X
	Checkout		X
	Acquisition		X
	Validation		X
	Compression/decompression		X
	Formatting		X
	Distribution		X
	Ground commands		X
Mission Planning	Inventory management	X	
	Scheduling	*X	X
	Operational/Emerg Proc	*X	X
	Personnel/skill scheduling	X	X
Subsystem Application Routines	Overall subsystem monitor	X	X
	Power supply control	X	
	EC/LS control	X	
	Subsystem status display	X	X
	Configuration control	X	X
	Data acquisition	X	
	Communications center	X	
Engineering Data Management	Test data processing		X
	OCS summary data process.		X
	Subsystem modeling		X
	Performance evaluation		X
	Performance enhancement		X
Guidance, Navigation & Control	Integrators	X	
	Latitude/Longitude	*X	
	Accelerometer	X	
	Rendezvous sensor	*X	
	Attitude control & ref	*X	
	Maneuvers, orbit keeping	*X	
	Transmit G&N, tracking data	X	
	GNC Crew critical commands		X
	Balance management	X	
	Artificial G	X	
	FFM G&N	X	X
	Experiment pointing	X	
	SII disposal		X

*Indicates Apollo onboard computerized tasks.

Table D-2
 PROCESSING FUNCTION COMPARISON CHART
 APOLLO VS SPACE STATION (Continued)

Allocation	Processing Functions	Location	
		Onboard	Ground
Checkout	Mode control	X	X
	Subsystem monitoring	*X	
	Status display	*X	
	Data acquisition	X	
	Calibration	X	
	Stimuli control	X	
	Trend analysis	X	
	Data storage/retrieval	X	X
	Diagnostics	*X	
	Fault isolation	*X	
	Failure effect	*X	X
	Component switching	X	
	Limit checking	*X	
	Display (fault/instruction)	X	
	Data/status summary	*X	X
Inventory systems	X		
Experiment Control/Processing	Control	X	
	Data acquisition	X	
	Decommutation	X	
	Formatting	X	
	Conversion	X	
	Reduction	X	
	Editing	X	
	Buffering/storage	X	X
	Retrieval	X	X
	Analysis	X	X
	Archival	X	X
	Summation	X	X
Distribution		X	
Image processing	X	X	

*Indicates Apollo onboard computerized tasks.

To satisfy this job environment it has been found that development must take place in onboard processing equipment and use that is not in being for today's aerospace systems. Processors carried onboard aircraft, (eleven for the Douglas DC-10) and Apollo, are classified as special purpose in the sense that we think of computers today. They are limited in core memory, have extensive memory project logic, bulk memory is held to a minimum and the instruction sets and the companion data entry devices are designed to solve a family of particular applications. In most cases these processors are dedicated in nature and operate independently. For comparison purposes Table D-2 and Appendix C show processing functions anticipated for the Space Station with that presently done onboard Apollo spacecraft. It is not hard to visualize from an examination of the interactive operations that the demand on the processors and user alike onboard the 1973-1980 spacecraft configurations will be enormous, in comparison to present flight operations. Where flight operations such as Apollo using a "procedure oriented" language approach for astronaut/computer communication was sufficient in the past, the operational environment and crew of future spacecraft dictate that a "problem oriented" approach is needed. Computational needs during extended periods of deployment may arise unexpectedly without advance knowledge (or planning) of the requirements. Efficiency of solution demands fast communication with a responsive processing system. Designing and formulating the problem for processor solution must be done rapidly and, to keep with the goal of autonomy, with minimum support. Debugging has to be made easy and efficient; results must be obtained and displayed fast and the operators data entry devices reconfigurable to meet the changing situation. Where and when malfunctions and warnings are indicated, user/system interaction and reaction must take place with the same ease, speed and responsiveness as in the original environment. To meet this condition, mathematical and scientific computations, querying and updating data bases, alternate path selection, adaptive filtering applications, etc., can all provide problem-solving and fast response. However, the above requires the use of intermediate or large scale processing systems of the general purpose class as opposed to those processor configurations now in use. The use of such a processor in an interactive, highly responsive, problem-solving environment implies fast and easy selection of (1) data, (2) computer application programs, and (3) instructive type information, coupled with visual queues and "graphics" to allow the user the latitude and responsiveness to act and/or react to any given situation.

Neither the present day processing configurations nor the "vocabularies" used aboard aircraft/spacecraft have been found to possess the computational "power" and ease of use to meet a problem oriented situation, divorced from extensive support facilities.

Based then, on the crew compliment and skills expected onboard and the applications to be performed onboard, considerations were given to a class of languages that are used today for computer communications in ground support operations to assess suitability for space application.

D.2.1.1 Communication Considerations

Higher-level languages can be segregated into two categories, problem-oriented, in which the problem to be solved is described, and procedural, in which the procedure or algorithm for solution of the problem is specified. In these two categories, languages associated with graphic displays can be separated again into two forms, the written form and the pointing form. The first form of latter category is the most familiar and in common use, where, for example, inputs to a compiler are statements consisting of strings of alphanumeric characters. The second form provides a much better facility by which the operator can, in the case of procedural language, specify program operation by written statements combined with the use of the function keys and light pen as additional inputs. The vocabulary syntax associated with the written statements and the command structures that interface with the display devices are critical to responsive user interaction with the computer for space applications.

Some languages in use today (e.g., SPL and CLASP), have been found to be space oriented in the sense that they contain procedural oriented functions which are useful for professional programmers for space applications, (e.g., bit string manipulation) but these functions still belong to the well known function set of general purpose languages, oriented to program design and maintenance in extensive ground processing facilities. Functionally, it is highly unlikely that the scientists and other crew members onboard the space vehicle will consent to the burden of using a universal, multipurpose

language of this nature. Also the support equipment (keypunch, card readers, etc.) will not be available onboard. As previously identified, the variety of disciplines and tasks in space operations will require specialization and diversity of function, because while a general purpose language, by its very definition, can handle a wide variety of diversified applications it cannot do so in a problem-solving operational environment where the price to be paid, namely, considerable design, programming, and mental concentration, is too high.

Therefore, it has been concluded that the operating environment requires problem-oriented languages and structure vocabularies wherein functions are specialized according to the conceptual needs of the user and his equipment and wherein the activation and execution of the functions do not require language expressions to deviate substantially from the natural way the user would state his problem in a noncomputer environment. Neither a space oriented general purpose language (in the sense of SPL mentioned above) nor specified subsets of well-known general purpose languages (PL/1, JOVIAL, ALGOL, etc.) nor language used in aircraft or space systems to satisfy these requirements.

The fact of excluding universal, general-purpose languages from problem solving usage does not mean, of course, that one or more universal language (e.g., a space-oriented language) would not be incorporated into the total vocabulary structure for use by professional programmers. The overall properties and diversity of problem-solving usage point, however to a versatile and flexible language capability, coexisting with the language for professional programmers and capable of meeting two basic requirements needed for future space operating environments, ease of learning and ease of use.

Based on the above, Section 3 presents a means for establishing a communication method and preliminary test result using the recommended approach.

D.2.2 Processing Environment

The magnitude of the software available onboard the spacecraft will be of such a nature that an assortment of library subroutines that an operator may call will offer a number of options depending on the job at hand. This is

particularly significant in the area of experiments, an example of which is given in Appendix E. Not only will an operator/user have his choice of processing options but the structure of the processor control program (supervisor, executive) will allow control over processor resources and job scheduling similar to that found in today's time shared systems. For example, in the processing environment, if at any time the program (and operator) currently in control of a processor-shared device cannot use the resource due to dependency on the completion of some requested service, then the processing system must be capable of shifting control to some other program (user) contending for the sharable resource. A similar situation could occur in the case of a piece of malfunctioning equipment. It has been determined also that the user will need to accomplish a set of initializing conditions at a terminal prior to starting the processing operation. Initializing will be required for both the console operations and the computer programs he desires (e.g., clear memory, set up job procedures, load). For each user, initializing would include entries of such data as name, task descriptor, password entries (in the case of secure data), identification of console and processing functional category. It may also include his processing control communications such as offline output file identifiers, job priority, job timeline, processing limits, etc.

His major effort prior to job execution will be in establishing data base parameters and "linking" together programs to perform the specified job. In most cases the linking together will be through a "tree" arrangement where the stored computer programs are grouped into successively more detailed functions. The function categories and the resultant structures given in Appendix A are reflective of this arrangement. With a hierarchy of processing structures it is postulated that a means of problem designation will precede any procedural method for formulating and solving the task at hand. Obviously, a graphic display device which presents the "procedures" and informs him of actions required/verified in a processing environment of this complexity is highly desirable. Thus, setting up the "problem" and the subsequent execution by the processor it can be seen that incorporation of a communication method

(vocabulary structures), with the processor must take in account and provide facilities for definition and management of the following:

- A. Display device input/output and display data structures.
- B. Images, views and geometrical entities.
- C. Problem (associative or relational) data structures.
- D. Asynchronous events.
- E. Processor resource allocation and communications.

In addition, the complexity of the processing configurations and the size of the computer programs, dictates that the vocabulary allow for some of the more basic programming oriented tasks such as:

- A. Construction of display files similar to computer programming.
- B. Relative addressing to permit relocation of files and data anywhere in memory.
- C. Indexing for interations counts, data retrieval, etc.
- D. Direct display access to all computer process registers.

The document function categories influencing the processing environment have been determined to be experiments, mission control, operational status, maneuver management, and mission independent crew functions.

The processing environment of this magnitude has been determined to contain all the characteristics of that associated with definition of a large scale information processing system, mainly data handling and routing, data formation, data manipulation, data entry and data output/presentation, and as such is best described in these terms for further definition of communication methods.

D.3 REQUIREMENTS

The derivation and recommendation of methods by which the crew can communicate with the onboard computer to perform the functions determined during Phase A of this study (ref Appendix A) has been shown in the preceding sections to require considerations of a broad range of technologies associated with an information system.

As such, we have found it necessary to establish a broad definition of the term "method" as used in the context of this study. Method relates to the integrated operation of the processing equipments, the graphic-display terminals and the computer software, driven by operator commands to provide for, and allow, crew members a systematic means for performing their assigned tasks. The combination of the equipment and software is considered as the operating system needed to interpret and act upon the operator commands.

By this approach, we have found it feasible to define a structured vocabulary around which can be formulated equipment operational characteristics and software functions. In this manner, assessments of the interaction of each can be made in relation to overall system requirements.

D.3.1 System Requirements

The allocation of performance requirements and operating features to a method for accomplishing onboard communication has been derived from the general communications requirements identified in Phase A of this study. To reiterate:

- A. Flexible in Design--Methods must be "open-ended" for future enhancements and readily susceptible to change for ease in modification.
- B. "Straight-Forward" in Use--Methods must be easy to learn and easy to apply so as to minimize the training process requirements.

- C. Consistent in Approach--Methods must be similar in performance for different functions in all function categories, and methods for performing identical functions must be similar for different space vehicles.

A structured vocabulary approach is one means of providing for an effective computer communication operation, capable of satisfying the above requirements. The ultimate implementation of the language used in the structured vocabulary, is centered around the need for a versatile data entry/viewing-screen combination type device. The data entry and display equipments associated with this device has been found to be varied and, depending upon performance features, can have a significant impact upon the language design. An analogy would be that used in the Apollo (verb/noun via a numeric keyboard) versus a space language (PL-1 via a typewriter terminal). Thus, the analysis of equipment and the operating system has been predicated on a set of user operating conditions (designated system requirements) that are considered imperative for future onboard operations and consistent with the findings and assumptions previously given. These are:

- A. Communication methods should provide for both problem solving and procedures oriented tasks.
- B. Display concepts and data presentation methods should negate the need for providing to the operator extensive onboard dedicated display/instrumentation devices.
- C. Data entry approaches should minimize operator vocabulary memorization and checklisting. Adaptive instructions and tutorial type information displays should be provided for each identified function category and operator action.
- D. Maximum use should be made of audio and visual cues to ensure operator awareness and proficiency.

- E. Data entry characteristics as common as possible between computer input devices, console interaction, and a "roped" means for data gathering should be provided.
- F. Provisions should be made for simple correction and response methods for incorrect data entries.

D.3.2 Interactive Data Requirements

To identify the effects of crew functions on a operating system, in this study we have choosen to identify and categorize processed information that will allow the operator responsive action and reactions. The classification of information to be processed and controlled (automatically or by manual input actions) is historically varied. Depending on the application(s) and the individual identifying such data is confusing at best. There can be file data, ephemeris data, quick look data, command data, and an infinite number of data sets (e.g., data base).

In this study, the categorization of the information to be processed has been substantiated in part by the MDAC 90-day manned test. The categories are based on recommendations for "minimum scanning requirements" by observers and from an analysis of the function categories in Appendix A. Four types of information "formats" are identified:

- A. Current Status--Data or arrays of data associated with the operation (real-time in most cases) of onboard subsystems and individual components.
- B. History--Data sets associated with each subsystem and related function category that can be correlated to indicate trends and are useful for performance prediction.
- C. Diagnostic--Data, presented in a form, that will be useful in isolating a fault or in other cases pinpointing an error or errors.

- D. Alarm--Information that alerts the operator and/or other operations to nonnominal conditions.

In considering a system configuration and features of the control and display equipments that could best meet the above and satisfy the system requirements, the approach used was to separate the processing problem into the component parts of an information processing system, distinguishing between categories of information (data) to be either displayed or entered, and the function required for processing of that data. Having taken this approach, the performance characteristics relative to both the hardware and software could be assessed. The functions associated with interactive processing are defined as (1) data entry, (2) data handling, (3) data formation, (4) data manipulation, and (5) data presentation. The data that the operating system must be capable of interpreting and displaying has been separated into two broad areas (1) display categories (output) and (2) command actions (input). The characteristics of this data has been derived from features inherent in the operating systems studied under this contract (aircraft, ground control stations) and that which has been determined to be applicable to the information formats presented above. The requirements and data characteristics are given in Table D-3. Performance characteristics which best satisfy these requirements are discussed in the subsequent paragraph, and are presented in Table D-4 in terms of information processing functional components.

