SIMULATOR VERIFICATION TECHNIQUES STUDY

TASK REPORT 3 (TR-3)

INTEGRATED SIMULATOR SELF TEST SYSTEM CONCEPTS

20 SEPTEMBER 1974

MDC E1149

G. MONTOYA

T. H. WENGLINSKI

PREPARED FOR NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

JOHNSON SPACE CENTER
HOUSTON, TEXAS 77058

SUBMITTED UNDER CONTRACT NAS 9-13657

MCDONNELL DOUGLAS ASTRONAUTICS COMPANY - EAST
HOUSTON, TEXAS 77058 (713 488-5660)

TABLE OF CONTENTS

LIST OF TABLES

LIST OF PAGES

PRECEDING PAGE BLANK NOT FILMED

SECTION 1

SUMMARY

This document is Task Report 3, TR-3, of the Simulation Verification Techniques Study, Contract NAS 9-13657. Specifically, this report presents system level self test concepts and requirements for simulator Self Test.

Testing techniques for training simulators are discussed in Section 3. Three categories of testing are covered: Readiness Tests, Fault Isolation Tests, and Incipient Fault Detection Testing. Readiness Tests are generally end to end functional tests. Fault Isolation Tests are performed at a lower level of hardware detail and therefore point out the faulty LRU when a failure occurs. Various techniques for Incipient Fault Detection are discussed in light of their applicability to various simulator subsystems. Incipient Fault Detection Testing was found to be most useful in decreasing unscheduled maintenance of the Motion Base System, Visual Simulation, both video and model components, and servo driven instruments. Readiness Tests are intended to be performed on a daily basis.

Self Test software requirements are discussed in Section 4. The proposed software structure is based on an executive that controls test operations, including test software loads for various subsystem tests, checks test results to determine the next test step, and assembles test results for output to the operator, to hardcopy devices for permanent records or to mass storage files for updating of the incipient fault detection data base. The executive is a batch program that allows the subsystems test software to establish data rates and sampling rates. Accurate timing data is obtained by time tagging data words at the time they are sampled.

The total hardware requirements for implementation of the Self Test System are listed in Section 5. Largest test hardware requirements in terms of cost is in the Visual Simulation CCTV subsystem. In terms of numbers, the instrumentation for the various controls and displays on the Crew Station and IOS are most demanding.

Section 6 discusses changes and modifications, primarily in the vehicle and payload complement, and the impact that these changes will have on Simulator Configuration, Self Test System, and Data Base make up. Changes discussed here are only those that would take place after installation and acceptance of the simulator.

Relative cost data of the Self Test components to the basic simulator subsystems and to each other are presented in Section 7. Self Test of Visual Simulation CCTV subsystem and the Crew Station and IOS were found to represent the most significant portions of the total Self Test System cost. Conclusions and recommendations, as they pertain to the overall Self Test System Requirements and implementation are documented in Section 8.

SECTION 2

INTRODUCTION

This report presents software and hardware requirements for implementing hardware self test as defined during the Simulation Verification Techniques Study being conducted for NASA's Johnson Space Center under Contract NAS9-13657. This study is being performed by McDonnell Douglas Astronautics Company - East at its Houston Operations facility. Keith L. Jordan, Simulation Development Branch, FSD, is Technical Monitor of the contract for the NASA.

The Simulation Verification Techniques Study is one of a number of studies being conducted by NASA-JSC in support of the development of training and procedures-development simulators for the Space Shuttle Program. The other studies consist of the following: Shuttle Vehicle Simulation Requirements, NAS9-12836; Space Shuttle Visual Simulation System Design Study, NAS9-12651, performed by McDonnell Douglas Electronics Company; Development of Simulation Computer Complex, NAS9-12882; and Crew Procedures Development Techniques, NAS9-13660; both of the latter were performed by McDonnell Douglas Astronautics Company - East at its Houston Operations facility.

The present study is concerned with the development of self test techniques for simulation hardware and the validation of simulation performance. A preliminary report for the Hardware Verification Task has already been published. The present report presents the results of an analysis of the requirements of an integrated simulator self test system. This system consists of the additional hardware and software required for a training simulator in order to provide maximum reasonable self test capability. The results in this report will be incorporated in the final version of the "Simulation Self Test Hardware Design and Techniques Report" to be published shortly.

SECTION 3

INTEGRATED SYSTEM TEST DESCRIPTION

Testing of the simulator as an integrated system involves the sequential, and sometimes concurrent verification of simulator subsystems and functions. There are three hierarchical classifications of tests determined by the level of detail to which the test is performed and the level of confidence obtained after successful completion of the test. These classifications are Readiness Tests, Fault Isolation Tests, and Incipient Fault Detection Tests.

3.1 READINESS TESTS

Readiness Tests are used to verify the readiness of simulator subsystems to perform according to design and operational requirements. Generally these tests check each of the simulator subsystems and, whenever possible, use already verified subsystems to check other subsystems of the simulator. Ideally readiness tests are end to end tests. They are primarily concerned with testing a complete string of hardware.

The sequence of subsystem testing developed in the course of this study is based on the "building block" approach of establishing the operational readiness of a system element before that element can be used to test other portions of the system under test, in this case the simulator. Figure 3.1-1 shows the sequential and parallel arrangement for test execution during the Readiness Test. Individual tests shown at the same horizontal level can be executed simultaneously by the various relatively independent processors in the simulator. Checkout of the HOST computer is required prior to commencing with simulator checkout.

Each of the vertical strings indicate dependence of tests, lower on the chart, on successful completion of previous tests. For example, the simulator interface minicomputer must pass successfully its readiness self test prior to executing the routines for DCE self test. Failure of the minicomputer to pass this test would return information to the HOST regarding the point or function at which the test failed. On the other hand, a failure in the DCE would prevent

