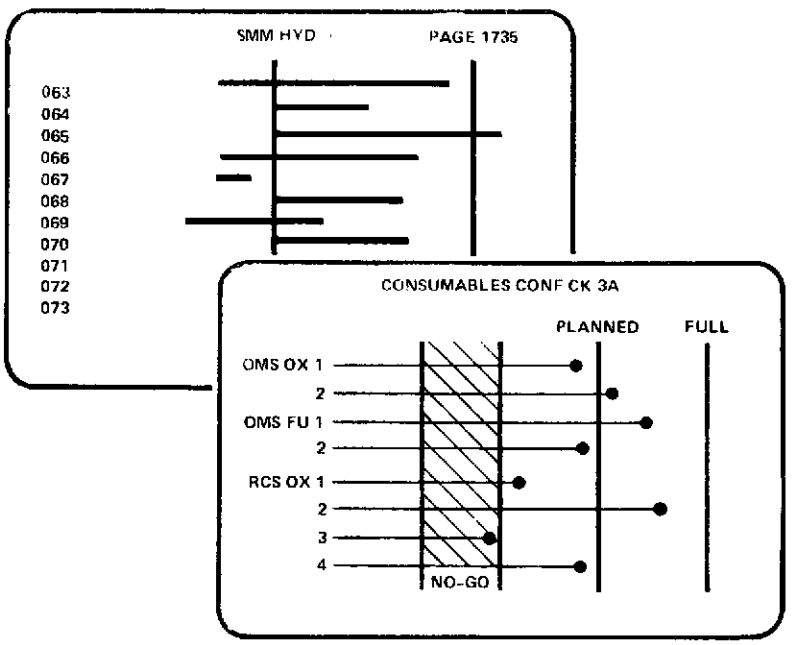
DRL NO. T-959
LINE ITEM 4
DRD NO. MA-129T

9330-75-005
FEBRUARY 1975
NASA CR-
141758

FINAL REPORT-
STUDY OF SPACE SHUTTLE ORBITER SYSTEM MANAGEMENT COMPUTER FUNCTION

VOLUME I
ANALYSIS, BASELINE DESIGN



Prepared for
LYNDON B. JOHNSON
SPACE CENTER, NASA
HOUSTON, TEXAS

Prepared by
ADVANCED PROGRAMS DEPARTMENT
UNDER CONTRACT NUMBER
NAS 9-13887

HARRIS
COMMUNICATIONS AND INFORMATION HANDLING

HARRIS CORPORATION Electronic Systems Division*
P.O. Box 37, Melbourne, Florida 32901 305/727-4000
*(formerly RADIATION)

9330-75-005
FEBRUARY 1975

FINAL REPORT
STUDY OF SPACE SHUTTLE ORBITER
SYSTEM MANAGEMENT COMPUTER FUNCTION

ANALYSIS, BASELINE DESIGN

VOLUME I

HARRIS CORPORATION, ELECTRONIC SYSTEMS DIVISION

Prepared for
LYNDON B. JOHNSON SPACE CENTER, NASA
HOUSTON, TEXAS
Under Contract Number NAS 9-13887

Distribution List

4 cys	C. T. Dawson Mail Code EJ5 Communications, Power & Data Systems Branch NASA Johnson Space Center Houston, Texas
1 Cy	C. S. Parks Mail Code BC731(40) R&T Procurement Branch NASA Johnson Space Center Houston, Texas
4 Cys	Retha Shirkey Mail Code JM6 Technical Library Branch NASA Johnson Space Center Houston, Texas
1 Cy	J. T. Wheeler Mail Code JM7 Management Services Division NASA Johnson Space Center Houston, Texas

ABSTRACT

A system analysis of the Shuttle Orbiter Baseline System Management (SM) computer function is performed. This analysis results in an alternative SM design which is also described. The alternative design exhibits several improvements over the Baseline, some of which are increased crew usability, improved flexibility and improved growth potential. It was discovered that analysis results were restricted so long as SM was regarded simply as a performance monitor. Not until SM was treated as a crew management infromation system, whose principal objective was to aid the crew, did analysis findings become meaningful. The analysis consists of two parts: an application assessment and an implementation assessment. The former is concerned with the SM user needs and design functional aspects. The latter is concerned with design flexibility, reliability, growth potential and technical risk.

The system analysis is supported by several topical investigations. These include: treatment of false alarms, treatment of off-line items, significant interface parameters and a design evaluation checklist.

The most extensive supporting investigation consists of an in depth formulation of techniques, concepts and guidelines for design of automated performance verification. The formulation outlines the design process and covers criteria for selecting functions for which verification is most effective, identification of performance measures, system partitioning, false alarm avoidance techniques, response time implications, treatment of status indications and methods for measuring performance. This formulation is supported by examples.

TABLE OF CONTENTS,

Volume I of 2

		Page
List of Acronyms and Abbreviations		iii
Conclusions		vii
Recommendations		ix
1.0	INTRODUCTION	1-1
1.1	Study Objectives	1-2
1.2	Report Organization	1-3
1.3	Study Program Synopsis	1-3
1.4	System Analysis Approach	1-4
2.0	DESIGN SUITABILITY — AN APPLICATIONS VIEWPOINT	2-1
2.1	A User Viewpoint — The First Assessment	2-2
2.1.1	Flight Crew Needs	2-2
2.1.2	Contrasts To The Baseline	2-6
2.2	Functional Analysis — The Second Assessment	2-8
2.2.1	Time Line Assessment	2-8
2.2.2	Design Independence	2-10
2.2.3	Process Criticality	2-11
2.2.4	Potential Process Ground Implementation	2-12
2.2.5	Credibility of SM Output	2-12
2.2.6	Functional Assessment Summary	2-19
2.3	Relationship To The Original Concept—The Third Assessment	2-20
2.4	Application Assessment Summary	2-23
2.5	An Alternative Design	2-25
2.5.1	Design Overview	2-25
2.5.2	Selected Processes	2-28
3.0	IMPLEMENTATION ASSESSMENT — THE DESIGNER VIEWPOINT	3-1
3.1	Technical Risk Assessment	3-2
3.2	Flexibility Assessment	3-5
3.3	Growth Potential Assessment	3-6
3.4	Reliability Assessment	3-8
3.5	Assessment Summary	3-8
4.0	SOME SPECIAL TOPICS	4-1
4.1	Treatment of False Alarms	4-1
4.1.1	Mechanized Avoidance	4-1
4.1.2	Procedural Avoidance	4-3
4.1.3	Mechanized Handling	4-4
4.4.4	Procedural Handling	4-5
4.2	Treatment of Off-Line Items	4-5
4.3	Sampled Data and Processor Decisions	4-7
4.4	SM Restarts and Initialization	4-9

Page

4.5 Ground Support Trades 4-10
4.5.1 STDN In The 80's. 4-11
4.5.2 Available and Required Services . . 4-15
4.5.3 Tradeoff Criteria 4-18
4.5.4 Candidates for Ground Support . . . 4-21
4.6 Critical SM Interface Parameters. 4-22
4.7 The SM Model Dilemma. 4-24
4.8 A Design Checklist. 4-25

5.0 TOPICS FOR FURTHER STUDY. 5-1

APPENDIX A SOME ADDITIONAL CONSIDERATIONS
APPENDIX B BASELINE DEFINITION — SYSTEM MANAGEMENT (SM) OPERATIONAL VERSION
APPENDIX C ORIGINAL SM CONCEPT — A PERFORMANCE SPECIFICATION

LIST OF ACRONYMS AND ABBREVIATIONS

ACM	Acquisition and Control Module
ACT	Action
A/D	Analog to Digital
ADI	Attitude Director Indicator
ADP	Automatic Data Processing
A/L	Approach and Landing
ALT	Approach and Landing Test
AMEC	Aft Master Events Controller
AN	Alphanumeric
A&P	Attitude and Pointing
APU	Auxiliary Power Unit
ASA	Aerosurface Servo Amplifier Assembly
BCE	Bus Control Equipment
BITE	Built-In Test Equipment
BPS	Bit Per Second
BTU	Bus Terminal Unit
B/U	Back Up
CBN	Cabin
CCD	Constants Change Display
DACBU	Data Acquisition Control and Buffer Unit, also called PCM master unit
DBN	Data Bus Network
D&C	Display & Control
DDU	Display Driver Unit
DEG	Degrees
DET	Digital Event Timer
DEU	Display Electronics Unit
DFI	Development Flight Instrumentation
DISP	Display Function
DMA	Direct Memory Access
DPS	Data Processing Subsystem
DP&S	Data Processing & Software
DSKY	Display and Keyboard
DU	Display Unit
EC	Events Controller
ECLSS	Environmental Control and Life Support System
EIU	Engine Interface Unit
EOF	End of File
EPS	Electrical Power System
ET	External Tank
ET	Elapsed Time
EU	Engineering Unit
EU	Electronics Unit
EXEC	Execute
FAA	False Alarm Avoidance
FC	Flight Computer
FCOS	Flight Computer Operating System
FCS	Flight Control System
FDA	Fault Detection Annunciation
FDI	Fault Detection and Isolation
FDIR	Fault Detection Isolation and Recovery
FI	Fault Identification

FKB	Flight Display Keyboard
FMEC	Forward Master Events Controller
FOF	First Operational Flight
FPFD	Functional Path Fault Detection
FS	Fault Summary
FSP	Fault Summary Page
FSSR	Functional Subsystem Software Requirements
FVF	First Vertical Flight
G&C	Guidance and Control
G&N	Guidance and Navigation
GN&C	Guidance, Navigation and Control
GPC	General Purpose Computer
GSE	Ground Support Equipment
GSFC	Goddard Space Flight Center
HAL	Houston Aerospace Language - Compiler
HSI	Horizontal Situation Indicator
IBV	Item Being Verified
ICC	Intercomputer Channel
I/O	Input/Output
IOP	Input/Output Processor
IPL	Initial Program Load
ISS	Inhibit/Override Summary Snapshot Display
IU	Interface Unit
JSC	Johnson Space Center, Houston
KBU	Keyboard Unit
KYBD	Keyboard
KSC	Kennedy Space Center, Florida
LDB	Launch Data Bus
LDS	Landing/Deceleration Subsystem
LPS	Launch Processor System
LRU	Line Replaceable Unit
M	Meters
ma	milliampere
MCDS	Multifunction CRT Display System
MCIU	Master Controller Interface Unit
MDM	Multiplexer/Demultiplexer
ME	Main Engine
MEC	Master Events Controller
MECO	Main Engine Cutoff
MET	Mission Elapsed Time
MIA	Multiplexer Interface Adapter
MM	Mass Memory
MMU	Mass Memory Unit
MOP	Measure of Performance
MPS	Main Propulsion System
MSBLS	Microwave Scanning Beam Landing System
MSC	Master Sequence Controller
MSC	Moding, Sequencing & Control
MSS	Mission Specialist Station
MSU	Mass Storage Unit
MTC	Master Thrust Controller
MTS	Magnetic Tape System
MTU	Master Timing Unit

MUX	Multiplex
NASCOM	NASA Communications Network
NSP	Network Signal Processor
N/W	Nose Wheel
OFI	Operational Flight Instrumentation
OFT	Orbital Flight Test
OI	Operational Instrumentation
OMS	Orbiter Maneuvering Subsystem
OPS	Operational Sequence
PCI	Program-Controlled Input
PCM	Pulse Code Modulation
PGS	Power Generation Subsystem
PHS	Payload Handling Station
PIC	Pyroinitiator Controller
PLD	Payload (Alternate P/L)
PLM	Payload Management
PMAD	Performance Monitor Annunciator Driver
PMS	Performance MonitoringSystem
PP	Partial Pressure
PROM	Programmable Read Only Memory
PVS	Performance Verification System
PVT	Pressure Volume Temperature
PSW	Program Status Word
RA	Radar Altimeter
RAM	Random Access Memory
RC	Recorder Control
RCC	Redundance Connection Control
RCS	Reaction Control System
RDD	Requirement Definition Document
RF	Radio Frequency
RM	Redundance Management
RMC	Redundancy Management Control
ROM	Read Only Memory
RREU	Rendezvous Radar Electronics Unit
RS	Redundancy Status
RTC	Real-Time Command
RWM	Read/Write Memory
SAIL	Shuttle Avionics Integration Laboratory
SBHC	Speedbrake Hand Controller
S/C	Signal Conditioner
SCC	SCM Comparison Display
SCM	Subsystem Configuration Monitoring
SCU	Sequence Control Unit
SDL	Software Development Laboratory
SGLS	Space-Ground Link Subsystem
SMM	Subsystem Measurement Management
SM	System Management
SMI	SM Initialization Display
SRB	Solid Rocket Booster
SRM	Solid Rocket Motor
SSDD	Software System Design Document
STDN	Space Tracking and Data Network
S/W	Software

TAFM	Terminal Area Energy Management
TBD	To Be Determined
TDRSS	Tracking and Data Relay Satellite System
TFS	Telemetry Format Selection
TLM	Telemetry
TTY	Teletypewriter
TVC	Thrust Vector Control

CONCLUSIONS

a. Checking the status of functional paths — as opposed to checking numerous parameters — is both feasible and desirable. The feasibility is supported by Volume II to this report and the desirability by Section 2.0, et seq in this volume.

b. The Baseline SM Design is both feasible and technically adequate. It can, however, be improved in the following ways.

- Usability by the crew
- Credibility of SM decisions
- Adding functional path status checking
- Reduced C&W response time
- Increased flexibility and growth potential

This is supported by Sections 2.0 and 3.0 et seq.

c. Needs of the user were not adequately considered in the design of SM. Taken in total context, SM is not just a performance monitor. Rather, it is a crew management information system. This is supported by Sections 2.0, et seq. and 4.6.

d. Off-line items are probably best checked on a scheduled basis (as opposed to attempted continuous checking) and this check is most practically implemented by successively switching them on line. This is supported by Section 4.2.

e. SM cold restarts are viable. This is supported by Section 4.4.

f. It is feasible, should the need arise, to implement all scheduled SM functions on the ground provided TDRSS is used. This is supported by Section 4.5.

g. The concepts and techniques of automated fault detection can be formulated into a design discipline. This is supported by Volume II to this report.

h. The critical SM interface is the crew interface. This is supported by Section 4.6.

i. The technology area of management information systems and performance monitoring have not yet reached a stage

Formal enough to warrant practical analytical cost/ performance prediction simulations. Analytical assessment techniques are, however, realizable. This is supported by Sections 4.7 and 4.8.

j. Precondition Steering does not affect the sample rate of subsystem parameters, only the necessity that state information be sampled close to the parameter sample time. This is supported by Section 4.3.

k. Step response is a universal performance measure for false alarm avoidance techniques. For the avoidance method used in the Baseline Design, its step response represents the fastest response it can possibly have. This is supported by Sections 2.2 and 4.1.

l. Failure to adequately consider the user apparently originated with failure to perform an operations analysis. This analysis should have been part of the RDD. In short, a specification step was omitted. This is supported by Section 2.4.

m. SM will require more than a performance monitoring function. It will also require a performance verification function, the distinction being that the latter not only monitors but also verifies acceptability of performance. This is supported by Section 2.2.

n. The alternative SM design developed in this study significantly improves the crew's ability to manage Orbiter resources and to effect positive action in recovery from emergencies. In terms of design measures, crew workload has been reduced, crew responsiveness (to unforeseen circumstances) has been increased and autonomy has been increased. To achieve these program improvements, additional SM design efforts will be required. This is supported by Sections 2.0, et seq. and 3.1. Particular support for the SM role is contained in Section 2.2.3.

RECOMMENDATIONS

a. In view of the time remaining until the First Operational Flight, the Baseline Design should be reassessed considering the potential improvements resulting from this study.

b. Consideration should be given to including a set of nominal and a set of emergency parameter limits which are crew-selectable.

c. Determination of parameter limits and smoothing constants should not be a real time crew task. This should be done by past-flight data analysis when all the facts are in.

d. A Functional Path Fault Detector should be included in SM.

e. A Status Table and Redundancy Table should be included in SM.

f. The SM function of C&W should be functionally isolated from the rest of SM.

g. Some SM tasks can be accomplished procedurally. A trade study should be instituted which clearly identifies procedural tasks and SM tasks.

h. Current precondition steering logic does not include a NOT operator. Consideration should be given to adding this operator.

i. An SM operations analysis should be performed.

j. The topics for further study identified in Section 5.0 of this volume represent major areas to be fulfilled before SM is committed to a final implementation. Each should be undertaken.

1.0 INTRODUCTION

This final report, prepared under Contract Number NAS 9-13887, presents the results of an Orbiter System Management Computer Function Study and was conducted by Harris Electronic Systems Division, Melbourne, Florida.

The study has not only provided an SM system analysis and a discipline for designing performance verification systems, but more importantly, an alternative SM design which significantly improves the crew's ability to manage Orbiter resources and respond to emergencies.

The System Management (SM) Computer Function is intended to monitor the various Orbiter and Payload subsystems and make these results available to the crew. The function is implemented exclusively in software, extracting data from the Operational Instrumentation System as well as GN&C.

While forms of performance monitoring have been in existence for some time, application of this concept on a real time basis across an entire sophisticated system is a challenging task. Achieving a practical, credible implementation of this concept requires more than instrumentation technology. It requires a totally different way of thinking. Along these lines, one of the purposes of this study is to provide a fresh, objective look at automated performance verification.

It should be noted that this study evaluates the Operational Version of SM rather than the current Approach and Landing Test (ALT) Version. Four factors influenced this decision.

- The Operational Version is far enough in the future that study results would be timely.
- The Operational Version will not be subject to the schedule pressures and development dynamics experienced by the Current Version.
- The Current Version is not being implemented with the same software constraints as the Operational Version.
- Operational missions, procedures and environment can be used.

Before departing these introductory comments, it is important that SM be placed in perspective. Roles similar to that of SM are being performed by other Orbiter Functions. First, emergency warnings such as fire or loss of cabin

pressure are handled exclusively by a separate, dedicated, hardwired system. Second, for purposes of fast reaction automatic redundancy fault recovery, Guidance Navigation and Control (GN&C) performs its own fault detection. Only the results of this detection may be passed on to SM. SM does not have direct access to GN&C subsystems data. Finally, alerts signifying potential hazards, such as high Active Thermal Control temperature indications, fall into a category known as Caution and Warning (C&W). These events are handled by a dedicated, hardware system known as C&W which drives panel annunciators. SM provides a software backup to C&W. Of the remaining vehicle parameters, those most used or those of greatest importance are displayed on panel instruments at the several crew stations.

This introduction has been divided into four major topics. Objectives of the study are outlined in Section 1.1. This section is followed by a description of how the report has been organized in Section 1.2. Section 1.3 of the introduction provides a synopsis of the study program, describing the actual course of the study. This section also responds to the objectives cited in Section 1.1. Finally, Section 1.4 outlines the approach to the system analysis which occupies the majority of this volume.

1.1 Study Objectives

The overall objectives of this study are to, (a) establish analytical procedures and (b), perform a comprehensive systems analysis of the Operational Flight Version of the Shuttle Orbiter System Management (SM) computer function. The latter objective requires no additional comment. The first objective consists of three specific analytical procedures, the first of which is functional concepts for performance verification. This procedure includes:

- Criteria for selecting functions for which monitoring is most effective.
- Monitoring techniques.
- Criteria for selecting subsystem and redundancy string data extraction points.
- False alarm avoidance and recovery.
- Computer self-testing.

The second procedure includes the definition of pertinent tradeoffs between real-time onboard analysis and control and real-time ground-based analysis and command using the STDN with and without TDRSS.

The final procedure includes analytical simulation programs capable of predicting results of proposed changes in system configuration, operating parameters, and capable of evaluating the effectiveness of fault detection techniques.

1.2 Report Organization

The report has been basically organized around the study objectives. Volume I is dominated by the SM system analysis with some supporting investigations. Volume II is dedicated to the formulation of automated performance verification concepts and techniques.

Sections 2.0 and 3.0 of Volume I, this volume, contain the SM Baseline Operational Version system analysis. The reader should expressly note that the Baseline Design is described in Appendix B to this volume. This was done to preserve report continuity. Also, Section 1.4 below provides the structure for the system analysis.

Section 2.0 identifies the user needs and assesses the design suitability from an applications viewpoint. Based on these assessments an alternative design is developed and described.

Section 3.0 provides an implementation assessment by contrasting the Baseline Design to the alternative design. Section 4.0 contains investigations supporting the system analysis. Topics believed to merit further study are enumerated in Section 5.0.

Appendix A contains some additional SM design considerations. These considerations were not included in the alternative design since the implications of their integration were not completely investigated.

The contents of Appendix B were discussed above. The original SM concept is believed represented by Appendix C. This appendix contains an excerpt from the SM performance specification.

Volume II of the report structures the automated performance verification problem by critically examining what is demanded of such an endeavor and then providing general concepts, techniques and guidelines for achieving these demands. The concepts and techniques are applied in an example verification design for the Orbiter Hydraulics System.

1.3 Study Program Synopsis

The study objectives were addressed by first defining the concepts and then applying these concepts to the system analysis. Formulating the automated performance

verification concepts occupied the first half of the study. Attention was then turned to applying these concepts to the system analysis. It was soon discovered that while these concepts provided significant insight and direction, they did not address the whole SM problem. Basically, the concepts provided the where with all to address how performance verification was to be done. They did not answer what to do with the verification information or allow judgments regarding selection of the appropriate technique to use. There were obviously requirements and constraints over and above performance verification. Not until SM was viewed as a crew Management Information System, instead of just a performance monitor, did the source and character of the additional requirements come into view. This placed SM in a larger context and allowed the total analysis problem to be addressed. For it is crew information needs which controlled the "what" as well as technique judgments. It came as no surprise when the analysis found the Baseline Design had not given this same area sufficient attention.

Management Information Systems are user-oriented. Their performance is judged by what information the user needs and how well the system serves those needs. The analysis was then restructured with a significant portion addressing SM application.

All the study objectives were met except one, the SM cost/performance prediction simulator. At the time the analysis was being restructured, attempts began at defining this simulator. After considerable definition and some trial runs it was concluded that systems such as SM have not reached a formal enough stage in design definition to warrant a practical simulation. The problems centered around quantification and lack of sensitivity to system design changes. In lieu of a simulator, an analytical assessment technique was developed.

Considering the alternative design and extensive application analysis contributions which were not part of the original objectives, the study is believed to have contributed a good deal more than was originally intended.

1.4 System Analysis Approach

There has likely never been a design that could not be improved. The spirit of the analysis is one of design improvement and is not intended to be a condemnation. In keeping with this spirit, a constructive alternative is offered.

As indicated earlier, the SM Baseline version being analyzed is the Operational version rather than the ALT Version. So that this system analysis could be completed

on schedule, it became necessary to freeze design evolutions in October, 1974. At this time no Operational Version had been specified. Rather than attempt an analysis using a temporary test version environment, an Operational Version was defined as part of the study. This approach is believed to offer the greatest value to the Government. The Operational Version Definition was built on the following observations:

a. The ALT version constitutes a subset of the total Operational Version capability.

b. The design is to evolve through its various revisions to achieve the final version.

c. In light of (b), the processes, capabilities, procedures and crew interface used for the ALT version would remain the same in the Operational Version.

d. The Operational Version would comply with the complete capability defined in the original RDD, SD-72-SH-0103-8, Vol. 5-8, Jan 1974.

The Operational Version Baseline Definition contained in Appendix B to this volume uses the ALT functional definitions for those functions to be included in ALT. The remaining functions were defined per the RDD.

Will the accuracy of the Operational Version Baseline affect the study results? The answer is, very little. First, based on the rationale for achieving the Operational Version, the resulting definition should be a good deal more than just a "straw man." Second, the vast majority of improvements recommended by the study are centered squarely at ALT version definitions. Finally, the system analysis is not limited to identifying Baseline weaknesses, but more importantly, it establishes requirements for an operational version by stating what the design <u>should</u> be.

Turning now to the system analysis itself, any such analysis is based on disciplines applicable to the system under study as well as the definition of the system itself. The disciplines supporting the analysis were developed as part of the study and appear in Volume II of this report. The system definition, however, requires further discussion.

System definition must be distinguished from design definition. The latter describes an approach for realizing a system. The former describes what the system is/does. It represents the basic concept of the system. The importance of this distinction should become clear in the following discussions.

SM has been tagged as a performance monitor. So long as this definition (and that is unquestionably what it is) persists, the full realization of the system will suffer. For example, under this definition the performance verification discipline developed during this study was of limited value in the analysis. This is not because the discipline is incomplete. On the contrary, it is a comprehensive treatment of how to do performance verification design. It does not consider what the results are to be used for. Neither does it provide the criteria for selecting which technique is best or for establishing design priorities. Such considerations are derived from a larger context. Thus, the performance monitor definition of SM, without a larger set of considerations, is inadequate. And, the larger set of considerations are not explicit in any of the SM developmental documentation.

If an SM system analysis is to be performed, the additional considerations must be defined. The obvious starting point is in the system definition itself. What describes the totality of SM contribution? What does it do? Looking over the spectrum of SM development, a significant point comes to light; SM is first a user aid to managing vehicle/payload resources and then a performance monitor. SM can then be defined in the much broader context of a user management information system. In this context, the identity of the larger set of considerations becomes clear. These considerations involve the operations of the system, its application, its user interface and, indeed, its design suitability. These are the considerations which dictate what is to be done with the performance verification results, which allow design priorities to be set, which allow tradeoff of techniques.

To recap, a significant analysis result has already been identified, i.e., SM must be evaluated in the broader sense of its application. The remainder of this discussion will deal with how this is to be done. As might be expected, a system analysis which includes a redefinition of the system will necessarily involve development of additional requirements. A new perspective must be developed. To do this, an applications assessment has been performed and is contained in Section 2.0. In order that a complete perspective be achieved, the applications assessment consists of three independent assessments, each with a different viewpoint. These are:

- A user assessment which provides the user's viewpoint and develops user requirements.
- A functional assessment which provides the operations viewpoint and develops

operations requirements on the SM functional relationships.

- An original concept assessment which pursues Baseline compliance with the original SM concept.

The theme of these assessments is developing requirements which are necessary for SM to fulfill its larger role and which will allow application of the performance verification discipline. The Baseline is then assessed against these requirements and improvement areas are cited.

So that the assessment findings can be integrated in a concise manner, an alternative SM design is presented based on the developed requirements. This accomplishes two goals of the analysis. First, it presents a constructive summary of results. Second, it desensitizes the results to the accuracy of the Baseline Design. Whether the Baseline is accurate or not, the alternative design provides a benchmark against which any design can be measured.

The analysis has thus far concentrated on applications suitability and has culminated in an alternative design. What about implementation? Implementation must be considered. It is obvious that the alternative design has not been carried to the point of implementation. It could be implemented in dozens of ways. As of the study design freeze in October, the Baseline implementation had not been specified either. The Baseline per se could also not be analyzed from an implementation standpoint. There is, however, sufficient material by way of detailed functional descriptions, core estimates, CPU loading estimates and user interface to gain a reasonably accurate description of Operational Version implementation needs. An implementation comparative analysis is, then, quite feasible. Such an analysis is contained in Section 3.0. Here, implementation of the alternative design is evaluated relative to that of the Baseline.

Section 4.0 of this volume contains special investigations supporting the system analysis. Avoiding and handling false alarms is the subject of Section 4.1. This subject is treated for both procedural and mechanized approaches. Sections 4.2 through 4.8 advance the treatment of off-line items, sampled data problems, SM restarts, ground support trades, critical SM interfaces and last but not least, SM analytical techniques. Sections 4.7 and 4.8 support the necessity of considering SM applications.

Attention is also called to Appendix A which contains several SM design options which resulted from the system analysis.

2.0 DESIGN SUITABILITY — AN APPLICATIONS VIEWPOINT

As indicated in Section 1.4, this section addresses three aspects of Baseline Design suitability. They are: (a) a user assessment, (b) a functional assessment and (c), an original concept assessment. Each assessment was performed as independently as possible. The theme is one of building three self-contained cases, each with a different objective in mind, to establish a composite and complete picture of Baseline improvement candidates. Each assessment not only identifies problem areas in the Baseline but also advances solutions by indicating what the approach should be. The assessments, then also establish design requirements for an SM design. This section is completed by an alternative design which fulfills these developed requirements. This design was produced for three reasons. First, it offers a constructive alternative. Second, it is the most convenient way to achieve reader identification and at the same time tie all the assessment finding neatly together. Finally, since the Baseline represents an extrapolation of the existing design, it is more equitable to indicate what an Operational Version of SM should be, rather than to indicate what the Baseline should not be.

Since the alternative design has been produced from these application assessments, what about implementation considerations? This is best answered by considering the purpose of a functional analysis. Such an analysis concerns itself with the functional relationships and input-output of the design constituents. Also, it is the first level at which constraints are introduced on the implementation resulting from application of the system, its environment, interfaces, susceptibility to change, options which should be held open, etc. Thus, gross implementation requirements are also established in this section. These requirements are evaluated in Section 3.0 by contrasting alternative design implementation considerations with those of the Baseline.

The reader is reminded that much of this assessment is based on findings from several special investigations which preceded it chronologically. These investigations are contained in Section 4.0 of this volume and Volume II to the report. These investigations should be consulted for additional supporting material. Some of the topics in Section 4.0 provide further elaboration on those contained in Volume II.

The first applications assessment concerns the user viewpoint. It is contained in Section 2.1. Section 2.2 provides the second independent assessment, the functional analysis. Here, design requirements are developed based on operational considerations. Baseline shortcomings are also discussed. Section 2.3 contrasts the Baseline Design to the original SM concept. This assessment is performed to test the specification evolution.

Findings of the three independent assessments are consolidated in Section 2.4. This section also advances the probable cause of the design shortcomings. Finally, Section 2.5 presents the alternative design.

2.1 A User Viewpoint — The First Assessment

The purpose of this section is to assess the Baseline Design in Appendix B from a user's viewpoint. What do the users need? When do they need it? In what form do they need it? Three SM users have been identified. They are the crew, the payload owner and Ground Maintenance. There is very little available information regarding the latter user and an assessment of his needs was not attempted. The most important information to Ground Maintenance will be that which relates to anomalies which occur in flight and whose conditions cannot be duplicated on the ground. Extending this general statement to specifics is a crucial task and should be a topic for further study. (For that matter, the entire problem of analyzing vehicle flight data to pinpoint failed LRU's should prove to rank among the most challenging problems of the program.)

Of the remaining two users, interests of the payload owner are assumed served by the crew Payload Specialist who has direct access to SM. These two users will then be treated collectively as the crew.

An assessment of this sort is best approached by first developing a picture of what the user will need and then contrasting that composite picture with the design to be assessed. Section 2.1.1 develops such a picture and Section 2.1.2 provides the actual Baseline assessment by way of contrasts.

2.1.1 Flight Crew Needs

In order to ascertain the needs of the flight crew, one must project himself into their position under a plausible flight environment. To do this, mission scenarios have been developed using viable crew roles. It is not necessary to develop scenarios for the entire flight since SM use can be characterized by two distinct applications: (a) vehicle assessment check just prior to committing to a new mission phase, e.g., changing orbit and (b) on orbit stay. So as not to be too general in the scenario, the first application was specifically performed for the Post Launch phase which occurs once the vehicle has achieved its initial orbit. The same scenario will apply to all phase-change checks, e.g., deorbit, except that points of interest will vary.

2.1.1.1 Post Launch Phase

Assume that launch and orbital insertion have been successfully accomplished and the Orbiter is now in a safe

parking orbit. The crew must have information on sub-system status, consumables and desired vehicle configuration to perform the next maneuver in the mission profiles in order to make a GO/NO-GO decision on whether to continue the mission or return to base.

Table 2.1.1.1 presents a task outline of possible post launch vehicle assessment crew tasks when it may be desirable to assess the condition of the vehicle equipment following the stress associated with launch and its capability to allow successful accomplishment of the next phase of the mission. A gross level task analysis is shown in Figure 2.1.1.1.* It is an attempt to identify some of the decisions which the crew must make to assess mission "GO/NO-GO" status and to reflect the kind of information which could aid the crew in making these decisions. Sketches of CRT displays have been included as exemplary of the kind of information which could aid the crew decision-making process and not necessarily a representation of information display implementation.

In an operational situation such as the one under consideration, there are several levels of information required by the crew with each level satisfying specific functional requirements. In the stress of a time-critical situation, it is mandatory that the crew be informed when a failure occurs that could jeopardize flight safety and which requires immediate crew response. Extraneous information displayed will tend to distract the crew and increase their time to detect, recognize and respond to a critical display. When the situation has stablized, the crew may then wish to assess the effect of the fault on the vehicle's ability to accomplish the intended mission. Now it becomes necessary to display information to aid the crew in recalling or identifying specifically what malfunction has occurred. As in the previous situation, presentation of information not specifically required to identify the malfunctions will tend to confuse or add "noise" to the overall display environment as well as increase crew reaction time and probability of overlooking a malfunction which has occurred.

The first level of information will necessarily concern C&W and vehicle functional paths. Therefore, "exception displays" which list only the failed functional paths or C&W out of tolerance identifiers were given as examples of a gross identification level display. Based on crew knowledge of vehicle systems and the identification of the fault which occurred, the crew may be able to determine, without further information, the criticality of the fault to subsequent mission phase performance. For instance, assume that a TACAN 3 fault

*This figure has been placed at the end of Section 2.1 for the reader's convenience.

Table 2.1.1.1 Post Launch Vehicle Assessment Task Outline

Crew Task: To decide GO/NO-GO for next phase of mission.