D.3.3 System Configuration Requirements

Based on the interactive data requirements and factoring in the tasks (reference Appendix A) required onboard, along with the skill, training, and professional background of the crew hypothesized to perform these tasks, it becomes apparent that a number of data entry and display devices (console) will be required onboard. The exact number, of course, cannot be determined at this time and is dependent on many other factors that are outside the scope of this study. However, because of the nature of the interactive operation, a system that is independent of an exact console configuration can be described.

Table D-3
INTERACTIVE DATA REQUIREMENTS

Data Characteristic	Applicability Information Type				Requirements
	I	II	III	IV	
Display Categories					
Forced*				X	To automatically present information to the operator or operators, notifying them of conditions requiring immediate or imminent action.
Discrete*	X				The presentation of a point, character or symbol independent from other data sets.
Attention*				X	Present information to the operator in a manner that will allow immediate recognition.
Tabular*		X	X		Present text type information in a narrative, summary form, peculiar to a particular function or task.
Situation	X				Present in geometrical or video form with symbolic annotations indications of the mission operation and present status picture.
Graphic	X	X	X		Present in the form of line drawings or graphs, scientific/engineering type information, suitable for quick-look analysis.
Relative				X	A combined presentation of video, graphics and text material, grouped in a manner that allows correlation of function.

*Information in either alphanumeric or symbology or both.

X Primary application to information formats.

Table D-3
INTERACTIVE DATA REQUIREMENTS (Continued)

Data Characteristic	Applicability Information Type				Requirements
	I	II	III	IV	
Command Inputs					
Category select	X				Choose a particular operation/function and/or procedure.
Point designate			X		Provide the ability to select a discrete data point from a bulk or display data, either symbolic or graphic in nature.
Trace			X		Provide means to allow line drawing or track a moving point or locus of points.
Data call	X	X	X	X	Provide the capability or retrieve preselected information from a particular source or command an action.
Parametric insertions		X	X		Provide the means to insert numerical or alphanumeric data in real-world units of measurement.
Mode select	X				Provide a capability to direct and initialize to a particular operating condition.

*Information in either alphanumeric or symbology or both.

X Primary application to information formats.

Table D-4
MULTIPURPOSE CONSOLE FUNCTIONAL ELEMENTS

Component Characteristics	Function	Operational Use
Data Handling		
Computer interface	To multiplex computer input data and to distribute data from onboard sources.	Automatic check of all onboard digital data.
Console memory	To provide buffering and display refresh for computer generated data.	Data retrieval.
Process unit	Decodes control signals received from console data entry devices or computer generated input commands.	
Data Formation		
Vector generator	Produce continuous line segments.	Mapping, heading determination.
Symbol generator	Produce special character sets.	Situation designation, targetting.
Character generator	Produce English alphabets and numerals.	Conversational language, parametric insertion.
Circle generator	Produce continuous line circles from center and radius.	Navigation, mapping, targetting.
Audio synthesizer	To convert digital bit streams into audio, voice recognizable signals.	Warning, direction.

Table D-4
MULTIPURPOSE CONSOLE FUNCTIONAL ELEMENTS (Continued)

Component Characteristics	Function	Operational Use
Data Manipulation		
Position generator	Locate and change placement of display image.	Editing.
Cursor	Point designate a location on display screen.	Figure drawing, trace, tabulation, editing.
Line structuring	Permit selection of dotted, dashed or dot-dash lines/curves.	Mapping, graphing.
Blink control	Timed flash of display image.	Warning, attention.
Rotation control	Rotate symbols/characters in circular increments.	Graphs, navigation.
Data Presentation		
Intensity control	Provide manual control to adjust foreground/background brightness.	Adjust to meet ambient conditions.
Expansion control	Provide magnification of particular display sector.	Data analysis.
Recorder	Provide permanent hardcopy record of input/output information.	Simulation, data reduction.
Annunciators	Provide discrete, event oriented indications to operator.	Warning, status, automated procedures (event actions).
Film Viewer	Project photo image.	Maintenance, checklists targetting.

Table D-4
MULTIPURPOSE CONSOLE FUNCTIONAL ELEMENTS (Continued)

Component Characteristics	Function	Operational Use
Data Entry		
Photoelectric reader	To interpret and decode digital information stored on tape.	Simulation, mission planning.
Function keys	Reassignable set of switch actions to designate a broad range of data handling tasks.	Category select, menus conversational action, display manipulation.
Mode control	Provide a discrete set of operating conditions for console performance.	Simulation, flight ready, start, checkout.
Keyboards	Switches assigned alphabetic, numerics and special characters codes.	Parametric insertions, editing, calls, procedures, simulation.

At this point in the study, it became clear that the processor configuration (multicomputer, modular multiprocessor, and distributed processors) that the consoles will interact with what must be considered. The authors are aware of the multiple problems and arguments, pro and con, regarding the centralized versus decentralized advantages in system design, but time and study limitations did not permit us to trade all alternatives. For this study we simply based our consideration on an operating system configuration that could satisfy one basic requirement, conservation of resources. It is a well known fact that in any one terminal interactive system, the processor via an executive control program will spend much of its time idling, waiting for the next operator instruction. In an environment where onboard activities have relationships to each other, such as where a vehicle maneuver is a precursor to an experiment a well integrated, interleaved processing need exists. Independent dedicated processors are out of the question. However, in interleaved processing a scheduling problem surfaces where idling time can still exist. Its duration is minimal and in most cases in times of high activity, it is nonexistent. To ensure proper scheduling, three factors must be taken into account:

- A. Priority of jobs and sequencing.
- B. Resource availability and management.
- C. Input data loading.

Therefore, to make the total onboard multiple console "communication" concept, efficient, flexible, and reliable, it would seem practical to share a processors resources among a number of consoles, and have a dynamic control program for scheduling. The alternative to this, as previously identified, is to have small special purpose processors dedicated to every console. However, this approach appears to be out of the question in that (1) the language processor (interpreter) associated with the structured vocabulary would consume far too much core space, (2) the decentralization of computer program library routines and recording equipments would impose an excessive weight penalty as well as increase complexity of the system, and (3) interconsole communications and automatic scheduling is not available. Still, to require the control program

to be capable of scheduling all interactive operations, especially those unique to the console graphics seems an unreasonable demand on its time. What is considered an optimum configuration is a compromise between the above two approaches, where every console contains its own special purpose computer for control and other fast response functions and is connected, via a data bus, to a large general purpose multiple processor which allows access to large data bases and common files and executes shared analyses programs.

It follows then, in a configuration such as this, that a requirement must also exist for standard interfaces with the onboard processors and the operator. In addition, with the size of the crew complement expected, and the estimated operator workloads (as given for each function category), the need exists for multiple, (possibly portable) data entry and display consoles in use throughout the spacecraft interior compartments, coupled to a central operating position for command type information and data gathering. The total hardware operating system must provide the operator with the resources to allow convenience of operation under all circumstances that may be encountered during launch and flight. We can therefore conclude that a requirement exists that the total operating system for crew/computer communications be multi-purpose in nature and capable of being reconfigurable with minimum of crew intervention required. With this in mind, the performance characteristics for a responsive processing system based on the interactive data requirements are given in Table D-3.

D.4 SYSTEM CONFIGURATION

To meet the requirements set forth herein and to accomplish interactive communications within each function category established by this study, an onboard information processing system with multiple terminals, a central viewing screen and supporting peripheral processing equipment are envisioned. Figure D-3 shows the arrangement of such an onboard system. The diagram is intended to show logical arrangement of a typical onboard processing system and its key elements, rather than depict a design solution. For example, in the processing equipment group, the processors can be configured into a

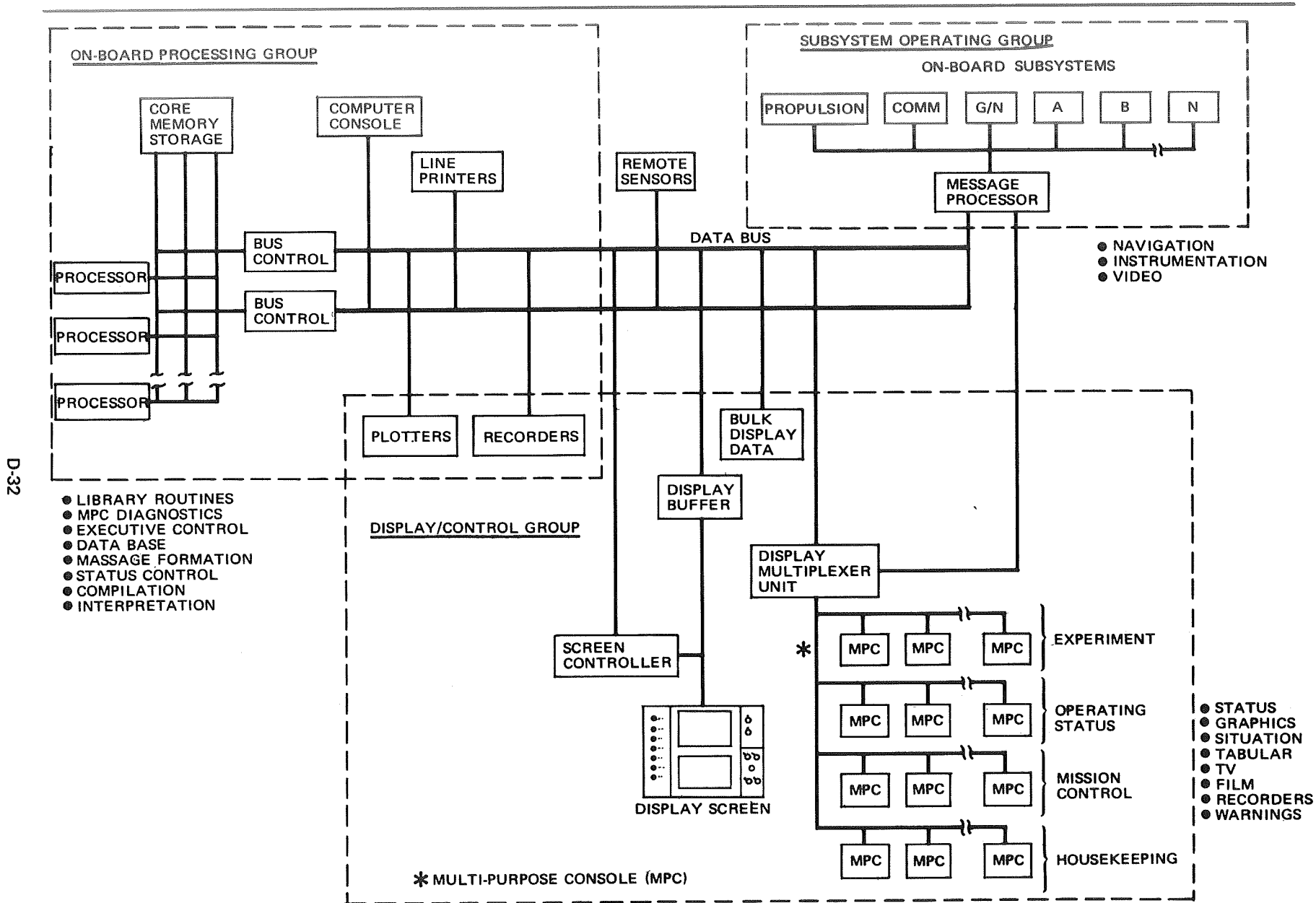


Figure D-3 On-Board Spacecraft Operating System-Interactive Communications

multiprocessing system (common memory sharing) or each dedicated to a particular processing element of the total system. In either case, the effects on the structured vocabulary are negligible, however, the processor control program, library routines, and compiler implementation concepts are effected. Regardless, this portion of the Phase B study is primarily concerned with the devices/consoles concerned with onboard computer communication. The equipment is depicted in Figure D-3 as the display/control group, operating in conjunction with the total operating system.

Display/Control Group--The display/control group consists of a functional grouping of input terminals, designated multipurpose consoles (MPC), bussed together in a manner to satisfy each of the function categories identified by this study. While each of the consoles within a category are bussed together to satisfy reconfiguration and intercommunication requirements so are each tied together through a data distribution unit to establish a standard interface with the processor. All information emanating from the processor or from the consoles passes through this unit. The unit should accept, decode, check, and distribute information received from the processors, over the data bus and/or direct from the onboard sensors or subsystems. The multipurpose consoles are envisioned to consist of a standard equipment complement throughout to allow:

- A. Interchangeability in the event of a malfunction or increased workload in any one function category.
- B. Standardization in operator actions and onboard procedures.
- C. Standardization in processor interface.
- D. Facilitate design production and ground and onboard maintenance.

While this approach appears to needlessly force additional equipment features on operators or crew members whose jobs may be less complex in nature thereby complicating operational equipment (as in the pilots controls for the G&N function) and thus increasing change of error, the simple expedient of hardwired mode control in evidence in todays display systems can circumvent operational problems. Mode control coupled with function keys can provide the operator with the broadest range of interactive facilities he will need to do his job, gives him a large degree of flexibility in operations, and allows maximum latitude in design of a structured vocabulary. Included then, in the concept is a dependence on a variety of data entry devices to accomplish control over both the operating environment and the processing environment that the operator will encounter in space. The board usage of these devices include function switch actions, equipment mode control switches, data editing, and formatting keyboards, stepping controls for graphic data manipulation, photo-electric readers for automatic operations and fiber optic sensing devices.

Display/Control Console--The multipurpose consoles (MPC) contained with the control and display groups are structured such that each can be reconfigured, via the structured vocabulary and the display multiplex unit, to assume the identity of another like unit (Figures D-4, D-5, and D-6). The limitations on assignments of course, are a function of operations and are not possible to identify at this time, however, this is not felt to be a significant factor in that the structured vocabulary can easily implement console "legality" assignments. The functional block diagram of the MPC is shown in Figure D-7.

The history of the development of display terminals/consales is a varied one, but primarily one of terminals custom-designed for specific computers and particular applications, such as military command and control, process control, and engineering designs. Only recently have data entry terminals been designed to interface to a variety of processing systems, without the need for customer engineering. Standardization has been brought about primarily through the use of minicomputers as part of the overall terminal design and the use of higher order language compilers. Such computers can be locally or remotely interfaced to a larger simplex or multiprocessor and provide the capability for self-contained graphics within the terminal rather than using the interfacing computers library.