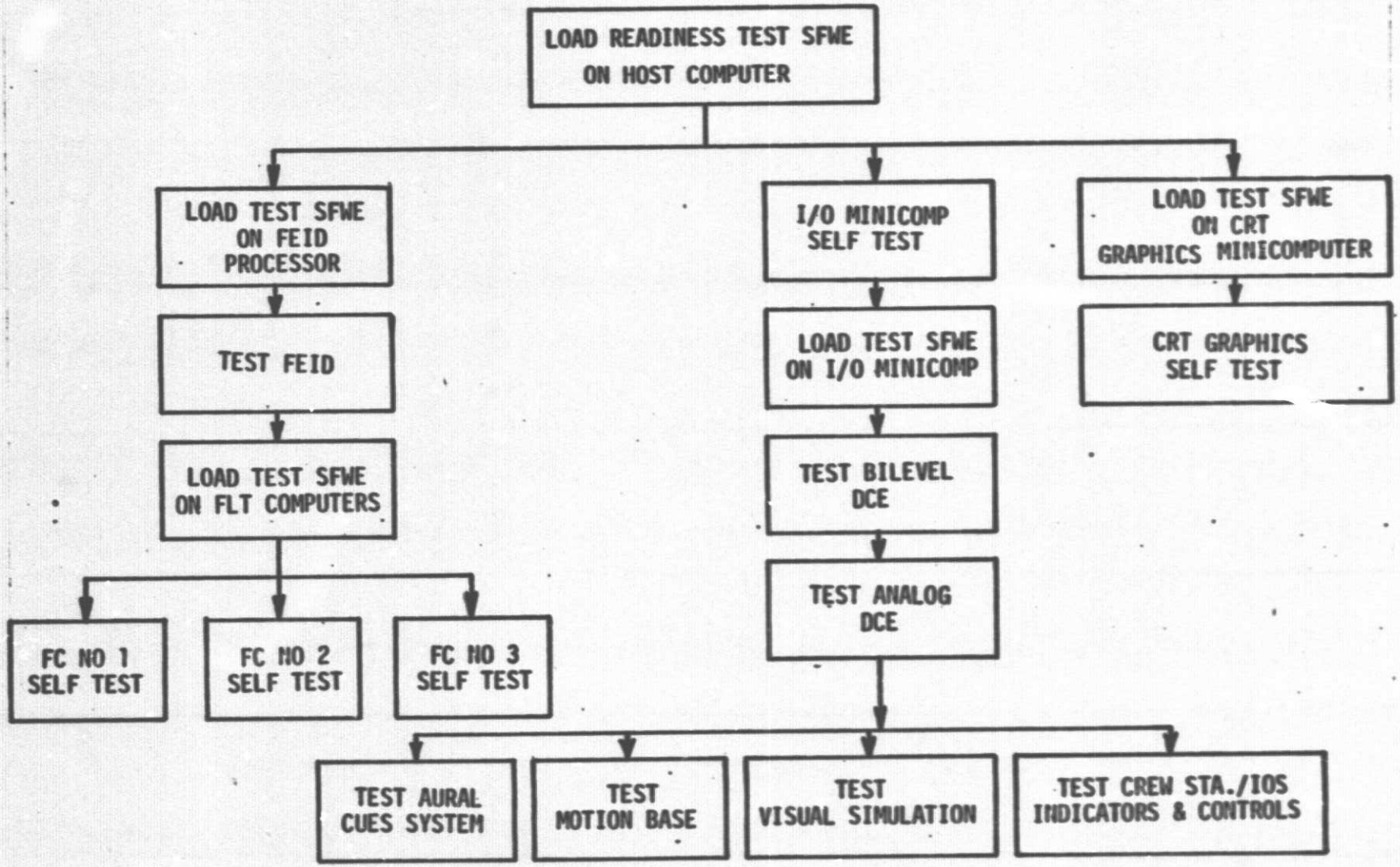

FIGURE 3.1-1 READINESS TEST ELEMENTS AND EXECUTION SEQUENCE

Crew Station or Visual Simulation system checkout; and it would be up to HOST computer software to determine what automatic fault isolation tests should be run on the DCE.

The self test software is initially in mass storage elements of the HOST computer complex, and is loaded into the system under control of and through the interface facilities of the HOST computer. At the conclusion of the load, each processor returns to the HOST a message indicating that the load has been completed, and some other information such as a checksum which the HOST can then use to verify that a successful, accurate load has been accomplished.

After the loading process is completed, and the HOST has verified that all went well, each processor is commanded by the HOST to execute the subsystem Self Test programs.

The simulator processing elements that receive this self test software load are the Flight Equipment Interface Device processors; the simulator interface for the DCE, crew station, visuals, and motion base; and the IOS Graphics minicomputer. Testing of the Flight Computers is not performed until after the FEID is thoroughly checked out; therefore, loading of the Flight Computer self test software does not take place until after completion of FEID Self Test. This particular sequence is necessary because FC Self Test requires correct performance of FEID hardware not only during the load, but also in checking FC capability to communicate with avionic components external to the FC and simulated by the FEID.

Completion of each level of Readiness Tests results in control being returned to the HOST, which then logs any subsystem operational data gathered during the test, issues appropriate messages to the Test Operator, and determines the next logical step in the test sequence.

The following sections discuss briefly the nature and extent of the Readiness test for each of the major simulator subsystems.

3.1.1 FEID Readiness Tests

The major elements of the FEID tested during the Readiness Test are the processors and the various interfaces. The types of tests performed on these elements, as well as the Central Buffer Storage, are such that thorough and progressive checks are made on each portion of the FEID. Failure to pass one of those tests immediately points to a fault in the LRU being tested, and therefore these tests, by their thoroughness and level of detail, obviate the need for fault isolation tests.

At the completion of FEID Readiness Test, the HOST is informed of which LRU's, if any, have failed and require repair or replacement action. The HOST can then report this information to the test operator, together with possible diagnostic information that would aid the maintenance personnel in repairing faults detected.

3.1.2 FC Readiness Tests

The Flight Computer elements, as is the case with the FEID, will be thoroughly checked during the Readiness Tests. The subsystem level self test analysis indicated that automatic testing of the FC's should rely on using vendor supplied diagnostic software. This software will be available, and will yield a higher level of confidence on readiness of the flight hardware than would be likely to be achieved by software developed especially for simulator application.

Use of test software supplied by the FC manufacturer will provide an adequate level of fault isolation. For the FC's, therefore, no additional fault isolation tests are recommended.

3.1.3 Simulator Interface Computer and IOS Graphics Minicomputer Readiness Tests

The simulator interface computer and the graphics computer are commercially available computer systems. These computers will be newly procured for the specific simulator applications. As part of that procurement, it will be necessary to assure that these computers have available adequate software for accomplishing functional checks of the basic computer elements. In implementing these functional tests, the faults which may be detected are immediately related to the LRU level.

Consequently no special fault isolation tests are required for these computers. The test capabilities required for the functional test software are discussed as part of the analyses of the ancillary computers for the simulator configurations considered for this study.

3.1.4 DCE Readiness Test

The DCE Readiness Test, as developed during the subsystem analysis portion of the study, consists primarily of switching output bilevel and analog channels on to input channels. Generation of predetermined outputs is followed by measuring corresponding inputs and then determining that input patterns, for bilevel DCE, or values, for analog DCE, agree with the outputs. This type of test adequately verifies the health status of each individual channel, and if a failure occurs, can point to or isolate the pair of channels, input and output, where a fault might exist.

3.1.5 Crew Station/IOS Readiness

The crew station contains a variety of display, control, and instrumentation components. These are both digital and analog devices, and within these categories there are significant differences in the physical, electrical and performance characteristics of the many components used. This variety prevents the use of functional performance testing of groups of components and requires that each individual meter or switch be tested individually to ascertain Readiness. This component by component test approach during the Readiness Test leads directly to fault isolation and identification of the faulty components without any additional Crew Station Fault Isolation Test.

3.1.6 Motion Base Readiness Tests

The motion base readiness testing consists of two basic elements. The first of these is the monitoring of sensor levels during power on and power off static checks. The second element is the analysis of dynamic test data to assess motion base dynamic performance adequacy. The dynamic test data may be obtained either from specially executed motions as part of the readiness test or it may be data accomulated on-line during the previous operational period.

The sensor data directly identifies faults in the systems since it monitors specific parameters such as hydraulic fluid level or pressure drop across a filter. There is no additional requirement for fault isolation with respect to these parameters.

However, the dynamic data obtained for a synergistic motion base can be checked for proper performance without any insight being obtained into the likely sources of out of specification performance for the actuator servo systems. Fault isolation analyses may be applied to obtain the desired insight.