Must be able to:

Vehicle

1) Ascertain current status of sub-system performance.

   - Are all functional paths "up"?
   - Extent of options and/or redundancy available?
   - Are all parameter measurements within nominal limits?

2) Ascertain current status of consumables.

   - Remaining quantity?
   - Quantity required for safe return?
   - Quantity required for performance of next mission phase?
   - Predicted quantity remaining after completion of next mission phase?
   - Predicted quantity for safe return after completion of next mission phase?

3) Ascertain vehicle configuration is correct for successful performance of next mission phase.

Payload

1) Ascertain current status of sub-system performance.

2) Ascertain current status of consumables.

IF: The above conditions are determined to be "GO", initiate next mission phase.

indication occurred during launch. This equipment would have no effect on the vehicle's ability to perform an orbital translation and could be relegated as not critical to next mission phase. The crew could then divert their attention to faults which might be critical in the upcoming mission maneuver.

Simple name identification of a fault might be insufficient for the crew decision on fault criticality. They may need further information indicating the consequences of a functional path which has malfunctioned. For example, assume that the functional path HYD 1 has failed. What vehicle functions does it service? Nose wheel steering? Main landing gear extension? Flight control actuation? And further, are there backup systems which could be switched or are there even redundant suppliers of these functions? This next level of display provides information on a specific functional path rather than the entire vehicle. This is illustrated by the display labeled "identify effect." In Figure 2.1.1.1, HYD 1 services nose wheel steering and HYD 3 is redundant. With this information, the malfunction could be judged not critical to the next mission phase by the crew in the present scenario.

If the malfunction were in a system that was determined to be critical to the next mission phase, a requirement might exist for a higher detail level of information about the elements which make-up a functional path. It may be necessary at this point to display actual measures, limits associated with each measure and interaction of the elements within the functional path. It may be desirable to present measurement readings of the parameters on the CRT and at the same time provide functional path block diagrams by means of a film viewer to allow the mission specialist to assess element interaction. The capability should exist to scan across systems, where several are performing identical functions, to make a comparative check, e.g., the three APU turbine RPMs or a comparison of the three catalytic convertor bed temperatures.

In each of the information display situations thus discussed, it is important not to present too much information. That is, information not required for the immediate decision at hand. However, it is equally important to provide information display which will aid the operator in assessing a particular performance measure. In the examples presented, this was done in one instance with a "lubber line" — the planned consumption bar shown on the hypothetical Consumables Quick Scan Display. The quantity indicators for the various consumables should track or exceed the planned consumption bar, which is positioned as a function of mission time for a planned mission profile, as estimated consumption rates. Not to exceed limits are also shown in the minimum safe return quantity bars.

Presenting performance measure indications such that desired performance will be located within a specified region of the display or will track a command line, considerably

enhances the operators ability to detect out-of-tolerance conditions; even in rapid scan situations with diverse performance measures — inches, pounds, gallons, psi. For the in/out tolerance level of information, this type of display enhances operator performances due to the fact that it is much easier to distinguish position location differentials than it is to mentally read and compare numerical values. The baseline design presents information such as LINE TEMP = 68.9°F on the System Measurement Management Display. If this is an out of limit condition, a "bug" is presented adjacent to the readout. No information is presented to indicate how far out of limits this indication might be or if it is in limits, whether it is approaching an upper or lower limit.

In the post launch vehicle assessment, if all the sub-decisions are favorable and a "GO" decision is reached for initiation of the transorbital maneuver, it might be desirable to initiate a payload checkout to determine its status. However, its quite conceivable that the payload checkout may not be accomplished until just prior to payload eject after performance of the transorbital maneuver. In either case, information would be required which indicated functional path performance within the payload, consumables status and payload system configuration prior to eject to prohibit ejection of malfunctioning payload. If the crew determines that payload capabilities have been significantly degraded, then the payload may be returned to base for repair.

2.1.1.2 On-Orbit Phase

During the on-orbit phase, crew information needs, other than those served by dedicated vehicle instruments, would probably be primarily concerned with long term performance monitoring and declaration of functional path out of tolerance conditions. It is expected that the Mission Specialist would periodically request the quick scan displays to allow an updated assessment of vehicle conditions and capabilities. The SM quick scan capability would relieve the crew of the rather monotonous and tedious task of reading extended numerical displays.

The continuous monitoring capability, viz., C&W and functional path fault detection, would reduce necessary crew vigilance. This would enhance the probability of rapid detection and recognition of a fault which would otherwise be reduced, resulting from crew performance degradation over time due to fatigue.

2.1.2 Contrasts To The Baseline

When comparing the above-developed needs with Baseline capabilities, the first observation to be made is, unfortunately, a rather sweeping one. There is very little similarity between

the crew needs portrayed in the scenarios and the Baseline Design. The information needed by the crew is available alright, but its presentation leaves something to be desired. The two single characteristics which are violated are information levels and content separation. In the Baseline, C&W and alert conditions, even though their occurrence is signaled differently, are intermixed on the same display. Assuming the crew must rely on this backup C&W, they will want to work out C&W problems first. These should not be cluttered with information from potentially dozens of parameters, many of which they will not even be interested in at the moment.

A significant level of information to the crew is the status of functional paths. Furthermore, this must be the next level of concern following C&W. They should not have to pour through parameters to determine this information which they need straight away. Once functional path status has been determined, the crew, at some convenient time, can investigate the whys and wherefores. The Baseline does not recognize this separation or level of information.

The Baseline parameter displays do not show how far a parameter is out of tolerance nor do they indicate how close an in-tolerance parameter is to its limits. In addition, the "digital type" presentation will be hard to track when parameter variations are of interest. Finally, it will be quite difficult for these displays to be scanned while station keeping.

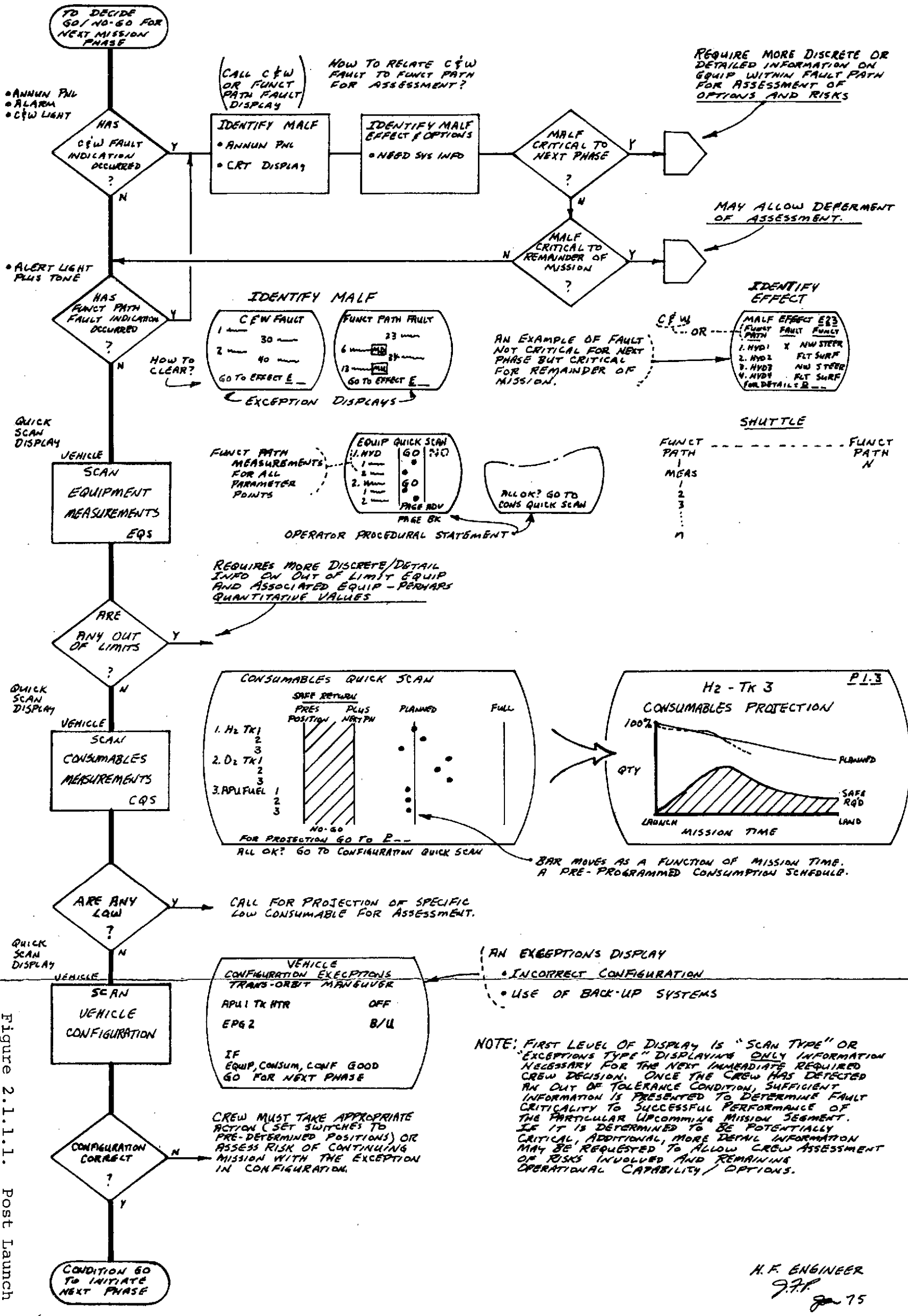
Remaining consumables should be part of a configuration check. There is currently no way for the crew to assess just how far they are out from minimum recommended levels.

When a functional path is identified as having failed, the Baseline presents no information (at least in defined displays) as to the consequences of this failure. Failure of one redundant path often leads to different consequences than the failure of another path in the same group.

If the crew is to actually assess the extent of their problem, they will need to know more than just the state of a heater or switch or the value read by a transducer. They will need to know their location and their relationship to other readings and functions. This can only be accomplished by system block diagrams (stylized to whatever degree). Placing these diagrams on mass memory is wasteful of storage and resulting CRT display will be of questionable quality. Besides, the crew member pursuing the problem should have the diagram and the SMM display simultaneously. These diagrams should be either on indexed cards or on microfilm which can be viewed while viewing the SMM display. The latter is considered a better choice since there will likely be other data which will be of value to the crew, e.g., functional descriptions, which could be economically carried on board on this medium.

FOLDOUT FRAME

ORIGINAL PAGE IS
OF POOR QUALITY



NOTE: FIRST LEVEL OF DISPLAY IS "SCAN TYPE" OR "EXCEPTIONS TYPE" DISPLAYING ONLY INFORMATION NECESSARY FOR THE NEXT IMMEADIATE REQUIRED CREW DECISION. ONCE THE CREW HAS DETECTED AN OUT OF TOLERANCE CONDITION, SUFFICIENT INFORMATION IS PRESENTED TO DETERMINE FAULT CRITICALITY TO SUCCESSFUL PERFORMANCE OF THE PARTICULAR UPCOMMING MISSION SEGMENT. IF IT IS DETERMINED TO BE POTENTIALLY CRITICAL, ADDITIONAL, MORE DETAIL INFORMATION MAY BE REQUESTED TO ALLOW CREW ASSESSMENT OF RISKS INVOLVED AND REMAINING OPERATIONAL CAPABILITY / OPTIONS.

H. F. ENGINEER
J.F.
Jan 75

Figure 2.1.1.1. Post Launch Vehicle Assessment Task Analysis

ORIGINAL PAGE IS
OF POOR QUALITY

FOLDOUT FRAME

In concluding this assessment, it is recognized that some of the shortcomings identified here could easily be handled by procedures. It is recommended that a trade study be instituted to document which items will be handled by procedures and which by SM. In effect, this would trade paper for software.

2.2 Functional Analysis — The Second Assessment

Section 2.1 provided a user's assessment of the Baseline Design assuming scenarios to characterize user demands. This section will examine the defined Baseline processes from a systems viewpoint. In effect, the preceding section assessed the system Input/Output and this section will assess the internal system processing. The Baseline Design is defined in Appendix B to this volume.

The assessment will consider five design criteria: time line, design independence (with respect to vehicle/payload design), process criticality, potential implementation of process on the ground and credibility of system output. Results of the individual criterion assessments are collected in Table 2.2.6 in the summary of this section.

2.2.1 Time Line Assessment

A generic mission profile for the Orbiter is shown in Figure 2.2.1. The purpose of this assessment is to investigate when the SM functions will be used on this profile for purposes of core overlay and loading considerations as well as process partitioning considerations. The results will also be used in Section 2.2.4 when potential ground implementation is considered.

Each process will be classified as to whether it takes place:

C - continuously throughout the profile or at unscheduled or random points in the profile, or

P - at predetermined or scheduled points on the profile, or

S - during predetermined or scheduled periods on the profile.

In the first classification, no distinction has been made between a function being scheduled for the entire mission and one that will be required at unpredictable times. This is based on the observation that the underlying cause of functions being required randomly is the occurrence of an out-of-tolerance parameter in the Orbiter subsystems. Since it is the purpose of SM to detect such conditions, there is obviously no way to schedule the appropriate function within any reasonable call-up time. The function will, then, effectively be required on a continuous basis.

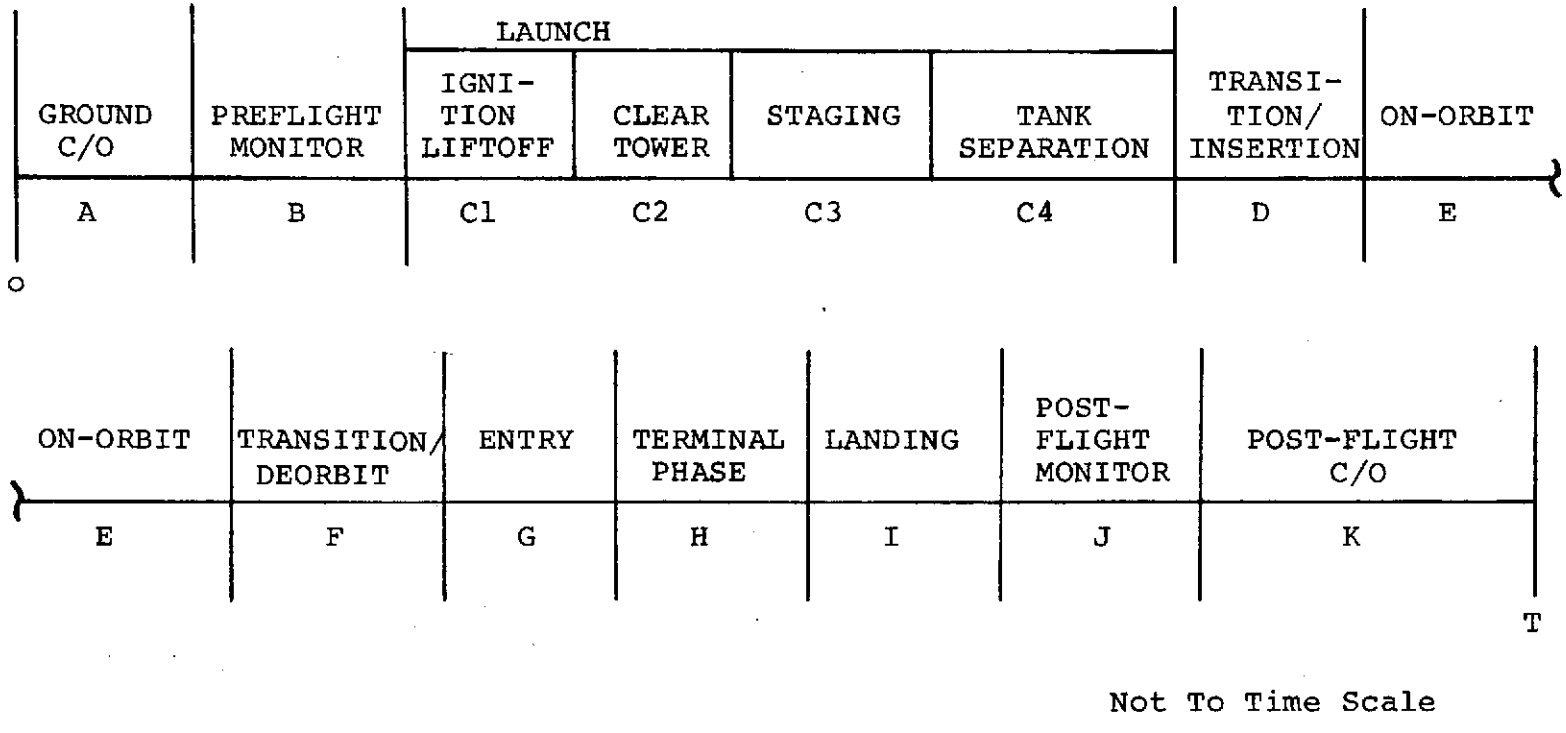


Figure 2.2.1. Orbiter Mission Profile

A distinction has been made between points in time and periods of time. The purpose of this distinction is to allow an assessment of the degree to which residency and background/foreground requirements may conflict. For practical purposes, a point in time is considered to describe an operation which is very short compared to the mission duration (several minutes). A period of time is not necessarily short (several hours).

Fault Detection and Annunciation (FDA), Data Recording Management (DRM) and portions of Payload Support are clearly classified as continuously required processes. The fact that only portions of Payload Support, viz., the counterparts to vehicle FDA, are continuous processes indicates that this process can be partitioned into two processes for overlay and ground support considerations. This option should be left open for further trades, implementation considerations and growth assessments. Since the Baseline makes no time line distinctions regarding partitions of Payload Support, the entire process must be classified as taking place in continuous time.

Subsystem Measurement Management (SMM) is used by the crew in the Baseline Design as a necessary adjunct to FDA. For, without SMM, FDA results are far from conclusive. Therefore, SMM will also be considered to be required on a continuous basis.

Subsystem Configuration Monitoring (SCM) and Telemetry Format Selection (TFS) are used at predetermined points in the mission. Referring to Figure 2.2.1, SCM might be used at the end of Mission Phase D, several times during Phase E and certainly just prior to Phase F. TFS usage would somewhat correlate with these same events.

On the surface, Consumables Management (CM) would appear to be required on a continuous basis due to "random" call-ups by the crew. This argument is valid, not due to random crew call-ups, but rather due to that portion of CM which is required to log consumables consumption for historical consumption rate displays. That portion of CM which generates displays is another matter. The fact is that, upon call-up, when this portion of the process is actually performed is not critical (so long as a "reasonable" response time is realized). Thus, this portion of CM can exploit call-up time and be considered as being required at scheduled points in the mission (there is sufficient notification time to "schedule" the event). CM, like Payload Support, should be partitioned into two processes —and for the same reasons. The Baseline does not make a distinction for CM. The entire process must then be required continuously for purposes of the Baseline assessment.

Ground Checkout Support is clearly a scheduled period which corresponds to Mission Phase A.

2.2.2 Design Independence

This design criterion is intended to reflect the degree to which the design of an SM process is free of the effects of a design change in the Orbiter Subsystems. This section will discuss the considerations in general. A scoring of this criterion appears in Table 2.2.6.

Two factors must be considered in this assessment: (a) the volatility of the Orbiter subsystem design and (b), the degree to which the SM design is "tied" to that design. The first factor is quickly disposed of. There is but one Orbiter "subsystem" which is designed to be volatile or flexible. That is the payload. Any portion of Orbiter avionics which interfaces with the payload must be able to accommodate changes. This directly implies that a separate software partition be allocated to payload interface so that it may be changed without affecting design of the remaining SM processes, e.g., FDA, SMM. This is an additional reason for a further partitioning of Payload Support as indicated in Section 2.2.1. It should be noted that, while a Payload Support process is identified in the baseline, there are no restrictions as to how this process is to be handled.

Returning to the degree to which SM design is tied to Orbiter subsystem design, it will be expedient to dispose of those processes which are the most independent. (Payload Support will not be considered again.) Data Recording Management, Telemetry Format Selection and Consumables Management are relatively insensitive to Orbiter design changes. If for no other reason, this can be argued on the basis of their simple interfaces. FDA, SCM and SMM pose quite another problem. These processes operate on a large number of diverse subsystem parameters as well as logical relationships among them. These processes can be quite sensitive to Orbiter design changes. Furthermore, any routine for assimilating this vast amount of data can change with crew data demands and even mission phase. To minimize this susceptability to change (and to increase growth potential) a data acquisition process should be defined. Such a process would place a standard interface at all other SM processes and would also provide the advantage of buffering.

The Ground Support process will also be very sensitive to extra-SM design changes, viz., those occurring in the Launch Processing System (LPS). Since this purpess is operating in an entirely different regime, it will not be pursued here.

Two remaining processes are an Uplink Service and a Downlist Control. The former services commands/data from the ground (R.F. or Umbilical), accommodating communications protocols, queueing and routing. The latter controls the SM data to be sent to DACBU for inclusion into the telemetry stream. These data are currently manually selectable. Both

these processes have manageable interfaces and should be rather insensitive to Orbiter design changes.

2.2.3 Process Criticality

It is best to begin a discussion on criticality by placing the subject in perspective. Loss of the entire SM function will in no way impede the flight capability of the Orbiter. Furthermore, it will not have a significant effect on the inherent safety of the craft. The likelihood of a safe mission abort will be but slightly diminished. Such a loss will, however, have a noticeable effect on successful mission accomplishment. The crew can still fly the craft but they cannot be informed about its health or condition. In addition, payload checkout will be much more complicated, if at all possible, since it would have to be accomplished using a spare CPU tailored (on the spot) for this task. Finally, ground turnaround could be extended due to insufficient data on the maintenance recorder. Depending upon the implementation, this could be overcome by the crew manually controlling the recorder. This would imply that manual recorder control not be accomplished through software. A panel switch with talk-back should be used.

Depending on your point of view, loss of SM either results in almost total loss of autonomy (information processed and analyzed on the ground) or a diminished mission success probability. In actuality, the result will likely involve both extremes.

Loss of SM will limit the crew's knowledge of Orbiter status to that which is displayed on panel instrumentation (which is predominantly flight-critical). They will be able to respond to emergencies as they arise but will not be able to plan, to alter their behavior or that of the craft in advance of these emergencies. They lose a great deal of control of their own destinies. While not expressly stated, SM, in addition to apprising the crew of vehicle health, status, etc., also affords the crew the capability to effect fixes (not repairs, however). They can actively plan and improve their condition. What are the advantages of man over machine in a spacecraft? What are the advantages of manned vs. unmanned vehicles? Man adds three important ingredients which no machine has thus far been able to duplicate on the same time base. They are rapid adaptability, spontaneity and resourcefulness. The price paid for these commodities is the weight, volume and power required to sustain life as well as the weight and risk of the men themselves. SM is a major link between the man and the machine. Without it, arguments for a manned Orbiter are necessarily diminished.

In summary, if the Orbiter were unmanned, there would be very little need for SM. But, since it is manned, the role of SM is far from trivial. It is difficult to attach a true value to its contribution, however, since its

role is almost exclusively passive and well removed from the mainstream of activity. An attempt has been made to rank SM process criticality in Table 2.2.6. Those processes which affected safety were scored the highest with the remaining processes being scored on the amount of information lost when the process was lost. This lost information directly affects crew capability and mission success.

As a result of this assessment, two significant observations can be made. First, that portion of FDA which serves as C&W backup is the only part of SM that can be considered to affect safety. Its criticality ranking is far above all the others. FDA should be partitioned into two processes: C&W and alert. This will allow C&W to be treated individually and also not require C&W restrictions to be passed on to alert class information.

The second observation deals with a point already made in Section 2.2.1. That is the implicit but real association of SMM with FDA. In the Baseline, SMM must be used to interpret FDA results. It is, then, just as critical. It should not be. SMM should be used to refine FDA information when and if the crew desires to do so.

2.2.4 Potential Process Ground Implementation

It is recognized that none of the SM processes are planned to be implemented on the ground as this would decrease autonomy. This topic has been addressed solely on the basis of a design contingency measure; if it doesn't fit on board, can it be implemented on the ground? The analysis is contained in Section 4.5 and is easily summarized by stating the processes identified in Section 2.2.1 as continuous processes are not likely ground implementable. The remaining ones can feasibly be implemented on the ground. Adhering to the repartitioning recommendations in that section as well as those in Section 2.2.3 will make this an even more viable option. The reader should refer to Section 4.5 for more details and to gain more information about the impact of TDRS on this option.

2.2.5 Credibility of SM Output

What is credibility? It is ones willingness to believe a statement when presented as factual. And the ... "as factual" is important. When a statement is qualified in some manner such that it is presented as supposition and not fact, ones credibility is not reduced if the statement turns out to be erroneous. This argument is crucial to achieving credible SM results. SM is a purveyor of facts and as such should be correct the vast majority of the time. It is unreasonable to expect that SM will be without error and equally unreasonable to penalize the design which admits

to errors. It is, then not without reason to require such a design to qualify results which it has reason to believe are suspect. The credibility problem, then, can be approached from two ends: reduce the likelihood of error and qualify results which are suspect. The former approach is the subject of Sections 2.2.5.1, 2.2.5.2 and 2.2.5.3 while the latter is discussed in Section 2.2.5.4.

If SM is to fulfill its expectations, its results must be credible. Is credibility a problem with the Baseline Design? This question is best answered by another question. Why is virtually every SM operating parameter changeable in-flight? This capability adds a great deal of programming complexity, consumes core and increases I/O traffic. It complicates the use of SM and could well decrease credibility since no discipline exists for controlling or auditing changes. The change capability contributes little, if any, to flexibility and will probably be of marginal use in helping to establish SM parameters in actual flight conditions. It is recognized that it is not possible to establish firm limits and false alarm constants based solely on paper analyses. It is also recognized that uncontrolled "fiddling" with these parameters will not go much farther. This is a task best left to postflight ground computer analysis using telemetry data. Determine the values in non-real-time with the power of a ground processor when all the facts are in. By the time of the first operational flight, the values should be pretty well established.

While on the subject of SM parameters, there is a gross distinction that must be made regarding the limits set for parameters. Regardless of how they are arrived at, these limits represent someone's belief of what constitutes the dividing line between acceptable and unacceptable. The crew, however, will likely see this judgement differently. They may go along with a set of nominal limits so long as everything is running smoothly. When they get into a tight place, they are willing to judge performance less critically. They are simply not going to give up if a value is declared unacceptable under nominal conditions when it is still good enough to get them home with reasonable safety. Furthermore, it is under just such circumstances that SM could be their best aid in determining a strategy. To fulfill these requirements, each critical parameter should have two sets of limits, one representing nominal performance and the other representing emergency performance. The latter should be set such that it is admittedly riskier but not disasterous. The crew will likely want to assess their situation with the entire vehicle being judged under one or the other sets of limits. This precludes individual parameter selection, allowing a single decision to be made. SM simply uses either all nominal limits or all emergency limits.

2.2.5.1 C&W Performance Vs. Alert Performance

C&W class conditions and alert class conditions are handled together in the common process of FDA. These two conditions vary dramatically in their criticality and response time requirements. C&W conditions are quite critical and demand rapid response time with very low false alarms. Response times for alert conditions are not so critical and some false alarms can be tolerated. It may seem that both conditions could be accommodated by the single process with adjustments in limits and false alarm avoidance constants. Unfortunately, this is not the case.

Before advancing this argument and offering a solution, some ground work must be layed. First, this assessment is based on the analysis contained in Section 4.1 and material in Section 8.0 of Volume II of this report. Second, the definition of a false alarm is self-explanatory but that of a miss is not often as well understood. A miss occurs when a condition is not recognized within a prescribed notification time. Given enough chances, almost any scheme will eventually detect an out-limit condition. Thus, any process which replicates decisions on the same parameter (such as FDA) will, with high probability, eventually discover an out-limit condition. A miss, then, is tied to response time and the two terms are often used to describe the same behavior (although they are strictly two different performance criteria of a decision process). More will be said about misses in Section 2.2.5.3.

Returning to the argument advanced at the outset of this section, it will be appropriate to examine the FDA process. Here, decisions are made on the respective parameters and these decisions are passed on to a smoother which demands "N" consecutive decisions before it declares the parameter as out-of-limits. The issue is the smoother or false alarm avoidance. As the name implies, the scheme is quite effective at avoiding false alarms. It also, unfortunately, has a very high miss probability or response time for certain kinds of parameter behavior. C&W conditions are almost exclusively parameter value checks which will be typified by a gradual crossing of the parameter limit. Furthermore, the value of this limit is usually quite critical, i.e., there is little latitude in changing its value. The smoothing scheme employed for C&W will have a step response of "N" counts which is usually quite good for data which is typified by jump behavior such as catastrophic failures.

When a step response is cited for a smoother, it is usually assumed that it will do better for all other kinds of inputs. This is not so for the "pump-up" or post-decision scheme used in FDA. In fact, its step response represents the shortest possible response it will have to

any input and any value of "N" greater than unity. Thus the use of post-decision schemes as a smoothing device for C&W is a questionable practice. It will typically result in extrapolated response times or increased miss errors.

Since ample arguments have already been advanced for treating C&W separately, it is worth while to consider an alternative method of smoothing. The method is first order recursive smoothing and it has the advantage of guaranteeing a maximum response time. The step (worst case) response of this smoother is shown in Figure 2.2.5.1 which is reproduced from Volume II of this report. The smoothing constant, alpha, is selected to achieve the maximum reduction in parameter variance (minimum probability of false alarm) while not violating the response time requirement. The reader is referred to Volume II for further performance details, and to Section 4.1 for implementation considerations.

In operation, the smoother is placed before the decision device, thus the name predecision smoothing which appears in Section 4.1. The decision device or limit checker then operates on the smoothed parameter and once it finds the limit has been reached, annunciates the condition.

Recursive smoothing has one drawback that post-decision smoothing does not. It is sensitive to time jitter in the input data. This should not pose a problem with C&W data since its criticality justifies some extra care in data acquisition.

2.2.5.2 Error Reduction And The Number Of Decisions

It has been noted that C&W should be removed from FDA. What else can be done to reduce errors? In the Baseline Design, FDA is limit checking well over 1,000 parameters at an average rate of two per second. Are this many decisions necessary? It is recognized that misses are not of particular concern since the limit checks are replicated and response time is not too critical for alert conditions. It is further recognized that the false alarm avoidance constant can be set arbitrarily large, thus forcing false alarms, even for a seven day mission, to be few in number. The true issue involves both characteristics. The greater the number of decisions made, the higher will be the likelihood of a false alarm. Thus, the smoothing constants will have to be increased in value. This results in possibly unnecessary delay if the number of decisions can be reduced. Furthermore, FDA is deciding when a parameter goes out of limits and again decides if and when it comes back in limits. Since the crew are not told when a parameter becomes good again, its only utility is in helping to decide when the parameter

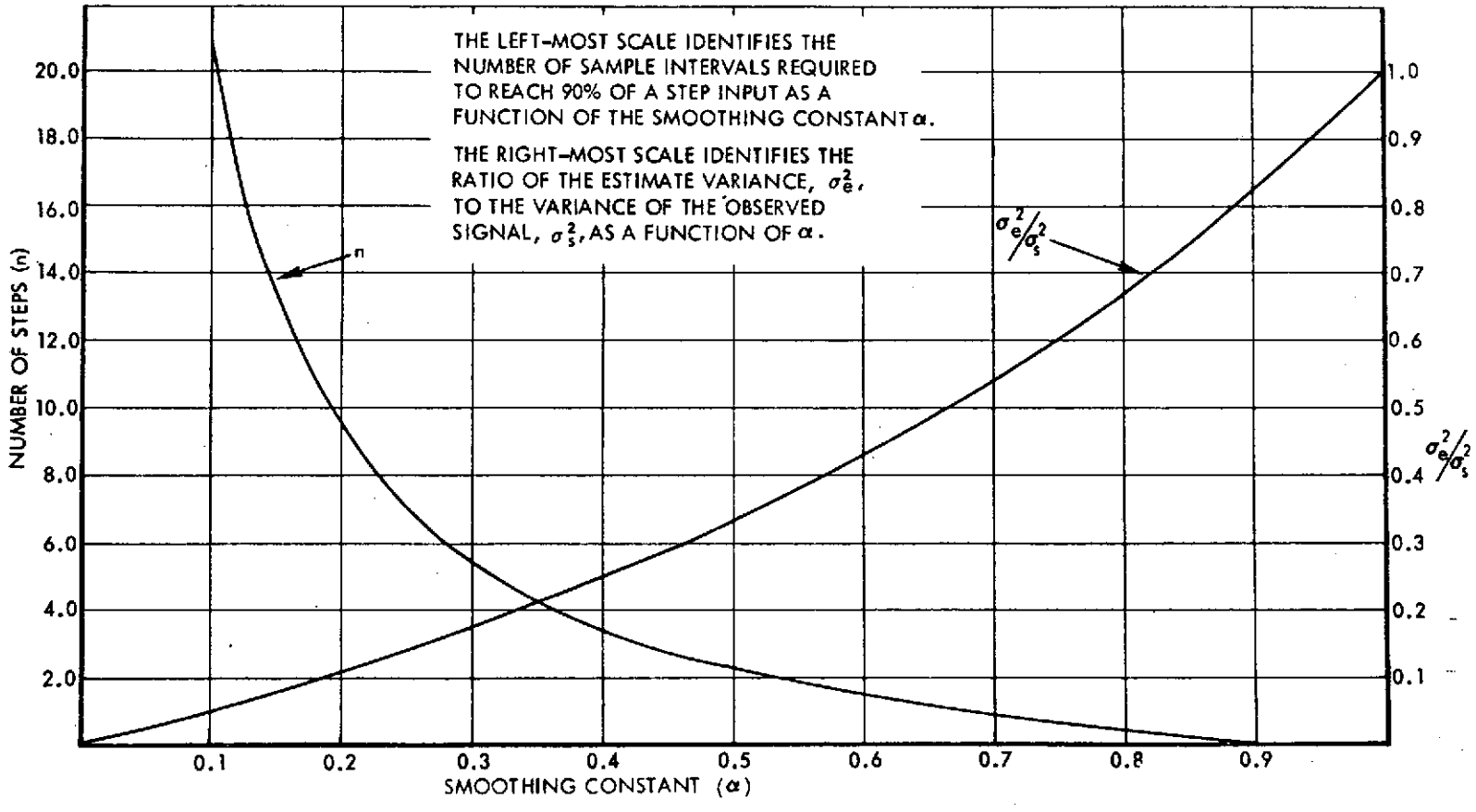


Figure 2.2.5.1. Step Response for a First Order Recursive Smoother

goes bad again. What results is a chain of conditional decisions which further complicates the error problem.