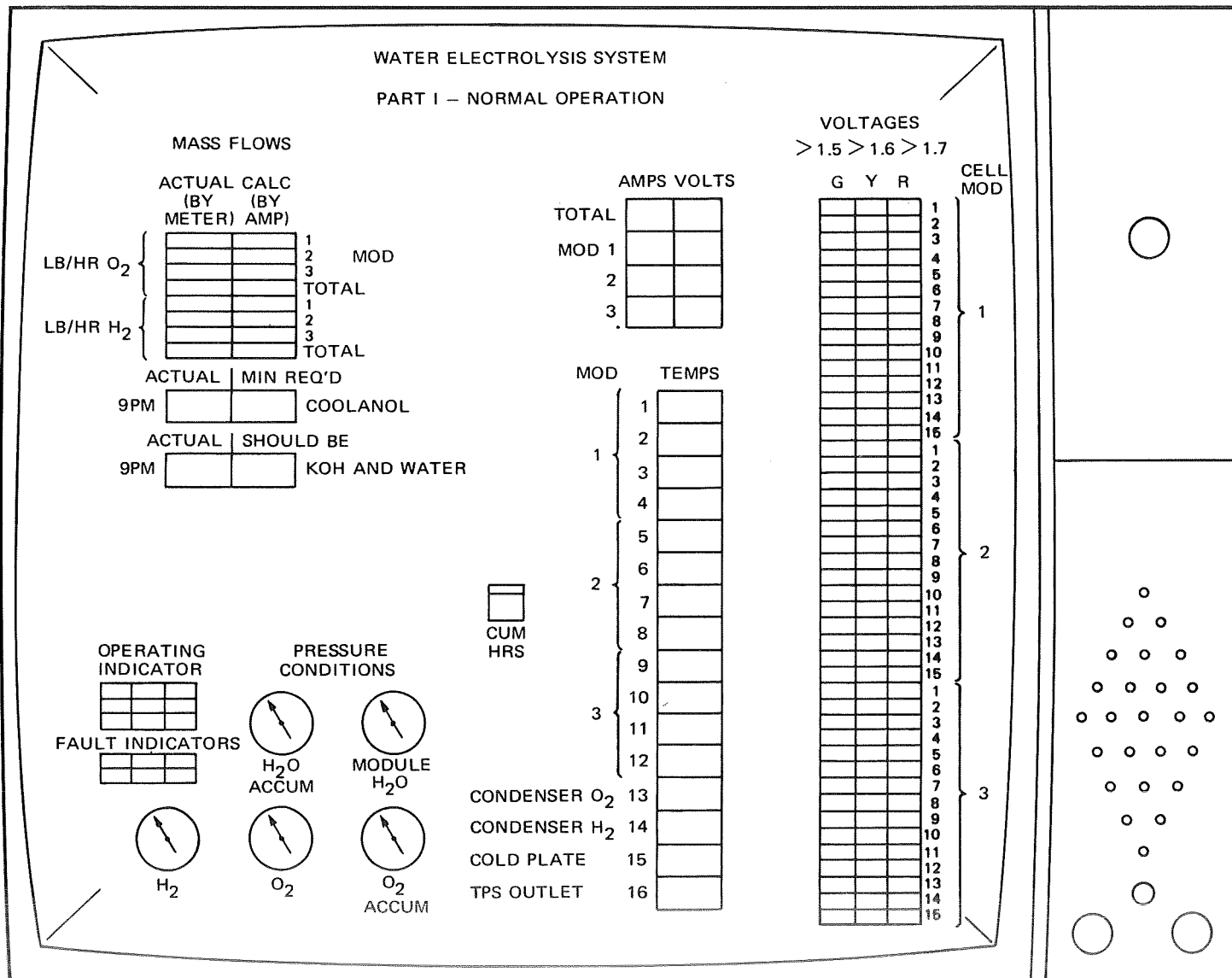


Figure D-4. Systems Status Display Monitor Station Panel 1

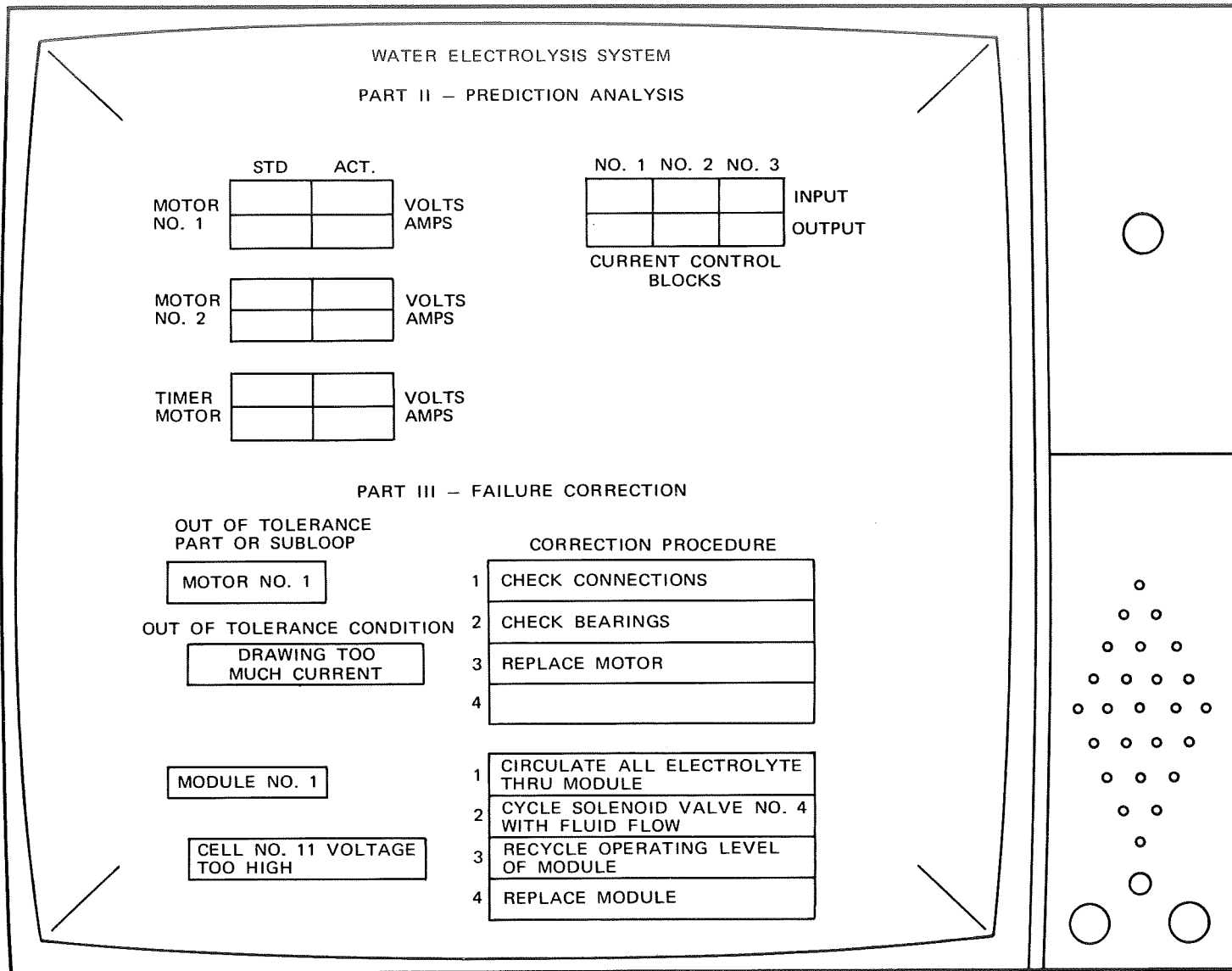
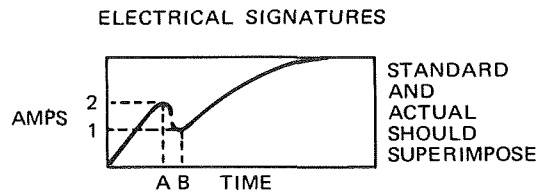


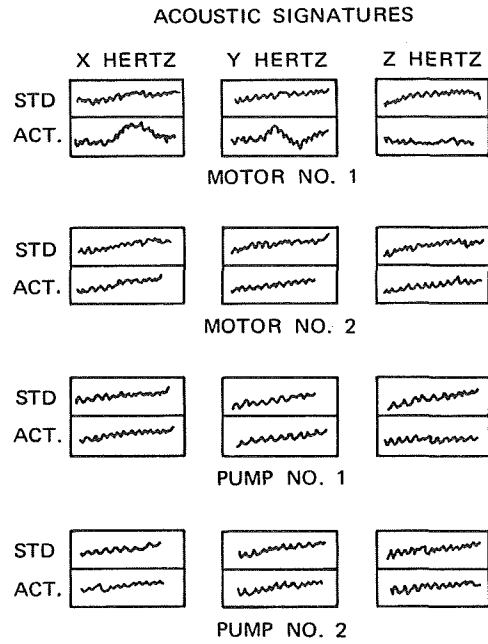
Figure D-5. Systems Status Display Monitor Station Panel 2

WATER ELECTROLYSIS SYSTEM
PART II - PREDICTION ANALYSIS

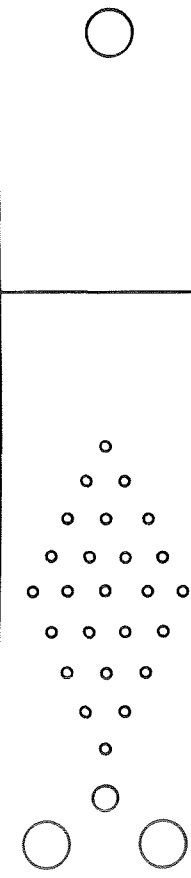


	ACT.	STD		ACT.	STD	
1						1
2						2
A						A
B						B
1						1
2						2
A						A
B						B
1						1
2						2
A						A
B						B
1						1
2						2
A						A
B						B

SOLENOID VALVES



	PRESSURE DROP CHANGES			SEPARATOR	CHECK VALVE & FILTER	
STD						
ACT.						
	MOD NO. 1	MOD NO. 2	MOD NO. 3	COOLANT		



D-37

Figure D-6. Systems Status Display Monitor Station Panel 3

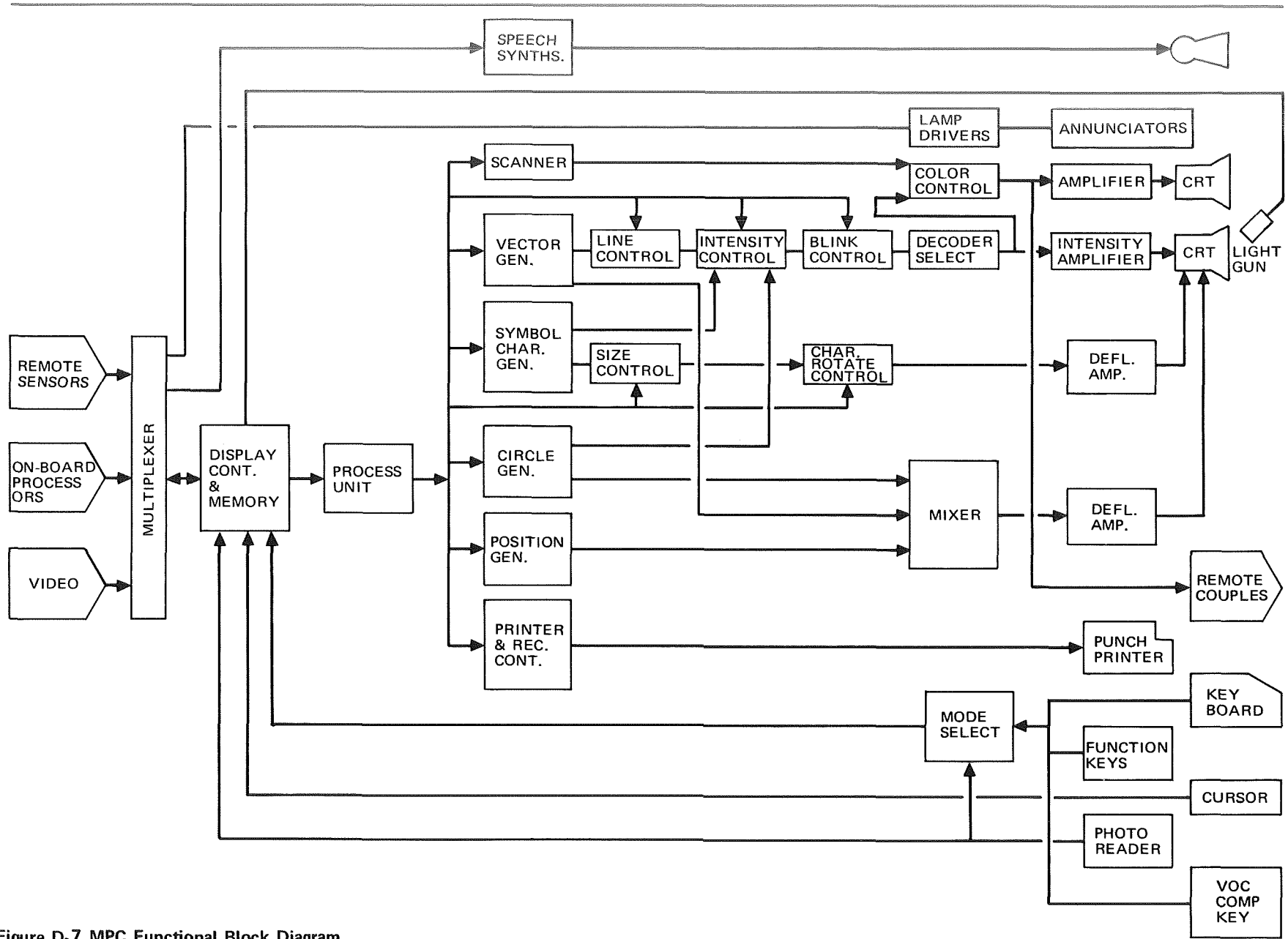


Figure D-7 MPC Functional Block Diagram

With respect to the terminal display devices and logic itself, the concepts used (which all center around a graphic CRT) have been found to fall into one of three categories, video, storage tube, or programmed scan. Storage tube and program scan techniques used to draw vectors, curves, characters can be either digital or analog. With digital logic, the path of a vector or curve between its end points, or the components of a character, are formed by the computer and displayed either by closely spaced dots or by short strokes. With analog logic only the end points of a vector or curve are required to define the path. It has been found that digital logic is slower and often more complex and expensive than analog logic, but permits a versatility in application not readily available in analog devices. For example, an operator has the ability to access the coordinates of any point along the path of a vector or curve directly, a useful capability in scientific or engineering work. Thus, in this study of hardware requirements for astronaut computer communications we have concentrated on digital techniques (raster scan, digital stroke) associated with CRT's and considered the data entry devices that will provide the best performance with these techniques. However, we have recognized the need and potential usefulness of other devices to beneficially augment both the CRT display and the data entry method (vocabulary) to satisfy the requirements.