3.1.7 Visual Simulation Readiness

End to end testing can be used successfully in establishing operational readiness of the Visual Simulation System. Enough information can be gathered by inserting a test signal at the CCTV camera and taking measurements at CRT grids to determine whether performance of the video equipment is acceptable to proceed with normal simulator operations. The servo systems that control probe, camera, and model motion can also be checked using functional end to end testing. As in the other subsystems, fault isolation testing is only initiated when a fault is detected during the Readiness Test. Failure data is furnished to the fault isolation software to determine which tests to run, and which strings of hardware to test during the fault isolation process.

3.2 FAULT ISOLATION TESTS

Fault isolation tests are performed following a failure during normal operation or after detecting a fault during the Readiness Tests, in order to localize the fault or source of failure to a line replaceable unit (LRU). This localization is required to permit prompt repair by replacing the failed LRU. Whereas Readiness Tests can be performed at the functional level (the ability to generate an audio tone with the Aural Cues System), the Fault Isolation Tests are performed at the hardware LRU level (verifying that the analog signal output from a DAC corresponds to the digital pattern input to it). Fault Isolation Tests, because they go to a lower level of hardware detail, are more exhaustive, take more time to run on any particular subsystem, and achieve a higher level of confidence on the health of the hardware than achieved by Readiness Tests.

There are subsystems that because of their characteristics, must be tested in detail in order to ascertain an acceptable level of readiness. These subsystems are not addressable by higher level, functional testing as part of the daily Readiness Test. These subsystems include the Flight Equipment Interface Device, FEID, the Flight Computers, the computers for the Simulator and graphics interfaces, and the Crew Station and IOS instrumentation, displays and controls. The readiness tests for these subsystems, as discussed in the previous section, effectively isolate faults to the LRU level required.

There are, however, several subsystems that are suitable for extended testing for fault isolation or in fact, require quite extensive additional testing in order to achieve identifacation of defective LRU's. These consist of the Data Conversion Equipment, the Motion Base, the Visual Simulation Subsystems, and the Aural Simulation subsystem. The Fault Isolation Test requirements for these subsystems are discussed in the following paragraphs.

3.2.1 DCE Fault Isolation Tests

At the end of the Readiness Test, the HOST is informed of the pairs of channels, if any, where faults were detected. The Test Operator is given this information and the option to terminate DCE testing or to command the HOST to initiate the DCE Fault Isolation Test. It is possible that the channels

that failed readiness testing are spare channels, not in use, and therefore no further fault isolation is required. On the other hand, if the channel is in use, the operator may patch around the faulty channel by making the necessary hardware and software data base changes to permit the use of alternate channels. This course of action may expedite activation of the simulator and obviate the need for fault isolation and maintenance operations, which would then be deferred to a more convenient, lower interference time.

If the Test Operator chooses to perform automatic fault isolation, he notifies the HOST and initiates loading and execution of the DCE Fault Isolation Tests. The objective of these tests is to determine which LRU in the DCE is faulty and therefore should be replaced or repaired.

The first step in the DCE fault isolation process is to determine whether the faulty channel in the suspected pair is the output or the input. This can be accomplished, as discussed in previously documented DCE Self Test analysis. If the fault is determined to be in an output bilevel channel, the fault isolation process continues to determine whether the faulty LRU is the digital output routing and address decoding logic, or the memory and drive circuitry. If the fault is in the bilevel input channel, then the isolation process will localize the fault in the input multiplexer and address decoder logic, or the bilevel signal receiver circuit. Problems in a pair of analog channels will cause the fault isolation testing procedure to determine whether the faulty LRU is the DAC or analog driver in the output, or the Buffer Amplifier or ADC in the input. The fault isolation process for both bilevel and analog channel LRU's has already been documented in the Self Test Design and Techniques analysis portion of the study.

3.2.2 Motion Base Fault Isolation

Although many of the components that may require periodic servicing can be measured by direct sensing, the isolation of faults based on degradation of hydraulic actuator system performance is a factor that may be amenable to analysis. Fault Isolation Tests for the motion base require consideration of the frequency response of the servo systems or servo loops of which the

hydraulic actuators are the major components. Definition of the frequency response enables comparison of breakpoint data with fault dictionary information which is established by simulation of the motion base actuator servo systems. Analysis of this data then permits identification of the particular components or likely components which are causing the degraded performance.

3.2.3 Visual Simulation Fault Isolation

In the Visual Simulation System there are two major types of hardware: one is the primarily electronic elements which make up the Closed Circuit Television system and include camera, camera control, video processing, display switching and the displays themselves. The other type is the electromechanical, servo controlled elements used to move and position the various components of the camera/model based simulation. This type includes the spherical earth models, the camera gantry and its driving system, and the probe pointing attitude control servo systems. The fault isolation approach for the CCTV components is based on a progressive test of succeeding LRU's in the video string until the failed LRU is switched in and unusual performance degradation is detected. On the other hand, fault isolation for the model drive components concentrates on analyzing that control string which exhibited unacceptable performance during the Readiness Test. Characteristic parameters of each LRU in that string are measured in order to ascertain operational status of each LRU and determine which is the one that has failed. The actual sequence of fault isolation tests for each string of hardware has already been documented in the Visual Simulation System Self Test Analysis part of the study.

3.3 INCIPIENT FAULT DETECTION

This section covers incipient fault detection testing techniques and their applicability to simulator hardware. First of all, a set of criteria to determine which components can profit from this third class of testing is discussed. Then various methods to perform incipient fault detection are presented with emphasis on the one selected, that of gathering historical operational data and looking for trends that point to eventual failure of a component. The applicability criteria were applied to all the simulator subsystems, and a set of relevant parameters, along with processing techniques were defined for each of the subsystems that could benefit from incipient fault detection. These parameters and techniques are documented for each subsystem at the end of this section.

3.3.1 Incipient Fault Detection Criteria

The primary objective of incipient fault detection testing is to identify potential failures before they occur. This objective becomes quite important in components that require extensive simulator downtime to repair or replace. For example, an electronic module can be replaced in seconds, and if it fails during simulator operation, only a few minutes would be lost from the training schedule. On the other hand, if a hydraulic fluid accumulator ruptures, it would take hours to replace, if a spare were available, and even longer to repair, reassemble the plumbing, and recharge the hydraulic system to replace the lost fluid. In the first case, the payoff from incipient fault detection implementation is measured in minutes saved when the weakening module is replaced during non operational time. In the latter case, incipient fault detection analysis can point to the sticky valve, the surging pump, or the clogged filter that could eventually result in rupture of the accumulator. This information could be used to replace or repair the faulty component during a more convenient maintenance shift, instead of having to face the situation described above that would cause the loss of one or several training shifts, following rupture of the accumulator.

This example illustrates the primary criterion used in determining which subsystems should be analyzed for possible application of incipient fault detection techniques: if an LRU in a system requires more than one hour for

replacement or repair, the operational effectiveness of the system can be increased by the use of techniques that will identify a potential failure of that LRU, so that replacement or refurbishment can be effected during scheduled maintenance time. This criterion would apply to devices that, although easily replaced, a spare unit is not normally kept on hand because of its high cost coupled with high reliability of the device.