Added to the above considerations are those of the crew and the payload owner. Do they need all these decisions? Can they even use them? Is it even practical to attempt them? Section 2.1 suggests a no to the first two questions and following material will show that it is not practical, or at least not desirable, to attempt them.

Dismiss FDA for the moment and consider a process which determines directly the status of Orbiter functional paths or redundant legs. Using approximately 200 well chosen parameters, this process could determine such status using the same processes now existing in FDA. The "pump-up" decision smoothing technique is well suited for this task since principal interest would be with parameters which changed suddenly. And, if a value (indicative of how well the particular functional path was operating) stayed close to the limit for an extended period of time, any delay encountered in annunciating this condition would be tolerable. This comes about since the status indications would still represent alert conditions and it is not crucial to know exactly when a limit is exceeded.

The possibilities of having quorum, i.e., n of m decisions before annunciating, post-decision rules was given a very light treatment and should be pursued further. Based on this brief examination, however, it appeared that any advantages to be gained would be outweighed by requirements for more processor resources.

In contrast to C&W which is concerned with values of parameters per se, the above process is concerned with status of functional paths directly. The distinction seems trivial but the approach to the two processes is entirely different. The former is a parameter checker and the latter is a fault or failure checker. Volume II of this report is devoted to methods of achieving such failure checkers. Also, the Appendix to that volume contains an example worked for the Orbiter Hydraulics System.

What about the remaining 800 or so subsystem parameters? Neither the Functional Path Fault Detector nor C&W need results of limit checks on these parameters to accomplish their tasks. In addition, the functional path information already processed provides the crew with sufficient information at the minimum level for which they can take corrective action.* The remaining parameters are then

*There are some exceptions to this, e.g., OMS-RCS fuel cross-feeds. These cases should be treated as exceptions and solved individually.

of value in assisting the crew (or payload owner) to determine the extent of the functional path loss or to gain a more complete picture of Orbiter condition when the need arises. The crew does not need routine decisions on these parameters. In fact, such routine decisions can only aggravate the false alarm problem.

Pursuing the crew assessment case, out of limit parameters constitute only a portion of the information needed by the crew in determining the extent of a problem. Equally important is the behavior of a parameter and its relationship to other parameters. Is the parameter fluctuating? Is it remaining constant just at a red line value? Is it way out of bounds with similar or related parameters? It is impractical to ask a machine to do this extensive kind of analysis, especially when there is no way of knowing which question will be of interest at which time. Why, then, should the machine even check for out-limit conditions? It should not. The parameters and their limits should be displayed and the crew can make this assessment along with those discussed above. This kind of interpretation is done routinely by crews of conventional aircraft. They perform meter scans and watch for red line conditions. The Orbiter crew is not asked to do this routinely, only when they are troubleshooting. If a display is properly designed, this will be no burden. The function which provides these services is only a slight modification of the existing SMM.

If these parameters are not checked, what about LRU data on the Maintenance Recorder? This approach will definitely reduce the number of entries on the recorder. The issue is whether maintenance data has been jeopardized. If C&W and Functional Path Fault Detection drive the recorder control, all parameters (in the telemetry format) will be recorded for each major event. What will be missing are the interim behavioral characteristics of the LRU data. It is questionable whether these events add much more information to the maintenance picture, especially when one considers the cost and scope of a ground-based data analyzer required to take advantage of the information. In the final analysis, if it turns out that these limit check indications are of great value, it would not be difficult to incorporate a full-time resident checker for the purposes of triggering the maintenance recorder. The results of these checks should be transparent to the crew. They should not enter into the displayed data or alarm system. In addition, such a limit checker should use the predecision or recursive smoother described in Section 2.2.5.1

In summary, the above approach decreases the number of decisions to about one-fifth of those for the Baseline Design

and the most aggravating sources of potential false alarms have been removed. This will result in a more viable and cleaner design as well as a more meaningful output. As an additional bonus, it is not prudent to declare a functional path bad and some time later declare it good again if and when the performance measure for that path drifts back to "good." Such occurrences should be checked by the crew. In the fault detection system described above, functional paths can only be declared bad so there is no need (or desire) for chains of bad-good-bad... decisions. If the crew, upon checking the condition (on this small subset of parameters), believes the functional path good, they must have the capability of manually overriding the process decision.

2.2.5.3 Configuration Monitoring and Miss Errors

In all the process discussed thus far, replicated decisions and misses were of little concern. Subsystem Configuration Monitoring (SCM) in the Baseline Design is a one-shot process. It takes its samples when called and makes decisions based on single values of these samples. If the crew wishes to recheck these results, they call SCM again. Since SCM is an exception reporting device, misses are rather important. An improvement would be to have SCM, each time it is called, take, say 10, consecutive samples of each parameter and declare a disagreement if any of the ten do not agree with the check list. This is simply exploiting the post-decision technique used in the Baseline for FDA.

2.2.5.4 SM Self-Check for Disclaiming Factual Output

The preceding three sections discussed methods of reducing errors to improve credibility. This section will discuss how to improve credibility in the face of those errors which remain. Section 2.2.5 discussed the need for disclaiming factual statements if the statement is known not to be factual. It will then be necessary to incorporate an SM self-tester. Implementation of this tester is a topic for further study. The tester must have the following characteristics:

a. Be responsive to data quality monitoring from the DACBU, RAM, Mass Memory, Payload MDMs, CRT/keyboard and ICC to SM. This includes all transfers on buses as well as the I/O processor and CPU.

b. Be responsive to data timing and missing data/responses in the above.

c. Be responsive to I/O Processor and CPU hardware/software faults.

d. Report a suspect condition by hardwire to a panel, not through software channels which have already been declared suspect.

Features accomplishing much of this list already exist. They have but to be integrated and polished. The IBM CPU fault detection scheme will undoubtedly play a role but it should be recognized this scheme does not test the IOP which is at least as complex as the CPU. The CPU fault detection scheme is a good one but its direct application should be scrutinized.

The importance of SM self-test can only be appreciated when one compares the reliability (or unreliability depending on your frame of reference) of the IOP/CPU combination with that of the individual items being checked by SM. It is seen that the checker is, in many cases, no more reliable than the items it is checking.

2.2.6 Functional Assessment Summary

Five criteria were used to assess the Baseline Design. Results of the first four appear in Table 2.2.6 as scores or classifications. The fifth criterion, credibility, will be discussed in a later paragraph.

- Process Time Line Use

  The classifications for this criterion appear in the first column of Table 2.2.6. The code is:

  C - continuous or random use

  P - used at scheduled points in the mission

  S - used during scheduled periods of the mission

- Design Independence

  This criterion is actually scored as design dependence in the table. Processes with the highest score are the most sensitive to extra-SM design changes.

- Criticality

  In Table 2.2.6, the highest score represents those processes which are most critical to the mission and/or safety. Those affecting safety, notably C&W, were scored the highest. The FDA process was partitioned here to indicate the

Table 2.2.6 Baseline Design Scoring

Process \ Criterion	Time Line Use Code	Design Dependence	Criticality	Potential for Ground Implementation (3)
FDA	C	5	10/5 (1)	N
SMM	C	5	5	N
SCM	P	4	4	Y
CM	C	2	3 (2)	N
Recorder Mgt.	C	1	2	N
TLM Format Select	P	2	1	Y
P/L Support	C	10	4	N
Uplink Service	C	1	2	N
Downlist Control	C	1	2	N
Ground C/O Support	S	N/S	N/S	N/S

Notes:

1. Top quantity is for C&W, bottom for alert
2. This function can be accomplished other ways with additional effort.
3. Y - Yes
   N - No

N/S - Not Scored

contrast between C&W and alert conditions. The scores should be interpreted only in the context of SM functions.

- Potential for Ground Support

  This criterion was evaluated simply yes or no. Modifications to the Baseline partition will alter these results.

Four methods were identified for improving the credibility of SM results. These were:

- Treat C&W conditions separately from alert conditions and employ a different false alarm avoidance technique for C&W.

- Reduce the number of parameter decisions made by SM by:

  a. Employing a Functional Path Fault Detector.

  b. Make no decisions on SMM data.

- Force Configuration Monitoring to replicate list comparisons automatically and report disagreements for comparisons which don't agree for all replications.

- Incorporate a SM self-test to flag questionable output.

2.3 Relationship To The Original Concept — The Third Assessment

The purpose of this section is to relate the original SM performance specifications to the Baseline Definition. The reasons for this retrospection are simple. First, it revitalizes the views of the original Orbiter architects. Such views often tend to fade in the day-to-day efforts to "make it work." Second, it provides perspective for an objective view of a design milestone. Where does it stand with respect to achieving original intents and purposes? It is recognized that implementation may dictate departure from original concepts. It is also recognized that in systems such as SM which may be classified as user-oriented, users' needs seldom change from those initially identified in gross performance requirements. Once contrasts between the original concept and the current approach are identified, so will be the areas on which to concentrate constructive improvement recommendations.

The original concept is believed best represented by two sources. The first is Johnson Space Center document

0700, Vol. X, the applicable portions of which appear below:*

Performance Monitor

A performance monitor function shall be provided utilizing elements of the instrumentation, display and control, and data processing and software subsystems. This function shall provide to the flight crew information concerning health status, configuration status, and fault detection and isolation status for flight vehicle subsystems. This function shall also support redundancy management to the level required in flight; onboard fault detection, isolation and anomaly recording; management of orbiter data recording; and monitoring and management of certain other inflight functions. An interface shall be provided for use of the onboard capabilities in support of ground operations.

Payload Data Processing

The Orbiter shall have the capability to checkout, monitor, and command payloads. The Orbiter must be capable of performing this checkout, monitor, and command at all times after liftoff. A capability for payload monitoring shall be provided for all flight phases and ground operations. Payload caution and warning signals shall be displayed to the flight crew and at the mission specialist station. The capability shall be provided to display payload parameters in real-time to the mission specialist station.

Note, the reference to C&W signals should be interpreted as software back-up C&W signals.

The second source is the Rockwell International document SD 72-SH-0103-8, Vol. 5-8. The pertinent portions of this document have been reproduced in Appendix C to this volume.

Relating the Baseline Design in Appendix B to the original concept described by JSC 07700 and the Performance

*Source: JSC 0700 Vol. X, Rev. A., Space Shuttle Flight and Ground System Specification; Johnson Space Center, Jan 2, 1974.

Specification in Appendix C, reveals a rather good correlation. There are, however, three rather important differences which should be examined. Taking them in order of occurrence:

- JSC 0700 requires that SM support redundancy management to the level required in flight.
- The performance specification (Appendix C) states that the purpose of (FDA) is to detect subsystem failures at the functional path level (level at which corrective action can be taken in flight) and inform the crew that the failure has occurred.
- The performance specification also states that the purpose of (SMM) is to provide the crew with access to data from which the degree of a problem (detected by FDA) can be assessed.

SM currently does no redundancy management. One reason may simply be due to differences of opinion as to what constitutes redundancy management. A second reason is that subsystem failures are not detected at the functional path level. And, unless this is done, automated redundancy management can never be implemented. It may also be that redundancy management was not included on the grounds that automatic fault recovery for C&W as well as alert class conditions was neither necessary nor desirable. The exclusion of automatic fault recovery is a sound decision. The relationship of redundancy management to this decision, however, is remote. Redundancy management consists of those functions and/or processes which keep track of the operational status of the entire redundancy network and executes those algorithms necessary to determine what to do in the event of a failure in any element of that network. Or, stated another way, redundancy management for System Management would employ automatic fault detection to the functional path level, a table of the operational status of each functional path and a means of displaying this table. In addition, where recovery algorithms are complicated, i.e., what to do in the event of a failure is predictable but depends on a long list of the state of other functional paths, the algorithm solutions could also be displayed to the crew. This is a viable description of redundancy management for SM. Note that automatic switching is not considered. Automatic switching is a recovery technique and the purpose of redundancy management is to tell recovery what to do — be it manual or automatic.

Detecting subsystem failures to the functional path level is the next subject in the list of differences. As mentioned above, the current approach does not actually perform this task. Instead, an attempt is made to identify

each parameter displayed on the fault summary with a functional path. The crew can then infer a fault in that path. Unfortunately, a very large list of parameters is being checked and it is not always possible to conveniently accomplish this association. Also, not every parameter which falls out of tolerance is indicative of a functional path failure. This distinction may appear subtle but the fact remains that the crew, not SM, decides whether a fault exists in a functional path. When considering the large number of parameters tested under the current concept, this approach is probably the only reasonable one. If the original concept is to be met, it appears that fewer (much fewer) parameters should enter into the decision. This notion is pursued in Section 2.5.

The third difference identified above is closely related to the first two. The original concept cast SMM in the role of providing information from which the degree of an FDA-annunciated fault could be ascertained. This implies a second level of information which can be used at the crew's discretion. If the crew either desires or needs to know "how bad the fault is," they may consult SMM. Under the current concept, SMM is not a second level of information. Rather, it must actually be used on a routine basis as part of the man-machine process of identifying faults to functional paths. The reasons for the current concept are quite likely similar to those mentioned earlier for other differences. And, under the current concept, this approach is not without justification. It is contended, however, that until SMM can be truly relegated to a secondary role, the value of SM (or at least FDA) will be diminished.

2.4 Applications Assessment Summary

Three different assessments have been made on the Baseline Design and a lot of material has been covered. What can be concluded? Each assessment used a different frame of reference and evaluated a different aspect of the design. Do they reveal similar findings? Are there portions of the Baseline which could be improved? Is the design suitable for its application? Beginning with the most chronic conditions, each assessment found problems with two processes (and their outputs). These processes were FDA and SMM. The theme which developed across the three assessments can be summarized as:

- Treat the C&W process, as well as its display, separately.
- Use a different false alarm avoidance technique for C&W.
- Do functional path fault detection and display status of functional paths.

- Use SMM expressly as an aid to determining extent of problems.

- Reduce the number of parameter decisions made by SMM.

- Provide redundancy information.

It is worthwhile to use some 20-20 hindsight and determine possibilities of why these problems exist and where they may have originated. Considering them collectively, the most prominent cause was lack of sufficient user considerations. The Baseline seems to have been developed without a thorough understanding or appreciation of how SM was to be used. While the user assessments given in the report (especially the scenarios of Section 2.1.1) are believed sound, they should be reviewed by JSC flight specialists. It is recommended that a similar user assessment be made by JSC.

Where did the problems originate? It seems clear from the third assessment (comparison to original concept) that the JSC document 0700, Vol. X and the Rockwell International RDD, SD 72-SH-0103-8 both considered the user. Furthermore, it is not within the scope of these documents to detail how the user is to be considered. The RDD, however, continues on to specify functions well below the recognized level for a design definition. The problems apparently originated in the preparation of the RDD and subsequent specifications, viz., in the definition of the baseline. The cause of the omissions does not, then, point directly to the Baseline Design. This design was simply done prematurely. A level of specification was omitted between the JSC specification and the Baseline which should have detailed how the intent of the original concept was to be carried out. This specification should have (and undoubtedly would have) considered the use of SM. This level of specification routinely includes an operations analysis and is the role which should have been fulfilled by the RDD.

Numerous other areas, not to mention the specifics behind the problems identified above, were indentified as improvement candidates. Rather than simply provide a list of deficiencies, a more positive approach is to propose something which overcomes these deficiencies. This is the purpose of Section 2.5 below where an alternative SM design is described. Methods for overcoming the cited problems have already been identified in the applications assessments by indicating what the approaches should be. The alternative design, likewise, should be no surprise. If the reader will pardon a lighter note, the alternative design turns all the "should be's" into "are's."

2.5 An Alternative Design

This section describes another approach to SM design based on the assessments in the preceding sections. It is not claimed that this approach is the approach nor is it claimed that it solves all the problems. It is claimed to be both feasible and reasonable. And, this is its intent. The design level is the same as that of the assessments. That is, applications are functionally oriented. Implementation considerations are addressed in Section 3.0.

The design functional flow diagram is shown in Figure 2.5.* Section 2.5.1 discusses the design overview from this diagram. Selected processes in the design will be discussed individually in Section 2.5.2. These discussions contain more detailed block diagrams of the processes as well as display formats. Neither the block diagrams nor the display formats are intended to dictate an implementation. Their purpose is to provide continuity to the functional flow and highlight operations as well as information presentation techniques. It should also be recognized that implementation will require the introduction of tables and communication not specifically identified in the block diagrams. For example, it will be very likely that all values will require parameter ID tags, either implicitly or explicitly. Data control and precondition tables will likely be required in addition to data base maintenance and initialization.

2.5.1 Design Overview

Figure 2.5 identifies 18 SM processes. Some of these processes may not be contained within SM by the time the Operational Version is implemented. Payload Commanding is a good example of this as it could logically be grouped with several other payload operations under a single software process which is designated payload peculiar. Whether these processes are actually a part of SM or not is hardly an issue. The important point is that they have been identified as necessary processes and the partition will allow them to be placed wherever is the most appropriate.

This overview is divided into three areas: (a) administrative processes, (b) design structure and operation and (c), overlay structure which are discussed in Sections 2.5.1.1, 2.5.1.2, and 2.5.1.3 respectively.

2.5.1.1 Administrative Processes

Of the 18 identified SM processes, five qualify as administrative. These are Keyboard and Display service, Local

*For the convenience of the reader, this figure appears as a foldout at the end of Section 2.5.

Exec/Control, Uplink Services and Data Acquisition. Whether the first four even exist depends on the finalized FCOS interface. All are undeniably operating system functions. They have been placed on the block diagram in the belief that it is better to identify a process and have it be absorbed elsewhere than to find out it is required at software integration. In this way, the processes can be defined and a verification made that FCOS does indeed meet the requirement.

The Data Acquisition process is a data routing, buffering and fetching operation. Constant streams of data are entering SM and being routed to the various processes. Some processes are resident and some are overlayed. In addition, as indicated in Section 2.2.2, certain of these processes are sensitive to Orbiter subsystem changes. A central data acquisition process will serve as a design buffer and allow the individual SM processes to continually fetch data from a central location with implied identification. When data requirements change, Data Acquisition can be instructed by the other processes. Most importantly, Data Acquisition should be a partition since it is clearly a foreground task.

2.5.1.2 Design Structure and Operation

In contrast to the Baseline, a new process, Functional Path Fault Detection (FPFD), has been defined. In addition, two new tables, along with their management function, have been defined. These are Functional Path Status and Redundancy Management. The Back Up C&W process has been made an independent operation to include its own precondition steering. SMM does not interface with FPFD, the old FDA process having been eliminated. SMM does no limit checking. Consumables Management (CM) has been partitioned into two processes and provides data to Subsystem Configuration Monitoring (SCM). The Redundancy Management process provides an input to Configuration Monitoring. An SM Performance Monitor has been added as well as a Payload Services process. This latter process provides for fixed SM design with a single process which is payload peculiar. The remaining processes are unchanged.

The processes of SMM, FPFD, CM, SCM and C&W have been made independent from data input to display. It is entirely possible that common elements can be defined for these processes in implementation, thus increasing processing efficiency. Precondition steering is a likely candidate. Whether common processing can be defined and implemented depends on the final overlay structure and detailed level process specifications. The point here is that, until each element of the independent process is completely specified, no attempt should be made at combining operations. The processes were made independent for

this reason. Design work-arounds to realize commonality are rarely a good substitute for understanding each problem and then deciding if commonality exists. More will be said on this subject in Section 2.5.2.

FPFD operates on a small subset of functional path data. Its purpose is to make status decisions regarding these functional paths and to drive the Status Table. The process has its own set of precondition steering logic. The Status Table is a permanent record of the state of each functional path in the vehicle (and payload as required). These paths include those in the GN&C subsystems. This table in turn drives a Redundancy Management table whose purpose is to keep track of what capabilities are lost when a functional path fails.

FPFD cannot declare a failed functional path good again. The system is initialized with all paths good. Unless manually overridden, the paths, once detected as having failed, will remain in the failed state in the status table.

Any non-GN&C entry in the Status Table can be manually changed. A functional path which is considered operative by FPFD can be manually changed to the down state. Likewise, a path indicated as having failed can be manually declared UP. This can be accomplished in two ways. First, the path can be declared UP with an override which excludes FPFD from again declaring it DOWN. Second, it can simply be declared UP; in which case FPFD can again declare it DOWN. Since GN&C paths are reconfigured automatically, manual changes to these paths are locked out.

Functional paths which are not being tested at the moment, e.g., de-energized equipment, are assumed operative. The Status Table assumes a path innocent until proven guilty.

The Redundancy Management table feeds SCM for the purpose of checking a minimum equipment/redundancy complement before continuing to a subsequent mission phase.

FPFD is capable of evaluating the functional path status using two standards, viz., nominal operation and emergency operation. This is accomplished by changing the limit values used in the limit checker. The change is initiated manually.

SMM performs no limit checks. It accepts all vehicle and payload data (via Payload Service), scales and/

or performs engineering unit conversions and displays the data. Data to be displayed is manually selected. To aid in data interpretation, a microfilm viewer (not part of SM) is included as an ancillary item. This viewer could be installed at the Mission Specialist Station. Applicable view page numbers are included on SMM and SCM displays.

2.5.1.3 Overlay Structure

The identified processes are considered to be either core resident or overlay. With regard to the latter, "overlayable" might be a better term. It is not intended that these processes be decreed as overlay, only that if overlay is exploited and solved now, integration of SM secondary functions (see Appendix C) will move more smoothly. The following processes must be resident.

Data Acquisition
FPFD
B/U C&W
SMM
Funct. Path Status Mgt.
SM Performance Mon.
Redundancy Mgt.
Recorder Control
Downlist Control
Payload Service
Uplink Service
Calculations for CM

The following processes are overlay:

SMM
SCM
Displays of CM
TLM Format Select

2.5.2 Selected Processes

This section provides additional detail on those processes which were most affected by the alternate design. Some of the operations appearing in the process flow diagrams may not be required for some implementations. The blocks have been identified for completeness. No CRT displays are forced. Each must be called from the keyboard.

2.5.2.1 Functional Path Fault Detection

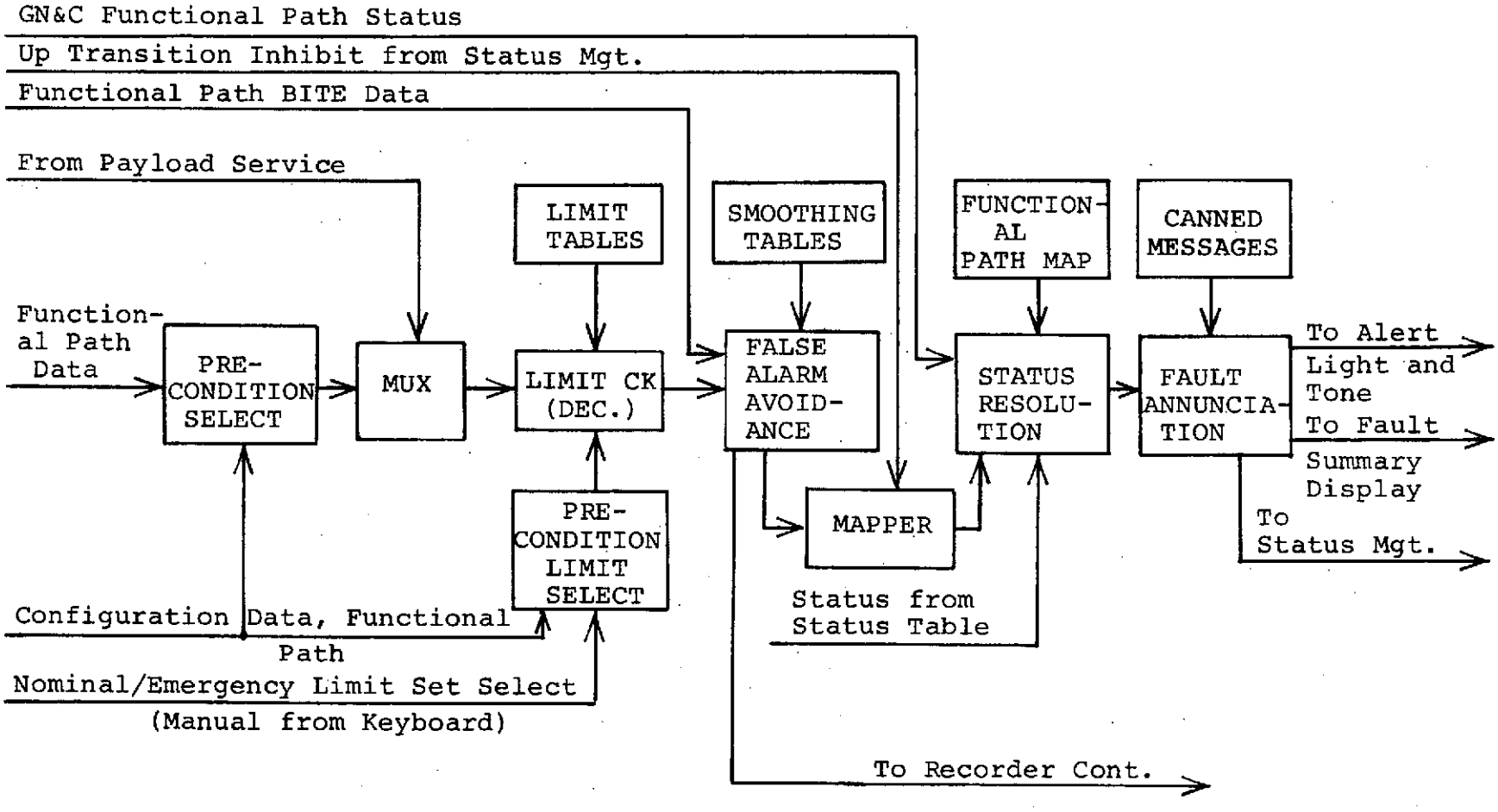
A detailed flow of this process is shown in Figure 2.5.2.1. Starting from the input, functional path data are received by Precondition Select from Data Acquisition. Precondition Select, as in the Baseline, determines whether the specific data are to be processed based on the functional path configuration. For example, if a particular functional path is de-energized, that path would not be examined. Alternatively, if a path were energized but in a "keep-alive" state, e.g., hydraulics in the on-orbit phase, a different parameter(s) may be examined. This functional description of Precondition Select might be modified in the final application. It is likely that Precondition Select would direct Data Acquisition as to the data it actually needs.

There is a counterpart preconditioning function. This is Precondition Limit Select. Its function is similar to that above except that it controls the limits used on the functional path status parameters as determined by system configuration. The second input to Precondition Limit Select is the manual Nominal/Emergency limit set select. Due to the significant reduction in the number of parameters to be checked, it is expected that both precondition functions will be simplified from those of the Baseline. This is due not only to a simple quantity reduction but also to a sizeable reduction in configuration/parameter interactions which would have to have been accounted for.

Since functional path faults are being detected, it is entirely possible that another function will have to be performed by Precondition Select. This function is special operations or computations. For example, it may be determined that the operation of a particular functional path can best be characterized by the sum of two of its parameters. Or, that the peak value of a parameter from another path is best representative of its operation. Such operations (here summing and peak detecting) would be a part of Precondition Select. If such operations become extensive, a block following Precondition Select should be included.

The next block in the sequence is the multiplexer which handles all functional path data from Payload Services. Payload Services accomplishes its own preconditioning functions. In operation FPFD would cycle through its vehicle data and, to maintain design independence, would enter a Payload cycle. On entering this cycle, Payload data would be multiplexed in from Payload Service. The data would be entirely determined by Payload Service.

Limit checking is the next block in Figure 2.5.2.1. The function of this block is self-explanatory. The limits can be one-sided or double-sided (high and low). They should



THIS PROCESS IS FULL TIME RESIDENT

Figure 2.5.2.1. Functional Path Fault Detection

never, however, be selected to describe more than two states of a path. Stated another way, limits should be selected which indicate the bounds between acceptable and unacceptable operation, not, acceptable-marginal-unacceptable. The Nominal/Emergency limit sets have already been discussed. It is worthy to note that this selection can also be used by the crew to determine how bad a functional path is. The Limit Tables simply contain the set of all limits used throughout the mission, both Nominal and Emergency. These values are not crew changeable.

There are two inputs to False Alarm Avoidance. One is the subsystem BITE indications. Depending on the character of these indications, some may be routed to Precondition Select for limit checking. The second input to False Alarm Avoidance is the results of the limit checks. False Alarm Avoidance or smoothing operates exactly as it did in the Baseline FDA, i.e., output after "n" consecutive out-limit indications and not output again until "n" consecutive in-limit indicators followed by "n" consecutive out-limit indications. Recall from Section 2.5.1 above that FPFD does not declare a path good again once it has been declared bad. Therefore, FPFD does not need this complicated smoother. The only reason the multiple transitions exist are to drive the Recorder Control each time the path is declared DOWN. Since C&W parameters are the only others driving Recorder Control, this step seemed advisable. The final design will depend on how much data is needed on the recorder. Furthermore, since the recorder is the only reason for the multiple transitions, the algorithm should be re-examined for possible simplification.

The next block in Figure 2.5.2.1 is a Mapper. This block serves two functions. First, as the name implies, this operation maps or combines the results of False Alarm Avoidance into the status of functional paths. For example, suppose the integrity of a functional path was to be determined by three independent parameters. Suppose further that if at least one of these parameters were to be declared bad by False Alarm Avoidance, the path would be declared down. The Mapper would perform the implied logical OR in this example.

The second function of the Mapper is to inhibit DOWN-to-UP status changes that may be reflected from False Alarm Avoidance driving the recorder. The Mapper receives functional path states from the Status Table to effect this control. In this way a manual change in the Status Table will be reflected directly back for re-evaluation. There is a counterpart to this up-inhibit control. This is the mode of manual Status Table change which involves a "good" state override. This will be discussed in the following paragraphs.

Status Resolution is the next operation of concern. The function of this block could be thought of as post-conditioning. To understand the need for this operation, consider the following situation. Several on-line functional paths are functionally in series. That is, each receives outputs from the immediately preceding path. Now suppose the first functional path fails for some reason. The next path in the chain could well receive an erroneous input (or no input). This next path, given its erroneous input could also be declared bad. And, in principle, this condition could ripple down the chain. The only path which failed was the first and this is the information the crew needs, not the other false indications. The purpose of Status Resolution, as the name implies, is to resolve such situations. To do this, the function will require a Functional Path Map showing how the paths are connected. It can then reason, based on information flow, the most "up stream" path and hold the down decisions on subsequent paths until corrective action is taken on the up stream path. The problem is not a simple one but it is not as bad as it first appears. Proper placement of BITE will aid in this decision process. Furthermore, there appear to be very few long chains of functional paths on the Orbiter. For strings of just two paths, an easy way out of this problem is to ignore Status Resolution and, should the problem arise, declare both down. The crew can then switch one or the other to see which corrects the problem. Under such conditions, both paths would have to be manually declared UP after the switching action.

There are three other inputs to Status Resolution. The first comes from the Status Table and consists of a manually declared UP condition which is also instructed to be an override. When an override UP is manually entered into the status table, FPFD cannot declare this path DOWN. As such, this entry will be the only one read by Status Resolution. This condition is contrasted to a conventional UP manual entry which will allow FPFD to again declare the path DOWN if and when this occurs.

Another input to Status Resolution is the identity of the functional path which is on line. This input is not shown in the flow diagram but is derived from Redundancy Management.

The final input to Status Resolution is the status of the GN&C subsystems. Manual changes to these indications are not allowed.

The output of Status Resolution is the resolved or unconditional status of each functional path. When a path is declared DOWN, Fault Annunciation accomplishes the distribution and driving necessary to get this condition to the

several destinations. The first destination is the Status Table wherein a state change is made. The second source is the Functional Path Alert Light and tone. The tone and light are self resetting.