Equipment Considerations

A summary of state-of-the-art hardware which can satisfy interactive data requirements and the characteristics envisioned for the structured vocabulary follows:

Commercial Interactive Graphic Display Systems--Commercial interactive graphic display systems similar in characteristics to those required for spacecraft use offer the following typical specification:

5" to 21" CRT, 32 function keys, programmable memory. 1 to 64K words, 8 to 32 bits (stores, activates, refreshes).

Can display: Charts, graphs, line drawings, 3D projections, full page typewritten copy. Can accommodate up to six separate display screens, up to 32K memory.

CRT resolution - 0.01 to 0.15

Positioning time - 14μ sec.

Accuracy ± 3 percent

Function generators - 62 to 120 symbols

Vector generator and circle generator - continuous line, intensity compensation

Size, intensity, blink control

Line structure - solid, dash, dot

Character rotate

Direct memory access to PMU providing small percentage of PMU's time for display generation.

Character sizes $1/8$ to $1/4$ produced by dot matrix, stroking or monoscope stroke characters easier to read, monoscope characters most natural, between 32/ and 80/line - 40 to 60 lines

Usually use storage CRT

Those using core storage are usually refreshed at 60 frames/sec

CRT's--CRT units used for numeric and alphanumeric displays are of four basic types, conventional TV-type raster scan units, higher linearity x-y positioning units, and storage type CRT units, all of which require external character generators, plus charactron shaped-beam CRT, in which the characters are produced by an aperture mask.

Commercial character generators for standard, TV monitor CRT's can provide displays with composite video for displaying numerals 0 through 9 on dot matrices of from 8 by 13 to 108 by 85. These displays can produce from 1 to 160 characters on from 10 lines of 16 characters each to 20 lines of 50 characters each.

Charactron Tubes

Produce characters by directing electron beam through individual characters etched in micromatrix.

Pictorial images time shared with alphanumerics 5000 to 8000 lines per screen.

Rear-window Tube

Random access slide projector.

Static data on slides projected onto face of tube.

As area changes operator of computer selects another slide.

Through 1/2 silvered mirror, photographs of tube face can be taken from rear part which display is observed from front.

Two-gun Tube

High speeds (25 revolutions/min or more) may not allow sufficient time to form alphanumerics and symbols.

One gun produces computer generated information other generates high resolution video images.

As many as 5 guns plus rear-window display have been built.

Recorder--Cassette recording systems are very popular with minicomputer makers they enjoy all the advantages of incremental magnetic type transports except computer type compatibility and they are useful in many speed buffering applications. They are generally useful in recording data from manual entry keyboards, digital data logging equipment, and program generation equipment for automatic checkout or numerical control.

Keyboards--Almost all computer control systems involve at least one keyboard at the computer console, usually with a number of special function keys not found on any standard keyboard. Almost all keyboards are semiconductor logic ranging from slide matrices to LSI/MOS circuits. Manufacturers use a multitude of switching means, such as, hall effect elements, magnetic cores, capacitive coupling, and other proximity transducers. Many keyboards contain coding circuitry within each individual key module, making it possible to rearrange keys at will. Other options are multikey interlocks that permit readout of only one key at a time and two-key rollover that provides the correct sequence code outputs when two keys are depressed at a fast burst rate.

Rand Tablet

Visual feedback separate from stylus positioning.

Requires scanning and processing.

Operates on horizontal surface.

Joystick

Spring loaded, vertical handle which can be positioned in any direction up to approximately 30 degrees thus positioning potentiometers.

Requires scanning and processing.

Visual feedback separate from positioning device.

Activating switch on handle.

Mouse

A small block on wheels at right angles to each other which turn potentiometers when rolled across a flat surface.

Requires clear horizontal work space.

Requires scanning and processing.

Visual feedback separate from positioning device.

Activating switch on unit.

Track Ball

Ball mounted in cup drives potentiometers at right angles to each other.

Requires scanning and processing.

Visual feedback separate from positioning device.

Slower than joystick positioning.

Separate switch for activating.

Light Pen

The light pen consists of a fiber optic light pipe with photomultiplier for sensing light on the face of a CRT, an aiming circle of light to facilitate selection of the display element to be manipulated and a finger tip switch to allow independent computer interrupt. It can be used by the operator to compose and edit displays and programs by drawing input data pictorially on the screen and allows retrieval and manipulation of stored data. Gives direct position information without additional scanning or processing. Activating switch on pen.

An interactive graphic display system can be very useful in providing parametric information in condensed tabular form for subsystems having a multiplicity of readings to be monitored. This offers a great advantage over monitoring many discrete readout devices.

One approach to presenting analog information to the crew with a graphic display system would be to provide replicas of meter faces on the CRT screen along with numerical readouts corresponding to pointer positions. Assignment of function keys could be such that individual or related functional groups of meters could be called up for display.

Caution and warning indicators can be replaced by using a CRT display to provide appropriate information on the screen to indicate when a critical parameter approaches or exceeds tolerance limits. Current development is aimed at a large central display, the most important information of the moment appears on this panel. In emergencies, messages in red would interrupt normal and caution data sequences and would convey corrective action advice. A few examples of possible display representations are included as figures.

As has been previously stated the keyboard is the basic means of man/computer communications. Data processing control functions should be centrally located on the operators console. Control functions will be implemented through an alphanumeric keyboard and a group of function switches. The function switch keyboard could provide capabilities similar to the following:

- A. One group of switches should provide for selection of the address for data being inserted.
- B. Another group of switches should identify the data parameters.
- C. Address switches should be interlocked with one another so that only one at a time can be engaged. They should also be lighted upon engagement. Parameter identification switches should also be interlocked and lighted in the same manner.
- D. The switch keyboards could provide capabilities similar to the configuration described in Appendix D.

Section 3 briefly describes the hardware requirements for a minimal type remote graphic display terminal CRT, line select keys, function keys, vocabulary/command keys, numeric keyboard, light pen/joystick automatic cursor. Appendix E describes an example operation of the keyboards.

Special purpose switches can be added to the console for power, display parameter control, mode or subsystem control as required.

Potentiometers can be utilized for brightness, contrast, color, and audio adjustment control.

Trackball, mouse, joystick, and light pen devices could be used for cursor control. However, from information gained through this study the most efficient and accurate method of hooking, changing, and manipulating pictorial, graphic, and alphanumeric information on an interactive graphic CRT display is through the use of a light pen.

To provide an interactive display system that will present an optimum display with the least amount of memory space, hardware function generators should be used to produce alphanumeric characters, vector, circle, and other special symbol representations. This will conserve programming effort, time, and memory. Using this approach will allow single-command high level control of hardware features. With a hardware character generator the programmer need only specify the particular mode followed by data words (one or two characters per data word). With software generation, characters are stored as individual subroutines in memory. Programming may then involve at least three steps, a character mode work, designation of a character subroutine, then a branch instruction to the particular subroutine. With hardware provisions for sub-routining, only two commands are required (1) a jump to the specified routine and (2) a jump back to the main display file. The alternative involves rewriting the subroutine many times in the display files.

The employment of other devices has been traded against the capabilities of the CRT to accomplish the same requirement so as to keep the number of displays and controls down to the absolute minimum and circumvent the conditions prevalent now in aerospace systems. Over the years the pilot and co-pilot

of the DC-3 were confronted by a total of 109 displays and controls. The same two men in a DC-9 must contend with a total of 453. In a much shorter period of time, a similar increase in complexity has occurred in spacecraft. Mercury (spacecraft 20) contained 102 displays and controls. With Apollo (spacecraft 12) this had risen to 448 different display functions, 350 of which are on display simultaneously. Although the same number of functions and possibly more, will be required on future spacecraft it will be desirable to display simultaneously no more than 150 of these. If for no other reason, the sheer lack of panel space will force the use of multifunction displays which use the same basic display for many functions.

During the performance of a requirements analysis of aerospace display/control/computer interactive hardware many devices were investigated. It was found that control hardware including data entry devices as related to the man/computer interface have not changed appreciably over the last four or five years and are not expected to improve drastically in the near future (to mid 1970's). The most dramatic improvement which we can look forward to is computer control by use of the spoken word. This advancement, however, is not expected to be perfected for spacecraft use prior to 1980.

Speech input has a number of drawbacks that must be overcome before it will become practical for communications with computers. Calibration of machines to particular voices is the greatest problem. Word recognition capabilities are presently in the area of 90 percent accuracy. Speech input rates are twice that of key punching by inexperienced operators but no different when experienced key punchers are used. Long duration keypunching is reported to be less tiring than continued speech input. The major advantage to speech communication is that it can free the hands for work with graphic components.

The forerunner to voice operated computers is the voice synthesizer which has been tested under simulation conditions and under controlled flight conditions to determine its feasibility as an interactive tool in spacecraft. This device which is mainly being developed as a voice warning system is expected

to be of practical value for aerospace usage during the mid 1970's. Some conclusions from these tests are:

- A. Voice warning systems have definite advantages over tone warning systems.
- B. It offers added flexibility by allowing a pilot to evaluate a failure in terms of mission and safety requirements before acting.
- C. It does not merely serve as an alerting signal but provides direct information that enables the pilot to take immediate corrective action.
- D. The benefits are more apparent under high task load conditions.

Until the state-of-the-art reaches the level of the voice operated computer we will probably have to be content with the following standard control and entry devices:

Keyboards

Functional

Alphanumeric

Special purpose switches

Rotary

Thumbwheel

Pushbutton

Toggle

Potentiometer

Trackball

Mouse

Joystick

Light pen

The most prominent component of the graphic interactive display system is the display device itself. For years industry has been striving to develop a replacement for the CRT which does not have the disadvantage of the great volume of space required for the envelope of the tube. Many relatively flat innovative display devices are available or under development in the field of aerospace interactive displays. In determining what type of interactive displays will be available in the mid 1970's and early 1980's the following devices were uncovered.

CRT: Single Gun	Electrostatics
CRT: Multiple Gun	Frustrated Internal Reflection
Field Effect EL	Laser (Scanning)
Carrier Injection EL	Laser (holographic)
Plasma (Internal Electrodes)	Piezoelectric
Plasma (External Electrodes)	Scotophor CRT
Inorganic Thermochromics	Thermochromic CRT
Organic Thermochromics	CRT - Plasma Hybrid
Photochromics	Thermochromic - EL Hybrid
Fluidics	Fluidic - Plasma Hybrid
Magnetics (Simad)	Fluidic - Thermochromic Hybrid
Magnetics (Nematic)	Magneto - Optical
Plastic Performation	

Rapid advances are taking place in this expanding area of technology, however, certain of the listed devices have undergone only limited development or have

other characteristics which render them more useful for future ground-based applications than for more immediate aerospace usage.

After reviewing available information on the listed devices it is evident that for present day interactive display systems (up through the mid 1970's) the systems designer faced with the problem of presenting anything more complex than a few alphanumeric characters must limit himself to cathode ray tubes, electroluminescent or plasma displays, although certain of the hybrid technologies should result in useful devices by the early 1980's.

Despite the current supremacy of the CRT for aerospace applications, some consideration of two other areas is warranted, plasma display and electroluminescent. Plasma display in its simplest form consists of neon glow lamps. A matrix display can be created using orthogonal electrodes either internally or external to the cavity. Present problems with this type display deal with gradual decrease in light output due to deposition of material from electrodes on cavity walls. It is expected that as the state-of-the-art is further advanced solid state matrix displays will be used in spacecraft interactive display systems. Solid state displays may permit significant improvements in the cost, reliability, and accuracy of spacecraft displays by the early 1980's.

Carrier injection EL offers a real potential should its rate of development be fast enough to meet the demand. Carrier injection EL depends upon completely different principles, and is based upon semiconductor technology. Like transistors, the lifetime of carrier injection EL devices is essentially unlimited.

Several companies have been working on the development of solid state matrix displays utilizing electroluminescent cells with ferro electric transchargers. The ferro electric, in conjunction with a diode provides a selection threshold which permits the circuit to be addressed in a matrix by voltage coincidence. This technique has provided considerable success but is still in need of further research in certain areas such as brightness and resolution. Efficiency is low and cost is high at the present, but because of its low weight, volume, voltage,

and extended lifetime promises to become competitive with the CRT in the next generation of space vehicles.

Other companies are developing plasma solid state displays which may offer a realistic alternative. The present problems similar to those previously stated including requirements for high voltage drive, but can help in handling lifetime limitations. It is possible that displays of this type could be perfected to a useful state within the next few years, however, from past experience with similar devices, such as EL, it would be optimistic to expect full usage of such displays prior to the late 1970's.

Another new readout device which promises low power and practical large-size displays is the liquid crystal device. This type display offers good contrast even in excessively high ambient levels. It is not self-illuminated but depends on reflection of ambient light from areas of the display which are made opaque through dynamic scattering caused by an electric field. Despite recent advances several improvements such as temperature sensitivity must be accomplished before this device will be practical.

The choice between EL and plasma for spacecraft becomes one of weighing conversion efficiency versus power supply and switching circuit complexity for the intended application.