The second criterion in determining applicability of incipient fault detection is the physical and functional nature of the LRU involved. Some components, notably solid state electronic circuit elements, do not exhibit degradation characteristics that can point to eventual failure. These devices usually fail as a result of a single overstress condition that causes fusion or breakage of the delicate semiconductor material, thereby changing the electronic characteristics of the unit. It is not possible to predict a failure in this type of circuitry, and even though it may be desirable to do so because of maintainability considerations, no known incipient fault detection technique can be effectively applied here.

On the other hand, there are LRU's, notably electromechanical devices, that do age and wear out either because of mechanical function or by chemical changes within the device. For example, an electrical motor may exhibit erratic starting characteristics. This condition may be an indication that the bearings or seals have worn down, or that capacitors in the starting circuitry have aged to the point that eventually a failure might occur that could require extensive overhaul of the motor. Early detection of the incipient fault permits scheduling of overhaul as part of normal maintenance operations. However, this early detection can only be effected on components that degrade or age to the point that may eventually result in a failure.

To summarize then, the two basic criteria for determining applicability of incipient fault detection techniques to a particular subsystem are:

1. Maintainability characteristics of the LRU must be such that repair or replacement time would exceed one hour.

2. Physical, functional and aging characteristics of the LRU are such that unit integrity degrades with time to an unacceptable or failure level, rather than failing catastrophically and unpredictably.

3.3.2 Incipient Fault Detection Technique

Various techniques for performing incipient fault detection were analyzed during Subtask 1.2 of the study, and are documented in more detail in the previously published Self Test Techniques Report. They are reviewed here for reference. The four techniques analyzed were:

1. Overstress Testing. The LRU is operated at the high end or beyond the high stress limit of the operational band. This overstress operation should be within the safety margin normally encountered in hardware design. If operation is faulty, it may be a symptom of possible future failure. Examples of this type of testing are frequency response tests at higher frequencies than required in normal operation, or testing of electrical components at higher voltage than the maximum specified for normal performance. The greatest disadvantage to this technique is that it may force a failure at a time when maintenance would impact simulator operational schedule. If properly scheduled, however, this technique could be a useful tool to detect weaknesses in some simulator components.

2. Marginal Testing. The LRU is operated at the low end or beyond the low stress limit of the operational band. This technique, like overstress testing can be used to detect degradations in components that may lead to a failure. Examples of marginal testing would be running the motion base with lower than normal hydraulic fluid system pressure, or slewing a camera gantry slowly enough to find sticky points along the range of travel that indicate a possible accumulation of dirt on the track, or flaws in the driving mechanism that can lead to a future failure.

3. Grey Area Performance Detection. The LRU is tested within its normal band of operation, but a warning is issued when performance drops below a predefined level. This technique assumes progressive performance degradation which becomes a failure below a predefined level. If a higher level is defined to create a warning or "grey" area, then performance within this area, although acceptable, is an indication that the unit has degraded to such a point that a failure will soon occur. Simulator operations personnel should then schedule maintenance of the LRU in order to preempt the actual occurance of the failure.

4. Rate of Degradation. Measure and record historical data on a parameter that indicates degradation of performance in a particular LRU, and use this data to compute rate of degradation. If this rate increases or changes markedly, it may point to an imminent failure of the unit. The amount of time during which historical data is recorded depends on the nature of the component and the slope of the normal or average degradation function for that type of component.

   If degradation takes place very fast, as in the case of a slightly ruptured hydraulic filter that through usage might enlarge the rupture and eventually break down, then there is no need to store data for months. Information gathered for the last five days may be enough to indicate that something has happened to the device, and that its performance is fast degrading and that soon a failure will occur. On the other hand, a slowly degrading component such as an electrolytic capacitor may require processing of data acquired for a period of weeks to indicate that the component has reached its "last leg" in the degradation function, and that soon it will fail totally.

Techniques number 3 and number 4 are the ones that offer the greatest value for simulator incipient fault detection testing. Their advantages lie in that no unusual operational conditions have to be created, and therefore, normal operation or readiness testing exercises can be used to gather the necessary data to detect incipient faults. Therefore, these techniques do not need additional test hardware or control software beyond that which is required to perform the Readiness Tests or Fault Isolation Tests.

3.3.3 Incipient Fault Detection Implementation

Using the Incipient Fault Detection Applicability Criteria in analyzing simulator subsystems, it was determined that incipient fault detection techniques would benefit the operational utility of the following subsystems:

- Hydraulic Motion Base System
- Visual Simulation Video Components
- Visual Simulation Model Drives and Controls,
- Servo Driven Instrumentation

These are the system elements that exhibit progressive degradation characteristics and would require the simulator to be out of service a substantial period while maintenance takes place. Table 3.3-1 summarizes the use of incipient fault detection techniques on each subsystem. Integration of these techniques is a part of recommended self test procedures.

TABLE 3.3-1 INCIPIENT FAULT DETECTION TECHNIQUES APPLICABILITY

SYSTEM ELEMENT	INCIPIENT FAULT DETECTION TECHNIQUES			
	Overstress	Marginal Testing	Gray Area Performance	Rate of Degradation
Hydraulic Motion Base System	X	X	X	X
Visual Simulation Video Components	X	X	X	
Visual Simulation Model Drives And Controls	X		X	X
Servo Driven Instrumentation	X		X	X

3.4 TEST SEQUENCING

The three types of testing discussed above have been integrated into a recommended simulator self test concept. This self test approach is based on several assumptions and on the above description of the basic nature of these tests when applied to specific simulator subsystems. The general test approach and test sequencing are most evident in the discussion of the Simulator Self Test Software Description presented in Section 4. In this section, we will review the rationale for test sequencing and distribution of test operations as it relates to the three types of testing involved.

The readiness tests are intended to be overall functional tests for the various simulator subsystems and as such should require only a limited amount of time to execute. It is assumed that the readiness tests would nominally be executed every day or at the beginning of each training shift. In the event that a failure or fault is detected by the Readiness Test, it may be necessary to execute a Fault Isolation Test in order to identify more precisely the LRU which is the source of the problem. As previously discussed, this is only necessary in the case of the DCE, the motion base, the visual system and the aural simulations.

If the fault isolation test is required, then a question arises as to whether it should be executed automatically or whether the operator should be required to initiate the additional test by further positive action. In design of the test software, Section 4, automatic execution is assumed since this establishes the necessary basic sequence elements. The introduction of pauses or overriding interrupt logic is a relatively minor and optional problem when the software is implemented.

Figure 3.4-1 illustrates the operational sequencing of Readiness, Fault Isolation, and Incipient Fault Detection Testing. The basic data for the Incipient Fault Detection Test can be obtained in most cases during the readiness test execution- as shown in this Figure. However, if a fault is detected during the readiness test, then the data obtained should not be entered into the Incipient Fault Detection data base. Tests for this contingency are shown in the software flows.