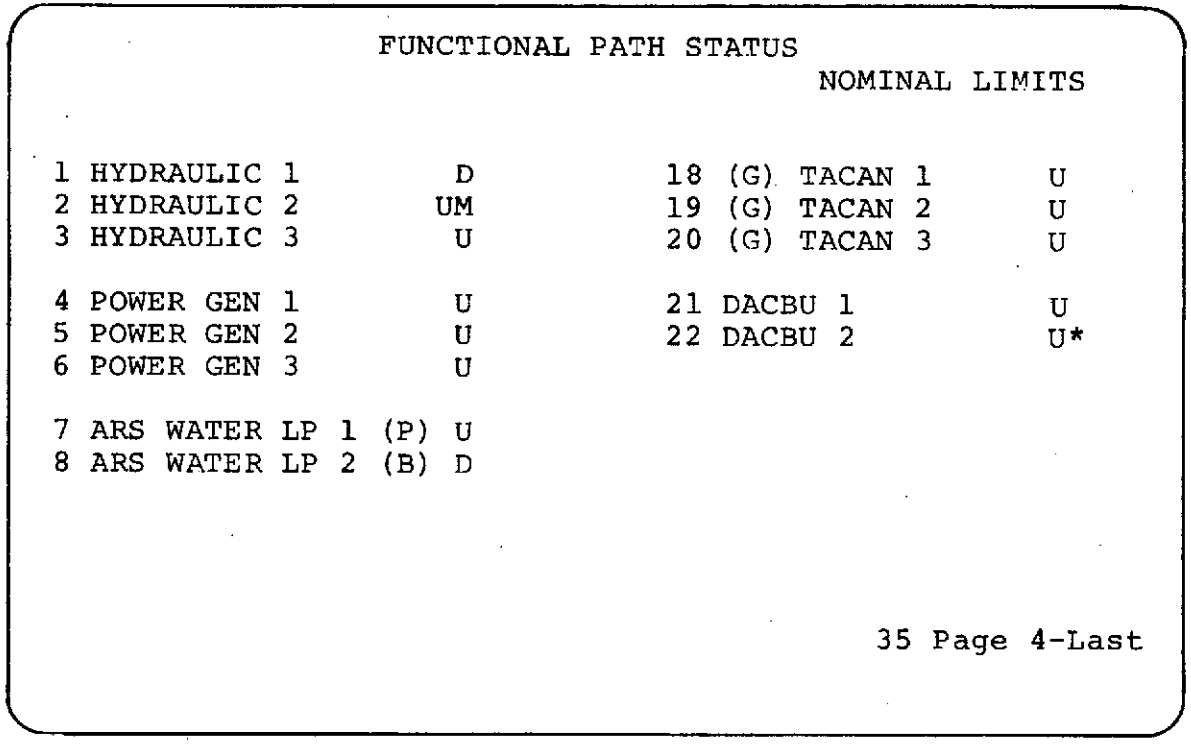
The final destination of Fault Annunciation is a Fault Summary Display Buffer. Fault Annunciation generates a display message which identifies the failed path and places it in the push-down buffer. When the Fault Summary Display is called by keyboard, the contents of this buffer are displayed in order of failure detection time. The purpose of the Fault Summary Display is to cue the crew about recent status changes. The Status Display discussed in the next section provides a complete status picture of the vehicle. Once the crew is apprised of recent status changes (which should be few in number) they have no further need for the display. Once the Fault Summary Display is recalled from the CRT, the Display Buffer is dumped (except for conditions which may have taken place during the recall time) to begin filling again. While the display is on the CRT, additions to the buffer will be automatically displayed.

2.5.2.2 Status Table and Redundancy Management

A possible format for the Status Display is shown in Figure 2.5.2.2-1. The purpose of Functional Path Status Management is to generate/service this display as well as to manage the Status Table. The display contains the names of each functional path and their status. Display contents always remain in the same locations. The status indications consist of four designators: U, D, UM and U*. These are identified in the illustration. The legend NOMINAL LIMITS at the top right indicates these status results are based on the Nominal Limit set. It is advisable to retain the old Nominal Limit status table in storage when Emergency Limit indications are desired. This will give the crew the capability to switch back for comparison. Only one table should be active at any one time.

The (P) and (B) on the ARS WATER LOOP legends indicate Primary and Back up. (G) indicates these paths are under GN&C control and their status cannot be manually changed.

Figure 2.5.2.2-1 also indicates a possible mechanism for entering changes and for paging. To change the status of a functional path, the line number is keyed in first followed by the change. For example, to change the status of ARS WATER LOOP 2 from DOWN to UP with the manual override option, the crew types in 8 UM. The display will then show this change which was entered into the Status Table. Paging is accomplished the same way using the line number, in this example, of 35. Note that page 4 is indicated as the last page so the crew will not attempt a page-forward.



LEGEND: U - Path UP as declared by FPFD
UM - Path UP as manually declared with override
D - Path DOWN as declared by FPFD
U* - Path not checked since launch, assumed UP
(G) - Designates GN&C equipment, manual status change locked out.

Figure 2.5.2.2-1. Status Display

The purpose of Redundancy Management is best seen by the display sample in Figure 2.5.2.2-2. Here, the remaining vehicle (and payload as necessary) redundancy is indicated by functional path along with the consequences of a lost functional path or paths. The remaining redundancy is simply a quantity, e.g., 2 of 3, if failure of any functional path in that set results in identical consequences or identical loss of capability. Otherwise, the remaining functional paths are individually identified. In the display example Hydraulic System number 1 has failed. This results in loss of nose wheel steering and landing gear actuators. To aid in crew scan, display entries always appear in the same location. This display cannot be manually changed. Display updates are automatically made to an active display. The display is called by keyboard.

In the sample display, the asterisk by the DACBU entry has the same meaning it did on the Status Display, i.e., at least one of the items has not been checked since launch and is assumed operative.

The purpose of Redundancy Management is to maintain the Orbiter Redundancy Tables, store the effects of functional path loss, generate the Redundancy Remaining Display and keep track of which functional paths are on line. This latter item should also be included in the redundancy display.

As a side note, if a convenient implementation can be devised, it would be beneficial to alert the crew through FPFD whenever redundancy has degraded to a single functional path.

2.5.2.3 Back Up C&W

The Back Up C&W process is shown in Figure 2.5.2.3-1. The precondition functions, limit check and Mapper perform the same as those for FPFD in Section 2.5.2.1. The Smoothing operation is entirely different from FPFD. Note it occurs before the Limit Checking. Before addressing the details of Smoothing, however, the overall operation should be reviewed. An out-limit condition detected by C&W will cause a start signal to be sent to Recorder Control. This condition will also be annunciated and an entry made in the C&W Out-Limit Table. This entry will remain in the Out-Limit Table so long as the condition exists. If the condition becomes back in limit, for whatever reason, that entry is removed from the table. The table contents may be viewed at any time using the Back Up C&W display. The C&W display is called by a single, dedicated key on the keyboard. The display is described in more detail below.

The smoothing operation used here is a first-order recursive smoother. Limit decisions are made on these smoothed results. Grossly speaking, the recursive filter

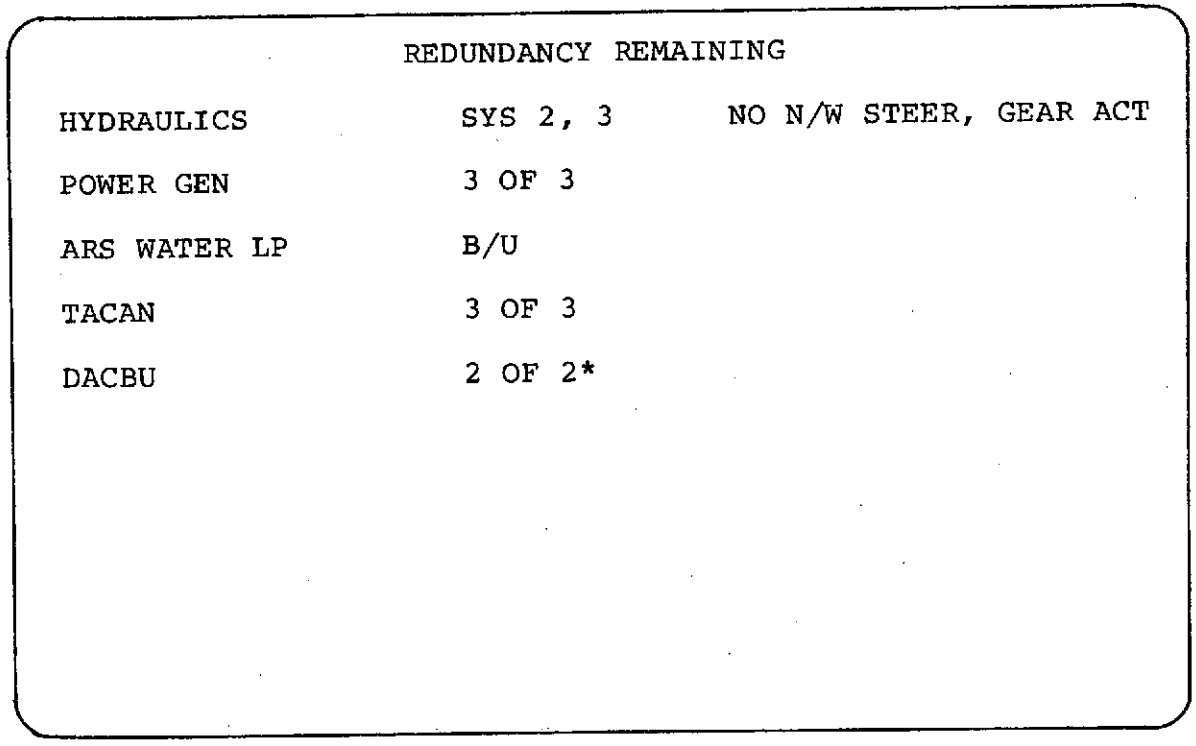
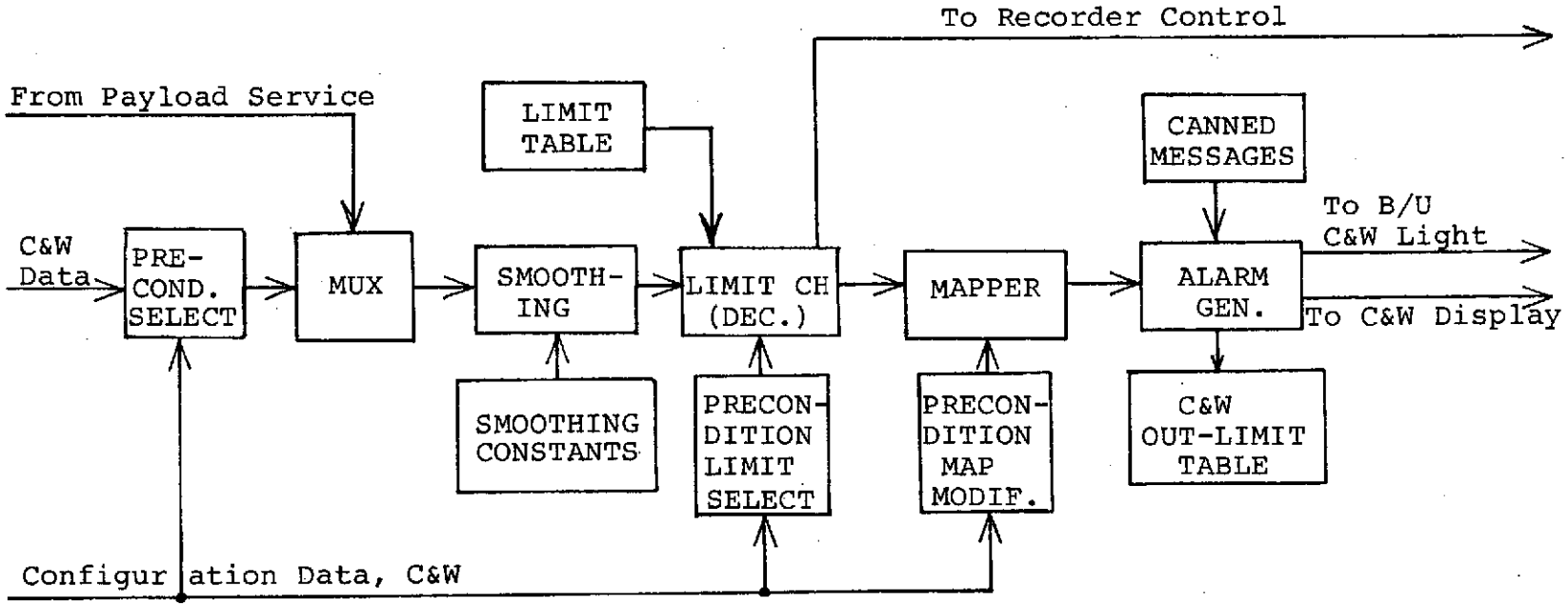


Figure 2.5.2.2-2. Redundancy Management Display



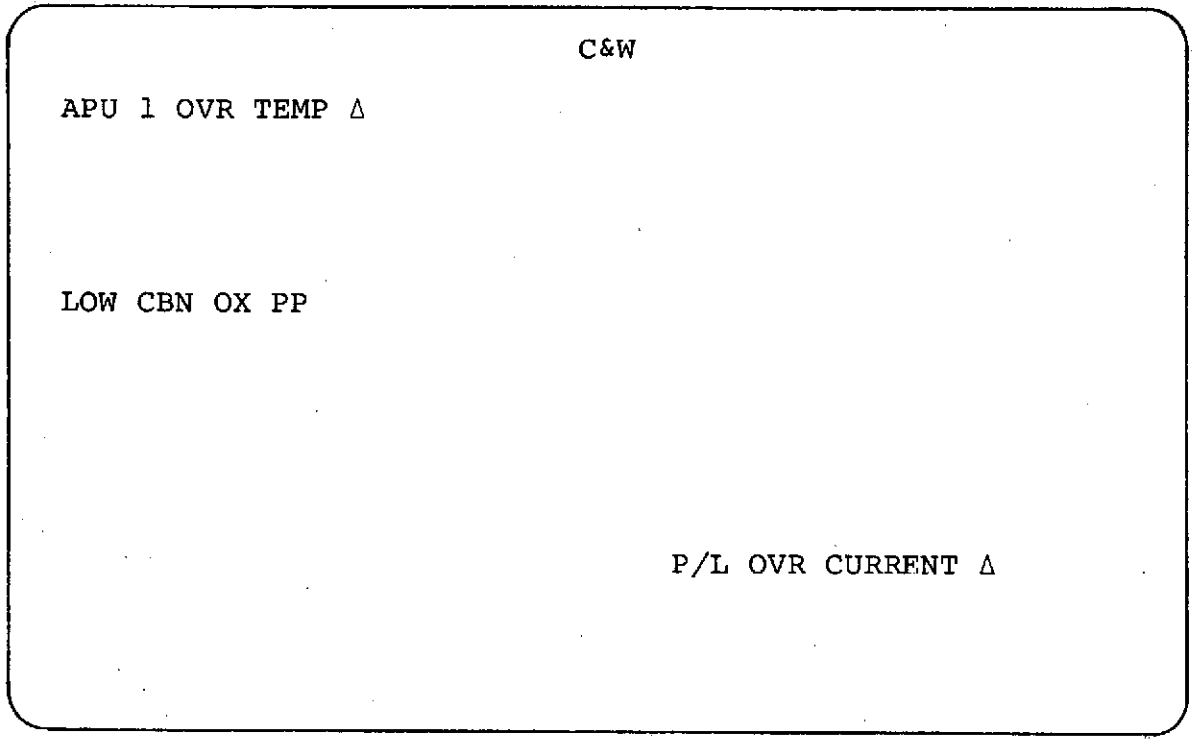
THIS PROCESS IS FULL TIME RESIDENT

Figure 2.5.2.3-1. Backup C&W Process

behaves very much like a single-pole low pass analog filter. It weighs each new sample by the smoothing constant and sums this result with the weighted value of all previous samples. It can be seen that this mechanization is not equivalent to a moving window averaging device commonly seen in these applications. In fact, it requires less storage than the moving window or running average technique. Since the smoother is essentially a low pass filter, it can be used to decrease both false alarms and repeated alarms with no increase in complexity. As with the post-decision smoother used in FPFD, this smoother will require initialization. Initialization can be effected by two extremes. First, nominal values of each parameter can be used for initial values to start the recursion. This requires a complete complement of nominal values be stored but results in a very short transient response — usually one step. The second method is to ignore initial values and let the smoother reach a steady state. This does not require storage but could take anywhere from 10 to 80 steps to become stable. The number of steps depends on the value of the smoothing constant, the initial value and the range of the check limits.

A good compromise to the initialization extremes is to initialize the smoother such that it is always "saturated" in the "good" region. For C&W this could be done with probably less than a dozen values. For example, consider a parameter which has an upper limit of +8000 on the PCM scale and a nominal value of +3000 (no lower limit). Initializing the smoother for this parameter at zero will certainly decrease the possibility of initial transients triggering a false alarm. In addition, if the subsystem from which this parameter originated had just been turned on, this technique would allow time for the hardware to settle.

A Back-UP C&W display is shown in Figure 2.5.2.3-2. This display is composed from the C&W Out-Limit Table and driven by the Back Up C&W Alarm Generator. Both the table and the display are fixed location, i.e., entries are not moved or pushed down. This allows the crew to become accustomed to a location for a given condition. The display consists of one page, double column and it cannot be manually changed. Operationally, when an out-limit condition exists, that condition, or group of conditions, is/are entered into the Out-Limit Table and the B/U C&W light is illuminated. The crew will then call the B/U C&W Display by its dedicated key. The display will show all conditions which are out-limit, including some from past alerts. The current conditions will be indicated by a "bug" adjacent to the entry. When the crew depresses ACKNOWLEDGE, the bug is removed. The display will reflect condition changes from the table so long as it is active.



Δ - Indicates condition has not been acknowledged

Figure 2.5.2.3-2. Back Up C&W Display

2.5.2.4 Subsystem Measurement Management (SMM)

The SMM process is shown in Figure 2.5.2.4-1. The process operates in two basic modes: Scan and Evaluate. The operation of these processes will become obvious in the discussion of SMM displays below. SMM is in effect no more than a mechanism for displaying Orbiter data. The process has access to all vehicle (and payload as applicable) data. In each of the above operating modes, the crew can call standard pages or identify specific parameters to be displayed. Achieving this option is the purpose of the first third of the process.

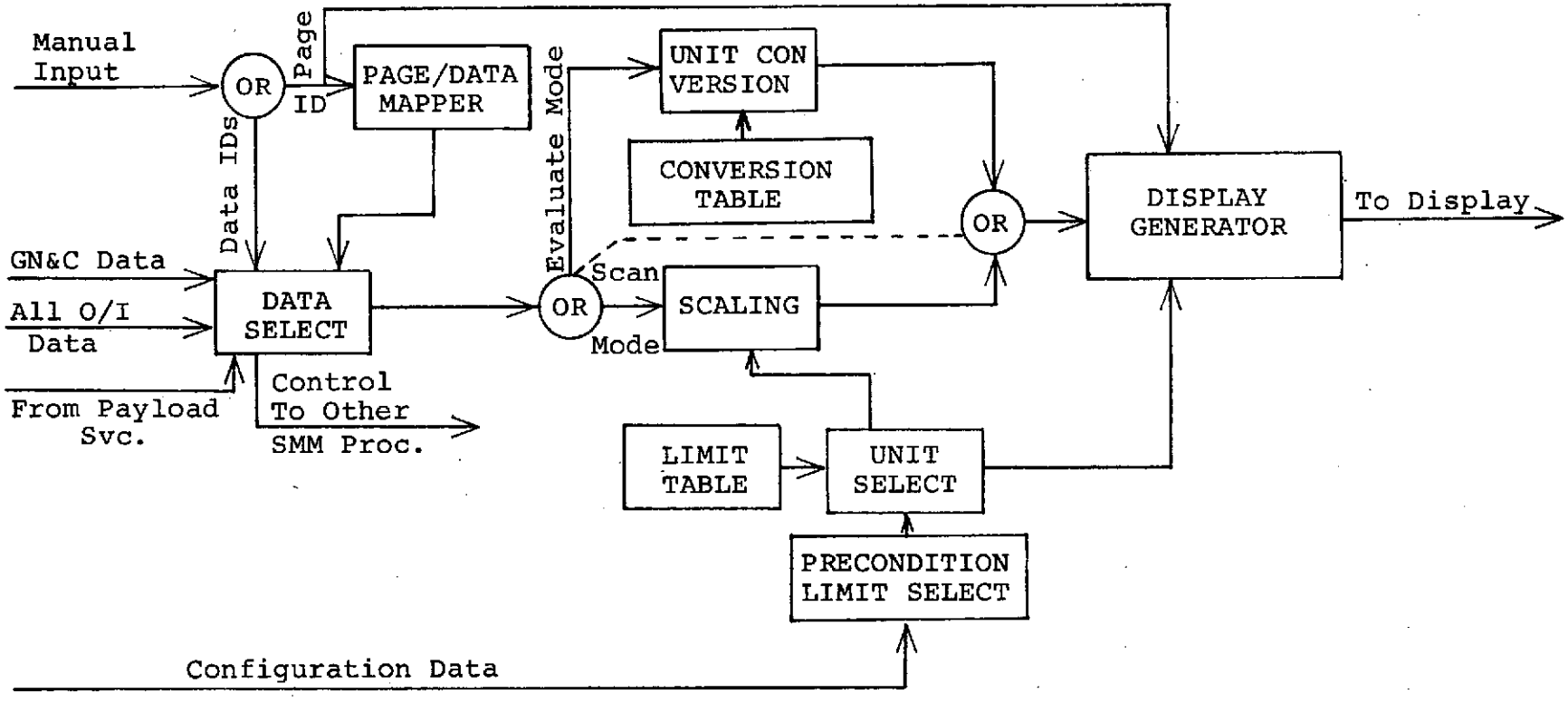
The SMM displays for each operating mode are shown in Figure 2.5.2.4-2. The display values cannot be manually changed. Each display is manually called from the keyboard. The displays are automatically updated so long as they are active. In the Scan Mode, a bar display is used. If any bar falls below the left hash line or above the right hash line, that parameter is out of nominal limits. The crew need only scan the bars to detect this condition and can rapidly page through a large list. The bars not only indicate out-limit conditions but also the extent of this condition. Marginal in-limit parameters may also be detected.

Note that the bars start at three different places. This is to accommodate the three versions of parameter limit conditions: upper limit only, lower limit only and two-sided limits. Upper limit parameters are scaled such that the left hash line represents the minimum PCM count. They can, therefore, never go below this line. Lower limit parameters are scaled such that the maximum PCM count corresponds to the right hash line. Two-sided parameters are scaled between these two extremes. In the case of the latter two, two different origins are defined to the left of the lower hash line.

The scan display contains (at bottom) the microfilm viewer page number where this subsystem block diagram is found. Note that discretes are not shown. They appear on the Evaluate Display shown in Figure 2.5.2.4-2(b).

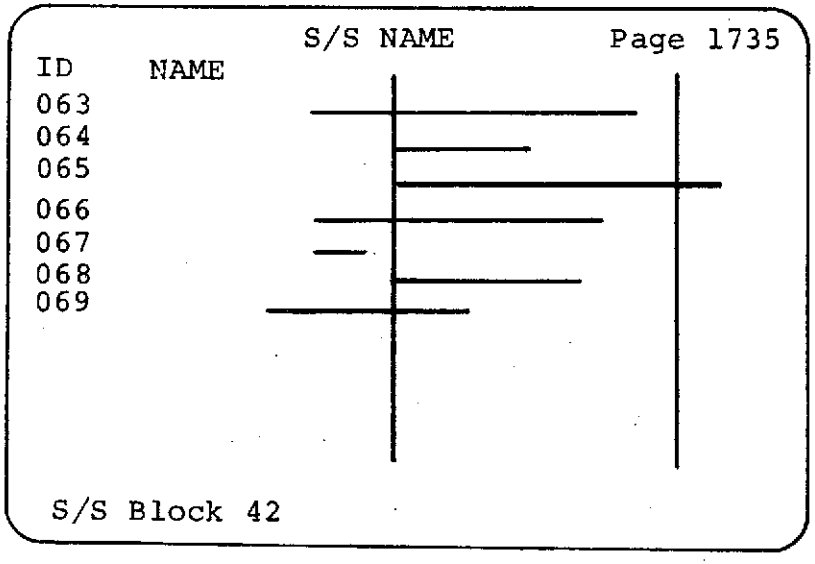
When the corresponding Evaluate Display is called, it will contain quantitative information in engineering units about parameters which were in the Scan Display. In addition, it will contain the state of all applicable switches, valves, heaters, etc. (the discretes).

The preceding described the page operation in both SMM modes. For purposes of correlating parameters on different displays, the crew may define a display in either mode by entering the desired mode and the parameter ID numbers recorded from the page operation.



THIS PROCESS IS OVERLAYED

Figure 2.5.2.4-1. Subsystem Measurement Management



(a) Scan Mode

Parameters 065 and 067 are out of limits.
Parameters 063, 067 and 069 have lower and upper limits.
Parameters 064, 065 and 068 have only upper limits.
Parameter 069 has only a lower limit.

S/S NAME Page 173E

ID	NAME	VALUE	LIMITS LO	HI	Units
065		685	-	600	PSIG
067		18	25	40	GPM
070		OPN			
071		ON			
073		OFF			
074		ON			
078		CLSD			

(b) Evaluate Mode

Figure 2.5.2.4-2. SMM Displays

2.5.2.5 Consumables Management

The flow diagram for this process is shown in Figure 2.5.2.5. The primary purpose for this diagram is to indicate the overlay option. The process of calculating consumables, e.g., PVT computations and table update, must be resident and operate in real- or near-real-time. Consumables consumption records must be available for the consumables displays and current consumables levels should be available to Configuration Monitoring. The rate at which the consumables level is updated is considered to be a programmable entry. This value could vary from once every five minutes to once every two hours. A call for Configuration Monitoring should initiate an immediate calculation.

2.5.2.6 Subsystem Configuration Monitoring (SCM)

This process is essentially the same as that of the Baseline, only its content has been increased. This is best described by the Configuration Monitoring Exceptions Display which is shown in Figure 2.5.2.6. This display lists all exceptions encountered in a check against a predefined checklist. The display is called, which in turn calls SCM, and the checklist ID is entered. This initiates the configuration check. In implementation, a READY indication on the display would be advisable when the process is ready for the checklist ID. The sample display identifies the ID of the configuration checklist being used as well as the exceptions. Beside each exception is listed the CRT page number for the applicable SMM display as well as the corresponding microfilm view page number on which the block diagram can be found. Note that the exceptions contain not only switch/valve positions but also consumables and redundancy information.

The second entry in the display has a question mark entered to the right of the condition. To explain this notation it will be necessary to recall that SCM will replicate its checks, say, 10 times. Of the displayed exceptions, all checked as disagreements all ten times except for OMS PORT SIDE OXIDIZER. This value disagreed at least once but not all ten times. The condition should be checked by the crew using the page index to the right of the display. Alternatively, they can reinitiate the SCM check.

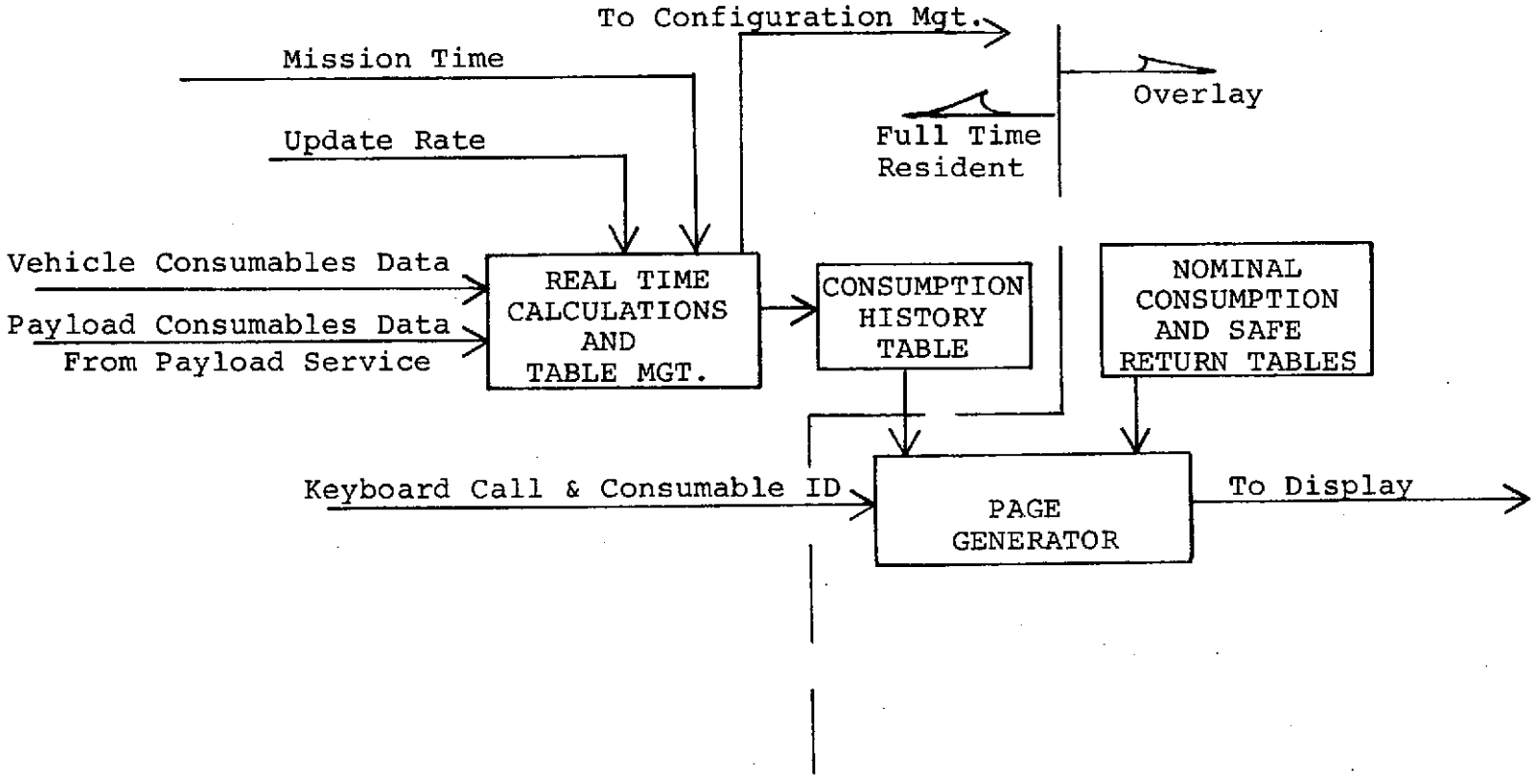


Figure 2.5.2.5. Consumables Management

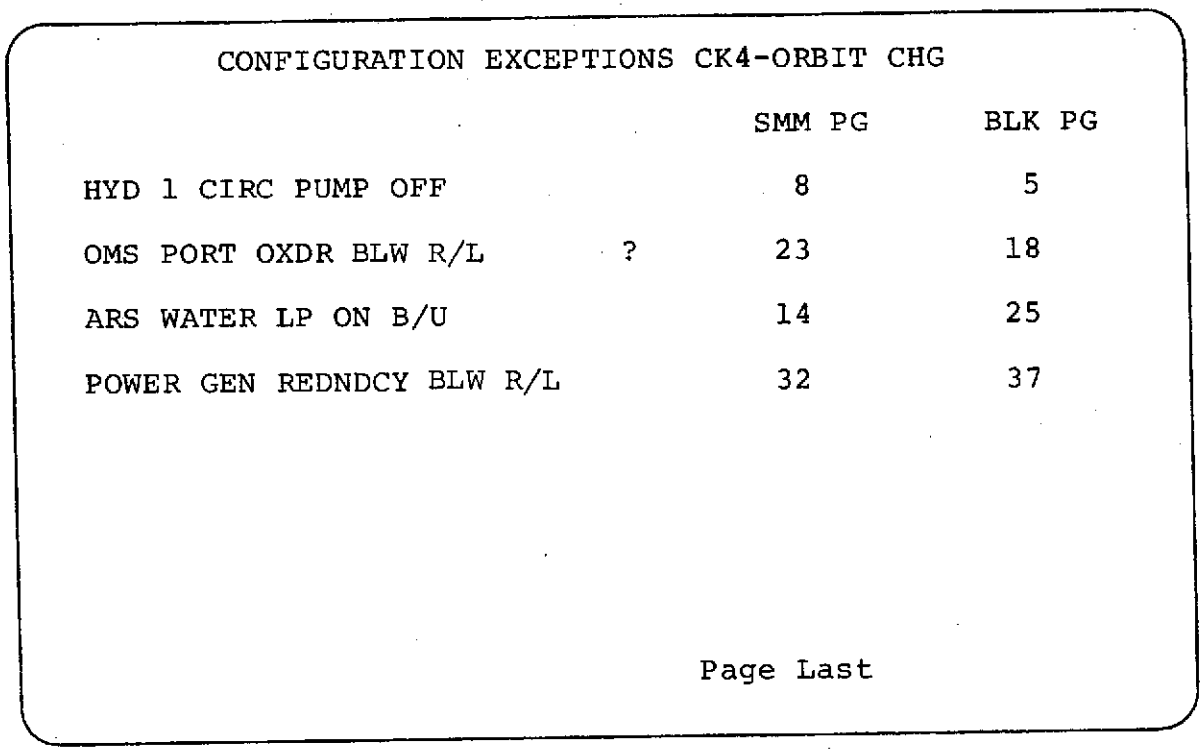
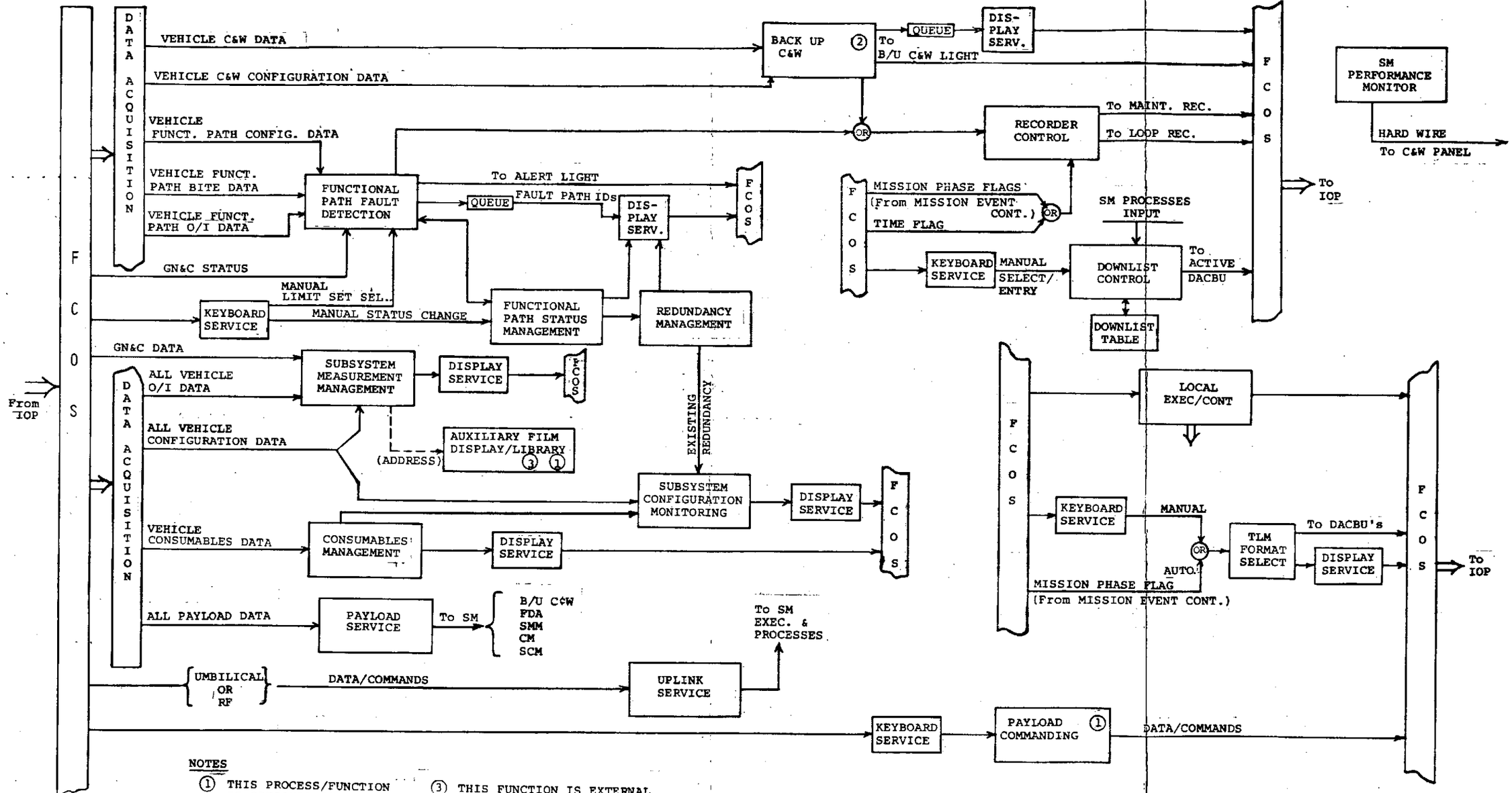


Figure 2.5.2.6. Configuration Monitoring Display



NOTES

(1) THIS PROCESS/FUNCTION IS OPTIONAL TO SM.