It is expected that for future aerospace display applications, a variety of display media will be needed to satisfy the variety of requirements. In the not so distant future, a number of hybrid technologies offer promise for new and improved displays. CRT's have been fabricated containing a mixture of conventional phosphors and thermochromics and a thermochromic. EL hybrid has also been fabricated using the same philosophy. The entire area of hybrid display is quite new, but rapid progress is being made.

Long range plans for future spacecraft should include growth oriented command and control terminals incorporating sophisticated controls and displays such as solid state panel readouts, integrated displays for reducing readout error, three dimensional displays for improved depth perception, and the use of normal speech for processor information exchange (costly but allows operation of

computer while remote from keyboard). However, after reviewing hardware developments to determine the state-of-the-art and trends towards future aerospace display and control capabilities and requirements, it appears that present and near at hand space programs (through the mid 1970's) should plan on utilizing a computer driven CRT type graphic display system for their main man/machine interface. Although breakthroughs in the development of EL and plasma type display hardware will continue to make these new devices look more enticing. The use of solid state hardware will be a gradual process with conversion of a few of the discrete displays at a time to multifunction solid state displays.

Appendix E
CREW/COMPUTER COMMUNICATION'S VOCABULARY

A crew/computer communication's language is required to overcome the man/machine interface barrier. Attributes of the language must satisfy the interactive requirements of the graphic-display terminal and computer without placing undue hardships on the crew. The approach adopted by this study as the solution to the above problem is to formulate a function oriented structured vocabulary. The objective being to provide each onboard discipline a unique command vocabulary for performing related interactive functions and eliminate the need for all crewmen to learn a general purpose language and adapt it to their respective job duties.

E.1 VOCABULARY STRUCTURE

The structured vocabulary has four distinct levels which parallel the breakdown of the function categories presented in Appendix A. Figure E-1 shows the relationship of the levels and a description follows:

- A. Category--Refers to nine function categories described in Appendix A, e.g., Mission Control.
- B. Function--The function level is first level breakout of a function category, e.g., Mission Control Planning.
- C. Subfunction--Subfunction, as expected, is the next level breakout of a function, e.g., Mission Control Planning Timeline Event Analysis.
- D. Vocabulary--The vocabulary level is the detailed working level and is subdivided into two operating modes, to be discussed later.

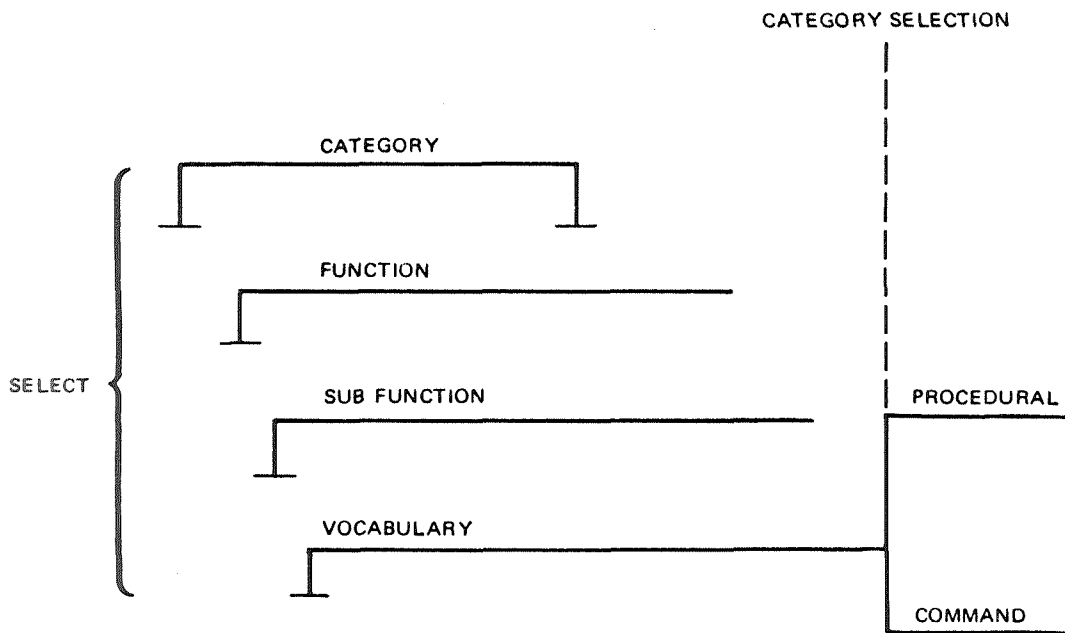


Figure E-1. Vocabulary Structure Breakdown

The purpose of the structured vocabulary is to expedite the location and access of the desired function language. This is accomplished through keyword association of job functions to technology language. Another advantage of a structured vocabulary is that it is easy to program, and adapts to an interactive environment.

The vocabulary level operating modes, "procedural" and "command," permit the user the option of constructing a complete function for automatic execution or performing each task serially. The latter affords the user greater control over the function being performed by providing intermediate task results for analysis before proceeding to the next task.

Example:

Procedure Mode

Telescope Experiment

Point telescope 1 at star 4, track for 2 hours, photograph
at 15 minute intervals. (execute)

Command Mode

Free Flying Module Experiment

Undock module from port 1 (execute)
Undocking accomplished (evaluate status)
Perform evasive maneuver (execute)
Evasive maneuver performed (evaluate status)
Fly to position "X," "Y," "Z"

The crewman continues to control the module in this manner
until it has achieved the desired operating state.

E.2 VOCABULARY METHOD OF USAGE

The intent of this discussion is to describe the procedure for using the remote terminal and vocabulary, documented in Appendix D and Subsection 3.1 respectively, in performing onboard interactive functions. Figures D-1 and E-2 are visual references for use in understanding the procedural tasks being described. Figures E-3 through E-7 are photographs taken of the St. Louis simulator tests described in Subsection 3.1

The objective, when initiating an onboard function from a remote graphic-display terminal is to select the communication's language and operating level required to perform the desired function. Figure E-2 is an example of an experiment using an x-ray telescope. Following this example the experimenter first depresses the category/select (C/S) key. The computer interprets this signal and in turn displays function categories on the CRT. The experimenter wishing to perform a particular experiment depresses the line select key corresponding to the line which accommodates the experiment category. (Note: Should the CRT be incapable of containing all entries, in this case function categories, the user continues to depress the appropriate function key for the next screen of data.) The computer then presents the various experiments until the experimenter selects the "grazing incidence x-ray telescope" experiment by depressing the line select key in the same manner as he selected the function category. The computer now responds with functions "initiate," "execute," and "terminate." Execute was the selected function as shown in Figure E-2. This selection was also made using the line select keys. The computer then displays the technology/command vocabulary on the vocabulary/command keys. (Note: Should the experimenter desire the procedural mode, he can select it by depressing the procedural select P/S key.) The experimenter is now ready to execute his experiment by entering the desired commands. This is accomplished by depressing the vocabulary/command key and supplying the arguments requested on the CRT. Arguments can be entered by selecting option using the line select keys, by positioning the cursor and using the numeric keyboard or by selecting arguments with the light pen. Regardless of the technique used the enter (ENT) key is used for partial data entries and the execute (EX) is used to signal command execution to the computer.