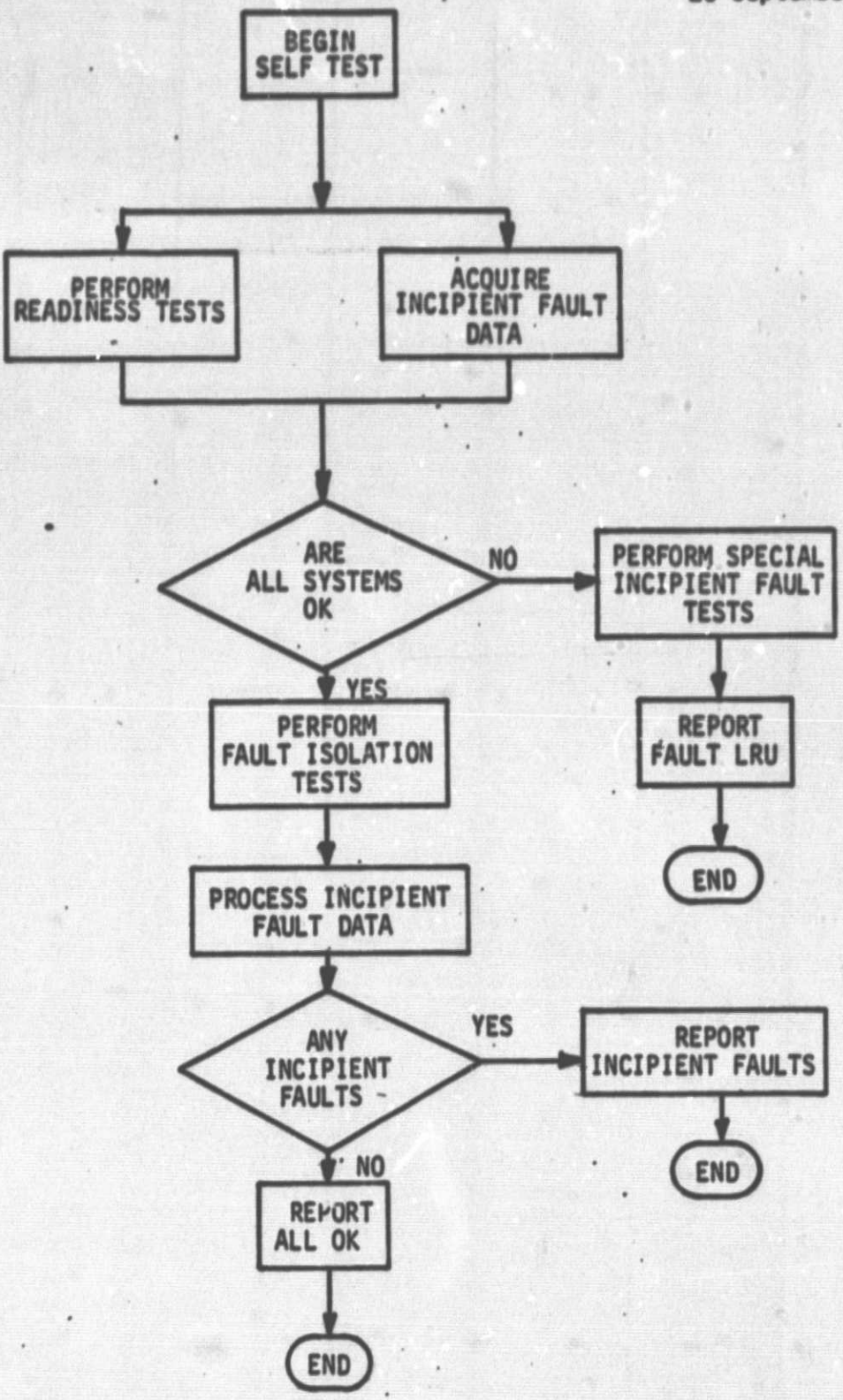

FIGURE 3.3-1 DAILY SELF TEST SEQUENCE

In summary, then, if the Fault Isolation Tests are initiated automatically, the operator will be provided with an output which identifies the following:

- o Subsystems tested that have successfully completed the readiness check
- o Sybsystems tested that have failed the readiness check
- o Results of Fault Isolations Tests for faulty subsystems
- o Results of analyses of Incipient Fault Detection Test data, including the following:
    - o LRU's that have moved into a gray area and are approaching an unsatisfactory level of performance
    - o Projected failure dates/times for near term critical LRU's

The operator must then decide whether to constrain the type of activities allowed in simulator usage, or to initiate maintenance or replacement of the components identified as failed. The output from self tests then is an indication of the readiness status of the system, including information helpful in trouble shooting or restoring any faulty functions or components of the Simulator.

SECTION 4

SIMULATOR SELF TEST SOFTWARE DESCRIPTION

The simulator self test software has been designed and structured so as to implement the test sequences discussed in Section 3 with respect to both the order of subsystem testing and the order in which the various types of tests are executed. Since the simulators of interest don't exist currently but are expected to be multi-computer systems with satellite processors available to handle input/output processing, data reformatting, and other miscellaneous operations, it is impossible to predict with much certainty the amount of computing power available in these peripheral processors. Therefore, it is also impossible to rationalize, at this point, the distribution of self test functions between the host computer and the interface computers. As a result, the approach taken in designing the software defers that problem to a later time and presents the software structure so as to define hierarchies of software modules, and the necessary sequencing of operations and channels of data communication.

A simulator self test software tree is shown in Figure 4-1. The software modules are identified by both an acronym, suitable for programming use, and a descriptive title. The modules shown in this tree do not have the relationship usually implied by a software tree in the sense that the modules at the two lower levels are not subprograms to the modules above them. Instead the implication of the structure of the tree is that modules at the same level may be exercised concurrently and independently, resources permitting; all modules in the same string but at a higher level must be exercised successfully before that module may be exercised; or, in other words, the modules in a string must be exercised in a top down sequence. All of these modules and programs will be controlled in their execution by the test executive and consequently in an ordinary tree would all be shown directly connected to the executive. Each of these modules is further described in detail in the final report for the hardware techniques task. In this report, we will limit ourselves to describing the test executive software.