(2) THIS PROCESS CAN BE UNIQUELY PARTITIONED IN GN&C CPU.

(3) THIS FUNCTION IS EXTERNAL TO THE CPU. PAGE ADDRESSES ARE PASSED TO CREW ON DISPLAYS.

FIGURE 2.5. FUNCTIONAL FLOW DIAGRAM, ALTERNATIVE SM DESIGN

3.0 IMPLEMENTATION ASSESSMENT — THE DESIGNER VIEWPOINT

The purpose of this section is to contrast the implementation aspects of the alternative design presented in Section 2.5 with those of the Baseline design in Appendix B to this volume. Since implementation specifications are available for neither of the designs, implementation per se can certainly not be evaluated. This assessment might be more appropriately termed "implementation-ability," or an assessment of how readily one design can be turned into a viable, mature product over the other. Four factors will be used in the determination and the two designs will be contrasted for each factor. These factors are:

- Technical Risk - The likelihood that an implementation cannot be realized within budget and/or on time.

- Flexibility - The ability of the design to accommodate changes in requirements and in the design itself. Since SM has a rather special application, features which enhance multi-use or generality are not considered to be of benefit.

- Growth Potential - the ability of the design to systematically accept additional requirements — both projected and unprojected. Emphasis must be placed on projected requirements.

- Reliability - the fault-free tendency of the software design.

It is obvious that the two designs will have to be compared on common terms. It will be necessary to assume that both designs will be implemented in the same hardware/software environment and that they will have identical software constraints. For example, if the Baseline is implemented with a C&W function in the GN&C computers, the alternative design would also be implemented in this hardware partition. If the Baseline is coded in HAL, so too will be the alternative. FCOS is assumed common to both designs.

It should be pointed out that the evaluations in this section will, of necessity, be independent of the suitability of each design as contrasted in Section 2.0. The intent here, as the section title implies, is to contrast the implementation pros and cons of two designs. Is one harder than the other, riskier than the other? Whether one design is operationally more appropriate than the other should not enter into the assessment.

3.1 Technical Risk Assessment

It is appropriate to begin by asking what makes a design a technical risk? Factors which are immediately brought to mind are requirements for state-of-the-art techniques, likelihood of using core/CPU capacity, lack of interface definition, likelihood of exceeding I/O capacity, extensive operator interaction, lack of processing definition, necessity for sophisticated algorithms, dependency on other hardware/software designs and implied extensive analysis of data to achieve defined processing.

Neither design will require state-of-the-art software techniques. Both should be rather straightforward implementations. This is not meant to imply, however, that no technical risk exists in the integration of SM. Orbiter software, all other terminology not withstanding, can only be classified as operating in a multiprocessor, multiprogramming, real-time environment. Such environments are not known for their lack of technical risk. To restate the initial claim above, there is no reason to believe that the designs for SM proper will pose a coding challenge and both should experience about the same integration risk.

CPU loading, core requirements and I/O loading can be lumped into a single data processing resource assessment. The implementation overlay structure will be one of the factors which influences I/O loading and core utilization. The problem is simple, the more programs/data which are resident, the less the I/O peak traffic. Or alternatively, the shorter will be the delay for program/display execution. This latter point is significant on two counts for SM software. First, the Mass Store is implemented with magnetic tape and is inherently slow. Second, at least one version of FCOS will allow a program to become resident or be posted upon call without its data being in core. It cannot become active until data also resides and the data fetch must wait in the core allocator priority queue. Neither of these characteristics is inherently poor, they are mentioned only to indicate a strong motivation for total residency.

Is core a problem if SM is totally resident? IBM has made core estimates for the Baseline Design. Applying a safety margin to these estimates it seems likely that SM can be implemented in 60K ½-words.* Based on the same estimates, FCOS and software housekeeping should require no more than 40K ½-words and COMPOOL no more than 6K ½-words. Total available core is 128K ½-words. Conclusion — no problem when SM is implemented

*Source: "Space Shuttle Orbiter Avionics Software Flight Software Memory Sizing and CPU Loading Estimates," IBM, FSD, October 1, 1974.

in a processor independent of GN&C. The above estimate includes all operating tables but no display formats. What about the alternative design? The alternative design has very little crew interaction and no more displays than the Baseline. Precondition steering and fault annunciation should be simplified even though some duplication of these functions exists. The alternative design adds the new function of status resolution in addition to two new tables. The latter are the Functional Path Status Table and the Redundancy Management Table. The additional processing code for these added features should absorb the reduction in complexity elsewhere. What is left are the tables for the added functions. If the Baseline can be implemented in 60K ½-words, the alternative design should require no more than 67K ½-words. This too poses no problem.

When consideraing CPU loading, there is a CORE/CPU trade which can become significant, especially for software with a high content of tables such as SM. This trade centers around core bit packing for program constants. Storing discretes, for example, as complete ½-words requires more core but eliminates CPU masking operations and thus reduces loading. Packing tables into core has the opposite effect. To avoid this additional variable in the assessment, it will be assumed that the two designs would be treated approximately the same in this regard. This assumption is not without foundation since the table structure for both is quite similar. The difference between CPU loading of the two designs can now be evaluated on the basis of required execution rates. Both designs will handle the same number of parameters. A significant reduction in operations on these parameters, however, is seen in the alternative design. Here only one fifth of the parameters are being limit checked. This reduces not only the checking operation but false alarm avoidance and preconditioning. This becomes particularly noteworthy when one considers these operations to be routine or cyclic in nature. In addition, the alternative design can be implemented to exploit these savings even farther. Since SMM, a demand operation, only needs the bulk of input data when it is active, data acquisition for SMM could be dispensed with when the process is not active. This will reduce the average load but possibly have little effect on the peak load. The processes which have been added to the alternative design require CPU time only in the event of a functional path failure or a table call-up. It can then be stated that if the Baseline Design operates within a safe CPU loading limit, the alternative design will comfortably do so.

Extensive operator interaction always adds risk to software development. This is not meant to imply that operator interaction is uncommon in designs, only that achieving a successful design requires care, additional

storage and often times extensive algorithms. Program/storage protection and lockouts must be implemented, especially if changes to the data base are allowed. A repertoire of illegal entries and a scheme for checking these will be required. To prevent the program from possibly acting on old and new (just entered) data at the same time, break points may be required in the code and certain programs may either cause the CPU to wait (pending a response) or relinquish their execution. System responses to the operator will have to be contrived and these should include notification of delays and acknowledgements. All this adds risk. And, the more extensive the interaction, the more the complications. Operator interaction in the alternative design is considerably less than that for the Baseline and does not include changes to the data base. The alternative design is a good deal less risky with regard to operator interaction.

To the extent that lack of interface definition causes risk, it can only be said that both designs will suffer equally in this regard.

With regard to dependency on other hardware/software designs, both designs rank approximately the same. The alternative design, with additional separation of functions, should experience less of an impact from changes.

For the purposes of this assessment, the factors regarding sophisticated algorithms and extensive data analysis to achieve processing can be combined. First, neither design will require sophisticated algorithms or extensive data analysis. It cannot be ignored, however, that even though the Baseline will require additional data analysis by way of precondition steering, limits and false alarm values, the alternative design will require a good deal of additional investigation in two areas. These are the development of functional path performance measures to be used for fault detection and the algorithms for status resolution. Not only will the performance measures have to be defined but their limits and false alarm constants as well. In this respect the alternative design presents more technical risk. There is simply more to define.

On the whole, the two designs represent approximately the same technical risk with one exception. The alternative design will require additional data analysis and algorithm development. This is not to say that there is excessive risk in technically achieving this development. The risk of achieving the alternative design within original cost and schedule is greater. The extent of this risk can best be grasped by considering that Functional Path Fault Detection represents 10 to 15 percent of the total design effort. This proportion considers the, approximately, 200 parameters it

checks as well as its complexity compared to the remainder of SM.

3.2 Flexibility Assessment

Factors which influence flexibility are modularity, excess core/CPU capacity, the operating system, code branching disciplines, use of a higher level language and the extent to which the code is table driven. Not considered to be a factor in flexibility is the ability to interact with program constants and data base from a CRT/keyboard. Such interaction is an operations policy. That is, it is a way of altering program data in an operational environment, not necessarily a software flexibility factor.

It has been presumed that both the Baseline and alternative designs will be developed under the same software environment. Thus, effects of a higher level language, i.e., HAL, and the operating system on flexibility will be the same for both.

Of the two designs, the alternative design is more modular. The modularity is more than an additional dividing of the processes, it provides functional isolation; C&W is separated from functional path alerts which are in turn separated from the gross problem of manual parameter evaluations. It is less likely that changes will affect all these areas and a change to one has a limited effect on the others. Design of the individual processes can be more directly related to requirements. The Baseline groups C&W and alerts under one operation and the manual parameter evaluation is also tied to this common process. With the in-flight data base change capability, the motives for such an approach are reasonable. The approach is, nonetheless, less flexible due to lack of modularity.

It is reasonable to group the factors of code branching disciplines and table driven code under a single assessment since they affect the finer structure of design. Branching disciplines deal with assembly code and the methods used to handle program branches. If these operations are not restricted to specific blocks of code, the end result can resemble a spider web and is very hard to change. Since the Baseline and alternative designs will be applied under similar software environments, design control for each should be the same. The additional modularity in the alternative design will tend to force some control over branching and is considered to be inherently more flexible in this regard.

The Baseline Design exploits table driven code. This is a definite flexibility asset in that only the tables need be changed to effect numerical (including flags) changes. While not explicitly stated in the alternative

design description, this same technique should be carried throughout. There is no inherent property of the alternative design which would restrict this in the slightest. In fact, some of the original tables would be partitioned which would ease changes somewhat and provide a separation of change responsibility.

From Section 3.1, the alternative design will use more core than the Baseline and less CPU capacity. There is still, however, ample core and for a real time system, CPU capacity is usually more critical. (CPU capacity is more critical from the standpoint that it is harder to get a handle on during design and the average capacity should rarely exceed 70% or so if a peak safety factor is to be realized.) Balancing these two factors, the alternative design is considered to be more flexible than the Baseline.

In summary, it must be concluded that when all flexibility factors are considered, the alternative design is inherently more flexible than the Baseline.

3.3 Growth Potential Assessment

This assessment considers the ability of the designs to systematically accept additional requirements, especially those which have been projected. From Appendix C, three projected requirements are:

Mission Profile Storage and Retrieval
Performance Evaluation and Trend Analysis
Contingency Planning Aid

The factors which affect Growth Potential are the same as those affecting flexibility. These are not going to be reiterated and the conclusions reached in Section 3.2 will apply here. The following material will discuss some additional features of the alternative design which have a bearing on growth potential.

Overlay. If the projected requirements identified above are to be planned for, core is an obvious consideration. The software overlay problem should be solved during the initial design so there will be more latitude to introduce the growth requirements. The projected functions are primarily demand functions. As such, they will require some resident core but considerably more core overlay area. Within the bounds of CPU loading and peak I/O traffic, demand operations in the present design should be designated as overlay to increase the growth potential. It is recognized that CPU loading will vary with mission phase and this must be considered in the overlay-I/O problem. Some contentions can simply be solved by operating procedures. The fact remains, however, that if overlay poses an I/O problem now, it is difficult to see how this is going to get better when demands for overlay increase.

The additional modularity of the alternative design makes it more "overlayable" than the Baseline with a lower minimum residency requirement. This is considered to be an advantage over the Baseline.

The alternative design inherently reduces routine I/O. In light of the foregoing, the advantages of this are obvious. The underlying reason, however, is subtle and will require some explanation. The Baseline uses a "pump-up" scheme for false alarm avoidance. Furthermore, this operation is applied to a large number of subsystem parameters. To achieve acceptable false alarm and repeated alarm rates, the counts used in false alarm avoidance will likely be large. This in turn causes extrapolated response times, especially in the C&W area. In an attempt to reduce response time, the sample rates are correspondingly increased. Now, what about the alternative design? First, the number of parameters which are being limit checked is significantly reduced. This decreases the aggregate problem of false alarms. Second, C&W is using an improved false alarm avoidance or smoothing technqiue which will not demand corresponding increase in sample rate. In short, both the quantity of limit checks and the smoothing technique used affect the sample rate. This rate will be inherently lower for the alternative design.

A final feature of the alternative design has to do with the projected requirement of trend analysis. Trend analysis falls under the body of practices known as forecasting. Before pursuing the details, a general observation should be made. It is very unlikely that viable forecasting can be developed and implemented for every subsystem parameter aboard Orbiter (and payload). Such analysis typically requires comparatively large amounts of data storage and, for large numbers of independent parameters, a wide variety of equations matched to the underlying processes.

Returning to the alternative design trend analysis feature, attention should be directed toward the added C&W smoothing technique. Recall that this technique employs recursive smoothing. Recursive smoothing is a vital portion of virtually any forecasting implemented on a digital computer. As such, one of the principal ingredients of one projected requirement already exists. And, it exists in the area which is likely to receive the most attention in trend analysis. It is not possible to utilize the current Baseline false alarm avoidance techniques in trend analysis.

Recapping the growth potential contrasts of the Baseline and alternative designs, the alternative is clearly superior.

3.4 Reliability Assessment

Factors affecting software reliability are currently the subject of heated debates in the industry. Without fear of too much wrath, it can be stated that these factors all seem to stem from three sources: complexity of the design, ability to control the design and ability to test the end product.

With respect to design complexity, it is difficult to determine a significant difference between the two designs. The extensive operator interaction in the Baseline Design adds complexity over the alternative. On the other hand, the status resolution operation and two new tables in the alternative design add more functions but not necessarily a corresponding increase in complexity. The complexity of the Baseline will be slightly more than the alternative design.

The ability to control and test a design directly relates to design partition, the independence of the partitioned members and the number of space/time states the design can assume. Since both designs are accomplishing essentially the same function using identical data sets, the number of space/time states should be approximately the same for both. The alternative design has not only more partitions but these partitions are a good deal more independent than those of the Baseline. The alternative design is, then, somewhat more inherently reliable than the Baseline.

3.5 Assessment Summary

In contrasting the Baseline Design to the alternative design, the following has been concluded:

- Inherently, the alternative design will result in somewhat improved flexibility and reliability.
- Inherently, the alternative design has a good deal more growth potential.
- Inherently, the Baseline Design represents a somewhat reduced technical risk when considering SM implementation alone. This result certainly should not be extrapolated to the larger context of Shuttle technical risk.

4.0 SOME SPECIAL TOPICS

Contained in this section is a collection of independent analyses which were used in support of the foregoing assessments. Volume II of this report contains extensive material used in support of the alternative Functional Path Fault Detection design.

4.1 Treatment of False Alarms

Before this subject can be properly addressed, it will be necessary to make some further distinctions. The treatment of false alarms has two parts: ways of avoiding them in the first place and ways of handling those which cannot ultimately be avoided. Each of these can be further decomposed into procedural treatments and mechanized treatments. An example of a mechanized avoidance treatment is the False alarm avoidance technique used in the Baseline Design. It should become obvious in the following discussions that the reason for this segmentation is that each of the four ways of treating the problem varies significantly in approach and system responsibility from the other three. Each method will be addressed below.

4.1.1 Mechanized Avoidance

This method of treating false alarms deals with hardware/software implementations which seek to eliminate, or at least reduce, the occurrence of false alarms. In a digital computer implementation, false alarms can come from three sources: (a) numerical/analog anomalies, (b) sample time variation or jitter due to random processor delays and (c), control transients due to state/data disagreements which occur in any sequential process such as a computer. The first source is the one most commonly recognized and the one which receives the greatest attention. The second involves variation in the time at which data samples are actually used in a decision, i.e., the decisions are not always made on samples exactly T seconds apart. The delay can come from the usual multiplexing phenomena but the most serious delays can be caused by bus or I/O contention and processor interrupt servicing. This error source is treated further in Section 8.0 of Volume II. Suffice it to say that, although the identified delay sources can represent large delays, their occurrence should be infrequent.

The final source of false alarms, control transients, has to do with the problem of "state" data which results from sequential operation. The computer does not "see" things all at one time. Its inputs are scattered over a complete cycle of input data scan. Thus, its decisions are not always based on correlated or up to date information. This subject is treated in more detail in Section 4.3 below and can be considered for present purposes to be simply a source of false alarms.

To reduce false alarms, effects of the above sources should be reduced and the final mechanization must be able to cope with the effects which are left. The most obvious mechanized method of reducing false alarms, and the one given most attention here, is to filter out or smooth the remaining errors. That is, attempt to present to the decision mechanism information which closely approximates the "true" value of the parameter in question.

Two methods of achieving parameter smoothing will be discussed. There are obviously others but the two selected are the most promising. These methods have been identified as predecision smoothing and postdecision smoothing. Concepts of these techniques are discussed in Section 8.0 of Volume II to this report. The intent of this discussion is not to repeat that material but rather to briefly describe each smoother and to address their implementation requirements. Each is implemented in software.

The predecision smoother addressed here is a recursive smoother, and in particular a first order or exponential smoother. Its purpose is to smooth the raw parameter values before decisions are made. Decisions of the limit checker are considered to be the state of the parameter.

The postdecision smoother operates differently. This device smooths by counting replicated decisions of a preceding limit checker. Many algorithms regarding the treatment of the decision replications are possible. The most simple, however, will be discussed. It is:

- Declare parameter BAD if N consecutive out-limit decisions are received.

- Once declared BAD, declare parameter GOOD again if N consecutive in-limit decisions are received.

Implementations of these two smoothing methods are pictorially represented in Figures 4.1.1-1 and 4.1.1-2. The predecision smoother must store two values, the smoothing constant, α, and the previously smoothed result. The indicated operation is performed on each sample; i.e., three Multiplies and two Adds.

The postdecision smoother must store three values, the previous output, the smoothing parameter, N, and the previous M-count. The method, as expected, is predominately logic. While it requires more program steps than the predecision smoother, not all the steps are used for each sample. If the parameter remains within limits the top-most path is taken and only a few rapidly executable steps are performed. On the other hand, if the parameter is approaching its threshold, in-limit indications will begin to be punctuated by out-limit decisions due to errors. The method

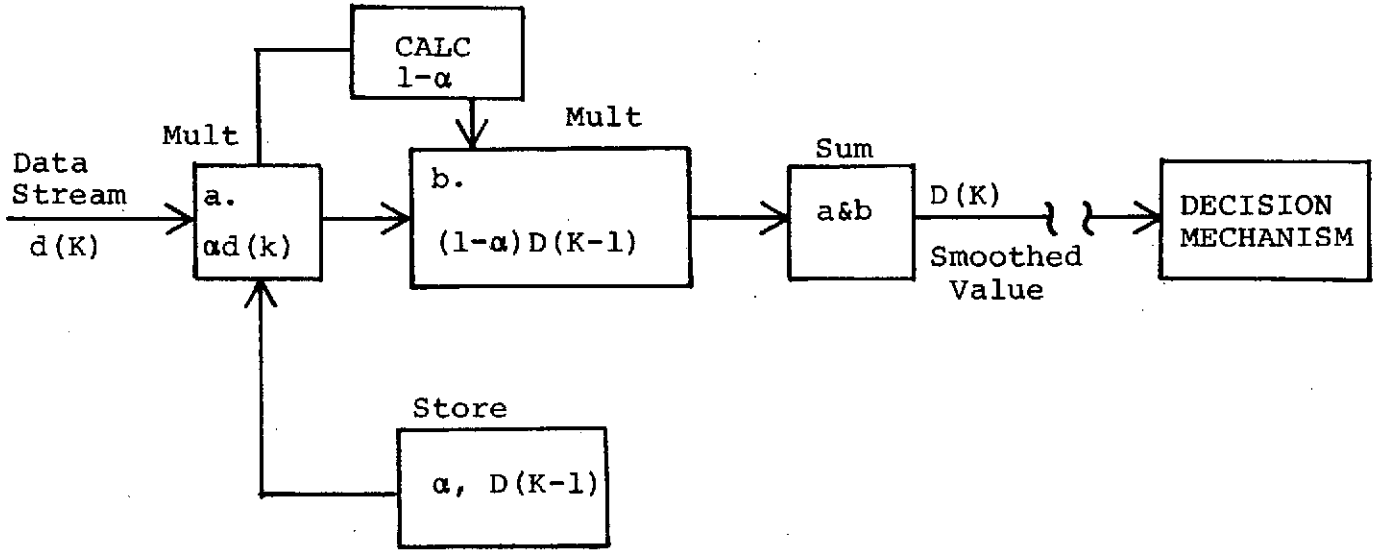


Figure 4.1.1-1. Implementation of a Predecision Smoother

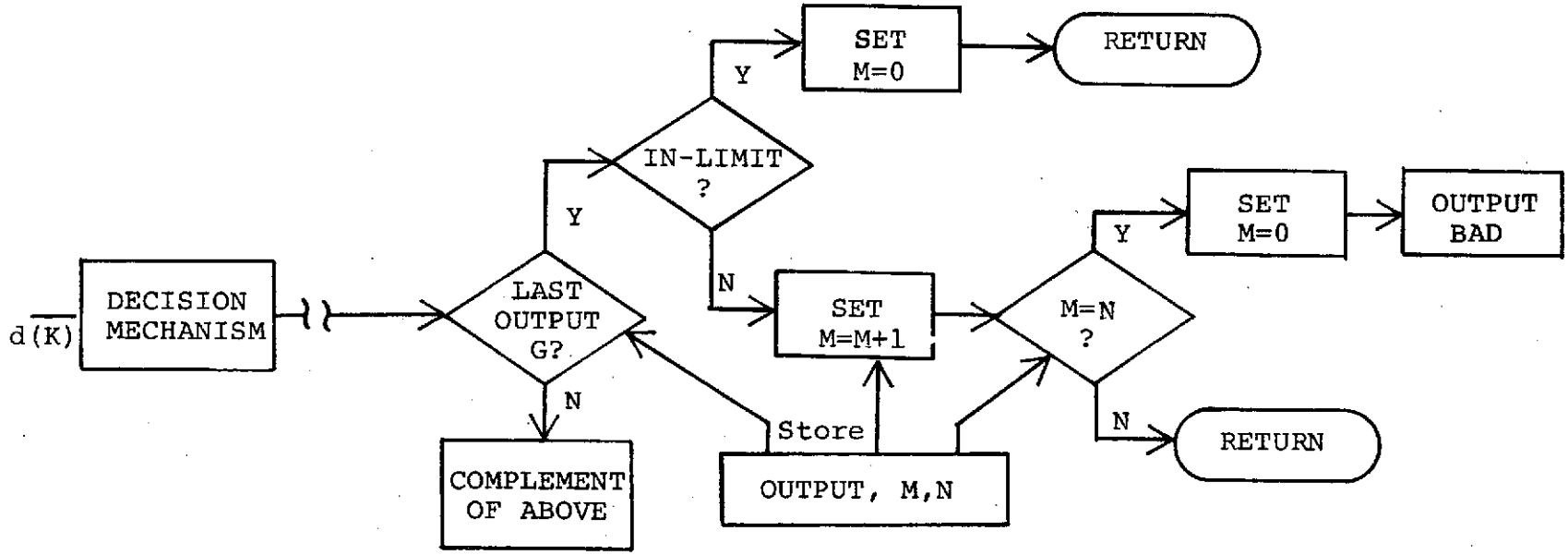


Figure 4.1.1-2. Implementation of a Postdecision Smoother

can then become quite busy. Also, as the number of parameters being checked increases, the higher will be the probability that the method will, on the average, consume more time than the minimal.

In contrasting the two methods, each can be implemented in a half-dozen ways. Also, the detail of accessing tables is not shown. Furthermore, it is difficult to predict how much average time will be consumed by the postdecision smoother. There is, then, little point in attempting actual CPU time estimates. It can be concluded that the predecision smoother can be implemented using less core than the postdecision smoother. And, on the whole, the postdecision smoother will execute more rapidly than the predecision smoother. For a very large number of parameters, the reduction in CPU loading using the postdecision smoother will be significant. This case represents a very strong implementation argument for postdecision smoothing. On the other hand, if a few (say 100) parameters are being considered, the CPU loading difference will often wash out or be taken care of by other factors. Under these circumstances it is reasonable to consider the two more on the basis of performance than on CPU loading.

Before turning to another subject, initialization of these methods should be discussed. Each method will require one value to be initialized. For the postdecision smoother this value is M. The solution here is easy, set all M's equal zero. The predecision smoother requires an initial value for its output (D in the figure). This value will require more consideration. Basically, parameters can be considered as having 2-sided limits, lower limits and upper limits. The initial condition for D must fall well within the good range of these limits. An approach to the 1-sided limits is to use two initial values, one which covers all upper limits and one which covers all lower limits. The 2-sided limits remain and they have to be solved on an individual basis. One of the drawbacks of this method is the initialization for a wide variety of parameters. In the worst case an initial value will have to be defined for each parameter. The only thing that keeps this from becoming a serious contention problem with the postdecision smoother is that the initial values do not have to be stored in core.

4.1.2 Procedural Avoidance

What can be done procedurally to reduce false alarms? Several sources of false alarms were identified in Section 4.1.1. There is another which was implied but not explicitly mentioned. This source is the sheer number of decisions which are made. If 1,000 parameters are each being checked twice per second, this amounts to one billion decisions on a seven day mission. Are they all necessary? Without belaboring the obvious calculations, it is reasonable

to expect that the more decisions that are made, the higher the likelihood of a false alarm. These decisions can be reduced in two ways. First, make decisions on fewer parameters. Second, of the parameters which are being checked, some portion will not require checking all the time. Such parameters can be spot checked at scheduled points in the mission. When on orbit, once a day checks may be reasonable for some parameters with very little reduction in crew information. These parameters could be deleted from the routine checking process and checked by a demand (manually initiated) process similar to Subsystem Configuration Monitoring.

4.1.3 Mechanized Handling

The preceding material dealt with ways of avoiding false alarms. This, and the next, section deal with ways of handling those alarms which cannot be avoided. Of interest here are hardware/software implementations.

If one is depending on a machine to make decisions, the least that can be asked is that it "tell" you when it is in doubt about its results. In fact, one may not be too concerned about the failure of such a machine so long as it announces that failure. This idea generally falls under the heading of "no information is better than bad information." To carry this argument one step further, it would not be unreasonable to define this machine as having failed only when it faulted and did not announce that fact.

The foregoing was intended to construct a performance measure for SM which focused on the handling of false alarms (and other errors). If performance were judged in this fashion, design interest would not be far behind. Any decision mechanism needs a way to check itself. The extent of this checking, and the cost which it warrants, depends on the extent to which the results can be corroborated by the crew and the criticality of the decision. These are the subject of Section 4.1.4 below and will not be pursued here. It is far better to have an error indication for each decision made. This will allow the crew to pick and choose while having the use of all decisions which are not in question. Such an approach can get costly and it assumes that all the decisions are independent. This latter point is seldom the case. In the final analysis, individual error indications are powerful and are practical on a limited basis for selected parameters. They are seldom practical when applied at large.

A compromise to this situation is to incorporate a single error indication. This notifies the crew to be cautious of all results and critical results, if not handled individually, can be double-checked. The duration of the error condition is also of interest. This implies that the self check has to be recoverable, i.e., when (and if) the error condition disappears, the indication should also

extinguish. A natural response to an error indication is to wait and see how long it lasts. If it is transitory, the results for that period are simply ignored. If it lasts for an extended period, the machine likely has a persistent problem and the crew can decide whether to continue, after further checks, using partial results or simply turn it off. A drawback to this technique is that, should another anomaly occur while the error indicator is on, the condition will go unannounced. The crew must, then, treat all results with care.

Implementation of a self-test for SM will involve an error recognition scheme as well as a CPU/IOP self-test. This latter test will undoubtedly exploit the IBM CPU self-test routines. One of the most important requirements for the self-test is that its results do not get handled by the software it is checking. The error indication must be a hardwire to a panel indicator.

A simplified and reasonably comprehensive alternative to the above self-test is described in Appendix A. This method is best used to augment the self-test and at the same time justify its simplification.

4.1.4 Procedural Handling

Within the bounds of crew workload, it is important to be able to corroborate SM decisions manually. Whether or not SM or GN&C find the control surfaces operative, the crew will test their response before leaving orbit. So long as an emergency situation does not exist, the crew should have the capability to corroborate a majority of SM results in a similar fashion. This can be done by observing effects on their panel instrumentation (especially those not software driven) or monitoring SMM displays. Since SMM is in fact serving in this corroborative role, it should (a) be free of machine-made decisions and (b), be easy to use. Once the crew has detected an SM error, they should have a means of either manually correcting it or identifying it as erroneous.

Manual checking of SM results is reliable and simplifies design (see Section 4.1.3). It cannot, however, be totally relied upon. Manual verification takes time, and if an emergency exists, time may be a precious commodity. SM should be of the most value when the crew is the busiest. In this regard, critical results, such as C&W should employ the schemes outlined in Section 4.1.3.

4.2 Treatment of Off-Line Items

What is an off-line item? For the purposes of this discussion, an off-line item is a functional path in a redundant network which has been "switched" out of the active functional role. In contrast, the on-line item(s) is/are

performing the work of the function: moving data, generating electrical power, controlling actuators or whatever. Off-line items are in effect spares or backup for on-line items. Aboard Orbiter, unless the off-line item needs to be constantly updated to allow rapid take over of the on-line job, it will usually be powered down to conserve consumables.

The subject this section wishes to address is how off-line items can be checked to verify operation. Before continuing, however, it seems obvious to ask why these items need be checked at all. They were checked just before launch and presumably found satisfactory. The stress of a launch plus the environmental extremes of space are the two most significant factors causing concern about off-line items. In addition, the crew should know the status of each functional path that will possibly be used in an up-coming mission phase before committing to that phase. Aside from identifying the needs, the preceding statement makes another important point. That point is that the crew does not need off-line status all the time, only at predefined points need this status be re-checked and updated. This point establishes the underlying approach to checking of off-line items — schedule their checking similar to the process/procedure used in Subsystem Configuration Monitoring (SCM). Such a checking scheme could be made a part of SCM. What if the crew switches in a functional path which has failed since it was checked? The consequences of this depend on the criticality of the functional path. For noncritical paths (alert class), the crew will quickly discover the path has failed and switch to the next. If the status of each on-line functional path is being directly checked, the crew will be notified immediately and unambiguously of this condition. Since GN&C employs automatic fault recovery, these more critical off-line items are presumably checked continuously by GN&C. The likelihood of an unknown failed path can be reduced by increasing the frequency of off-line status checks. Emphasis should be, however, on those paths which may be used in an up-coming exercise. It should also be pointed out that with scheduled checking of off-line items, any status indications to the crew should indicate when such an item was last checked.