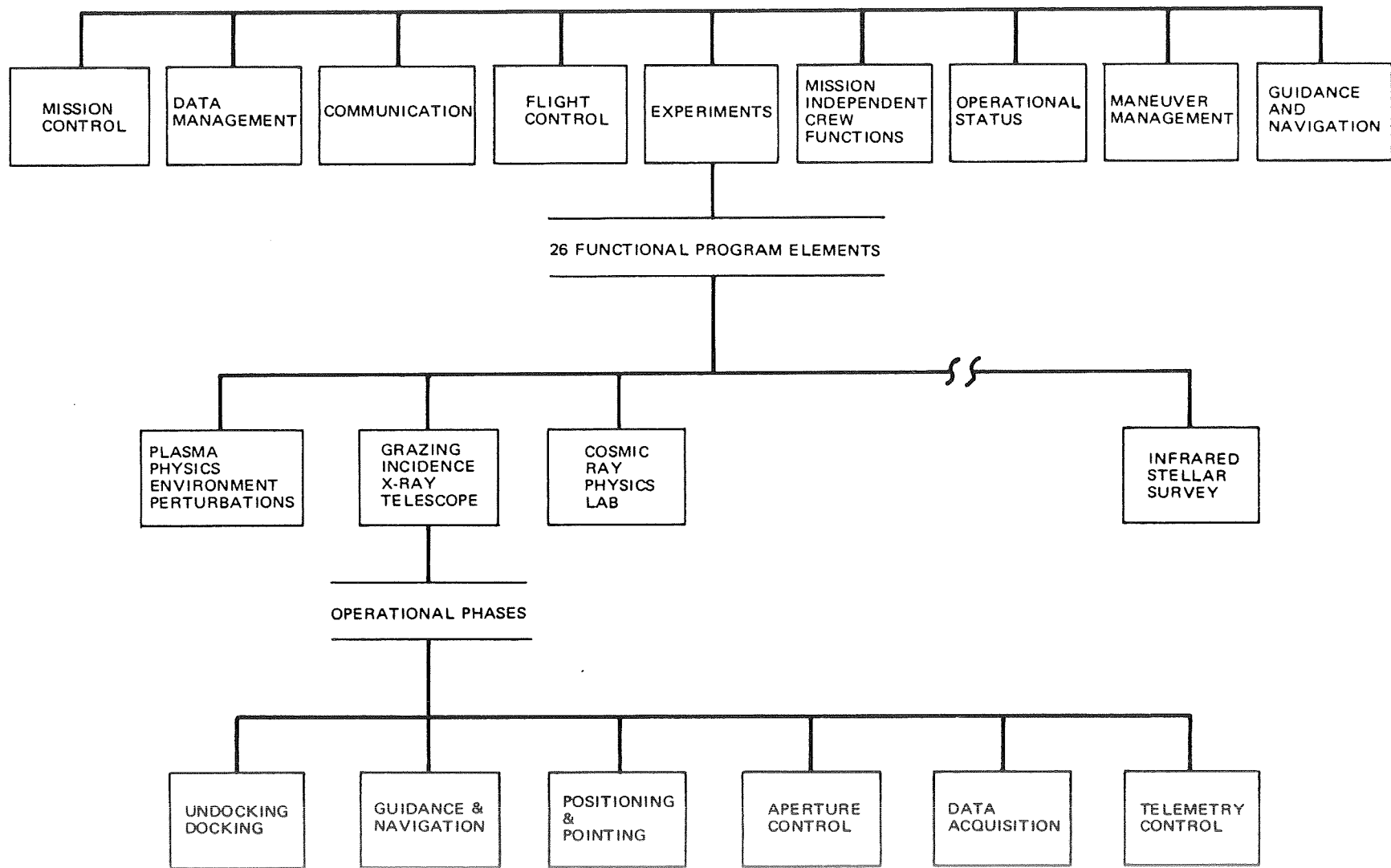


Figure E-2. Function Categories

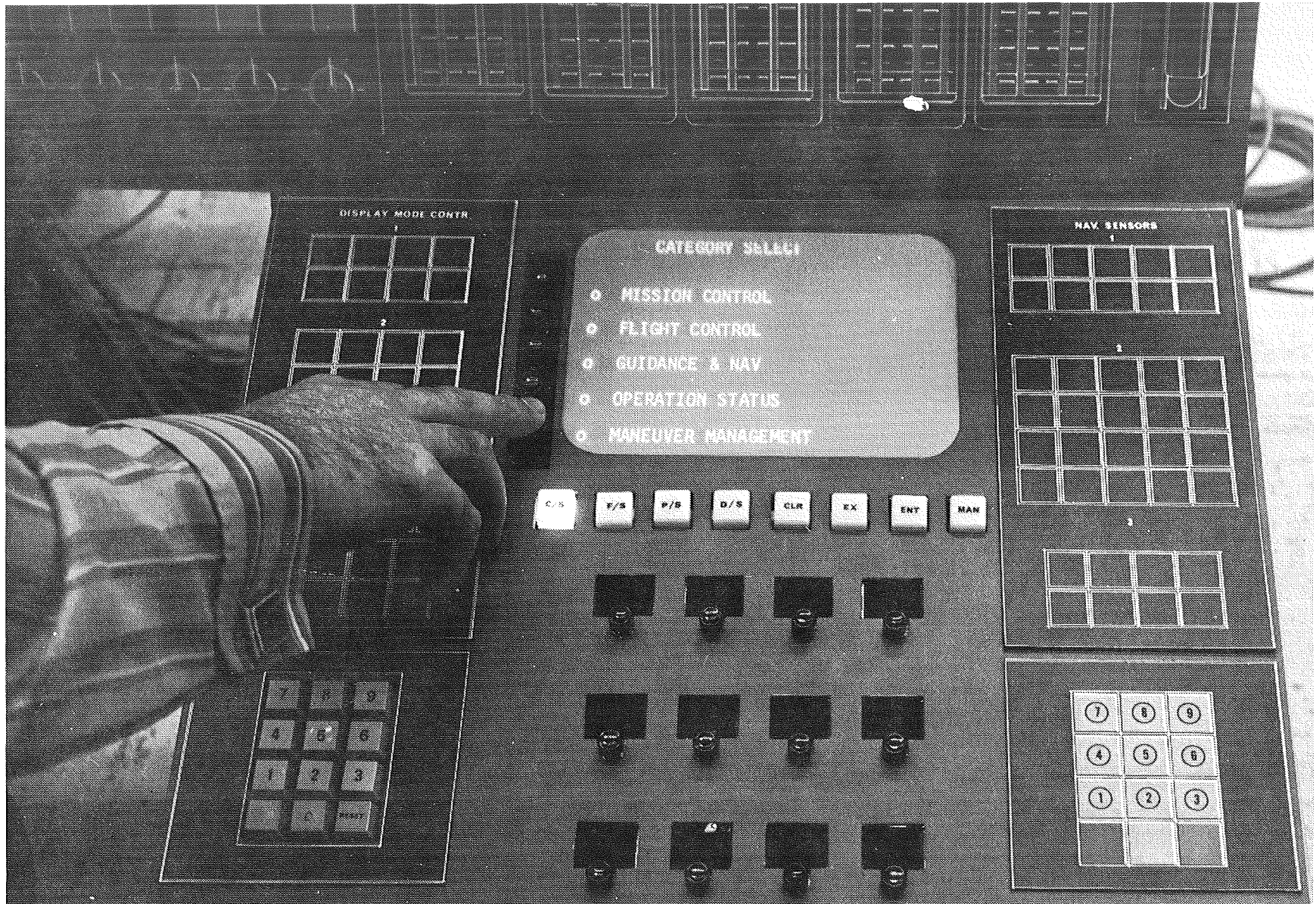


Figure E-3. Space Shuttle Cockpit – Category Select

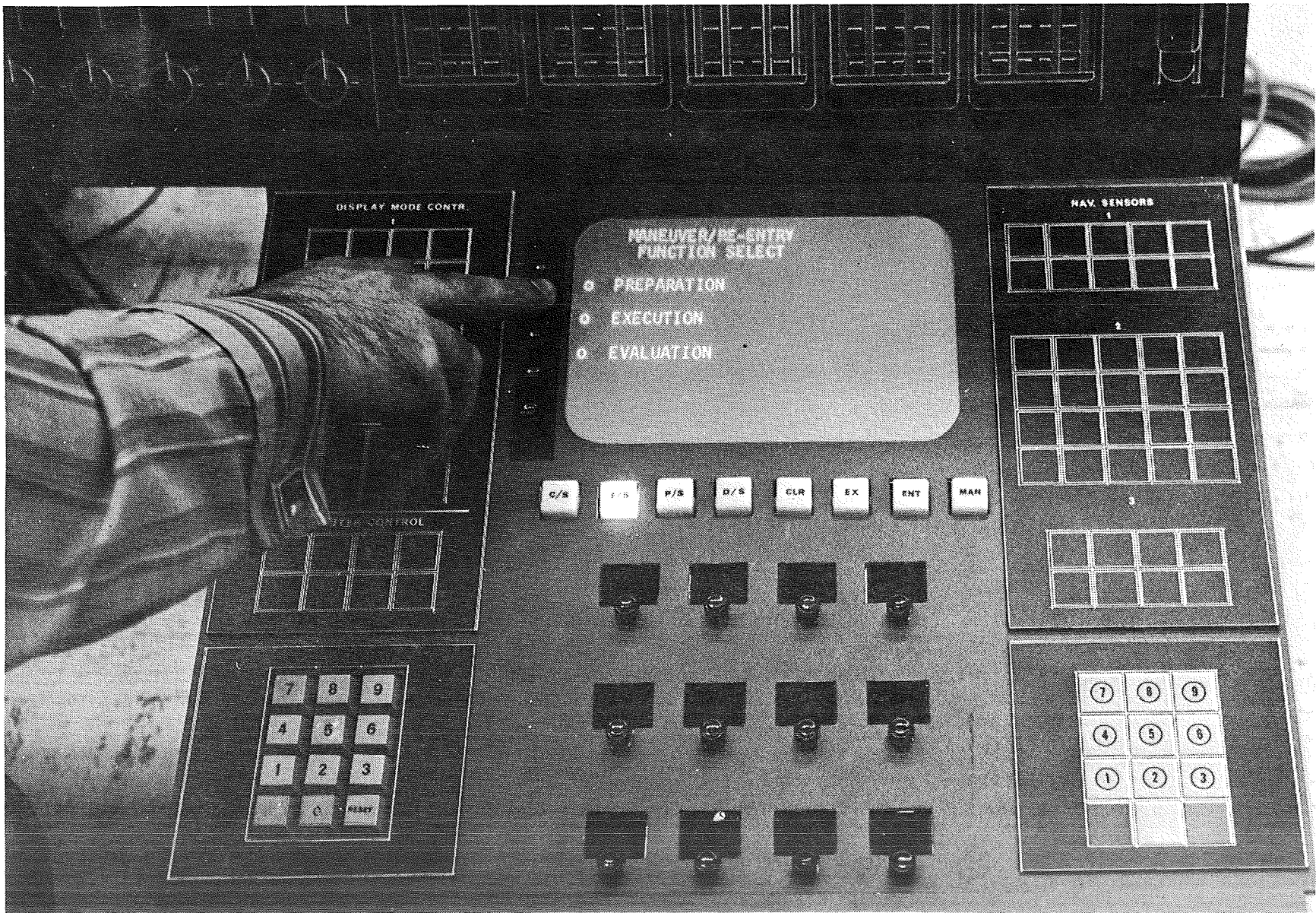


Figure E-4. Space Shuttle Cockpit – Function Select

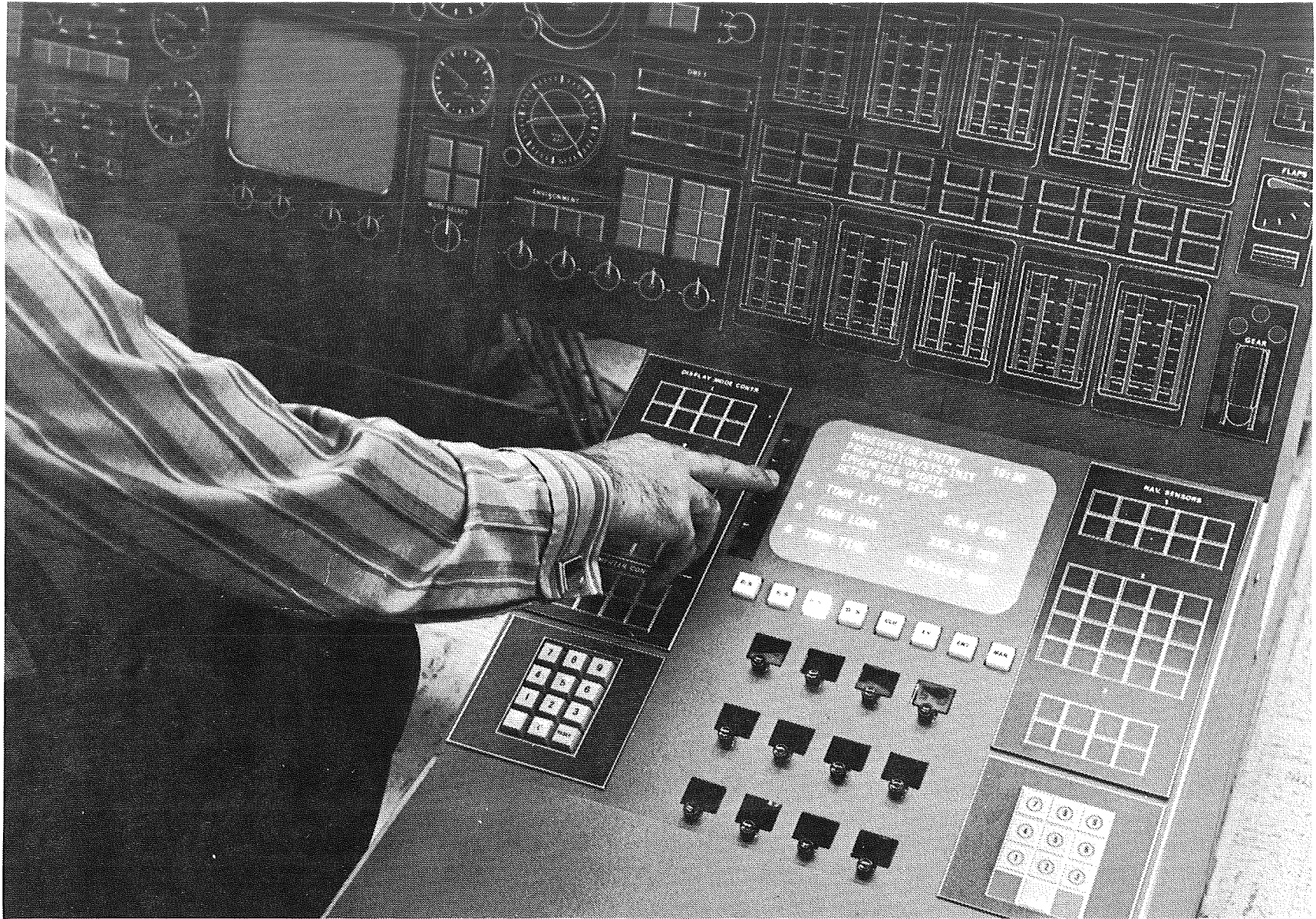


Figure E-5. Space Shuttle Cockpit – Command Sequencing

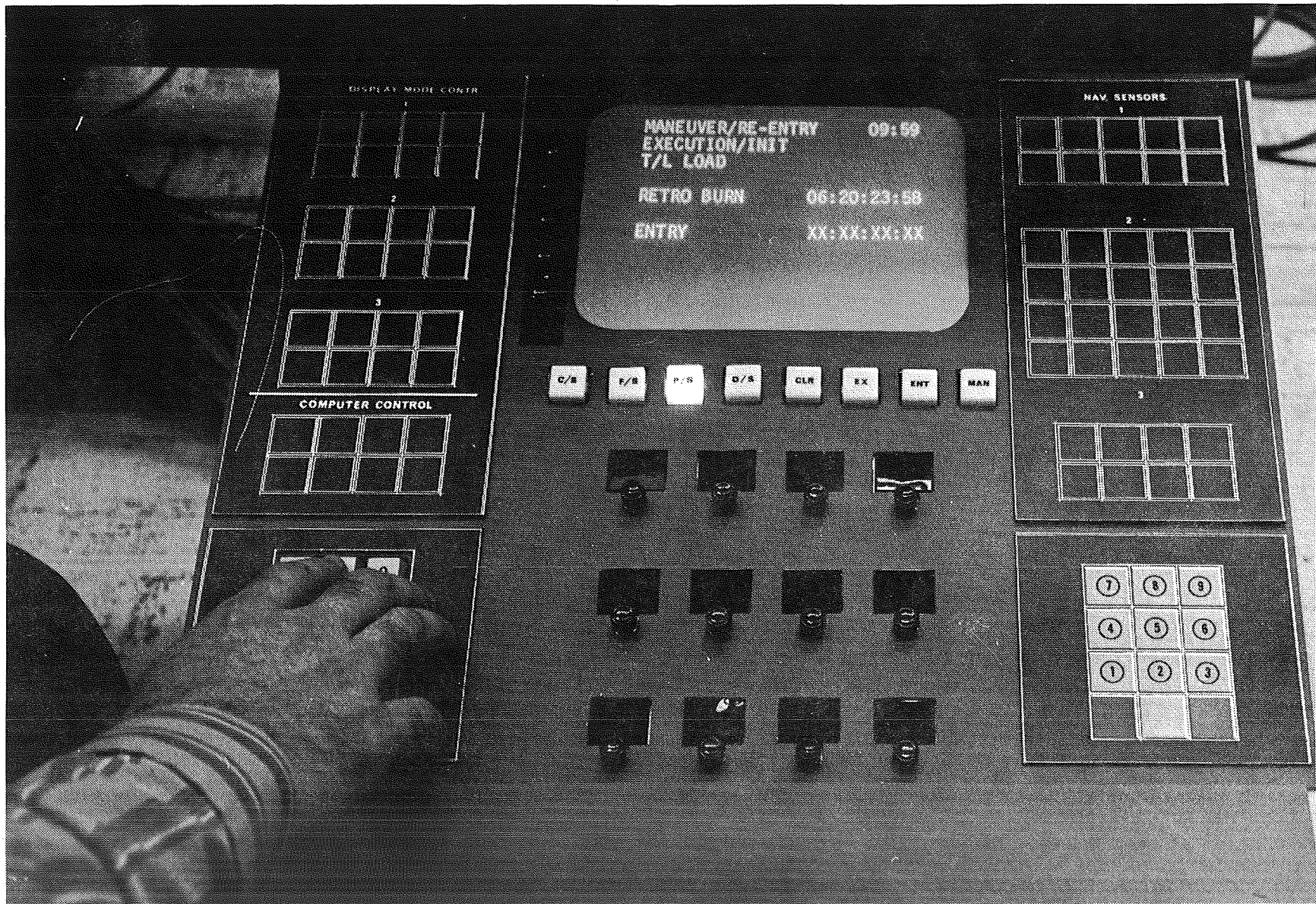


Figure E-6. Space Shuttle Cockpit – Data Update

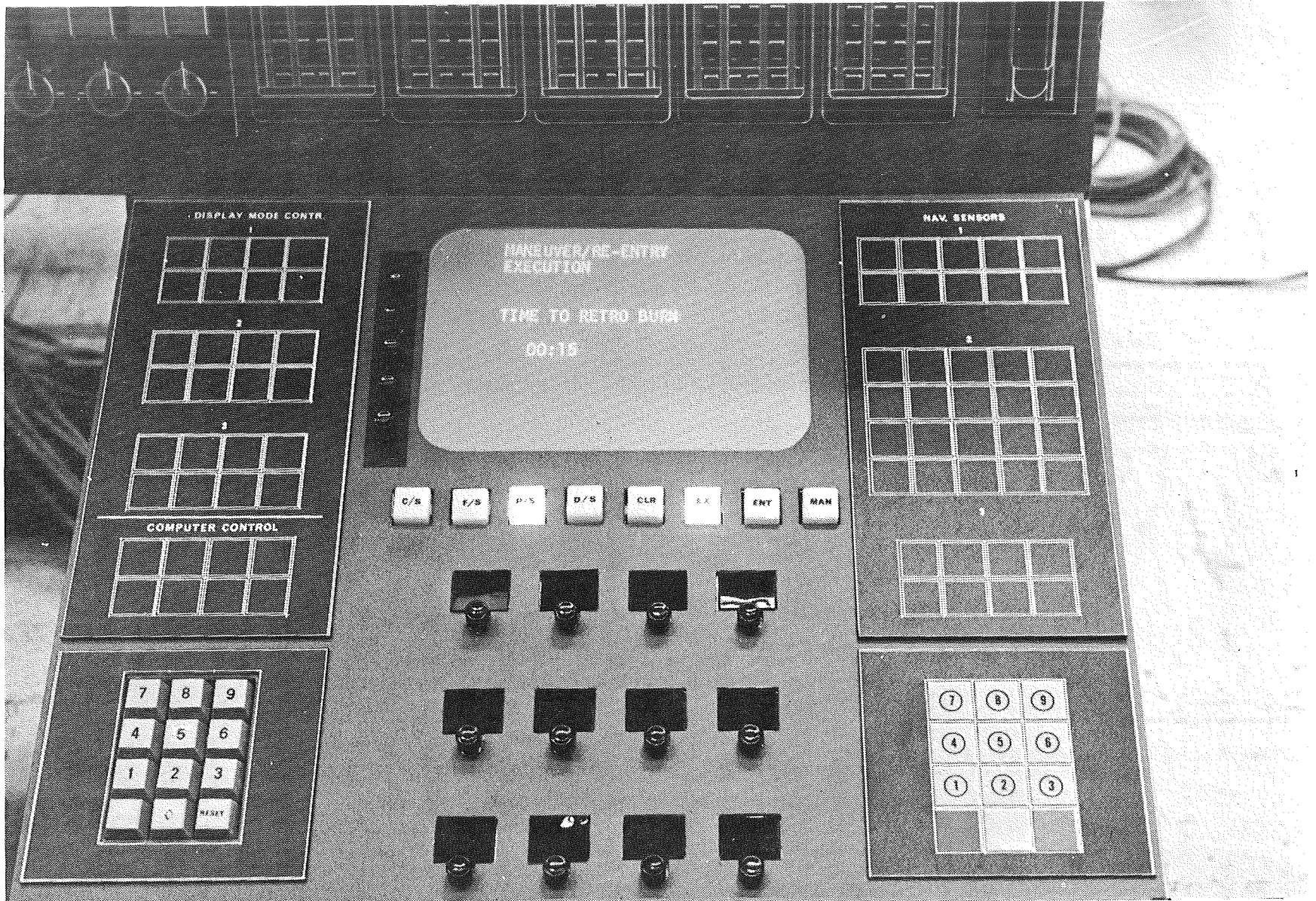


Figure E-7. Space Shuttle Cockpit – Function Execute

The manual (MNL) function key is used by the terminal operator to go directly to the desired vocabulary level without going through category and function select operations. This is accomplished by depressing the MNL key, the appropriate function key and entering control data through the keyboard, preferably an alphanumeric keyboard.

The data select (D/S) function key is used to engage the data management retrieval system. This is used to obtain general data from the computer data base.

The clear (CLR) function key is used to clear the last task signaled by the terminal operator so that it can be performed again (an error mode).

The clear (CLR) key on the numeric keyboard clears the last argument entered via the keyboard and position the cursor for reentry.

As can be seen, there is a need for additional general purpose as well as special purpose hardware in performing onboard functions. The alpha keyboard is the most apparent general purpose component. The alpha keyboard will be required by most functions in labeling data, but some functions will be unable to use it.

Examples of special purpose hardware can be identified with all functions but a few listed below:

- A. Flight control will need to use yokes, rudders translation and rotation devices, etc. All of these hardware components will communicate with the computer.
- B. Guidance and navigation could have a requirement to enter data base on optical settings of telescopes and/or sextants.
- C. Mission control and onboard checkout functions will require the use of dedicated and/or special purpose displays (i.e., electronic attitude direction indicator, map display, oscilloscope, etc.).

The point is, even though many functions will utilize special purpose hardware to communicate and control interactive functions via the onboard computer, modifications or changes to these functions can be accomplished using the procedural mode of the language described. This, of course, is assuming that the software is not hardwired.

E.3 ST. LOUIS VOCABULARY TEST

The following five photographs were taken as the structured vocabulary communication's technique was being demonstrated in the Space Shuttle Cockpit Simulator located in the McDonnell Douglas facilities in St. Louis, Missouri. The purpose of the experiment was to test the feasibility and "workability" of the vocabulary concept established by the study. Preliminary results from this test not only confirmed this approach as reasonable and advanced over conventional methods but also identified areas where the structure and technique must be refined. Solutions to these problems will be formulated during the follow-on effort to this study.

Figure E.3--Was taken after the computer had responded to the category select (C/S) key and before the operator had selected a category. As can be seen, the operator is selecting the maneuver management category using a line select key.

Figure E.4--Was taken after the operator had selected the particular maneuver to be performed, in this case reentry. The missing picture is the response to maneuver management selection which would have been a display of vehicle maneuvers. The operator then selected reentry which the computer responded with Figure E.4. This figure shows the operator selecting the preparation function of the reentry maneuver.

Figure E.5--Shows the operator using the line select keys to establish his maneuver reentry command sequence.

Figure E.6--Depicts the operator using the numeric keyboard to update and initialize data base elements in support of the anticipated reentry maneuver.

Figure E.7--Was taken after completion of all preparation functions and after the execution function key had been depressed to signal the computer to commence maneuver execution.

Appendix F

TECHNICAL DISCUSSION OF THE SYNTAX DIRECTED COMPILER

The development and extension of programming languages is a problem which increases with each new computing application. There are several good general purpose programming languages, but it is unreasonable to expect any one language to ever satisfy the requirements of people such as bankers, design engineers, and men in space. Consequently, a proliferation of languages is inevitable. The development of a translator or language processor for each new programming language can be a very time consuming and costly process if, for every new language, development must be started anew. This becomes even more significant if the application is such that language experimentation is needed prior to the actual processor development.

During recent years considerable effort has been devoted to the language translator, or compiler, development problem. Much of this work has been directed toward the development of generalized programs which, given the proper input, can process virtually any programming language. Further, much work has been directed toward techniques which will allow these programs to be as machine independent as possible. Input has generally consisted of a comprehensive description of the structures of the various elements of the language to be processed, referred to as syntax, in addition to the meaning or action associated with each element, referred to as semantics. In this context, a description of the syntax must include a detailed account of the form of each statement of the language. The semantic definitions must state precisely the actions to be performed by the language processor whenever an acceptable statement is encountered. Then, using both the syntax and semantic definitions it becomes possible, within certain limitations, to process any statement belonging to the described language.

Many approaches to this problem have been offered by as many people. This document describes one approach. To limit the discussion as much as possible,

the technique offered here will not be compared with alternative approaches. However, an attempt will be made to offer the rationale behind the more serious design considerations.

F.1 APPROACH

Before describing the technique in detail it is first necessary to identify certain major components of a language processor and introduce the functions of each. Then, discussion of the proposed language analysis technique becomes more easily understood. For this discussion, the language analysis process will be considered as involving source statement input, syntax recognition, parsing, and semantic execution. The first of these is merely an input function which can be trivial for some applications. It has been identified here because it will later help to describe more clearly the application of the available tool. The remaining three components will be treated in more detail.

As source images for a programming language are read into a computer they quickly become unwieldy. As a result, it is desirable to substitute a numeric code for each element of the syntax. This encoded form of the image can then be manipulated to determine whether or not the statement conforms to the rules of the grammar. The encoding process has historically been referred to as "syntax recognition," and the process of comparing the encoded image to the formal definition of the language is referred to as "parsing." Once it has been determined that a statement satisfies all syntactic rules, and if it can be determined which actions are required for the processing of the statement, those actions can be executed to constitute the language processing function. These actions, for the purpose of this discussion, are collectively called "semantics execution."

Up to this point, care has been taken to avoid referring to this type of language analysis processor as a "compiler." The more classical definitions of a compiler indicate that it translates a source language statement into machine language which can then be executed. For the work being described here, this definition is somewhat restrictive. If, in the language analysis process described above, the semantic actions were involved with the generation

of equivalent machine language code, then the process as outlined satisfies the classical definition of a compiler. If, however, the semantic actions are not restricted to this, the language analysis processor has capabilities above and beyond that of a compiler. An example of this might be an information storage and retrieval system in which the semantic actions, or procedures, might actually perform the data management functions without ever generating any machine code. For the sake of this discussion, this more general form of a language analysis processor will also be referred to as a "compiler," but it is an expanded definition of "compiler" which is being used as opposed to the classical definition. This does not appear to be an unwarranted liberty in view of the existence of such things as interpretive, incremental compilers. These will not be discussed here, but are mentioned to exemplify the scope of already existing compilation processes.