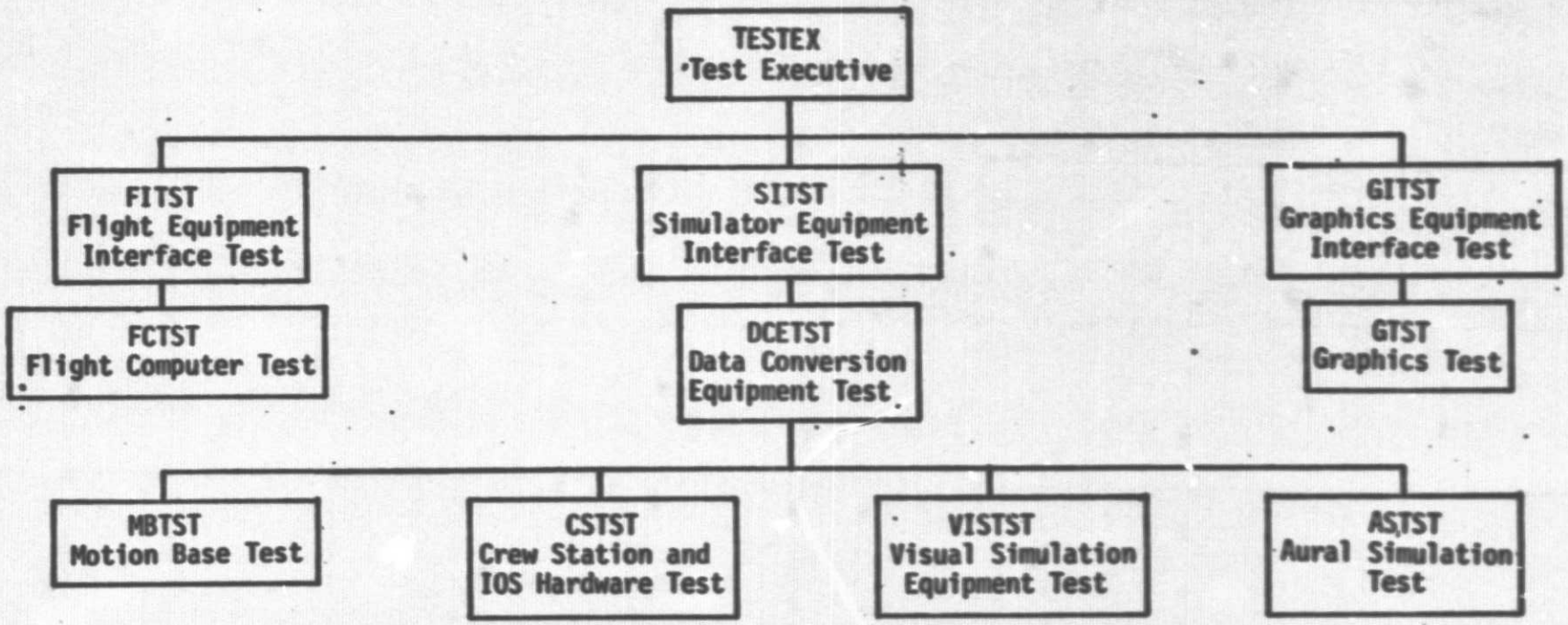

FIGURE 4-1 SIMULATOR SELF TEST SOFTWARE TREE

4.1 SELF TEST EXECUTIVE SOFTWARE

A software flow for the test executive is shown in Figure 4-2. The test executive is primarily concerned with controlling the order and extent of the testing executed and communicating the proper output messages and data for operator use as well as for storage in the data base for incipient fault detection. The test executive accomplishes its function by initiating and controlling the loading of all test software from mass storage facilities, commanding execution of the tests, reviewing the processed results from the tests and then selecting the proper course of action which will include in most cases, the presentation of data to the operator, the selection of additional tests, the storage of historical data in data base files, and the printing of test data reports.

It should be noted that the test executive does not become involved in the dynamic test timing function. The test data sampling rates are established by the test software for each subsystem being tested. Time tags for the data points are obtained from an external clock and are loaded into the controlling computer at the same time the data words are loaded. This minimizes the dependence of the subsystem tests on proper operation of all computer and interface timing functions and eliminates unpredictable interrupts and interrupt processing as an error source in test data timing.

The executive does contain the logic to decide on the course of action after each level of testing is completed. This logic requires access to the test results temporarily stored in a memory buffer.

When testing is completed or has proceeded to a standstill, the executive formats the data, if necessary, and communicates the appropriate information as follows:

- Operator display - Operator action requirements, including "system ready"
- Hardcopy printer - Daily test results summary and reference data
- Data base file - Incipient fault data base update

Figure 4-2 SELF TEST EXECUTIVE SOFTWARE FLOW

ORIGINAL PAGE IS
OF POOR QUALITY

Figure 4-2 SELF TEST EXECUTIVE SOFTWARE FLOW (continued)

4.2 SUBSYSTEM TEST SOFTWARE

The subsystem test software, such as FITST or MBTST, is loaded by the executive, TESTEX, and includes the programming which implements the tests, the characteristic parameter data associated with the hardware being tested, and the date processing software required for reducing the test data. The output of the subsystem test software is stored in a memory buffer for final disposition by the executive, TESTEX.

There are three different types of testing/processing at the subsystem level. These consist of readiness testing, fault isolation testing, and incipient fault detection processing, and are usually executed in that order.

The readiness tests are ideally end-to-end tests or total performance tests for a subsystem. If a readiness test indicates a fault and fault isolation tests are applicable, the decision to proceed with fault isolation can be taken automatically at the subsystem test software level, and the logic is shown accordingly as part of the subsystem tests. It is recognized that this decision could be exercised at the executive level where the results of other concurrent subsystem tests could also be considered. The decision can also be reserved for manual selection by the operator. Final choices can be made during software implementation without any significant impact on the software development.

Incipient fault detection data should nominally be gathered during the readiness tests. If a fault is detected at that time, the incipient fault detection data may be adversely affected. Consequently, the decision should be made at the subsystem level to delete this data and not add it to the data base. Otherwise, the incipient fault detection data is communicated to the executive through the memory buffer.

SECTION 5

SELF TEST SYSTEM HARDWARE

This section summarizes the total hardware requirements to implement the simulator Self Test System developed in the course of the study. The hardware listed includes only those components added for test purposes. A primary groundrule of the study was to maximize use of operational hardware in carrying out the Self Test job. This groundrule has been adhered to throughout the study, and hardware used in this manner is not specifically identified in this section. Cost data on Self Test System hardware is presented and discussed in Section 7.

Table 5.0-1 is a list of the various components added to each of the subsystems for Self Test purposes. Each photosensor used for galvanometer testing include a photodiode as a built-in light source. These devices are compatible with signal levels of logic circuitry used in the Data Conversion Equipment and therefore, no additional signal conditioning components are needed.

The current sensor circuits used to verify operation of lighted indicators are counted as a single component per indicator. These sensor circuits contain resistors, diodes, and transistors. However, each sensor circuit can be packaged as an individual module and using state of the art circuit integration technology, the circuits could be packaged right in the indicator assembly.

The loop closure switches listed for the Analog DCE are those solid state switches used to connect output channels to input channels in order to verify DCE operational status. Also under DCE is included the necessary components, ADC, switches, and buffer amplifiers needed to provide input channels for measurements taken during the Self Test process, measurements using instrumentation specifically added as part of the Self Test System.

Table 5.0-2 shows the breakdown of additional DCE channel requirements for Self Test. These requirements cover the data acquisition channels used specifically to measure characteristic parameters of simulator hardware during Readiness, Fault Isolation, or Incipient Fault Detection testing. No additional command channels were identified for applying test stimuli to operational hardware.

TABLE 5.0-1 HARDWARE ADDED FOR SELF TEST

COMPONENT	QUANTITY
Galvanometers (70)	
3 photosensors in each	210
Lighted Indicator Current Sensor Circuits	700
Analog DCE - Loop Closure Switches	
Closure to test input channels	200
Crew station isolation and connection of output channels to the 200 operational input channels	200
Analog DCE Test Channel Components	
ADC	1
Buffer amplifiers	120
Solid state analog switches	120
Bilevel DCE - Loop Closure Gating and Digital Input Multiplex Selection Gates	2500
Visual Simulation System Self Test Hardware	
Digital processing oscilloscope	1
Signal generator	1
Signal to noise meter	1
Video switches	16
Video buffer amplifiers	36
Motion Base	
Pressure sensors	5
Level sensor	4
Temp sensor	2

TABLE 5.0-2 ADDITIONAL DCE CHANNEL REQUIREMENTS FOR SELF TEST

COMPONENT TESTED	CHANNEL REQUIREMENTS	
	Bilevel	Analog
Galvanometers	210	
Servometers		16
Tapemeters		20
ADI (4 units)		
4 wire sin/cos, 3 channels each	48	
6 pointer position feedback signals from each ADI		24
Lighted Indicator	700	
Electromechanical Flags	400	
Switches - No extra lines for testing	---	
Visual Model Servos		
3 cameras x 6 servos		18
Terminal area transport servos		4
Orbital earth position resolver	30	
Cloud/sky/terminator altitude servos		3
Motion Base		
Sensors 11		
Position, actuators 6		17
	1388	102
With Spares	1500	125

However, in the case of the video components of the Visual Simulation System, the test controlling computer, with DCE, must issue the necessary commands to reconfigure the test hardware and control the test execution. These commands activate the DPO, signal generators, and measuring equipment used to test video elements. The number of these commands is relatively small (less than 20) and represent negligible impact on the total DCE.