Now that a scheme has been developed which solves <u>when</u> the off-line items are checked, it is now necessary to <u>turn</u> to <u>how</u> they can be checked. To do this on a general basis will require a grouping or classification of kinds of off-line items. These groups are first divided into whether the off-line item is powered or not. The next consideration deals with whether the items inputs/outputs are accessible in the off-line condition. Finally, the issue of whether the item can actually be tested off line must be considered. This latter point concerns itself with whether a practical input (either simulated or actual) can be provided to the item and whether the item can actually be exercised in the off-line state. In effect, concern is with whether on-line conditions can be duplicated.

Considering all the above possibilities, all kinds of off-line checking schemes can be devised. It is, however, hard to beat actually switching the item on line. This solves most of the problems which can arise by implementing actual off-line checking and does not conflict with the scheduled checking scheme previously developed. Any other method of checking off-line items should be used only in the face of compelling reasons.

The next best choice for off-line checking is to be able to place actual system inputs into the off-line item and check its performance in the same way the on-line item is being checked. In this mode of checking, output or effects of the off-line item cannot be allowed to enter the on-line operation. This is one of the drawbacks of this scheme since, sometimes the output "load" must be simulated. The use of BITE in off-line checking is a potentially powerful technique and represents one of its best applications. If properly designed, BITE can be used to verify off-line paths operating under reduced stress or power. The confidence in such results is necessarily reduced since the item is seldom being completely exercised. BITE should be of the most value when applied to the scheme being addressed here, i.e., applying actual on-line information to the off-line item.

The least desirable method of checking off-line items is to use simulated inputs. This involves the generation of the inputs and the attendant hardware/software. It should be pointed out that there will be some equipment aboard Orbiter which is not used until the final phases of the mission, e.g., TACAN, MSBLS. For practical purposes, all such equipment can be considered to be off-line until the deorbit check. If these equipments are to be actually checked at this time, a simulator will be required. There is no way of getting actual inputs to these systems in orbit.

4.3 Sampled Data and Processor Decisions

Consider a subsystem being checked by SM. When in the powered flight portion of the mission, the parameters will have a given set of limits used for parameter limit checking. Once the on-orbit phase is entered, some of these limits will change or a vehicle reconfiguration will necessitate a different parameter processing (such as may result from state changes to precondition steering). The processor is a sequential machine. Data are read in sequence and conditions can change immediately after a parameter was read. The processor will not be aware of this change until the next data cycle. Thus, decisions made by the processor are not always correct simply due to possibly "stale" information. This same principle applies to all Orbiter processors and the decisions which are made or commands issued. The GN&C voting schemes will necessarily be affected by this behavior.

It should be obvious that these conditions can trigger false alarms. One scheme for reducing this possibility is to assure all the analog-derived data is sampled frequently enough to detect changes of interest (whether this frequency meets the Nyquist criterion depends on what is to be done with the data) and then assure that all the relevant, discrete state or configuration data is read in the same cycle. Then, construct a system which is at least one data cycle error tolerant. This is the technique currently employed by SM. The key to the success of this approach is not necessarily the frequency at which the data are read, but rather that all necessary discretes are read in the same cycle as the analog samples. Ignore other factors which motivate high sample rates for the moment. A parameter could be sampled once an hour and, so long as the pertinent configuration data for that parameter were read within the configuration change response time of the data reading, the above principle would not be violated. For example, hydraulic pressure could be read twice an hour. So long as the configuration of the hydraulic system was checked within one millisecond of the pressure reading, there would be very little room for the stale data effect. This is obviously an exaggeration since a lot can happen to the hydraulics in half an hour. The point is that, notification delay, smoothing delay, subsystem behavior, etc. all dictate rapid data sampling for SM. False alarms due to configuration inconsistencies do not. They simply tie configuration sampling to analog-derived data sampling.

It should be noted that this is not the case with GN&C. This software is performaing operations and issuing commands based on a wide variety of analog-derived data inputs.

There are methods which reduce the tieing of configuration data to the analog-derived data. The motive for their pursuit is the reduction of I/O traffic by reading configuration data less frequently. The Orbiter configuration does not change rapidly nor do the operating modes of the subsystems. From this standpoint it seems fruitless to sample the state of a subsystem twice a second when it only changes state ten times in a seven day mission. And, when it does change state, it requires 120 milliseconds to do so. Two alternative methods of reducing the data-configuration tie will be discussed. They are not necessarily being recommended for general application in SM since each involves a more sophisticated decision mechanism. They are presented for consideration in special applications.

The first method to be considered is an adaptive configuration verification technique. It operates as follows. Consider that a decision to alarm is imminent for a particular parameter based on a 15-minute old configuration check (say that configuration is checked every 15 minutes). Before the alarm is issued, the decision mechanism immediately reads

(on demand) the most recent configuration. If this result agrees with its initial configuration information, the alarm is issued. Otherwise, the alarm is inhibited and the process will begin anew using the updated configuration information.

The second method deals with data autocorrelation or self-correlation. This method will not automatically verify the current configuration when an alarm is imminent. It first tests the reasonableness of the data based on a short history it has stored. If the data made a sudden jump which was considered unlikely for the parameter in question, the configuration data would be verified. If the data were erratic, the configuration would be verified. If the data were reasonable or within the limits of what could be expected, configuration would not be verified and the alarm would be issued. The method has very limited applications to analog data as it is seldom that "reasonable" behavior can be defined for a failing parameter. Discrete parameters, however, are another issue. They should be in one state or the other and should not alternate rapidly between states. As an example, consider a discrete indication which is derived from a device which is close to a source of vibration. The discrete data stream could be a series of random 1's and 0's. This is obviously not a normal condition and a brief data autocorrelation would detect it.

4.4 SM Restarts and Initialization

An analysis of SM would not be complete without a brief discussion of initialization. The issue which first arises in these discussions is the starting point of the initialization, or, in gross terms, whether that initialization should be cold or hot. The former implies starting at the beginning, the bootstrap loader. The latter implies a starting at the top structure by checkpoints or equivalent devices. It is in effect a "flywheel" concept that allows the processor to pick up approximately where it left off. Which of these techniques can be used depends on how "badly" the processor failed. At issue here is whether to design for a hot restart capability or not. Such a design imposes additional work on the processor as well as I/O to Mass Memory.

There appears to be little justification for a hot start SM capability, regardless of how it is implemented. If SM needs to be restarted, it is not a serious problem to lose all current data and start at the bottom. The following reasons are cited in support of this statement:

- SM is continuously updated. In one major data acquisition cycle it will receive a complete set of information. Within the response time

of its smoothers it will be about as current as it will ever get. What is lost is data on the maintenance recorder and SM downlist data. The former can be covered by manual recorder control and the latter is not a serious loss.

- SM does not have to maintain historical records or operate on long sequences of transactions where each result depends upon all transaction which preceded it.

- The DPS Mass Memory is inherently slow. This reduces the feasibility of checkpointing.

What should be of concern is the speed and ease of the restart process. First, the crew must have the capability to restart and they cannot be expected to be computer operators. The process should be automatic and hault only when a check condition is not met, e.g., DACBU not powered up. The crew should be notified of the conditions over which they have control, such as the preceding example, and the process should terminate for other reasons. Restart should be distinguished from initialization since the former is a crew task and the latter is an LPS task (or at least performed prior to checkout). The two have different problems and are initiated under different circumstances.

4.5 Ground Support Trades

The purpose of this section is to examine the trade-offs between real-time, onboard SM computation and ground-based, near-real-time SM computation with up- and down-links. The approach is that of determining the characteristics of space/earth communication in the 1980 time frame. These characteristics are then formed into a description of ground-based computation service and trade-off criteria identified. Finally, SM functions which are plausible candidates for the type of ground-based service are identified.

It is assumed that SGLS services will be limited to mission and payload information exchange and will not be used for the type of ground-based computational augmentation considered here. Thus, SGLS capabilities have not been factored into the analysis.

Space/ground communications services are presented in Section 4.5.1. Under the limiting assumption above, these turn out to be characteristics of the STDN. Section 4.5.2 contains a summary of the ground-based services offered as well as characteristics of services which may be required by SM. Section 4.5.3 identifies the trade-off criteria. Finally, Section 4.5.4 identifies SM functions which are candidates for ground-based service.

4.5.1 STDN In The 80's

STDN will consist of two subnets: the TDRSS subnet and the ground subnet. Current planning by the Goddard Space Flight Center includes Goldstone, Madrid, Orroral, Alaska, Merritt Island, and Rosman in the ground subnet with Bermuda and Tananarive providing only launch support. The ground subnet is intended to support users whose orbital geometry is not compatible with TDRS constraints, e.g., users whose orbital altitude is in excess of 12,000 kilometers, deep space, earth synchronous. Figures 4.5.1-1 through 4.5.1-4 show the coverage, for indicated user altitudes, for both TDRSS and the ground sites. Ground site coverage is included within the dashed ellipses. The cross-hatched area indicates the zone of exclusion for TDRSS. The zone of exclusion is the only area which TDRSS cannot cover. Figure 4.5.1-5 shows the percentage of TDRSS coverage for an average orbit as a function of user altitude. This relationship is plotted for various orbital inclinations. International requirements limiting flux density impinging on earth from a satellite (and presumably the Orbiter as well) can reduce the TDRSS coverage below that shown in the illustrations. Such reduction could amount to 20 percent for users with low earth orbits and omni antennae.

The TDRSS and ground subnets will present compatible interfaces to users. Thus, whether data is transmitted to TDRS or the ground is essentially transparent to the user. The ground stations will have the same signaling, acquisition, receivers, demodulation, decoding, and commanding as TDRSS.

Some additional characteristics of the two subnets are discussed in the subsections which follow.

4.5.1.1 The TDRSS Subnet

The description presented herein is based on the "TDRSS Users' Guide" published by GSFC, X-805-74-176, dated June 10, 1974. Projected NASCOM capabilities were obtained from a previous release of this document.

Operation with TDRSS will require acquisition of the first TDRS followed by a handover from the first to second TDRS when the user's orbit moves between the two TDRS's primary coverage bands. The user will, unless of sufficient altitude to preclude occultation, then move into the zone of exclusion and be reacquired upon emergence by the first TDRS. Depending on orbital geometry, this cycle could occur once per orbit or once per day for sun synchronous missions.

TDRSS is designed to offer real-time service for both telemetry and commands to user Operations Centers. Figure 4.5.1.1 shows a proposed approach which links the

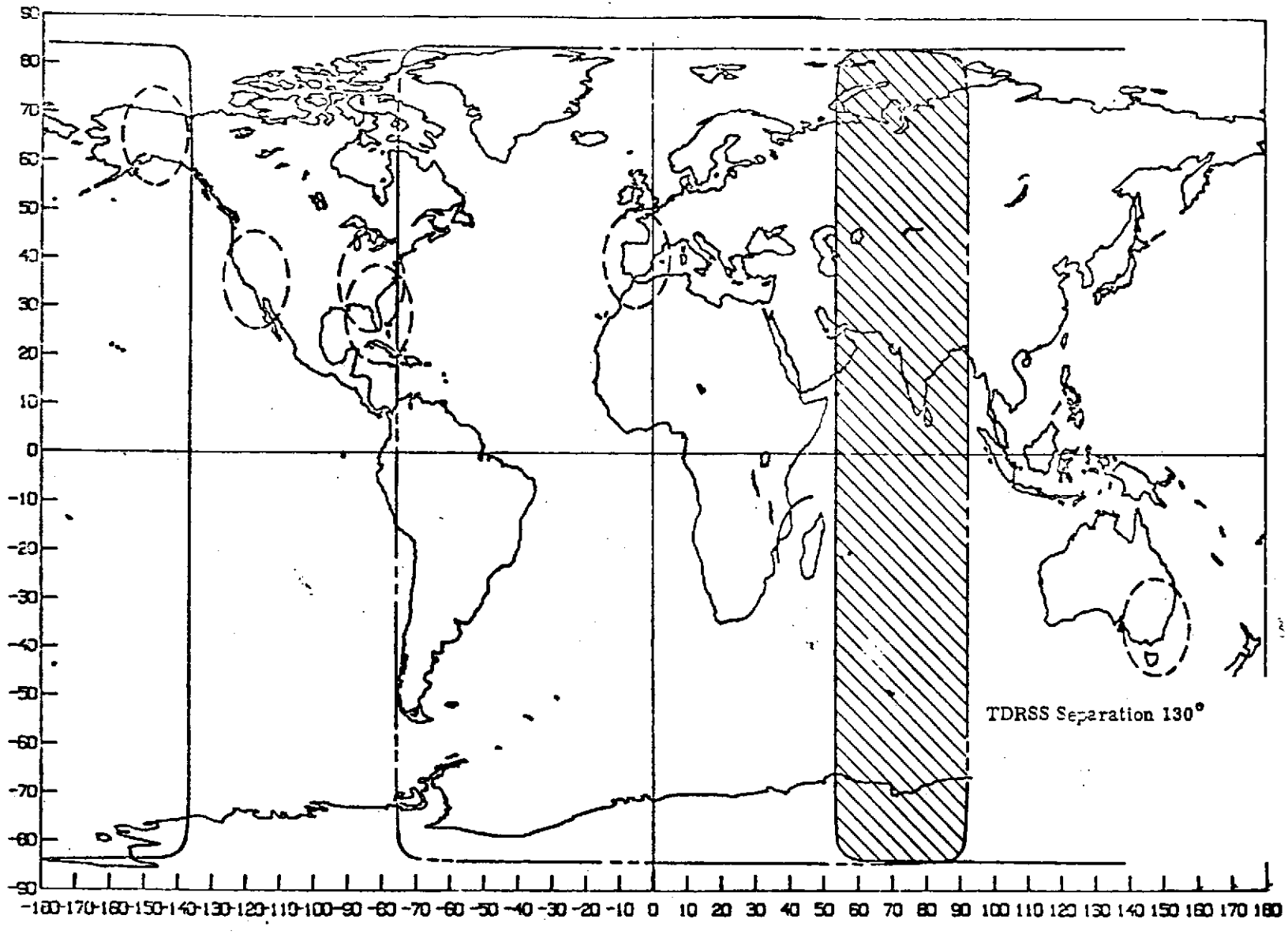


Figure 4.5.1-1. TDRSS Plus Ground Sites, 200 km

(Source: TDRSS User's Guide, GSFL, X-805-74-176, June 10, 1974)

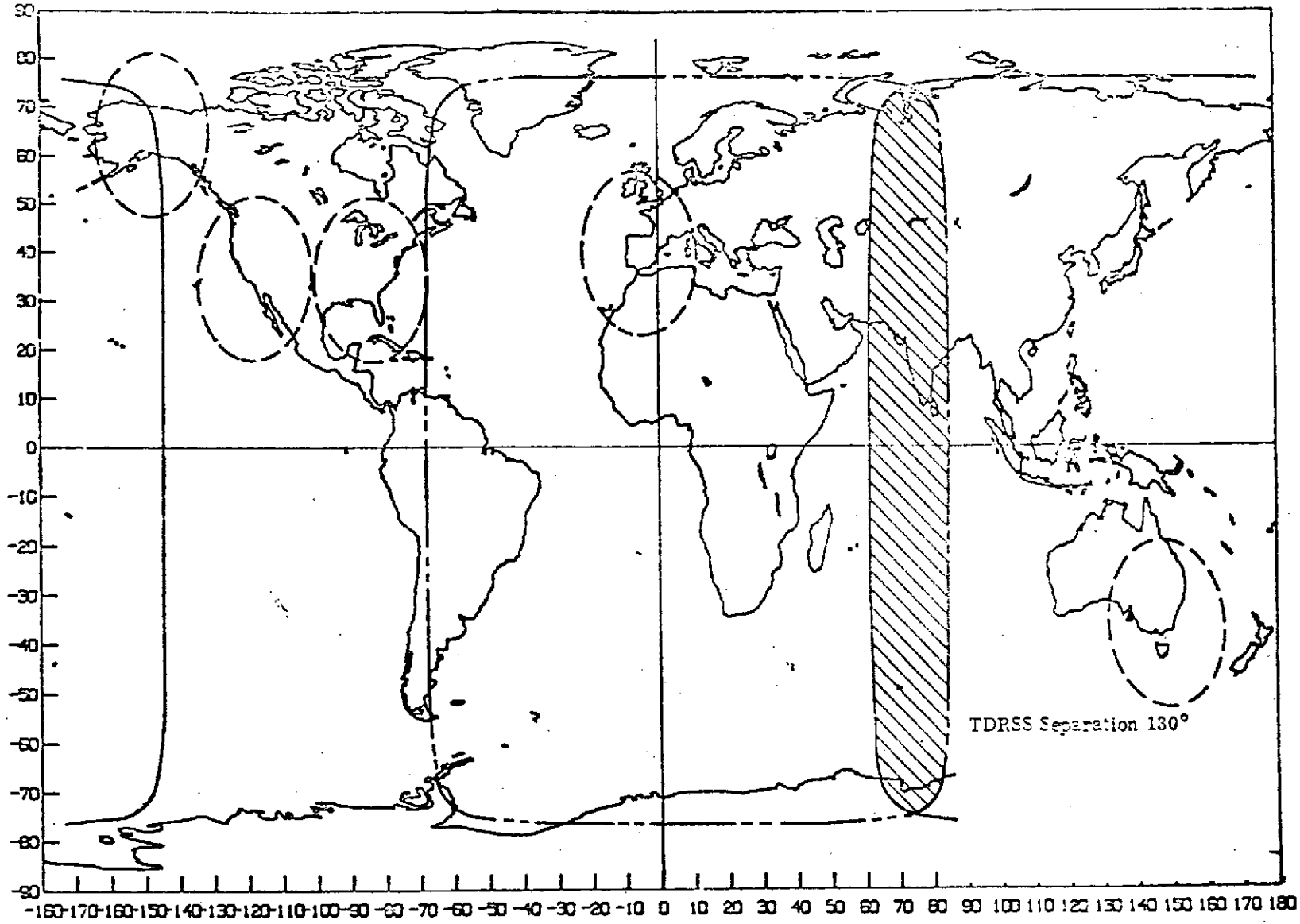


Figure 4.5.1-2. TDRSS Plus Ground Sites, 500 km

(Source: TDRSS User's Guide, GSFL, X-805-74-176, June 10, 1974)

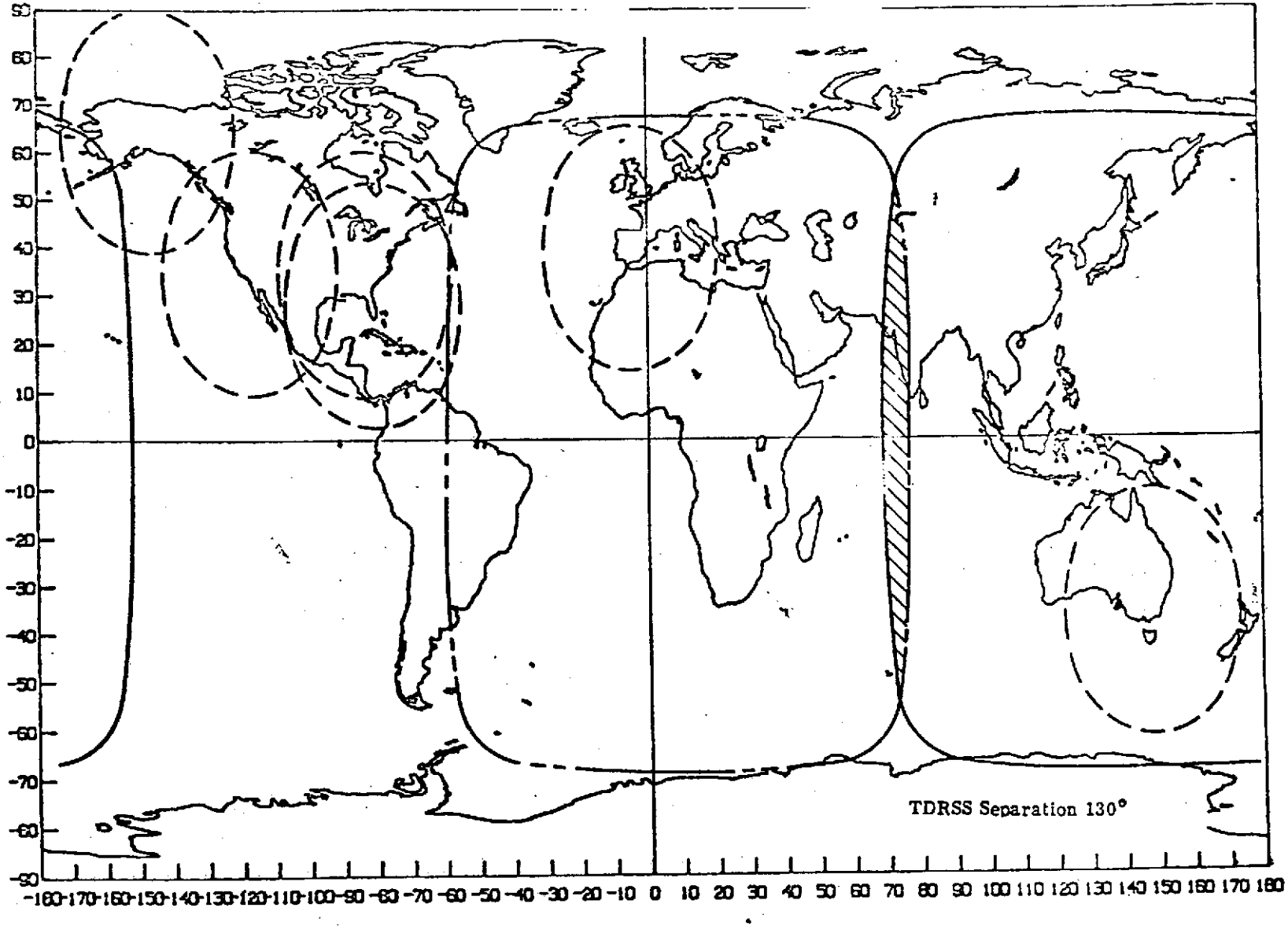


Figure 4.5.1-3. TDRSS Plus Ground Sites, 1000 km

(Source: TDRSS User's Guide, GSFL, X-805-74-176, June 10, 1974)

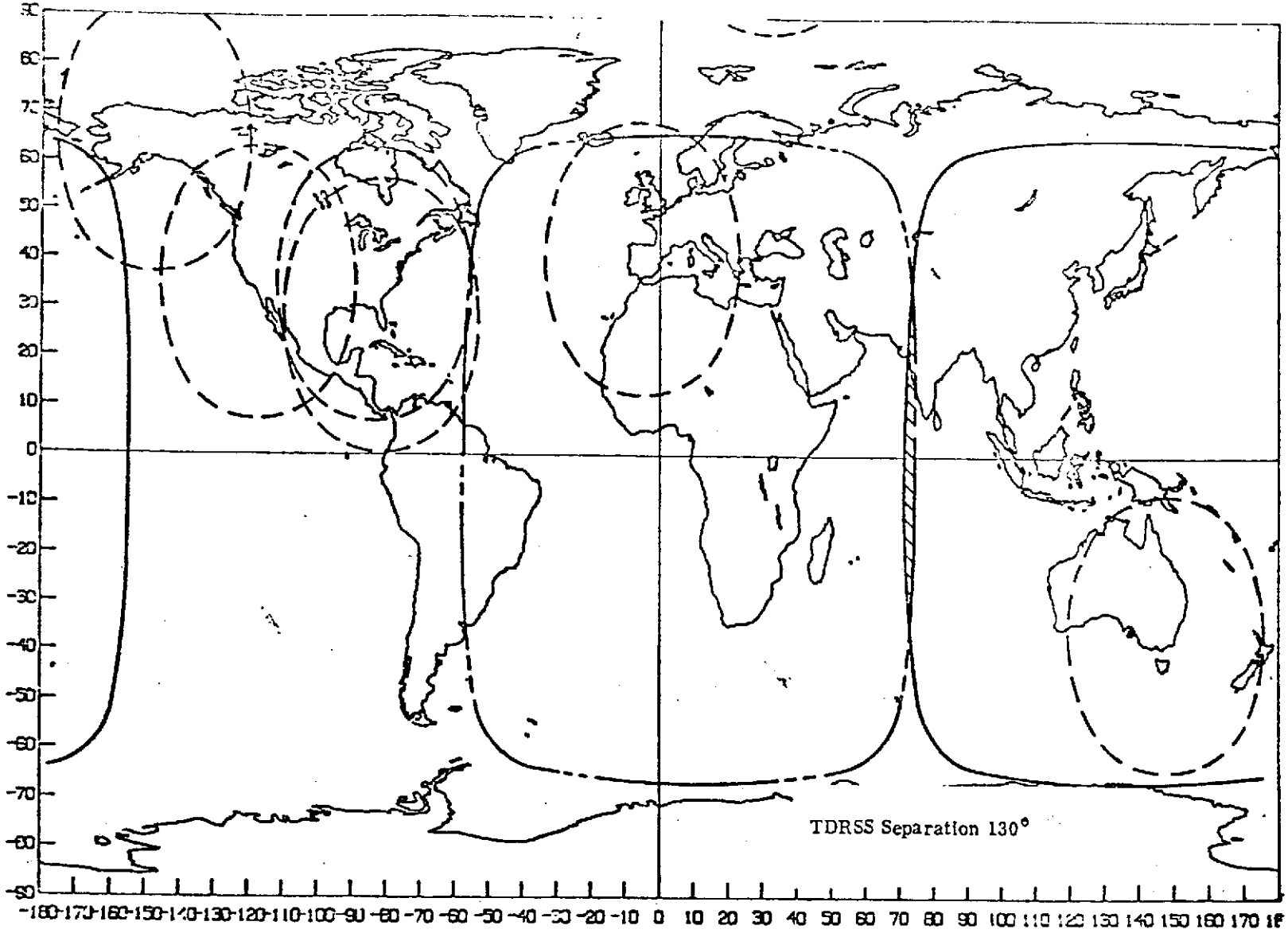


Figure 4.5.1-4. TDRSS Plus Ground Sites, 1200 km

(Source: TDRSS User's Guide, GSFL, X-805-74-176, June 10, 1974)

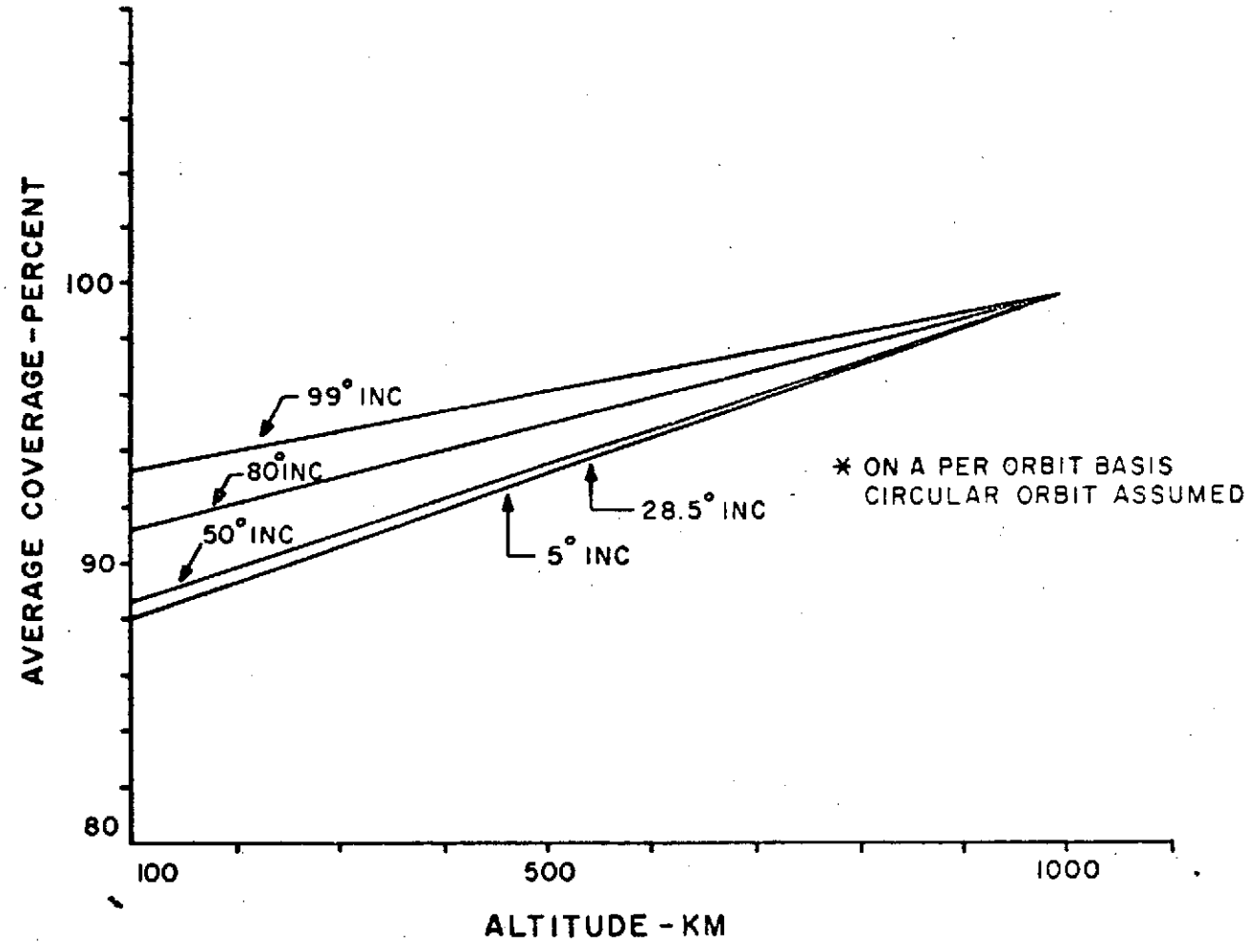
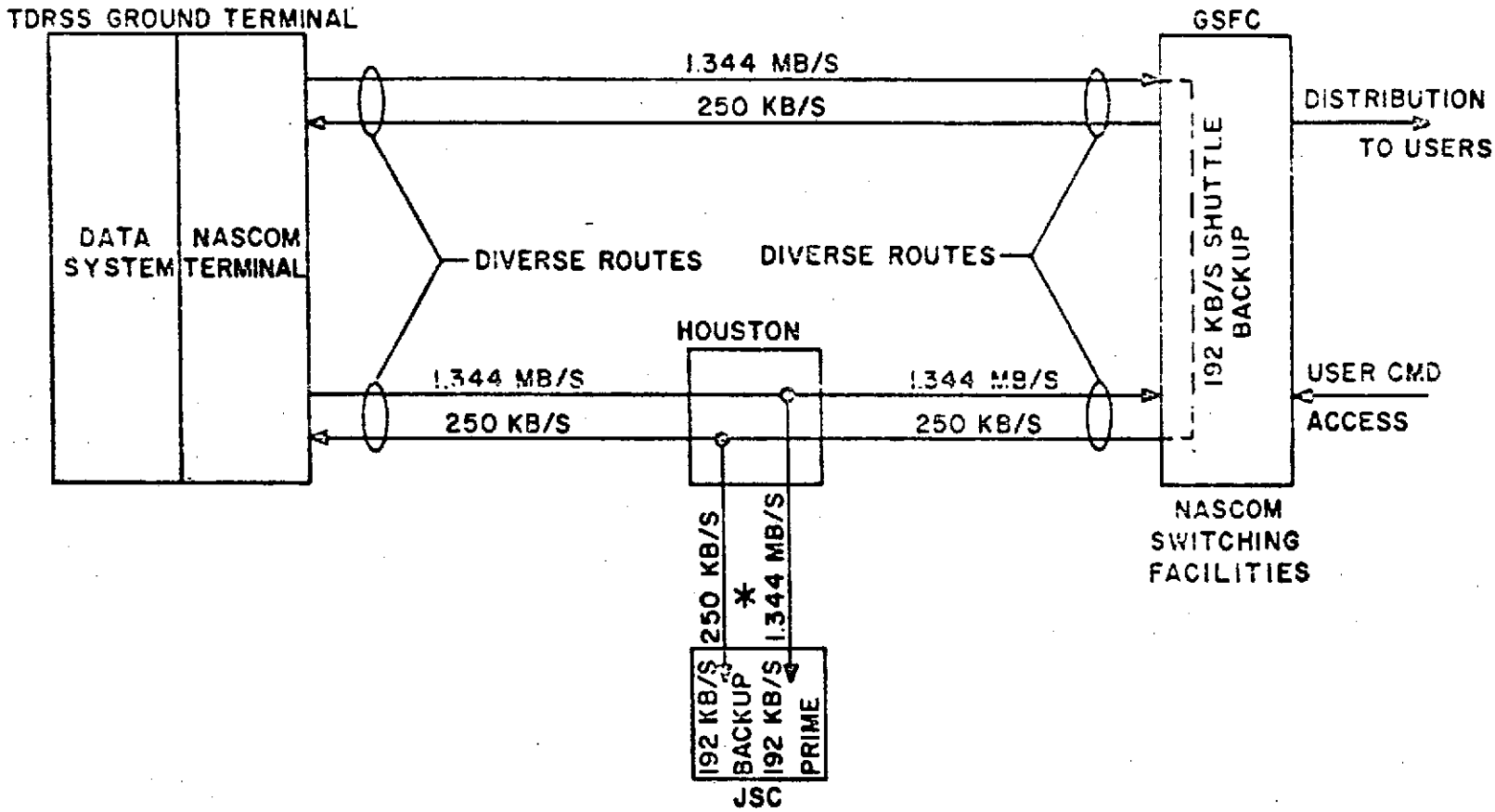


FIGURE 4.5.1-5. *Average Coverage vs. User Altitude, Various Inclinations

(Source: TDRSS User's Guide, GSFL, X-805-74-176, June 10, 1974)



* DIVERSE FACILITIES ASSUMED.