Consider now a program, a compiler, with the four components described earlier. It is theoretically possible to implement such a program in a manner in which the syntax and semantic definitions are expressed as interpretable data items which reside in tables. In this instance, the source input routine, the syntax recognition module, and the parser must be sufficiently general to process any language which can be adequately described and stored in the data tables, and conversely, any language which can be properly expressed and stored in the tables should be processable. This program could then be called a "table-driven, syntax-directed compiler."

There are some very serious problems involved with the development of table-driven compilers as just described. First, some technique must be made available to allow a user to express both the syntax and semantics of a non-trivial language in a format acceptable to the compiler. Presently, all known meta-languages for describing semantics are sufficiently limited so as to make them virtually useless for a wide range of applications. To circumvent this restriction, the method proposed here includes an extremely powerful meta-language for describing the syntax, as well as certain, very select semantics. This has been augmented with the facility for writing subroutines which can perform the more sophisticated semantic actions, and the facility for associating these subroutines, or procedures, with syntactic elements defined through the use of the meta-language. Using these facilities a

complete programming language can be efficiently described. The programming language might then be processed by a syntax-directed compiler which consists of a data input routine, a recognizer, and a parser, assuming that methods exist for making the meta-language definition acceptable to the parser and for associating semantic procedures with the appropriate syntax elements.

The generalized syntax-directed compiler just characterized has one severe shortcoming which greatly limits its utility. If the recognition phase is completed prior to the initiation of parsing, as implied by the above organization, the language analysis must necessarily be restricted to the processing of context-independent grammars. That is, a statement cannot be properly analyzed if the context of any of the syntactic elements influences the semantic actions to be performed on them. This can be quickly illustrated by considering the statement GOTO=A. In a context-independent grammar this might be recognized as an unconditional branch command written improperly. However, a context-sensitive analysis would reveal that in this context GOTO is a variable name which is to be set to the value contained in A. It is realized that this is a trivial example, but it illustrates clearly the point of concern.

In order for a table-driven language processor to be of real value, it should be potentially capable of handling context-sensitive grammars. To facilitate this it becomes necessary to integrate the recognition and parsing functions in a manner which does not preclude their generalization. This results in a compiler organization which consists of a data input routine, a recognizer-parser (referred to hereafter as the parser), the syntax definition (possible stored in a tabular form), and the semantic procedures.

It should now be clear that a generalized recognizer-parser can form the nucleus of any language analysis processor provided (1) it is sufficiently powerful to handle rich grammars and (2) it is easily used as a tool. Item 2 implies that there must be some convenient method for making available to it the syntactic and semantic definitions. Such a parser has been developed along with the necessary tools to effectively use it in the development of a language analysis processor. Further, these tools do provide for the implementation of context-sensitive grammars.

The remainder of this document is devoted to a description of parsing methodology, a brief overview of the operational software, and a description of how it can be used to implement a language.

F.2 PARSING PHILOSOPHY

Given the class of all characters input to a computer, there exists subsets which consist of structured character strings. One of the tools which has been developed and which is required to make a generalized parser usable is a method for defining all possible structures comprising a subset. A meta-language has been developed for this purpose, and it can be used to describe a language to be implemented. In this context, a "language" is any subset of the set of structured character strings.

A language structure can be expressed in terms of a network of terminal and nonterminal delimiters and parsing directives. A delimiter represents a defining node of a language structure. A terminal delimiter specifies a combination of one or more characters which can appear in the input string, and a nonterminal delimiter specifies any combination of terminal delimiters, nonterminal delimiters (this is a recursive definition of the term "nonterminal delimiter"), and parsing and semantic directives. If the language structure was completely free of ambiguities, each nonterminal delimiter would appear in the definition of one and only one other nonterminal delimiter. This structure is named the "inverted tree structure" in which the trunk represents the primary nonterminal delimiter (the head node), the branches and twigs represent non-terminal delimiters, the leaves represent terminal delimiters, and the forks represent parsing directives.

There are two fundamental methods for parsing an input image defined by a tree structure. The first method, referred to as a "top-down" technique, involves starting at the head node and following each of the branches to a leaf until one is found which matches an element in the input image. One major problem with this method is that in reality the leaves are not normally all unique. The result is that a network evolves in which multiple branch combinations can lead to the same terminal node. If improper search techniques are used during

the parsing process, nonunique terminal nodes will need to be examined for each combination of branches leading to it. The second search method for this type of structure involves first isolating the terminal delimiter and examining all places in the structure where these leaves can be attached, substituting the satisfied twigs for leaves, finding all places where those twigs can be attached, substituting branches for the twigs, etc. This method is referred to as "bottom-up."

For a completely nonambiguous structure it is easily shown that the bottom-up technique is the more efficient of the two methods. Further, for a language with highly ambiguous elements in which twigs and branches may be identical to others in the tree, the bottom-up method can still be shown to be more efficient than a pure top-down approach. The inefficiency of the top-down method is directly attributable to the redundant scanning of the input image during the processing of terminal delimiters. If, however, "truth" indicators can be maintained which record the status of delimiters during the parsing process, and if semantic information can be used to help "prune" the tree, then entire sections of the tree can be eliminated from the search procedure and delimiters once analyzed need never be re-examined. These and other parsing aids make the top-down method very competitive from a performance standpoint, and since it is easier to explain, more natural to use, and lends itself nicely to the implementation of programming languages, it has been chosen as the preferred parsing technique. This method has been refined and developed into an operational, generalized parsing program which can be used for compiler development. The following subsection describes the operation of this parser.

F.3 IMPLEMENTATION

A new definition of a compiler has been offered, and it involves an input facility, a parser, a syntax definition, and a set of semantic procedures. In an operational environment, the syntax definition could conceivably reside in a data table as a set of interpretable parsing commands. The remaining components could all consist of executable subroutines. The problem now to be confronted is that of automatically producing the data which represents the syntax definition in a form acceptable to the parsing program. This requires the development of a compiler capable of processing a language definition expressed in the form

of a meta-language. This compiler is called a "meta-compiler," and it produces all, or part, of an operational compiler for any defined programming language. Within the framework just described, such a meta-compiler would accept a definition of the syntax of a language expressed in the meta-language, and produce a table for the parser. A user could then develop an input routine and the semantic procedures, integrate the components, and an operational compiler for the new language would be available.