On the other hand, the additional input channel requirements of 1500 bilevel and 125 analog channels for Self Test purposes represent an effective doubling of DCE capability when compared with the Reference Configuration magnitudes of 2000 bilevel and 100 analog channels. Considerable increase in parts and manufacturing cost should be expected from the fact that the increase in DCE capability is more accentuated on the analog instead of the bilevel side of the DCE. Analog channel components are larger, more complex, and more costly than the gating required to add bilevel channels.

Additional test hardware for the Visual Simulation System consists of the Digital Processing Oscilloscope, a video test waveform generator, a signal to noise meter, and the necessary switching to isolate each LRU during the testing sequence. No additional hardware is expected for testing model scene generation equipment other than that listed under DCE.

The Motion Base test hardware consists of the sensors and signal conditioning circuits required to route characteristic parameter data to the computer. Normally these parameters are only routed to the maintenance panel either electrically, mechanically, or hydraulically. In order to implement the automatic Self Test capability developed in this study, it is also necessary to digitize characteristic parameter data and supply them to the computer conducting the test.

The amount of hardware discussed above is a significant measure of the impact on simulator design and maintenance by the addition of automatic Self Test. Cost data related to this impact is discussed in Section 7. However, at this point it is important to point out that parts cost is a small part of total impact when compared with the total cost that includes generation of

additional wiring instructions, manufacturing additional housing space for these components, actual wiring cost, and increase in acceptance and integrated testing of the resulting system. All of these costs, however, may be offset by the increase achieved in simulator availability and utility during the operational life of the system. They also represent a very small percent increase to simulator total procurement cost.

SECTION 6
SYSTEM IMPACT BY REQUIREMENTS OR DESIGN CHANGES

This section discusses various types of changes that can occur after the simulator has become operational and the sources of changes. Particular attention is given to the impact of these changes both on the Simulator System as well as in the Self Test System. This discussion is the basis for assessing the effect of changes on the operational simulator and Simulator Self Test systems, and for formulating some recommendations that could minimize impact.

6.1 SOURCES OF CHANGE

As indicated above, this section only deals with changes initiated after the simulator has been installed, accepted, and is in its operational phase. Changes during design or development of the simulator per se are not considered here, because these changes are allowed a greater impact than those during the operational phase, and also because when considering the implementation of development changes, the impact of these changes on other parts of the system is evaluated before permitting implementation.

The major sources of changes during the operational life of the Simulator are:

1. Mission peculiar requirements variation from flight to flight.
2. Vehicle design changes.
3. Advancements in state of the art and
4. Performance enhancements.

These sources of change are discussed below.

6.1.1 Mission Peculiar Requirements Variations

The primary mission of Shuttle simulators is to provide a training facility for the various piloting and mission oriented crews that will man Shuttle controls during the life of the program. Although the aerodynamic and vehicle performance characteristics will not vary considerably from flight to flight, it is reasonable to expect significent changes in mission objectives, payload management requirements, and general procedures as dictated by the type of trajectory, orbital characteristics, and manuevering requirements needed to accomplish these mission objectives.

Mission related changes manifest themselves in the Simulator in the need for special payload management controls, payload related math models, and development of procedures that permit efficient utilization of vehicle and payload systems in accomplishing mission objectives. A dramatic example of mission related Simulator changes is the addition of payload environmental control and life support management facilities in the crew station, as well as the addition of math model restrictions in payload manipulation when changing Simulator configuration from a Shuttle carrying only unmanned payloads, to one carrying a mix of manned and unmanned payloads.

6.1.2 Vehicle Design Changes

This source of Simulator changes is more difficult to visualize at this point in the Shuttle program. The design of the vehicle is an ongoing effort, and additions and changes are taking place on a daily basis. It would be ideal if this process could achieve an optimal configuration prior to the first space flight; however, experience in previous programs has shown otherwise, and therefore, it is reasonable to expect design changes to take place as problems become more apparent during early flights of the Shuttle.

Design changes may affect vehicle performance or payload accomodation facilities and consequently alter the control and displays layout in the crew station, as well as the looks, feel or sound of related cues.

6.1.3 State of the Art Advancements

Even more difficult to visualize than vehicle design changes are those modifications that result from advances in the state of the art of simulator hardware. Generally, these modifications result in the replacement of multiple components for an integrated system that performs the same job more effectively. Effectiveness in this case refers to fidelity of simulation as a primary intention, and cost of operation when compared with the price of implementing the change.

6.1.4 Performance Enhancement

The fourth source of simulator modifications is the need to improve performance of the simulator beyond that which was specified, and supposedly met as shown by acceptance test results. Again, the primary consideration in this

case is fidelity of simulation. The situation may develop such that although the simulator performs as specified, it does not present realistic cues to the crew so that positive training can take place in an environment that truly represents the vehicle in looks, feels, sound, and performance; consequently, modifications are needed to enhance the simulation capability of the system.

Performance enhancement changes would generally manifest themselves in the installation of new equipment, or replacement of existing components with units that have more capability. These changes would usually result in an increase of complexity in the simulator system, and therefore increase the need for control as well as data acquisition channels in quantity and sometimes in capability.

6.2 IMPACT OF SIMULATOR CHANGES

Implementation of changes on the simulator system requires additions, deletions, and reconfiguration of hardware and software. This impact will affect simulation as well as self test elements. In many cases, also, the characteristic parameter data base will be impacted. Table 6.2-1 itemizes the impact of various changes on the affected elements of the system.

The impact analysis for changes in interface minicomputers, visual simulation, and motion base is unnecessary since changes in these subsystems are unlikely after simulator acceptance. These subsystems are large, complex, and generally the subject of individually managed procurement efforts. This special attention in procurement yields comprehensive requirement specifications for components that are reliable, maintainable, and capable of meeting future needs of the simulator

The Crew Station and IOS control and display elements may be changed either by vehicle changes, mission requirement changes, or the need to improve fidelity of simulation. Flight hardware used as part of the simulator system may also be the subject of change by either of the reasons mentioned above. The changes shown on Table 6.2-1 for the DCE generally result from changes in Crew Station and IOS controls and displays or in simulator Flight Hardware. Minor changes may develop to take care of needs for additional control or instrumentation of simulator functions.

Notice that the changes discussed may originate from needs to modify the simulation system, or from needs to modify the simulator Self Test System. Changes in one system generally impact configuration of the other system. This relationship is shown in the Table. Also, it should be noted that the characteristic parameter data base, as expected, is only affected when new hardware is introduced into the system, or when some hardware is removed or replaced.