*** NASCOM LINE CAPACITY BASED UPON INITIAL TDRSS LOADING STUDIES. THESE CAPACITIES ARE SUBJECT TO REVISION AS USER REQUIREMENTS DEVELOP.

Figure 4.5.1.1. TDRSS/NASCOM Ground Communication Plan**

TDRSS earth station at White Sands, New Mexico to GSFC NASCOM switching facilities and to JSC over high speed NASCOM lines.*

TDRSS will offer two basic types of service: single-access service and multiple-access service. The capabilities and limitations of each are summarized below.

Multiple-Access - serves up to 20 low data rate users simultaneously for telemetry and time shares up to 20 users for commands. All links must be digital. All users are on the same frequency with separation being achieved by code division multiplexing.

Command Link

Bandwidth - 5MHz

Duty Factor - Continuous

Command Rate - 100 to 1000 bps

Modulation - PN spread spectrum PSK biphase.

Telemetry Link

Bandwidth - 5MHz

Max TLM Rate - 48 kbps.

Support Duration Per User - Continuous when not in exclusion zone

Modulation - PRN spread spectrum, PSK biphase

Single-Access - there are two single-access systems on TDRS, each normally supporting a single user for both command and telemetry. Each system can, independent of the other, operate at S- or Ku-Band or both. Each system can, however, simultaneously support two users, one at S-Band and the other Ku-Band, provided they are both within the beamwidth of the steerable antenna.

*DOMSAT has also been considered as a means of accomplishing these links.

S-Band Command Link

User Separation - Each user at a separate frequency

Bandwidth - 20MHz, tunable over 100MHz

Duty Factor - 100%, normal xmit power
50%, high xmit power (shuttle)

Modulation - PN spread spectrum PSK biphase

S-Band Telemetry Link

User Separation - Each user at separate frequency

Bandwidth - 10MHz

Max TLM Rate - 5 mbps

Modulation - PSK biphase nominal. Other modulation schemes acceptable.

Ku-Band Command Link

Bandwidth - 50MHz

Duty Factor - 100%, normal xmit power
25%, high xmit power

Modulation - PN spread spectrum, PSK biphase

Ku-Band Telemetry Link

Bandwidth

Narrow Band Mode - 88 MHz
Wide Band Mode - 225MHz

Max TLM Rate

Narrow Band Mode - 50 mbps. biphase
Wide Band Mode - 150 mbps. biphase
300 mbps. quadri-phase

Modulation	- PSK biphase nominal. Other modulation schemes acceptable.

4.5.1.2 The Ground Subnet

The description presented herein is based on the "STDN Technical Manual, Digital Data Processing System," MM-4287, published by GSFC. This document basically describes the 1976 capabilities of STDN. Compatibility with TDRSS has then not been considered but will be necessary by 1979. The overall impact of these compatibility requirements can only be speculated.

Ground stations making up the ground subnet were identified in Section 4.5.1. They are:

Goldstone, California
Madrid, Spain
Orroral, Australia
Alaska
Merritt Island, Florida
Rosman, North Carolina

Each site provides telemetry, command and tracking services. Since the sites will present the same interface to the user as TDRSS, the signaling and modulation characteristics identified in Section 4.5.1.1 will also apply to the ground sites. As indicated in Section 4.5.1, the ground sites do not complement low altitude satellite support; they are intended to support high altitude (greater than 12,000 Km) users or users with highly elliptical orbits.

The ground sites will normally store all non-real-time data (dumped from onboard tape recorders) for subsequent retransmission over NASCOM at NASCOM data rates. Real-time data will be forwarded immediately following processing provided the NASCOM data rates are not exceeded. Should a real-time data stream exceed NASCOM rates, critical parameters will be stripped or decommutated from the stream for real-time transfer. The remaining data will be stored for NASCOM transmission at the earliest opportunity.

Each site can handle up to four simultaneous PCM data streams with a short duration peak rate of one megabit per second total telemetry input rate. The total sustained input rate, including all overhead added by processing is 576 kbps. This rate represents the maximum recording (storage) rate.

The maximum real-time throughput rate per data stream is limited by the NASCOM lines. For most sites, this is 7.2 kbps including NASCOM blocking overhead. These circuits have the potential of being upgraded to 56 kbps.

Real-time commanding capability is being incorporated into the sites such that a Project Operations Control Center can command a satellite in real time. The real-time command rate is limited again by NASCOM at 7.2 kbps, including NASCOM blocking overhead. These circuits also have the potential for upgrading to 56 kbps.

4.5.2 Available And Required Services

This section characterizes the communication and computation services that might be demanded by the Orbiter SM of ground support. This is the subject of Section 4.5.2.1. Section 4.5.2.2 then summarizes the type of ground support, based on Section 4.5.2.1 above, which will be available to the Orbiter. This support is considered with and without a TDRSS. Support during launch poses a special set of problems and has not been considered.

4.5.2.1 Typical Services Required by SM

Section 2.2 identifies SM functions and indicates their use demand or time characteristics of operation. Some of these functions will require display generation while others require the retrieval of large quantities of data from files. Still others require minor computation and control. Most require retrieval of Orbiter parameters in real time which are to be displayed and/or compared against constant values on file. Some of the functions must be performed on a continuous or random demand basis while others are scheduled.

While generalizing the SM processes, it may also be helpful to state what they do not do. Not represented among SM computational chores are precision arithmetic, string manipulations, sorting and filing, file manipulation and maintenance, protracted recursions, closed loop control operations.

Except for display creation, SM computation can be characterized as a large number of simple and (almost) identical, independent operations. What then can be said about computational services which may be demanded by ground support? The characteristics are summarized below.

a. All services will have to be initiated by or through the onboard processor. This is a natural fallout since the Orbiter, and in particular SM, is the user demanding and controlling the service. This approach differs from that used on Apollo in that computation results are automatically communicated back to the onboard computer.

b. To fulfill the purposes of SM, whether a service is accomplished onboard or by ground support should be all but transparent to the crew. The ground support should offer only minimal constraints.

c. Some services can demand real-time turnaround of the ground if "transparency" is to be achieved.

d. Computer-to-computer simplex communication will be required as a minimum. This implies a set of time slots be allocated in the PCM stream (or equivalent link) for instruction and identification from the onboard computer to the ground computer. An equivalent up-link will also have to be established over command links.

e. Orbiter data will be passed to the ground over the PCM link, necessitating instantaneous stripping or decommutation of specific portions of these data to be operated on by the ground computer. The decommutating could be different for each task to be performed on the ground.

f. Constants will, of practicality, be stored in the ground computer.

g. Any SM tasks executed by ground-based computers represent special, albeit non-challenging tasks for that machine. The challenges will be in the timing and communications. Once these are solved, the potential of implementing SM secondary functions through ground-based computers poses a very real possibility.

4.5.2.2 Available Ground Support

Having surveyed the required services potentially required of SM, this section will summarize the services projected to be available to provide ground support. Due to the special nature of SM support and due to storage of all necessary constants, it will be assumed that:

a. All potential ground-base computational support will be performed at a single facility -- probably at JSC.

b. Based on the descriptions advanced in Section 4.5.2.1, execution of any or all SM functions at the central computing facility will pose no difficulty in storage, "horsepower," or exe tion time. Also, no difficulty should be experienced in the linkage and discipline involved with communication between the onboard computer and the ground-based computer.

It will then only be necessary to examine the capability of the communication network between the two computers, viz., the STDN. These capabilities will be considered with TDRSS and without TDRSS.

STDN Capabilities With TDRSS

This network is dominated by TDRSS with the ground stations potentially providing an augmentation role for the general range of Orbiter altitudes. As such, the contribution of the ground subnet will not be considered. The Orbiter will be an S-Band single access user and, with capabilities presently planned for this service, the telemetry and command links can readily be handled. The entire PCM telemetry stream can be routed directly to the ground central computing facility. The command link (up-link) should be able to be implemented with comparable facility. Programmable PCM decommutation can be implemented at the central computing facility where it is most cost effective. This network is depicted in Figure 4.5.2.2.

While Orbiter is in line of sight with a TDRSS, this network is not only feasible but affords a great deal of processor augmentation capability. Unfortunately, TDRSS coverage is less than 100 percent for orbital altitude less than 1200 kilometers (746 statute miles); see Section 4.5.1.1. Also, even if handover could be reasonably handled, no ground site "plugs" the coverage gap. This restriction would preclude the use of ground computer augmentation for rapid response tasks which are performed on a continuous basis or which occur in a random fashion. In effect, if a task cannot be scheduled to coincide with TDRSS coverage or if it cannot tolerate the queue incurred until the coverage gap is passed, it is not reasonable to plan it for ground support. For example, a 92-minute circular orbit at mid-inclination would experience a communication dead zone of approximately nine minutes on the average for each revolution. The occurrence and duration of this dead zone is predictable as soon as an orbit is known. Thus, it is not unreasonable, considering the large percentage of coverage, to schedule certain tasks within the coverage zone. A change of mission phase is a good example of such a scheduled task.

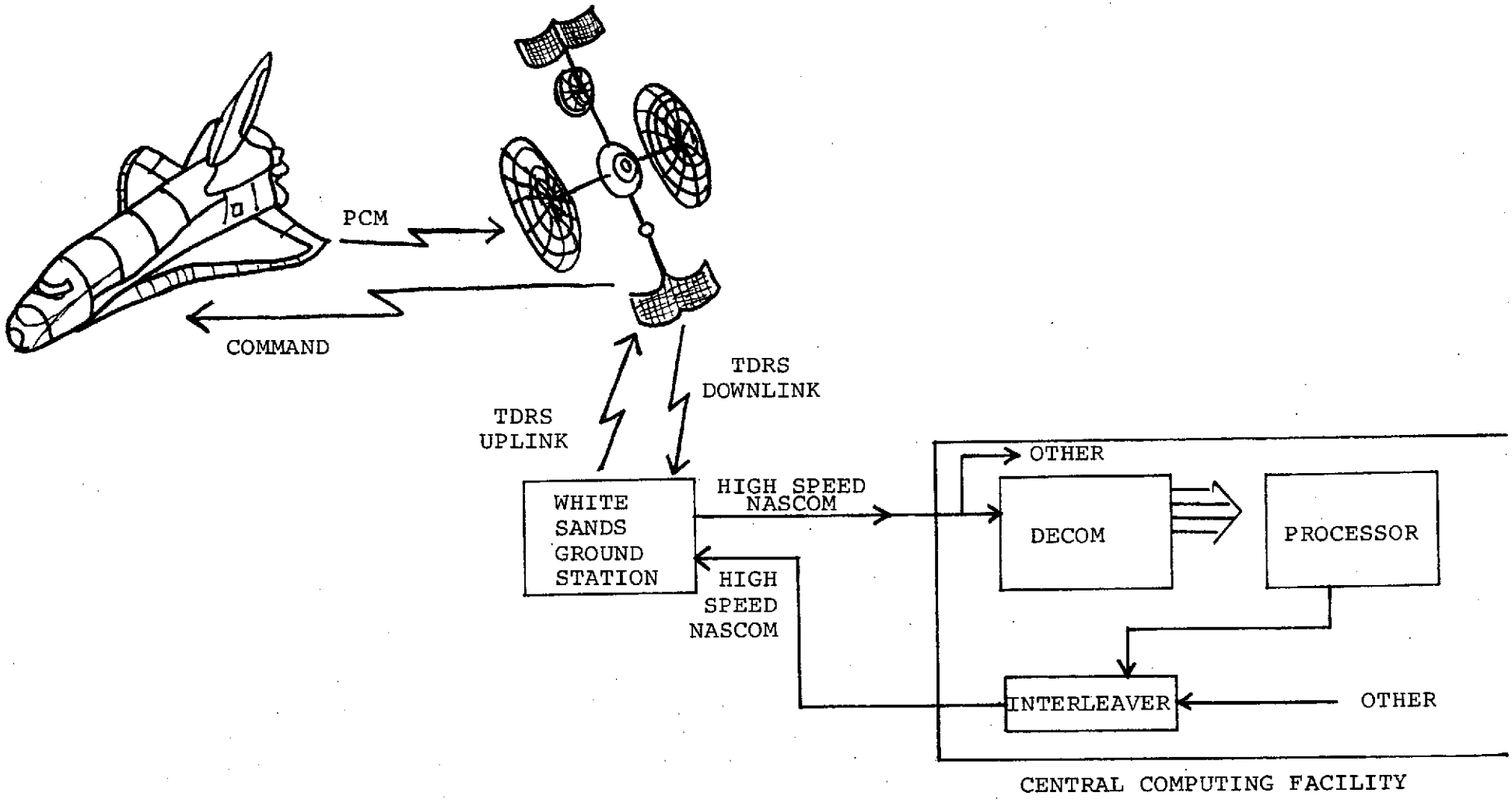


Figure 4.5.2.2. Central Computing Facility Communications Network

STDN Capabilities Without TDRSS

It is appropriate to begin this treatment by taking an exception to the number of sites identified for the ground subnet in Section 4.5.1.2. If TDRSS does not materialize by 1980, it is safe to assume the number of sites cannot be decreased to the six identified. The total number is more likely to be ten or twelve based on findings included in the GSFC document, "STDN Network Integration Plan," dated October 1972.

To achieve near-real-time response through ground sites, the relatively low data rate over NASCOM will dictate decommutation of the Orbiter PCM stream at each site. Furthermore, this decommutation will have to be programmable to accommodate the data (and computer instruction channel) of interest at the time. This restricts, with few exceptions, such operations to those which can be scheduled.

Once the data has been stripped, it can be sent over NASCOM to the central computing facility. The up-link (command link) is, in principal, accomplished straight away in near-real-time provided the ground computation tasks are partitioned such that a minimal amount of data need be transmitted to the Orbiter.

An additional scheduling constraint is imposed with the use of ground sites and this constraint applies to telemetry as well as command. In order that ground sites may connect directly to the central computing facility, they must be switched in synchronism with the user orbit by the NASCOM Switching Facility. This is scheduled to coincide with the period the user is in view of each site.

Communication coverage by the ground sites (even with a total of twelve) will be sporadic. For the 92-minute circular orbit considered earlier, a zenith pass of a site will result in approximately six minutes of communication coverage. Furthermore, the likelihood of contiguous site coverage is remote. Depending on the orbit inclination and phasing, only a single six-minute contact sould result for an entire orbital period. This "spotty" coverage together with the large amount of scheduling will considerably constrain the use of ground-based computer augmentation using only STDN ground sites.

4.5.3 Tradeoff Criteria

Why would the use of ground-based computer services be considered and what is being sacrificed by their use? Directly, the reasons or trades for off-loading SM computation to the ground are:

Reduce mass memory

- program storage
- data storage

Reduce core

program storage
data storage

Reduce processing time

If the realized savings either keep the core and/or mass memory from growing or, in the best of all worlds, allow them to be reduced in size, the tradeoff criteria can be translated directly into savings of weight and power. Unfortunately, such trades are usually entered into simply to keep from exceeding a fixed capacity. This will be especially true for the criterion of processing time. The trade in this case can be translated into whether the design can allow implementation of all defined functions or whether some must be dropped.

Reduction in processing time by delegating certain task to ground-computation must be treated with caution. Time can only be saved if:

a. The delegated task is large compared to time required to execute the additional interfacing operations.

b. The task is independent of ongoing onboard computations. Stated another way, the task can be well defined and initiated with no -- or certainly very few -- intermediate information transfers with the onboard computer.

c. The task has simple interfaces at initiation and completion. By this is meant that there are a minimum number of instructions and data passed and that these be direct (as opposed to indirect or referential) and independent.

d. Other onboard tasks can continue while the task is being executed on the ground, i.e,, the onboard processor does not, routinely, have to wait for results. This implies true parallel processing.

Most restrictions can be summarized in one short phrase: treat the ground computer as a batch processor.

While it has been alluded to in the above discussion, the notion that delegating tasks to a ground computer complicates onboard programming complexity bears explicit discussion. The fact is that such action, while possibly resulting in a net decrease in execution time or storage

requirements, will likely increase SM programming complexity. This statement is based on two observations; one, programming of SM for the onboard computer should not be complicated; two, establishing linkage to a second remote computer may not be as simple, especially under the Flight Control Operating System. Thus, the use of a ground computer will likely increase programming cost if the cost of programming the ground computer is also considered.

The likely increase in programming complexity leads to the next subject: the other side of the tradeoff coin. What does it cost to lighten the onboard processing burden? While each has been discussed previously, it will be well to identify the criteria which must be traded off collectively. They are:

- Increase in overall (net) programming cost.
- Reduction in autonomy.
- Additional time constraints on mission phasing, activity, alteration, and the like. This is due to lack of 100 percent communication coverage.
- Decreased response time.

If the last two items are to assume reasonable bounds, ground support will have to be achieved through TDRSS.* Ground subnet sites should be considered only for dumping onboard tapes. Their use for augmented computer support will quite likely impose a too severe constraint. If TDRSS does not materialize, augmentation of SM computation on the ground should not be considered.

It must be noted that the tradeoff criteria apply only to those SM functions which can feasibly be implemented on the ground. With lack of 100 percent communication coverage, functions which are executed on a random basis or which operate continuously cannot normally be deferred to ground. Thus, functions which are, or lend themselves to being, scheduled should be the only ones considered for tradeoff.

---

*The Orbiter S-Band quad antenna system can impose an additional scheduling constraint if the ground subnet is used during Air Force missions. Some attitudes of Orbiter will force a selection between SGLS and NASA operation for an optimum earth-directed signal. Manual antenna switching may be required to overcome this difficulty. This same constraint also applies to TDRSS operation but to a much lesser extent.

4.5.4 Candidates For Ground Support

This section identifies the SM functions which could be implemented in a ground computer. If a function is scheduled (or able to be scheduled) and can be treated as a batch process job which can run without holding up other SM functions, it is a candidate for ground support. This decision is based on results of Section 2.2 which are summarized in Table 2.2.6. It must be understood that no value judgment is being placed on ground implementation of these functions or for that matter, are they to be considered as design recommendations. The only claim made of identified functions is that their ground implementation is feasible. The functions are summarized in Table 4.5.4 and each is discussed below.

Three functions of the Baseline Design can be immediately eliminated as ground support candidates due to their involvement with random events, i.e., faults. These are Fault Detection and Annunciation (FDA), Subsystem Measurement Management (SMM) and Recorder Control. Note that, due to a division of information, SMM can be ground-implemented for the alternative design. The function of Payload Support can be divided into those tasks which must be executed in real time and those which can be executed deferred time. The former tasks must be assigned to the onboard processor while the latter are ground support candidates. Uplink Service and Downlist Control constitute real time tasks and are also eliminated from ground support consideration.

The function of support to ground checkout does not logically represent a tradeoff candidate. It is, by definition, to be performed by the onboard processor(s). The extent of this function will be decided based on tradeoffs with the KSC Launch Processing System.

Portions of the Subsystems Measurement Management (SMM) function in the alternative design are well suited for implementation on the ground. Since all measurements are being sent to the ground, engineering unit conversion can be implemented for those variables identified by the display page number at keyboard request. The converted variables can then be preformatted into a ground-stored display format or "skeleton" and transmitted to the orbiter for delivery to the Display Electronics Unit (DEU). Measurements not checked by C&W or FPFD could undergo more extensive checking by a ground processor and results of the tolerance check placed in the format before transmission to Orbiter.

Since the Configuration Monitoring function is being performed at fixed points in the mission and since the Orbiter configuration is being compared against a predetermined configuration, this entire function can be implemented on the ground. At keyboard initiation, instructions

Table 4.5.4. Ground Support Candidate Functions

FUNCTION	SUBFUNCTION[1]
Subsystem Measurement Management[2] (Alternative Design Only)	1. Engineering Units Conversion. 2. Tolerance checking for all measurements not operated on by FDA. 3. Preformatting of variables for display. 4. Fixed annotation for display pages ground-stored.
Payload Support	Non-real-time functions whose demand is manually initiated.
Configuration Management	Entire function can be ground-implemented. Correct configurations stored in ground processor.
Consumables Management	Non-real-time portion can be ground-implemented.
Telemetry Format Selection	Automatic portion of this function can be ground-implemented.

[1]Subfunctions not specifically identified are not ground support candidates. The reader is referred to Section 2.2 for a complete list of subfunctions.

[2]Due to variable telemetry formats, ground-implementation of this function may restrict the data which can be displayed at any one time. Furthermore, crew-specified comparison displays will complicate ground implementation.

would be sent to ground to execute this function for the indicated mission phase. The ground processor would then examine all configuration data and generate an exception display for transmission to Orbiter.

Consumables Management consists of a near-real-time pvt calculation segment and several displays, each showing actual consumable use history nominal consumption and an abort limit for that consumable. The presentation consists of time traces of each of these values with the latter two being fixed during preflight for the nominal mission profile. If the mission profile varies from the nominal, the value of nominal consumption and abort levels becomes questionable.

The display portion of Consumables Management (CM) can be implemented on the ground. Storing the constant traces for nominal consumption and abort level is a natural for the ground. The onboard-generated trace of actual consumption can be interleaved in the processor before relaying to the DEU. This will require pressure-volume-temperature calculations for each consumable on board at a fixed interval and storing the results to generate the consumable use history. When CM is called by the keyboard, the ground processor could generate the indicated display and insert the constant traces.

Telemetry Format Selection can be executed manually or automatically based on mission time and phase. It is a simple matter to execute the automatic portion in a ground processor.

In summary, with TDRSS, especially at higher altitudes, implementation of some SM functions in a responsive ground-based computer is feasible. Autonomy, responsiveness and some accuracy (due to RF link bit error rate) will be sacrificed in each case. Net programming cost will likely increase. Whether a function can be treated independently and its ability to be scheduled are the principal criteria for ground implementation.

4.6 Critical SM Interface Parameters

As its name implies, SM is a management information system. Its purpose is to assist users in the management of the Shuttle system. Three users can be identified. They are the crew, the payload owner and the ground maintenance operation. Since the crew is the only user with access to SM, it will be assumed that the payload owner's interests are served by the crew, in particular, the Payload Specialist. SM serves ground maintenance by causing Orbiter parameter and configuration data to be recorded at the time of an anomaly. The only control SM has over the contents of these records is what SM itself places there. Thus, the support to ground maintenance is an indirect one and will not be

pursued further. All that can reasonably be said of SM support here is that the operational state of functional paths, as determined by SM, should be of interest to ground maintenance and recorded.

It has now been established that the SM user is the crew and that SM exists to aid the crew in the management of the Orbiter/payload. If this is its purpose, then the only interface with any significance is the SM/crew interface. All other interfaces result from requirements at this interface. This includes the data processing interfaces with GN&C, Mass Storage, DEU and DACBU. Putting this in the form of a design guideline, "think crew, not instrumentation."

The physical crew interface is a keyboard and a CRT. This, however, represents only a vehicle for the interface of interest — the information content and form presented to the crew as well as that required to converse with SM via the keyboard. Parameters of such interfaces are difficult to define and fall directly in the lap of Human Factors Engineering. Some further characterizations of this interface will be advanced but a detailed definition must remain part of the system analysis. Suffice it to say that the critical interface has been identified and its definition, along with the implied definition of other interfaces, will be part of the system analysis task.

One objective of SM is to reduce the implied crew workload of managing the Orbiter/Payload. This objective is obviously met by automating what would otherwise be manually accomplished tasks. Another objective is Orbiter autonomy. Virtually all the tasks accomplished by SM have been historically accomplished (in one sense or another) on the ground by large computing facilities and an equally large contingency of personnel. Increased autonomy, then, is a two-edged sword, for it also implies a dramatic reduction in operating costs. It is worthy to note that the implication of SM "replacing" ground operations imposes an ambitious task on SM and an equally impressive requirement on the crew interface. To fulfill this objective, SM will have to do a good deal of data processing, sorting and assimilation. Information presented to the crew will have to be well organized and structured such that it is unambiguous, sufficient and not overwhelming.

A third objective of SM which has not been stated in the literature but which is strongly implied by the preceding is crew responsiveness. By responsiveness is meant the crew's ability to take timely corrective action or make contingency decisions necessitated by unscheduled or emergency conditions. If a mission went completely according to plan, SM would contribute to nothing more than a reduction of crew workload, which incidentally, was mostly imposed by autonomy. This is far from trivial, but SM's true value

becomes apparent during an emergency situation. It is the crew's link to the information needed by them to minimize their risk. SM contributes the most when the crew is the busiest.

4.7 The SM Model Dilemma

One of the objectives of this study was to construct an analytical model or simulation of SM capable of predicting change impacts and evaluating the effectiveness of fault detection techniques. Attempts to achieve such a model were unsuccessful. This statement demands some explanation, not only because this was an objective but more importantly to prevent this ground from being retraced. This is the subject of the following material.

It is important to begin by indicating the properties of a model or simulation for use in evaluating system design changes. First, it must be easy to use. Second, it must be sensitive to the causes which effect changes in the performance measures. It should, at the same time, be insensitive to the detail of how a function is implemented. Finally, it should operate on quantifiable variables and yield quantified results which are easy to interpret. It should indicate not only whether a change affected the design adversely or not, but also how much.

The performance measures of just an automated performance monitor such as exists, albeit widely distributed, in GN&C are conceptually straightforward since a known response will occur when a fault is detected, i.e., a path will be switched out. Performance for such a device is judged by probability of false alarm and the probability of a miss. The cost of each of these errors could also be included to provide proper weighting of the errors. Even here, however, a system level model poses problems. First, the two cited probabilities are very sensitive to implementation and the model, at least at this stage, should be reasonably insensitive to implementation. Second, the relationships between the probabilities and the performance monitor inputs, e.g., data identity, sample rate, information content, points of verification, are not as yet mathematically tractable, let alone quantifiable. It is easier to model a specific implementation than to take one step backward in generality to achieve a system model. And, it is not clear what value a model of a specific implementation would have.

SM is more than a performance monitor, it is a crew management information system. There is little utility in retracing the steps of the previous modeling description. Besides, while false alarms are of concern, there is no way of attaching a significance to their occurrence since a forced response does not result. The crew can choose to simply ignore the alarm. Misses have a similar problem.

Except for C&W, the response time of SM is not critical and it will eventually annunciate a hard failed condition. These two parameters would evaluate only a portion of SM anyway. In short, a model describing just performance monitoring is not only inadequate for SM, it is also misleading.

The measures of performance for a system should provide an indication of how well it is achieving its objectives. These objectives were stated in Section 4.6 and a are: reduced crew workload, increased autonomy and increased crew responsiveness (which could also be construed as a safety measure). How are these measures quantified? A satisfactory answer was not found. Nor, for that matter, was a satisfactory set of alternative measures which could be quantified ever found. Section 4.6 indicates that the output of SM is basically a Human Factors problem. The Human Factors approach to such a problem is to set up experiments using the designs to be evaluated and have subjects (in this case the crew) actually use the alternative designs. Subject fatique, error rate and the like are then measured and factored with their individual assessment of the designs. The models in this case are physical and quite specific. Analytical models are found in the Human Factors laboratory. It is, however, one thing to construct an analytical model for a laboratory but quite another to construct a system engineering model to evaluate system impacts from changes.

Some cause and effect relationships were discovered which are used in the system analysis in Sections 2.0 and 3.0. With these it is possible to determine the proper direction to take and assess whether a change has increased or decreased autonomy, crew workload or responsiveness. The next logical question that should be asked is, how much was the effect? This implies both quantization of the performance measures as well as a complete "audit trail" from cause to effect. This could not be done. If the sample rate of hydraulic pressure is reduced by 50%, how much does this affect reduction in crew workload?

This section should end on a positive note, indicating what has been learned. First, systems such as SM have not yet reached a stage formal enough to warrant analytical modeling. The subjective approach, guided by an appropriate checklist for the design, is the state-of-the-art. Second, a strong set of performance measures has been defined. These help considerably in reducing subjectivity and the scope or latitude of design evaluations. Finally, some cause-effect relationships have been defined. These are discussed in the following section.

4.8 A Design Checklist

Efforts at modeling SM resulted in the establishment of a design checklist. This checklist was used in the

system assessment in Section 2.0 and should be of aid in tracking and evaluating SM design changes. Each performance measure will be broken down into its constituent contributions to formulate the checklist. As may be expected, many of the factors are repeated for the performance measures.

The objective of autonomy is to significantly reduce the ground operations contingency. The extent that the crew, using SM, can accomplish this is a measure of autonomy. What factors contribute to this measure? Basically, the crew will need essentially the same information ground operations used to get. Furthermore, this information will have to be more compact and easier to interpret since the Orbiter will not be carrying along the team of experts that existed on the ground. The design parameters which directly affect this are:

- Sufficient information. Is enough information available?
- Information Accessibility. Can the information be retrieved by the crew?
- Information Organization. Is the information organized such that the decisions made by ground operations can be duplicated? Can a convenient assessment be made of current vehicle/payload status?
- Credibility. Are the results consistently believable?

Autonomy and reduction of crew workload are far from independent. They are in fact, dependent. If it were not necessary for Orbiter to be autonomus, the crew would have much less to do. Thus, in order for SM to increase autonomy while not increasing crew workload, it will necessarily have to do more tasks, become more automatic and have more clear cut data presentation techniques. A large portion of SM tasks could be performed manually. For a given level of autonomy, this is a pretty good indicator of how much SM is reducing crew workload.

The design factors which directly affect crew workload are:

- Functional Usability. Is SM easy to access and control? Will the task of updating, monitoring and controlling SM be almost as great as that of manually performing the Orbiter assessment? Can the necessary information be directly obtained?

- Information Presentation. Is the information easily and quickly interpreted? Is it concise? Is there too much at one time? Is a lot of manual analysis required for routine information? Must corroborative checking be routinely relied upon?

- Information Levels & Criticality. Is the information divided into segments which group those conditions which require immediate attention, deferred attention and interest-only attention? Does the information response time correlate with these criticality levels?

- Information Accuracy. Is the information accurate enough for the purpose? Is it sampled sufficiently? Is it sufficiently correlated with vehicle configuration? Does information accuracy correspond with whether a gross assessment is being performed or a detailed evaluation?

Crew responsiveness is concerned with the degree with which SM aids the crew in an emergency. The design parameters which directly affect responsiveness include those in reduction of workload plus the following:

- Credibility. In emergency situations, SM results must be believable. There may not be time to trace all the information. Have false alarms been decreased to a practical level? Can unavoidable false alarms be conveniently handled? Does SM have a viable self-checking feature?

- Troubleshooting Ability. Can the crew quickly get to the cause of a problem? Can they assess its extent conveniently. Can they easily associate parameter readings with physical/functional locations on the vehicle? Can they correlate data easily? Can they reliably determine not only if a parameter is in bounds or not but how it is behaving?

It can be seen from the above that performance monitoring is only part of the SM problem. Performance monitoring per se is embodied in only two SM functions. FDA and SMM. Requirements on how performance monitoring results are to be handled cannot be established until the crew is considered. The design factors for performance monitoring are extensively covered in Volume II to this report.

The distinction between performance monitoring and other SM functions is important. Performance monitoring determines the operational integrity of vehicle subsystems. It should be able to also determine if the subsystem functional paths are operating acceptably or not. In addition, it should provide the capability to verify the functional path determinations as well as the capability to assess the extent of a problem. All other SM functions cannot be considered as performance monitoring. Vehicle configuration, remaining consumables and C&W constitute operational information, not performance information. An exhausted consumable does not represent degraded subsystem performance. Although associated subsystems will soon cease to function, they are not necessarily to blame for the condition. This is operations information.

5.0 TOPICS FOR FURTHER STUDY

One of the important functions of a study is not only to accomplish the immediate objectives but to identify areas requiring additional investigation. These areas are identified below.

a. The purpose of SM Maintenance Recorder control is to provide a data window spanning the time a fault occurred. Depending on the behavior of the parameter at the time of Fault, using the results of Fault Annunciation could seriously shift this window due to False Alarm Avoidance. Using out-limit conditions prior to False Alarm Avoidance could cause an excess of data. Whether either of these is a problem depends on what data ground maintenance needs. The recorder control mechanism should be examined further.

b. This area directly relates to (a) above. Requirements for ground maintenance data should be established. This involves a development of a more detailed ground maintenance concept and a complete testing/isolation philosophy. In addition, requirements for a software data processor for analyzing Maintenance Recorder data should be established.

c. How SM operation is to be verified by LPS is bound to affect its design. This area should be resolved and design requirements placed on SM.

d. When subsystem functional paths change operating modes or are turned ON/OFF, transients are sure to result. It is not clear if False Alarm Avoidance will handle these situations. Several solutions have been advanced by the Baseline designer. Evaluation of these solutions is not clear cut. Transient-handling sould be examined in more detail.

e. The function of false alarm avoidance is used extensively in SM. Furthermore, it serves in a rather critical SM role. Two methods of false alarm avoidance are described in this report and each has been analyzed regarding pros and cons as well as

gross performance. In view of their role, the two techniques, including higher order versions, should be more carefully analyzed with simulation.

f. The alternative design presented in this report requires data analyses which are not currently required by SM. Specifically, these are (a) the functional path definitions, (b) functional path parameter and performance measure definition and (c), status resolution or post-condition steering definition. These analyses will be required to the extent that the alternative design is adopted and can be performed as described in Volume II to this report.

g. To the extent that the alternative design is adopted, additional implementation design and software functional specification will be required.