The basic approach just outlined is valid and has been used on various occasions. However, it is possible to modify the approach slightly to produce more efficient compilers in terms of execution. As outlined above, the parser must be capable of interpreting the contents of the syntax table. That is, the parser must interrogate the syntax table to get all parsing instructions, and then perform those instructions through the use of parsing subroutines. This procedure is indeed interpretive, and interpretive execution has a high rate of processing overhead. If possible, this should be avoided.

Since each entry in the syntax table represents a parsing directive, it is easy to image an executable subroutine, or macro, associated with each directive which performs the actual parsing function. This being the case, it becomes possible to substitute each of these executable components for the table entries. The result produced by doing this is an executable, rather than interpretive, parser for the language whose definition originally appeared in the syntax table. This method will, given the proper definition of the syntax of a language in a tabular form, automatically generate an executable parser tailored to meet the exact requirements of the described language. Furthermore, this parser will be capable of being the nucleus of an operational compiler for that language, and the resulting compiler will be syntax-directed. This technique has been used as the basis for an operational parser-generator which may be termed a meta-compiler (this must be qualified because the source input routine and the semantic procedures are not generated). Input to the program consists of a meta-language description of the language for which a processor is to be implemented, and a parser for that language is automatically generated.

Figure F-1 illustrates more clearly what is required to use this meta-compiler and all of the steps involved. The symbology consists of dashed boxes represent

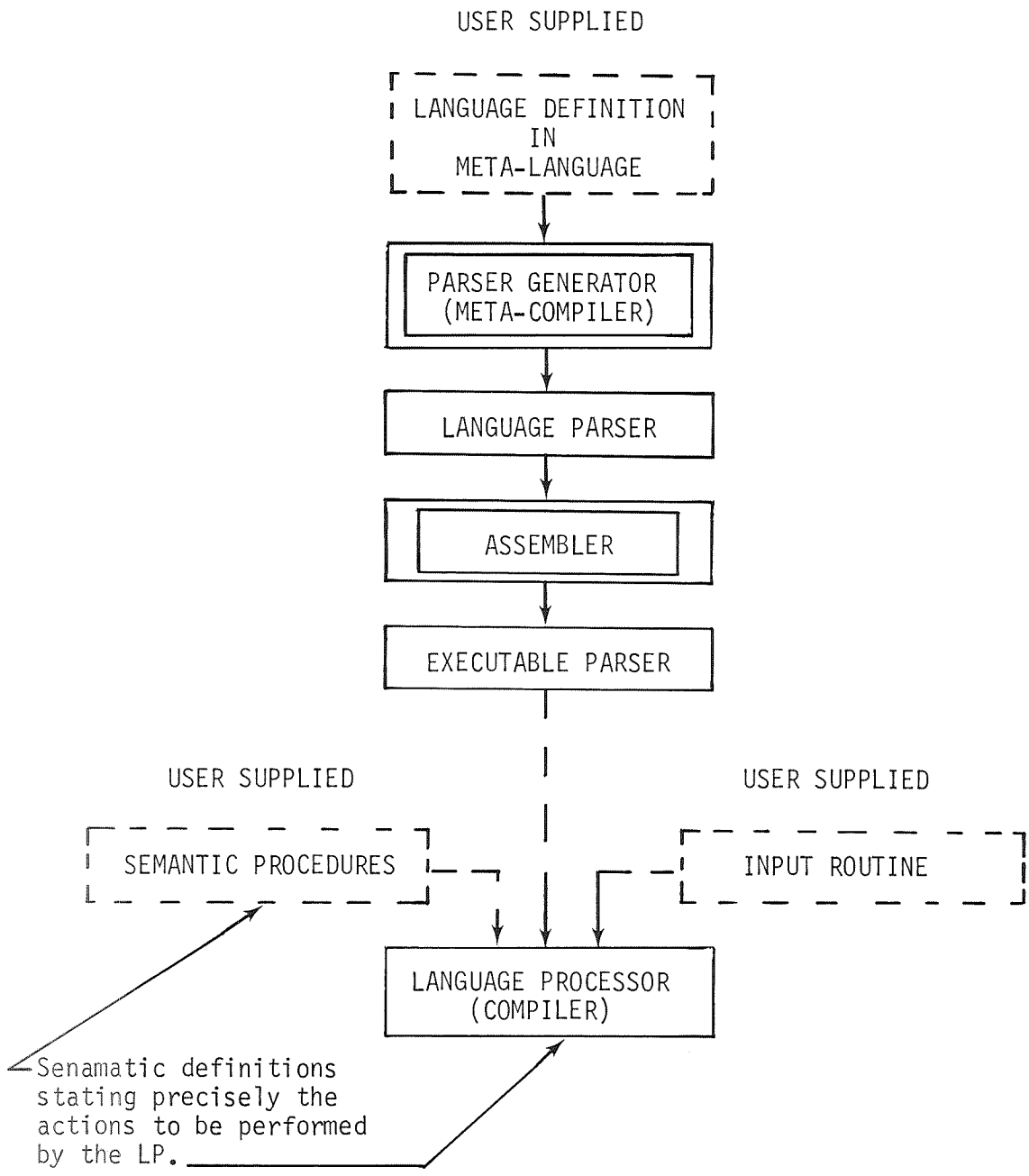


Figure F-1. Meta-Compiler Application

user supplied components, solid boxes represent automatically generated boxes, double solid boxes represent significant meta-compiler components, solid lines represent data flow, and dashed lines represent system integration functions which usually require nothing more than the application of certain of the computer's operating system utilities (e.g., the link editor or loader, FORTRAN compiler, etc.).

Beginning at the top of Figure F-1, the user must prepare a definition of the syntax for the new language using the meta-language acceptable to the meta-compiler. This meta-language has the facilities for specifying exactly which semantic procedures are to be associated with the various syntactic elements. So, the linkage between the syntax definition and the semantic procedures are specified by the user while developing the meta-language definition. Conceptually, this is very similar to stating in a main program that a subroutine must be called to perform a specific function. The syntax definition represents an abstraction of a main program, and the semantic procedures correspond to the subroutines.

The parser-generator accepts as input the meta-language description and compiles it into a parser for the language being implemented. This parser has built into it, the parsing directives needed to completely analyze the source input plus executable linkages to the semantic procedures. The output generated by the meta-compiler consists of a code which can be assembled by a macro assembler. This code is then passed to the assembler to generate an object time version of the desired parser.

The meta-compiler was designed so as to maximize its utility. As a result, no input routine is automatically generated for use by the parser. It was felt that this could often restrict its applicability. As a result, the user must provide the input facilities. The executable parser which has been developed so far in this discussion will interface with FORTRAN coded subroutines. All linkages have been organized in this manner. The result is that all semantic procedures, as well as the input routine, can be implemented as FORTRAN subroutines, and standard operating system utilities can be used to perform the integration indicated by the three dashed lines.

Before concluding this discussion there is one point worthy of mention regarding the implementation of the meta-compiler. In Figure F-1, the generated code which comprises the block labeled "language parser" will consist of a set of macro invocations. That is, in order to assemble the language parser, a complete set of macro definitions will need to be provided. These macro definitions will be made available as part of the total meta-compiler package, and need not burden the user. They are mentioned to demonstrate a very powerful feature of the package. The code generated by the meta-compiler consists entirely of macro invocations. This means that the generated code is essentially machine independent. In order to develop a language processor which can execute on a machine other than the host machine, it only becomes necessary to define the macros on the new machine, and assemble the new parser on its macro-assembler. This task is not trivial, but it does represent a significant degree of machine independence, particularly in view of the fact that the semantic and input procedures can be coded in FORTRAN, a relatively machine-independent language. It is also significant that the meta-compiler itself has been implemented using the same macro package. As a result, moving the meta-compiler from one host machine to another becomes greatly simplified.

F.4 SUMMARY

An overview has been presented of a tool which can be used for the development of language analysis processors. This tool consists primarily of a meta-compiler capable of generating operational parsers (possibly it should be called a "meta-parser") which in turn are potentially capable of processing context-sensitive grammars, and which are based upon syntax-directed techniques. Because of the apparent broad application for tools such as this, primarily the parsing problem has been dealt with. The development of the semantic procedures (actions which carry out the intent of the language) have been left as a task for the user, however, he is given considerable assistance in his task through the availability of a very powerful meta-language.

No attempt has been made in this discussion to provide a sufficient level of detail to allow a user to apply the tool. Rather, an attempt has been made to describe the concepts involved so that a potential user might assess its utility. Much of the tool's capability lies in the talents of the user, however, it does have a great deal to offer the language implementer in terms of time and cost savings.

Appendix G
USING THE SYNTAX DIRECTED COMPILER

As an explanation of a typical use of the syntax directed compiler, consider the language of the onboard experiment. Typically, this language would include a data base query capability as well as a command capability. For the purpose of this example, the syntax definition for one command, specifically a point command, will be illustrated and described.

A typical command format for the x-ray polarimeter, telescope free flying module experiment might be, "point telescope-1 at vega on 5th day from launch at 0343," where telescope-1 is one of several instruments which may be specified, vega is one of many targets which may be specified (alternately azimuth and elevation may be specified), and the day for pointing is optional.

It is desired that the point command be recognized as the 7th type of command, that the instrument code be validated against a predefined table of valid instrument codes, that the target be validated as one of a table of known targets or that the azimuth and elevation be of the proper form. The instrument, target azimuth and elevation, as well as the time of execution, should be validated to assure that the given instruction can physically obtain the desired target at the specified time. The system scheduler must then be given instructions to point the appropriate instrument to the target at the required time.

SYNTAX DEFINITION EXAMPLE

1. TOKENS TNAME, NUMBER,

2. SYNTAX . = CØMMAND/QUERY/... .

3. CØMMAND . = FLAG10-('POINT', 'CALIBRATE',...), (ØN FLAG10 POINT,CALIBRATE,...) '. '/FLAG10 EQØ,SAVE 'ILLEGAL CØMMAND'.
4. PØINT . = INSTRUMENT,(Ø,1'TO') TARGET,WHEN(3P1)(4P2,7).
5. INSTRUMENT . = (1,1Ø ALMERICS,1\$-,CHARACTER,(1PI)/ EXIT 'ILLEGAL INSTRUMENT'.
6. TARGET . = TNAME (2PI)/CØØRD/EXIT 'UNKNOWN TARGET'.
7. TNAME . = LETTER,(1,1Ø ALMERICS).
8. CØØRD . = AZIMUTH,FLAG15=FLAG14,FLAG16=FLAG17,FLAG18=FLAG19,ANGLE.
9. AZIMUTH . = "AZIMUTH" WHERE/WHERE.
10. ANGLE . = 'ANGLE', WHERE,/WHERE.
11. WHERE . = DEGREES,MINUTES,SECØNDS.
12. DEGREES . = FLAG14=1(Ø,1,'-',FLAG14=-1),FLAG12=NUMBER,FLAG14*FLAG12, (Ø,1 'DEGREES').
13. SECØNDS . = FLAG17=NUMBER,(Ø,1,'SECONDS').
14. MINUTES . = FLAG19=NUMBER,(Ø,1,'MINUTES').
15. NUMBER . = (1,4 DIGITS).
16. WHEN . = FLAG20=Ø,FLAG21=Ø,DAY,TIME/NULL.
17. DAY . = 'ØN',FLAG20=NUMBER,CONTR,'DAY', (Ø,1 'FROM', 'LAUNCH')/NULL.
18. CØNTR . = FLAG22=('ND','TH','ST')/NULL.
19. TIME . = 'AT' FLAG21=NUMBER/NULL.

The syntax example is a definition of the point command: "Point telescope-1 at vega on 5th day from launch at 0343." The general format would be:

POINT INSTRUMENT, TARGET, TIME.

POINT = Command to be performed

INSTRUMENT = Specific device to be pointed

TARGET = Direction to point (may be a specific target or coordinates of target)

TIME = Time at which command is to be executed.

INSTRUMENT will be recognizable by a name followed by a dash (-), followed by a character.

EX: DEVICE - X ,

The target argument will be recognized by the SDC by either isolating the target name or the target coordinates. The target name will be contained in a table of predetermined targets. The name will be a string of characters. Target coordinates will be recognizable by the numerics followed by the word degrees, and/or minutes, and/or seconds. The first coordinate will represent the azimuth and the second the angle of elevation. Both coordinates can be preceded by the word "AZIMUTH" or "ANGLE" as applicable. The coordinate may also be "plus" or "minus."

The TIME parameter is optional and is entered for the purpose of indicating the command execution time for automatic commands. If this parameter is omitted then "TIME-NOW" or "REAL-TIME" execution is assumed. The TIME parameter is entered by entering the day from launch, hour, and minutes of start time. The SDC will recognize the time by the format "a number followed by the word 'Day' followed by the military time representation."

When this command has been entered through an input terminal, the executive program will invoke the parser which was generated by the syntax directed compiler. The parser, when given this command, will recognize the form of this statement as satisfying the definition of a point command. In determining that the command is valid, two external procedures are executed. The first of these (procedure 1) verifies that the instrument code, which had previously been determined to be of the proper form, is indeed one of the available instruments. The second of these procedures examines the list of known targets to verify that the specified target is in the list. If the target is in the list, the azimuth and angle are stored as if they had been directly specified.

After the entire statement has been recognized as a valid statement, procedure 3 will be called which will verify that the instrument, target, and time are legitimate values and that the instrument can physically be pointed to the target at the specified time.

The fourth and final procedure will command the system scheduler to perform the required action. The call to the system scheduler will be of the form:

```
CALL SCHEDULE (POINT,INSTRUMENT,AZIMUTH,ANGLE,TIME)
```

where: POINT is the command type,
INSTRUMENT is an integer instruction code,
AZIMUTH and ANGLE are in SECONDS,
TIME is the time from launch.

The scheduling of this task completes the action required of the parser. Parser will then be ready to accept another input statement.

1000110

MCDONNELL DOUGLAS ASTRONAUTICS COMPANY

5301 Bolsa Avenue, Huntington Beach, California 92647 (714) 897-0311