TABLE 6.2-1 IMPACT OF SIMULATOR CHANGES

TYPE OF SIMULATOR CHANGE	IMPACT ON SIMULATOR HARDWARE	IMPACT ON SELF TEST HARDWARE	IMPACT ON SELF TEST SOFTWARE	IMPACT ON CHARACTERISTIC PARAMETER DATA BASE
CREW STATION/IOS				
o ADD CONTROL OR DISPLAY COMPONENT	o Panel Modification o Add command and response channels	o Add sensors o Add stimuli hardware	o Add calling sequence for self test routines for new component	o Add new component characteristic parameter data
o CHANGE CONTROL OR DISPLAY COMPONENT	o Minor.Maybe require changes in mounting hardware	o Minor, if any	o Minor, if any	o Change parameter data
o DELETE CONTROL OR DISPLAY COMPONENT	o Reconfigure Panels	o Minor removals	o Delete calling sequence for deleted component self test	o Remove parameter data
DATA CONVERSION EQUIP				
o ADD CHANNELS - ANALOG OR BILEVEL	o Add wiring, cabling, drivers/receivers,buffer amps & mux gating	o Add loop-back channels and switches to check new channels	o Extend channel check range in calling sequence to include new channels	o Add new channel numbers & special characteristics, if any
o REASSIGN CHANNELS	o Implement wiring or patching changes	o None --	o None --	o None --
FLIGHT HARDWARE				
o USE ACTUAL HDWE INSTEAD OF SOFTWE SIMULATED HDWE	o Add I/O channels o Possibly add I/O minicomputer o HOST I/O configuration modifications	o Minor, if any o Possible additional sensors & instrumentation	o If Hdwe added is different from other in simulator-need new Self Test routines o Otherwise only need test calling sequence	o Add parameter data for new hardware.
o CHANGE FLT HDWE IN SIMULATOR	o Only that required to implement flight hardware change	o May require different sensors	o Minor	o Change data base to reflect new hardware
o DELETE ACTUAL FLT HDWE.MAYBE REPLACE WITH MATH MODELS	o Decreased requirements by deletion of flight hardware	o Minor reduction	o Problem changes from hdwe verification to performance verification	o Characteristic parameter data becomes part of math model data base

SECTION 7

SELF TEST SYSTEM COST DATA

Relative cost data for addition of automatic self test capability to the Reference Configuration Simulator is shown in Table 7.0-1. No absolute dollar figures are shown because the selection of test component configuration, packaging, manufacturing, and installation techniques are dependent on the actual simulator design used, as well as the date in time when self test system implementation is to take place.

The first column shows cost of self test system elements relative to the basic cost of the subsystem to be tested. This rating ranges from Negligible (0-5%) to Moderate (20-30%). No "High" ratings were given to self test cost for any subsystem as this rating would have implied greater than a 50% increase in cost of the basic system. Consistent utilization of existing operational capability in the system for self test purpose is the basic reason for maintaining cost between Moderate and None.

The five columns to the right of the Table show cost of different factors in implementing subsystem self test as a percentage of total Self Test System cost. Some of these factors are estimated as less than 1% and therefore are shown only as *. The total cost of the Self Test System adds only to 98%; the remaining 2% is accounted for in the * items and others which exceed the whole number shown.

The largest percentage of Self Test System cost appears in Visual Simulation CCTV Self Test. This high percentage is due to the cost of the Digital Processing Osscilloscope, the special video pattern test signal generators, and the various signal and noise meters. The cost of visual simulation video component self test is further increased by test software development cost. This software is relatively more sophisticated than other elements of the test software in that it must provide processing capability for various complex waveforms from the video string. These waveforms have a variety of frequency components, require sampling rates of up to 50 Megahertz, and include test patterns such as a stair step and frequency multiburst signals.

TABLE 7.0-1 RELATIVE COST FACTORS OF SELF TEST SYSTEM

SUBSYSTEM	SELF TEST COST RELATIVE TO SIMULATOR SUBSYSTEM COST	SELF TEST COST RELATIVE TO TOTAL SELF TEST SYSTEM COST				
		DESIGN	FABRICATION & ASSEMBLY	HARDWARE COMPONENTS	SOFTWARE	TOTAL PER SUBSYSTEM
ANCILLARY COMPUTERS	LOW	2	*	1	5	8
FLIGHT HARDWARE	NEGLIGIBLE	1	1	1	2	5
CREW STATION AND IOS	MODERATE	3	3	5	10	21
ANALOG DCE	MODERATE	1	2	3	3	9
BILEVEL DCE	MODERATE	1	3	1	3	8
VISUAL SIMULATION-CCTV	NEGLIGIBLE	3	3	11	9	26
VISUAL SIMULATION-MODELS	NEGLIGIBLE	2	*	2	4	8
MOTION SYSTEM	LOW	1	1	3	5	10
AURAL CUES SYSTEM	MOD-LOW	1	*	*	2	3
					TOTAL **	98%

* Items that are less than 1% of total Self Test System cost

** 2% remain to cover costs not indicated numerically

Following closely in percentage of cost is the Crew Station and IOS self test factors. The high cost of controls and displays checkout is owed to the large number and variety of components to be tested, which in turn requires a fairly large characteristic parameter data base. In addition to the data base software, a generalized routine must be developed for each of the types of components used. Each component must be tested individually and therefore requires its own unique test calling and control sequence. With component count in the hundreds (thousands, if switches included), it can be seen that these sequences make up a considerable amount of software to be written, integrated, tested, and documented. Also, the large number of components affect design, manufacturing, and test component costs to make Crew Station and IOS self test a high portion of total Self Test System cost.

SECTION 8

CONCLUSIONS

The system level analyses of the simulator and automatic self test concepts have yielded several conclusions that help understand the self test process and the self test system as such. This understanding contributes to comprehensive, realistic and integrated requirements definition, while at the same time providing a picture of the cost of self test implementation relative to the operational advantages gained by its presence in the simulator system.

First of all, it was determined that three levels of testing are required to achieve effective operational utility of the Self Test System, and increase availability of the simulator. Readiness Tests yield information as to operational health of each subsystem. Fault Isolation Tests are called upon to determine which LRU, if any, has failed. Incipient Fault Detection Tests gather performance degradation data to schedule maintenance of degraded components before an actual failure occurs. The DCE, Motion Base, and Visual Simulation subsystems can be adequately checked at the functional level; only if a fault exists is it necessary to call the more detailed Fault Isolation Test. Other subsystems must be tested at the LRU level in order to ascertain readiness. Incipient Fault Detection techniques were found to be effectively applicable to the Motion Base System, the Visual Simulation elements, and to servo driven instrumentation.

A software structure for the Self Test system was developed. This structure conforms to the operational considerations brought about by the three categories of testing mentioned above, and depends largely on the use of utility test modules that minimize redundancy and maximize programming efficiency.

No significant increase in hardware was found to be needed to implement the simulator self test capability developed in the course of the study. On a subsystem basis, the relative cost of adding self test hardware was found to be from moderate to none. The greatest hardware impact is found in the DCE because of the more than 100% increase in number of analog channels, and in the Visual Simulation because of the Digital Processing Oscilloscope, signal generators, and signal meters needed to perform specially designed video component tests.

Change impact analysis indicated that changes occurring during operational phase of the simulator would be felt most in Crew Station and IOS controls and displays, in the Flight Hardware components of the simulator, and in the DCE. The magnitude of this impact cannot be assessed a priori, as the effect of a change greatly depends on the nature of the change and the components involved.