APPENDIX A

SOME ADDITIONAL CONSIDERATIONS

A1 INTRODUCTION

This appendix contains several SM design options which occurred during the course of the system analysis and which, for various reasons, were not included in the alternative design. Their direct application is not necessarily recommended since the implications of their inclusion were not completely pursued. The options are believed to have potential; if not in described methods, then at least in the principles they represent. It is likely that some of the options have been brought up during early SM development and were forced out by design trades. Since the alternative design alters many of the suppositions which were likely used in the Baseline Design, the options should be considered in this new light.

SCHEDULED FUNCTIONAL PATH STATUS EVALUATIONS

The basis for this option is discussed in Section 4.1. Its implementation will further reduce false alarms. Every functional path does not justify continuous status determination. This is particularly true for off-line paths. The functional paths should be divided into two logical groups: one requiring continuous status determination and the other, scheduled determination. The groups should be identified logically since the physical equipment can not be assigned to one group or the other. It will move between groups depending on whether it is on-line or off-line, whether it is operative or inoperative and on the mission phase.

The group containing continuous verification items is checked as always. The group requiring scheduled determination will be status checked on demand from the crew. In view of the correlation between crew requirement for this information and vehicle configuration checks, demand or scheduled status checking could be incorporated as part of SCM. Additional displays should not be necessary.

CONSTANT VALUE SELF-CHECK

One of the difficulties with verifying the operation of any function, especially the hardware/software combination of a digital computer, is the inability to evaluate the operations when they are handling unknown data. A way around this is to insert a known constant into the input data stream and check how the processors handle this constant. Since the input quantity is constant, the process should always handle it the same and produce the same result. For SM, this could be a known constant voltage analog applied at an OI MDM which was read just as any other parameter. The known parameter would be processed and limit checked. If this parameter ever checked out-of-limits, SM hardware (CPV/IOP) or software is very likely to be malfunctioning. Since the

decision processes within SM are serial operations, this parameter passes through the same operations as all other parameters, and, under very similar circumstances. A smoothed indication of this parameter being defective could be switched to a hardwire computer output.

The preceding represents one approach to a constant value self-check. There are numerous variations on the theme. Depending on the extent of checking desired and any resulting ambiguities which must be resolved, the known value could be inserted virtually anywhere, e.g., a PROM in the IOP, and checked at several places along the data route. In the above description, the known value was inserted before the DACBU. A DACBU verification of this value would be a viable BITE operation of the DACBU/OI bus.

MANUAL SM TEST FOR NO ALARMS

This option is an extension of the self-test described above. SM design has been dominated by ways and means to avoid false alarms. This concern has not been unwarranted but there is a complementary issue which should also receive attention - especially if all the efforts to reduce false alarms prove entirely successful. SM is an exception reporting device. Everything is assumed well by default. If all is well, SM does nothing. An entire shift could go by without a peep from SM. It would be natural to inquire if it is still working as such a failure mode could be possible. How does the crew verify this?

A way is to force an out-of-limits parameter and see if SM catches it. This could be done using the constant value discussed earlier. A panel switch or pushbutton could change this value (or some other known value) to a constant out-limit condition. SM should then respond to this condition. The scheme is a rather comprehensive checker since it follows the same information flow as the constant value check. In effect, the constant value check verifies that SM remains quiet when it should and this check verifies that SM responds when it should. If the known parameter were given a large smoothing constant, the panel button could also verify operation of the smoothing process. An extended depression of the button would guarantee a response if SM was working. A momentary depression should result in no response. As before, there are numerous variations on the theme.

CONFIGURATION MONITORING CONSUMABLES DISPLAY

It is not difficult to incorporate a consumables check as part of Configuration Monitoring. Such a check would give a GO/NO-GO to minimum acceptable consumables levels for the up-coming mission phase. If a NO-GO occurs for some consumable, it is not currently convenient for the crew to determine just how serious the condition is. The existing

nominal consumption display appears to be too coarse for this determination and the alternative method is to manually check hard copy. A consumables scan display incorporating bar plots of remaining consumables as well as a minimum acceptable level would not only indicate GO/NO-GO but the extent of the condition. Such a display appears on the cover of this report.

DATA PROCESSING STATUS PRESERVATION

One of the most important maintenance aids for data processing equipment is the contents of its registers at the time of the failure. In particular these registers include ADDRESS, STATUS, PSW and PC. The contents of these registers will aid ground maintenance and turn-around and a concerted effort should be made to get this information on the Maintenance Recorder.

APPENDIX B

BASELINE DEFINITION

SYSTEM MANAGEMENT (SM)

OPERATIONAL VERSION

B1.0 INTRODUCTION

This Appendix provides the conceptual level of the SM baseline design definition. The definition is, for the most part, based on Johnson Space Center Document Number SS-P-0002-430, Space Shuttle Program Orbiter Project, Computer Program Development Specification, Vol. IV, Book 3, dated July 5, 1974. While this document is intended to specify the Approach and Landing Test (ALT) version of System Management, there is a more-than-adequate framework upon which to build an operational version. Other documentation predating the above reference (for example, see Section 2.3 in the body of the report) was used as reference to complete the structure. It has been tacitly assumed that the portions of the design applicable to ALT will remain essentially unchanged for the operational version.

Section B2.0 provides an overview of the SM function while Sections B3.0 through B5.0 provide SM inputs, SM processes and SM outputs, respectively. Finally, Section B6.0 discusses the crew interface and operations in more detail.

B2.0 PERFORMANCE OVERVIEW

System Management (SM) is basically an operations and maintenance information system and is implemented exclusively in software. It has access to all the Data Processing System peripherals (keyboards, CRT's, Display Electronics Units and Mass Memories) as well as the PCM Master (DACBU).

SM provides the crew information regarding the vehicle health, configuration and performance as well as status of consumables. The automatic selection of telemetry format is also accomplished by SM as determined by mission phase.

SM contributes to Orbiter maintenance and turnaround by controlling the onboard maintenance recorder and inserting portions of its processing results into the telemetry/recorder data stream.

SM contributes to payloads or Oribiter users by providing the same services it does for the crew. SM also provides the capability to generate and transmit onboard commands/data to payloads.

B3.0 SM INPUTS

This section identifies SM input entities and the information moving from these entities. Three entities are defined: crew, Orbiter systems and uplink.

B3.1 Crew Inputs

The crew communicates with SM using the keyboards. No CRT-related communication (cursor, etc.) is available. The following types and operations enter this input.

- Payload data/commands
- SM display selection
- SM initialization control
- SM process control
  - change limit values
  - override parameter values
  - change smoothing constants
  - inhibit alarm for selected parameter
  - change program constants
  - inhibit alarm tone
  - turn FDA on-off
- SM process control display/function selection
- Manual telemetry format change (load)

B3.2 Orbiter Systems Input

The following information types and sources enter this input.

- DACBU
  - Vehicle Operational Instrumentation (OI) data. These are essentially all data not flight critical
  - Payload data
- GN&C
  - Flight data retrieved by GN&C and relayed to SM
  - GN&C subsystem status, i.e., results of GN&C automatic redundancy management
- Support from Data Processing System

B3.3 Uplink Inputs

There are two distinct uplink paths; R.F. and GSE umbilical.

B3.3.1 R.F. Uplink

The following are received via the R.F. uplink.

- o Parallel switch commands (vehicle)*
- o Other SM commands/data

B3.3.2 Umbilical Uplink

This path is used during ground servicing by Ground Support Equipment (GSE). Information received by this path has yet to be defined.

*This is not a normal operating mode.

B4.0 SM PROCESSES

This section describes the processes performed by SM. The processes identified are those necessary to fulfull the intended purposes of SM. It is recognized that implementation will demand further processes such as table management, display management and general software "housekeeping."

B4.1 Fault Detection and Annunciation (FDA)

FDA will detect that a parameter value is outside a defined tolerance, notify the crew that such a condition exsists and provide information about the location of the problem identified. To accomplish this, FDA has been partitioned into four further processes: Precondition Steering, Limit Sensing, False Alarm Avoidance and Fault Annunciation. Each of these is described below. FDA nominally operates in continuous time.

Operationally, when any parameter is determined to be out-of-limits, a flag is also set by FDA indicating the sense of the condition. This flag is used for display. An alarm is not sounded, however, until this condition exists for a predetermined number of consecutive samples of the parameter. These consecutive retrys constitute data smoothing. When the persistency has been tested, an alarm is sounded. At any given time, neither are the parameters necessarily limit checked nor are the limit values necessarily the same as the previous check. This is due to the various operational modes of the equipment as well as their changing configuration throughout the mission. The role of determining proper parameter conditions is that of Precondition Steering.

B4.1.1 Precondition Steering

Precondition Steering is a parallel FDA control process. It does not process parameters to be limit-checked. Rather, it controls which parameters are to be checked and the values of the limits used (based on a predefined, discrete repertoire). Precondition Steering uses (a) the value of the vehicle configuration parameters, (b) some selected, associated parameter values and (c) the status of crew-entered FDA controls (see Section B6.0) to effect this control, based principally on combinatorial logic.

B4.1.2 Limit Sensing

Based on conditions determined by Precondition Steering, Limit Sensing will determine for each parameter, in turn, whether the value of that parameter is within a prespecified value. If the parameter is out of limits, this condition is transmitted to False Alarm Avoidance and a flag

indicating the sense of the condition is generated. The flag is made available to Fault Annunciation and Subsystem Measurement Management (Section B4.2).

B4.1.3 False Alarm Avoidance

This process falls between Limit Sensing and Fault Annunciation. It effects a smoothing on the out-of-limits indications determined by Limit Sensing. The intent of the process is to avoid alarms for transient out-of-limit conditions and to avoid repeated alarms for the same out-of-limit condition. The smoothing is effected as follows:

a. Transient Avoidance

   An out-of-limit condition must exist for "n" consecutive Limit Sensing tests before a signal is relayed to Fault Annunciation. If the run is interrupted at all by an in-limit condition, the process begins anew at the first out-of-limit indication.

b. Repeated Alarm Avoidance

   Once a parameter out-of-limit signal has been sent to Fault Annunciation (as determined by the test in (a) above), this signal cannot again be sent (for the same parameter) until that parameter has transitioned to in-limit and remained there for "n" consecutive tests. Once this condition is met, the process in (a) is repeated.

The smoothing parameter "n" indicated above is parameter peculiar and is contained within the FDA data base. As implied by the notation, the consecutive counts for transient and repeated alarm avoidance are the same.

B4.1.4 Fault Annunciation

FDA is responsible for detecting two classes of conditions: a Caution and Warning (C&W) class and an alert class. As the name implies, out-limit conditions existing in the C&W class are more critical than similar conditions in the alert class. FDA serves as a software backup to a dedicated electronic (hardware) C&W system in that class. It is, however, the only indicator for the alert class of out-limit conditions.

When an out-limit condition is determined (by False Alarm Avoidance) for either class, a message is generated for the Fault Summary Display (see Section B6.0) identifying the parameter and other pertinent data. In addition, depending on the class, the following occurs:

- C&W Class

  A decidated FDA, C&W Light is illuminated (This in turn causes the Master C&W alarm to sound.)

- Alert Class

  A decidated FDA Alert Light is illuminated and an audible tone is generated.

The audible alert alarm de-energizes automatically. The indicators are extinguished by an ACKNOWLEDGE COMMAND initiated by the crew on the keyboard.

Fault Annunciation increments a cumulative counter each time a Fault Summary message is generated.

B4.2 Subsystem Measurement Management (SMM)

This process allows the crew to assess vehicle performance and health based on results of Limit Sensing and Fault Annunciation as well as values of parameters and configuration indicators which it fetches. The process contains the operations necessary for the crew to compose an SMM or Parametric Display. Composition consists of parameter identification, suppression options, units options, number base options and narrative/ graphics options. Two operations of significance to this process are engineering unit conversion and number base conversion.

Each parametric display can be unique (since it is defined or composed) and two or more such displays can exist on different CRT's.

B4.3 Subsystem Configuration Monitoring (SCM)

This process allows the crew to verify the vehicle configuration at predetermined points in the mission. Operationally, the current vehicle configuration, reflected by configuration indicators (switch scan among others) and selected parameter values, are compared against a stored configuration for this particular checkpoint in the mission. Exceptions to the check are displayed in English language on the SCM Discrepancy Display. SCM is thus an automated crew checklist.

Each callup of SCM causes a single comparison pass to be made. Repeated comparisons (to verify results) are achieved by repeated callups.

B4.4 Consumables Management (CM)

The purpose of this process is to provide the crew aid in establishing and evaluating status of consumables and their relationship to mission continuation. A graphic CRT display is defined for each type consumable. This display

plots, as a function of mission elapsed time, nominal consumption, actual consumption and minimum levels for safe return.

B4.5 Data Recording Management

The purpose of this process is to preserve significant data for use by Ground Operations in Orbiter turnaround maintenance. The data are stored on the Maintenance Recorder which is dumped over the GSE umbilical on return. Two modes of control exist. The first mode centers about FDA-detected faults. When a faulty parameter is detected, the process causes the most previous two minutes (nominally) of data to be dumped from the loop recorder to the maintenance recorder and also causes the maintenance recorder to record the next two minutes (nominally) of data. This provides a data window centered about the occurrence of the fault.

The second involves continuous recording during critical flight periods. During dynamic flight (e.g., boost, change orbit, reentry) the process commands the maintenance recorder to run continuously. Transfer of data from the loop recorder is unnecessary.

The data recorded is determined by the current format residing in DACBU.

B4.6 Telemetry Format Selection (TFS)

This process selects the 128 kb/s and 64 kb/s formats used by the DACBU for telemetry downlink and onboard data recording as well as the format used in support of GSE over the umbilical. A change of telemetry format is accomplished either automatically by this process or manually through this function by a crew interface option. In the automatic mode, the selected format is a programmable function of mission elapsed time, GMT or mission phase. When the proper time condition is met, the process automatically initiates a reload of the DACBU RAM used to store telemetry format.

In the manual mode, the format change is controlled by the crew using the keyboard and a display designed for this purpose. The process automatically verifies the data transfer from mass memory as well as the transfer to DACBU RAM. The display is used to apprise the crew of the process events and check results.

B4.7 Payload Support

This process provides the same SM functions provided the vehicle, i.e., those in Sections B4.1 through B4.5. In addition, it provides the capability to generate and transmit onboard commands/data to the payload.

B4.8 Ground Support

This process exists only during Orbiter umbilical connection to GSE. It has yet to be defined but generally works under control of, and in connection with, Ground Support to:

- Aid in LRU fault isolation
- Provide data acquisition format selection (see Section B4.6) for GSE
- Control subsystem configuration via command decoders
- Provide computational assistance to GSE using programming unique to this application

B5.0 SM OUTPUTS

This section identifies SM output entities and the information moving from SM into these entities. Five entities are defined: crew, DACBU, Recorders, Payload and Umbilical Downlink.

B5.1 Crew Outputs

SM communicates to the crew by the CRT and two panel indicator. The following information types are related by CRT.

- All parameter values (payload and vehicle) plus associated annotation to identify them. Parameters are expressed in Engineering Units and number base is an option in some cases.
- Parameter out-limit, out-scale and questionable results indications. The former indicate sense of condition and the latter are based on detected data transfer errors. The limit indications are unsmoothed.
- English language expressions of exceptions to a configuration comparison (from SCM).
- Time history of consumables consumption plus nominal, predicted consumption and safe return quantities.
- List of parameters which have been declared faulty by Fault Annunciation.
- Control indications for manual telemetry format selection (load).
- Control indications for various SM performance options and initialization initiated by the crew.
- SM software status which portrays the current condition of the processes and stored program numerical values (tables, data base, etc.).

Panel Indications to the crew are as follows:

- C&W dedicated illuminated indicator which is energized by Fault Annunciation.
- Alert dedicated illuminated indicator which is energized by Fault Annunciation. This indication is accompanied by an audible alarm.

B5.2 DACBU Outputs

SM provides three types of information to the DACBU: telemetry format load, downlist and table dumps.

- Telemetry Format Load

  Loads DACBU RAM(s) with new formatting routines fetched from Mass Memory.

- Downlist

  The downlist provided is interstaced into the DACBU telemetry stream. It constitutes SM's contribution to that stream. Information contained in the downlist is as follows:

  - Fault summary data
  - State of FDA on/off control
  - State of SM discrete outputs (indicators, recorder control)
  - Command verification results
  - Returned (echoed) commands for ground verification
  - SCM discrepancies, when SCM active
  - Vehicle/mission ID (keyboard entry)
  - Selected Data Processing System data, e.g., BITE results.

- SM Table Dumps

  On command, any table in SM can be dumped to DACBU for inclusion into telemetry stream.

B5.3 Recorder Outputs

SM provides an output to turn on the Maintenance Recorder and a separate output to cause the Loop Recorder to dump onto the Maintenance Recorder.

B5.4 Payload Outputs

SM has the capability to send commands/data to payloads from crew-composed messages.

B5.5 Umbilical Downlink

Information transferred through this entity has yet to be defined but will generally consist of LRU isolation data, command verification, BITE results, and selected parameter values.

B6.0 CREW INTERFACE AND OPERATIONS

This section outlines crew interface with SM and indicates the more significant crew responses. From Sections B3.1 and B5.1 (SM/Crew inputs/outputs), it is obvious that the vast majority of crew action involves the CRT and keyboard. Consequently, this section will address crew interaction with these two mechanisms. The discussion is centered about available crew displays which have been divided into two categories: operations displays and SM control displays. The former are used to present SM operational information to the crew while the latter are used to assess control and/or modify the SM processes and stored values, e.g., change smoothing constants. Each display must be called from the keyboard; no displays are forced. Data contained in an active display will, however, be automatically updated.

Each CRT has a scratch pad line and a message line. The former is used to display keyboard inputs or composition. The latter is used to deliver messages to the crew.

To place the SM displays in perspective, GN&C has its own set of display pages. SM displays, then, form a subset of the entire Orbiter display complement.

B6.1 Operations Displays

These displays represent output of the SM processes identified in Section B4.0.

B6.1.1 Fault Summary Page

This display is a pushdown list of the most recent 20 faults as determined by False Alarm Avoidance. This display is called by a dedicated key on the keyboard. For each fault, the following is identified.

- Parameter out-limit occurrence time.
- Alphanumeric location/description of parameter.
- CRT page number on which the details of the parameter can be observed using SMM parametric displays.
- For C&W parameters, parameter value and out-limit sense indicator.

B6.1.2 Configuration Monitoring Discrepancy Display

During a configuration verification, which is initiated by keyboard and performed by SCM, the crew enters the appropriate configuration list ID into the keyboard. This list ID determines the configuration to which the actual configuration is to be compared. Exceptions to the configuration list are displayed in English language on the Discrepancy Display.

B6.1.3 Consumables Management Displays

The crew will call up CM via keyboard and then indicate the consumable of interest. CM will then plot actual use history, nominal usage and safe return values (see Section B4.4).

B6.1.4 SMM Parametric Displays

This display is driven by subsystem Measurement Management and the displayed data status indications are those determined by Limit Sensing, i.e., they are unsmoothed.

The crew will call up SMM via keyboard and then indicate the page number on which the desired set of parameters is contained. This display will nominally contain the parameter name, ID, value, out-limit sense indicator and scale limit indicators. It is, however, very flexible and each display is defined by the keyboard.

B6.1.5 Telemetry Format Load Display (Manual)

To change the commutation format of the on-line DACBU, the crew, by panel switch, causes the DACBU to switch to its "Hard" format. The crew then calls up this display by keyboard and then identifies the desired format also by keyboard. The display then indicates the currently selected format ID as well as results of the data transfer error checks (see Section B4.6). On completion, the crew places the on-line DACBU back to the programmable RAM to resume operation using the new format.

B6.2 SM Control Displays

There are six of these displays and the total effect is that of allowing the crew to alter virtually everything in SM.

B6.2.1 Initialization Display

SM initialization is controlled by the keyboard and the purpose of this display is to monitor and indicate the status/progress of initialization. The sequence of initialization is described below and this display presumably plays an interactive role as manually initiated retry and skip options are available.

1. Load SM programs

2. Initiate data acquisition and check validity of:

    a. Payload MDM BITE
    b. DACBU BITE
    c. OI MDM BITE

3. Automatically turn on FDA

B6.2.2 Table Maintenance Display

This display is called by the keyboard using the parameter ID. While there is but a single page to this display, a unique display exists for each parameter in the vehicle as well as those checked on the payload.

This display contains all data peculiar to the identified parameter. The purpose of the display is to give the crew the capability to change, for every parameter, limit values and noise filter (smoothing) constants. The capability also exists to override the actual value read (from DACBU) for a parameter by a constant which will be used by all future SM processing. Also, the crew has the capability to inhibit annunciation of a parameter.

Changes are made by keyboard entry. The process consists of first entering the item number of the data item to be changed. (A change of limit values has a unique item number, override parameter has another, unique item number, etc.) The new value/instruction is then keyed in.

B6.2.3 Constant Change Display

The crew has the capability to change any computational constant used by SM. This display is used to aid in effecting this change. The display is called by keyboard. The constant to be changed is then placed on the display by keying its ID number. The constant value plus its ID are then displayed. To change the value, the item number of "change constant" is keyed in followed by the new value.

This display is also used to enter/change the vehicle/mission identifiers used in the telemetry format.

B6.2.4 FDA Controls Display

The crew has the capability to turn FDA ON or OFF, inhibit the alert tone for FDA alerts and to change the persistency with which a fault message appears on the Fault Summary page. This latter control is called interlock time and is discussed more fully below. It is, in effect, another smoother cascaded to the existing False Alarm Avoidance.

To aid in these changes, the FDA controls display is called by the keyboard. This display contains the status of FDA on/off, alert tone on/off and the value of interlock time. Changes are once again made by keying in item numbers.

The purpose of the FDA interlock is to prevent (further) the repetition of a faulty parameter indication

appearing on the Fault Summary page (when this page is displayed). It works as follows: when a faulty parameter message has been generated by Fault Annunciation, it will not reappear on the Fault Summary page if it has already been displayed there within the interlock time.

B6.2.5 Configuration Monitoring Comparison Changes

The crew has the capability to change the state or limits checked for each parameter in the SCM comparison. This display is called via keyboard to assist in this change. The display is then filled by keyboard entry of the configuration list ID, the item number of parameter ID and the ID of the parameter to be changed. These are then displayed and changes to parameter values can then be made.

B6.2.6 Data Controls Display

The crew has the capability to control the SM data output to DACBU (downlist). This display is used to assist in this control. The display is called by keyboard and changes are made by item number in the same way as the other displays. The downlist data options are identified in Section B5.2.

APPENDIX C

ORIGINAL SM CONCEPT
A PERFORMANCE SPECIFICATION

Preface

This appendix contains the System description portion of Rockwell International Document SD-72-SH-0103-8, Vol. 5-8, Requirements/ Definition Document, Performance Monitor dated January 18, 1974.

The designation, Performance Monitor appears throughout the document. This designation is synonymous with System Management, which is the newer designation.

The restriction that SM reside in computers 4 or 5 is no longer true. SM may reside in any of the five computers, and under current design, the C&W portion of FDA resides in all four C&W computers.

The PMS annunciator electronics unit referred to in Sections 5.2.1.1 and 5.2.2.1 no longer exists.

5.0 SUBSYSTEM DEFINITION

5.1 Introduction

The Performance Monitor Subsystem (PMS) provides an onboard capability for vehicle subsystem management during both inflight and ground turnaround operations. The primary PMS functions include: providing information to the flight crew concerning vehicle subsystem health, performance, and configuration status; recording the telemetry format control to accumulate a subsystem performance record of each flight; support for inflight consumables management; payload support; cooperative support with the ground system for turnaround maintenance and checkout. The secondary PMS functions include: mission profile storage and retrieval; performance evaluation and trend analysis; contingency planning aid.

The PMS primary functions will be developed and implemented for the first operational flight of the Shuttle System. The secondary functions may be developed and utilized subsequent to the first operational flight to the extent that they can be accommodated within the basic growth capability of the Data Processing and Software (DPS) subsystem.

The major functions provided by PMS are as follows:

Primary Functions

- Fault Detection and Annunciation
- Subsystem Measurement Management
- Subsystem Configuration Management
- Consumables Management
- Data Recording Management
- Telemetry Format Selection
- Payload Support

Secondary Functions

- Mission Profile Storage and Retrieval
- Performance Evaluation and Trend Analysis

- Contingency Planning Aid

These functions will be accomplished through the use of hardware provided by other subsystems (primarily computers and data processing, displays and controls, and instrumentation). The unique portion of PMS is the software loaded into computers 4 and 5 and the mass memories.

The inflight PMS functions operate in two basic modes: continuous operation and on-demand operation. The Fault Detection and Annunciation function, Data Recording Management function, Telemetry Format Selection function, and corresponding portions of the Payload Support function operate continuously. All other PMS functions operate only on demand.

With the exception of the Subsystem Measurement Management function, the software used inflight is not expected to be directly usable for ground turnaround operations. Separate software loads will be required for specific maintenance and/or ground checkout if these operations are accomplished through the use of onboard computers (as opposed to ground computers).

The PMS is one failure tolerant with respect to the hardware used to accomplish its functions.

5.2 Subsystem Description

5.2.1 Functional Description

5.2.1.1 Mission Operations

The PMS primary and secondary functions are accomplished in Avionics Computers 4 or 5 through software programs loaded into these computers specifically for this purpose. Measurement data required to perform PMS functions are acquired from the Data Acquisition Control and Buffer Unit (DACBU). Output data, as a result of PMS processing, are provided to the CRT's, the PMS annunciator electronics unit, the DACBU, and the tape recorders. Control of PMS functions is provided through the keyboards.

The mass memories are used to redundantly store the PMS programs that normally reside in Computer 4/5. They are also used to store subprograms and CRT display formats that are used only when requested by the crew.

At any given time, the PMS program is operating in either Computer 4 or 5. In the event of a failure, the computer not in use for PMS is manually switched in to replace the failed computer. Any program reloads from mass memory (if required) are also accomplished manually.

Provisions are included for control of the command decoders during flight operations, whereby the normal connection to the ground system is inhibited and Computer 4/5 control is enabled. This allows closed loop control of vehicle subsystems to the extent that parallel switch control is implemented; however, this control link will not be normally used for manned flights.

5.2.1.2 Ground Operations

Most of the hardware used for inflight PMS can also be used for checkout during ground turnaround operations. The inflight software is not expected to be directly applicable since it is tailored to vehicle subsystem inflight operational conditions that will not be duplicated on the ground. There are two extremes that can be accommodated within the PMS concept: (1) ground checkout can be accomplished entirely by the ground system computers, displays, etc., with measurement data provided from the DACBU and remote control exercised through the command decoders; (2) the onboard computers can be loaded with the ground checkout programs (through the command decoders) and checkout operations conducted from the cockpit using onboard CRT's for display and keyboards for control. The amount of automatic closed loop checkout that can be performed in either case will be limited by the extent to which parallel switch control is implemented, and by the interaction required with servicing type GSE (power, pressurant sources, etc.).

5.2.2 Major Component/Subsystem Description

5.2.2.1 Fault Detection & Annunciation (FDA)

The FDA function is implemented entirely in software. The purpose of the function is to detect subsystem failures at the functional path level (level at which corrective action can be taken inflight) and inform the crew that the failure has occurred. Measurement data are acquired on a cyclic basis from the DACBU and stored in Computer 4/5 main memory. From these data, the subsystem configurations for evaluation are determined and limits, tolerances, and calculations required are enabled. Each subsystem is then evaluated to determine if faults exist and the results are stored in discrete form. The results are then filtered for n consecutive occurrences of any given fault, where $n \geq 2$.

If the n consecutive fault occurrences criterion is met, an alarm indicator flag is stored for later processing. If n consecutive faults do not occur the filter flag is reset. Alarm inhibits for any given parameter (controlled by the crew thru the keyboards or dedicated switches) are then examined and the output annunciator format is determined. The output format consists of the following: a PMS Master Alarm signal that is enabled whenever any parameter is detected

out of limits (except when the parameter has been inhibited); a backup Caution & Warning (C&W) Master Alarm signal that is enabled whenever the failed parameter is a member of the C&W subset; individual failure indications to the subsystem level; a control signal for transferring prefailure data from the loop to the maintenance recorder and to start the maintenance recorder.

FDA provides a backup capability for the hardwired C&W subsystem to alert the crew to any detected hazardous or potentially hazardous conditions which require attention. Backup C&W limits are identical to primary C&W limits and are operative at all times when FDA is functioning.

5.2.2.2 Subsystem Measurement Management (SMM)

The SMM function is implemented entirely in software. The purpose of the function is to provide the crew with access to data from which the degree of a problem (detected by FDA) can be assessed. PMS CRT display formats are never forced; therefore, the crew must request all display formats. The crewman enters the desired display on a keyboard. Computer 4/5 then generates the appropriate read command to the mass memory. The static information in the display format is transferred to the Display Electronics Unit (DEU) and the variable portion plus appropriate linkages are transferred to the computer (straight "text" types of CRT formats may not require direct computer involvement since they have no variable portions).

When parametric data is required by the CRT format it will be obtained from the same set used for FDA in normalized form. It is then converted to engineering unit thru a scaling and/or biasing process. FDA results are obtained and used for generating out of tolerance indications adjacent to the displayed parameter and are bidirectional (hi, low). The output data is then formatted for transfer to the DEU/CRT equipment. The process described above primarily applies to alpha/numeric page type CRT formats when the result is typically a listing of related parameters by subsystem. Other summary type pages are also provided, such as failure summaries by functional path resulting from the FDA process. There are additional CRT formats such as graphics for consumables management and configuration management displays that are described within other PMS functions.

5.2.2.3 Subsystem Configuration Management (SCM)

The SCM function is implemented entirely in software. The purpose of the function is to provide an aid to the crewman in determining the correctness of the vehicle subsystems configuration at specific, predetermined points in a mission. The SCM function is operative only when called by the crewman thru the keyboards. It operates

continuously when called and is terminated when any other subroutine/display format is subsequently called by the crewman. A request for the SCM function is recognized by the computer and the required information is accessed from mass memory in a manner analogous to that performed by the SMM function. The parametric and discrete data required are obtained from the same set used for FDA. These actual status data (including both switch scan and subsystem function status) are then compared with a preprogrammed desired status map. Exceptions are then determined and stored as comparison results. The results are then scanned and English language statements defining these exceptions plus corrective action required are formulated. This information in then formatted for CRT display and transmitted to the DEU/CRT. Optional provisions are included for linking the results of an SCM subroutine with the command decoders whereby differences in desired versus actual subsystem configurations could be automatically corrected or altered within the limits of the parallel switch control provisions. This option will not be used during development or early operational manned flights.

5.2.2.4 Consumables Management (CM)

The CM function is accomplished entirely in software. The purpose of the function is to provide a crew aid in establishing and evaluating the status of consumables and their relationship to mission continuation. CM is accomplished for ECLSS, APU's, FUEL Cells, RCS, OMS, and EVA tankages as a minimum. The final result for each type of consumable is a graphic CRT display depicting nominal predicted consumption, actual consumption, and minimum levels for safe return (as a function of mission elapsed time). The CM function is operative only when called by the crewman thru the keyboard. A request for the CM function is recognized by the computer and required information is accessed from mass memory in a manner analogous to that performed by the SMM function. The parametric data required are obtained from the same set used for FDA. The current status data, consisting essentially of quantities remaining, are processed to determine total quantities (multiple tanks) remaining as a function of mission elapsed time. This data point is added to a set of data points stored in main memory that represents previous usage calculations. All of the previously stored data points plus the current point are then used to formulate the actual usage curve. These data are then formatted and transmitted to the DEU/CRT. The variable curve is then overlayed on the fixed portion of the display to provide the crew with a graphic representation of consumables usage.

5.2.2.5 Data Recording Management (DRM)

The DRM function is implemented in software. The purpose of this function is twofold: (1) to record information

prior to, during, and subsequent to a failure occurrence detected by the FDA function; (2) to initiate "snapshot" recording of significant subsystem data during dynamic periods of flight. The DRM function generates a control signal to transfer temporarily stored information (on the loop recorder) to permanent storage (on the maintenance recorder). The same control signal is also used to start the maintenance recorder. This action occurs whenever the FDA function detects an out of tolerance condition. The run time for each detected failure is programmable but will be constant at a total of 5 minutes (centered about the detected failure) for initial flights.

The "snapshot" data will be recorded during all dynamic flight phases such as ascent, rendezvous, reentry, atmospheric flight, etc. In this mode the maintenance recorder will run continuously with no transfer required from the loop recorder.

5.2.2.6 Telemetry Format Selection (TFS)

The TFS function is implemented in software. The purpose of this function is to control the format of data required for RF telemetry, onboard data recording, and the ground system (during turnaround operations). The selected format(s) is a programmable function of mission elapsed time, GMT, mission phase or direct crew control. At least 4 unique formats will be available for selection. All of the data required for onboard processing by PMS will be available in each format.

5.2.2.7 Payload Support (PS)

The PS function is implemented in software. The purpose of the function is to provide the same generic PMS functions that are provided for Shuttle vehicle subsystems, including backup C&W. The PS function also includes a capability to generate and transmit commands/data to the payload at a rate of 2 kbps when selected and initiated by the crewman thru the keyboard.